

# **Ostia Portus**

# Tutorials

Version 2012-12-17

December 2012



This document applies to Ostia Portus 2012-12-17 15:49:25 (MET) and to all subsequent releases.

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

© Copyright Ostia 2012. All rights reserved.

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

## **Table of Contents**

Tutorials	v
1 Adabas Tutorials	1
2 Using Java (Axis2) wrapper classes	3
3 Accessing Adabas using the PHP SOAP extension	9
What is PHP ?	
Installing the Eclipse PHP Development Tools	3
Accessing Adabas from PHP1	0
PHP Examples	2
4 Accessing a Portus Resource from a Ruby program	9
Running a Ruby program	0
5 Creating a sample C# application	3
6 Accessing Adabas through SoapUI	
7 Preparing for the LOBs (Large OBjects) samples 4	1
Loading the LOB file into an OpenSystems database	
Loading the LOB file into a Mainframe database	
8 Using LOBs (Large OBjects) with soapUi 44	
9 Accessing LOBs from PHP	
10 Accessing LOBs using a browser	
11 Accessing Adabas through Microsoft InfoPath	
12 Using Transactions with soapUI	
13 Portus - Configuration Versioning with Eclipse and CVS	5
Introduction	
Requirements	6
Requirements	O
Example Setup	
1	6
Example Setup	6 7
Example Setup	6 7 9
Example Setup	6 7 9 1
Example Setup	6 7 9 1 3
Example Setup	6 7 9 1 3 7
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php91	6 7 9 1 3 7
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10	6 7 9 1 3 7 1 5
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9117 ex02a_SoaGatewayEmpListDescending.php9118 ex02a_SoaGatewayEmpListSorted.php10119 ex06_SoaGatewaySpecial_SubDescriptor.php101	679137157
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10219 ex06_SoaGatewaySpecial_SubDescriptor.php10220 ex06_SoaGatewaySpecial_SuperDescriptor.php102	6791371579
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9216 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10219 ex06_SoaGatewaySpecial_SubDescriptor.php10220 ex06_SoaGatewayEmpAdd.php10221 ex03_SoaGatewayEmpAdd.php102	67913715791
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10119 ex06_SoaGatewaySpecial_SubDescriptor.php10220 ex06_SoaGatewayEmpAdd.php10221 ex03_SoaGatewayEmpAdd.php10322 ex04_SoaGatewayEmpGet.php103	679137157913
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10019 ex06_SoaGatewaySpecial_SubDescriptor.php10020 ex06_SoaGatewayEmpAdd.php10021 ex03_SoaGatewayEmpAdd.php10022 ex04_SoaGatewayEmpDel.php11023 ex05_SoaGatewayEmpDel.php110	6791371579135
Example Setup70More Information8114 Tut_02_List.java8115 ex01_SoaGatewayFirst.php9116 ex02_SoaGatewayEmpList.php9217 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10119 ex06_SoaGatewaySpecial_SubDescriptor.php10220 ex06_SoaGatewayEmpAdd.php10221 ex03_SoaGatewayEmpGet.php10222 ex04_SoaGatewayEmpGet.php11223 ex05_SoaGatewayEmpDel.php11224 ex10_SoaGatewaySimpleForm.php112	67913715791359
Example Setup7More Information814 Tut_02_List.java815 ex01_SoaGatewayFirst.php916 ex02_SoaGatewayEmpList.php917 ex02a_SoaGatewayEmpListDescending.php918 ex02a_SoaGatewayEmpListSorted.php1019 ex06_SoaGatewaySpecial_SubDescriptor.php1020 ex06_SoaGatewayEmpAdd.php1021 ex03_SoaGatewayEmpGet.php1022 ex04_SoaGatewayEmpDel.php1123 ex05_SoaGatewayEmpDel.php1124 ex10_SoaGatewaySimpleForm.php1125 ex15_SoaGatewayUpdateForm.php11	679137157913595
Example Setup7More Information814 Tut_02_List.java815 ex01_SoaGatewayFirst.php916 ex02_SoaGatewayEmpList.php917 ex02a_SoaGatewayEmpListDescending.php918 ex02a_SoaGatewayEmpListSorted.php1019 ex06_SoaGatewaySpecial_SubDescriptor.php1020 ex06_SoaGatewayEmpAdd.php1021 ex03_SoaGatewayEmpGet.php1022 ex04_SoaGatewayEmpDel.php1123 ex05_SoaGatewayEmpDel.php1124 ex10_SoaGatewayUpdateForm.php1125 ex15_SoaGatewayUpdateForm.php1226 empMiniList.rb (Ruby)12	6791371579135957
Example Setup74More Information8514 Tut_02_List.java8615 ex01_SoaGatewayFirst.php9716 ex02_SoaGatewayEmpList.php9717 ex02a_SoaGatewayEmpListDescending.php9718 ex02a_SoaGatewayEmpListSorted.php10019 ex06_SoaGatewaySpecial_SubDescriptor.php10020 ex06_SoaGatewayEmpAdd.php10021 ex03_SoaGatewayEmpGet.php10022 ex04_SoaGatewayEmpDel.php11123 ex05_SoaGatewaySimpleForm.php11124 ex10_SoaGatewayUpdateForm.php11125 ex15_SoaGatewayUpdateForm.php11227 ASGDemo.cs12	67913715791359579
Example Setup77More Information8714 Tut_02_List.java8815 ex01_SoaGatewayFirst.php9916 ex02_SoaGatewayEmpList.php9117 ex02a_SoaGatewayEmpListDescending.php9218 ex02a_SoaGatewayEmpListSorted.php10019 ex06_SoaGatewaySpecial_SubDescriptor.php10020 ex06_SoaGatewayEmpAdd.php10021 ex03_SoaGatewayEmpDel.php10022 ex04_SoaGatewayEmpDel.php11123 ex05_SoaGatewaySimpleForm.php11124 ex10_SoaGatewaySimpleForm.php11125 ex15_SoaGatewayUpdateForm.php11226 empMiniList.rb (Ruby)12227 ASGDemo.cs12228 MySQL_City.cs122	6791371579135957913

# **Tutorials**

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.

There is a tutorial demonstrating how IBM MQ Series can be used as a transport for SOAP messages to Portus.

- Adabas Tutorials
- MySQL Tutorials
- Usage Governance Tutorials
- MQ Tutorial

# 1 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 Center functions required to achieve this before continuing the tutorials trail.

#### **Basic Tutorials**

The basic tutorials cover the following areas:

- Using Java (Axis2) wrapper classes
- Using the PHP SOAP Extension
- Using Ruby
- Using C#
- Using soapUi
- Using Microsoft Office : Infopath

#### **Advanced Tutorials**

Dealing with LOBs (Large OBjects)

- Preparing the LOB file
- Using LOBs with soapUi
- Using LOBs with PHP
- Using LOBs with your favourite browser

Dealing with Transactions

**Transactions using soapUi** 

#### Samples

This section also includes the following demo files and code samples that you can try out:

- Tut\_02\_List.java
- ex01\_SoaGatewayFirst.php
- ex02\_SoaGatewayEmpList.php
- ex02a\_SoaGatewayEmpListDescending.php
- ex02a\_SoaGatewayEmpListSorted.php
- ex06\_SoaGatewaySpecial\_SubDescriptor.php
- ex06\_SoaGatewaySpecial\_SuperDescriptor.php
- ex03\_SoaGatewayEmpAdd.php
- ex04\_SoaGatewayEmpGet.php
- ex05\_SoaGatewayEmpDel.php
- ex10\_SoaGatewaySimpleForm.php
- ex15\_SoaGatewayUpdateForm.php
- empMiniList.rb
- ASGDemo.cs
- Other Samples

# 2 Using Java (Axis2) wrapper classes

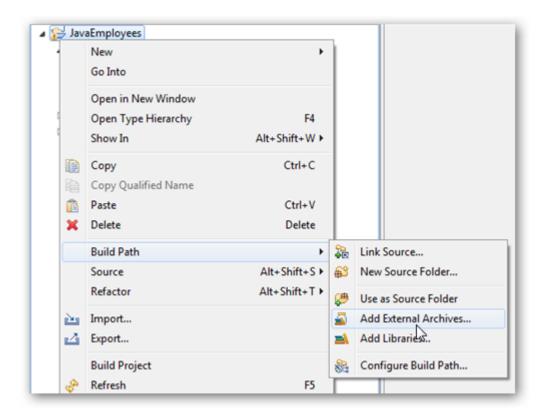
### Tutorial: Generate a Java wrapper for the "Employees" file

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..:"

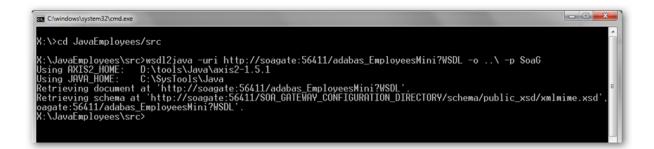


Add all .jar files from the axis2 "lib" directory to the project's Build-Path.

Organize 👻 New	folder			8	i • 🔟	8
🔆 Favorites	4	Name	Date modified	Туре	Size	
Nesktop		activation-1.1.jar	08.10.2009 15:31	Executable Jar File	62 KB	
🐌 Downloads		axiom-api-1.2.8.jar	08.10.2009 15:31	Executable Jar File	241 KB	
Recent Places	=	📧 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	
🥽 Libraries		🖃 axis2-adb-1.5.1.jar	19.10.2009 10:58	Executable Jar File	759 KB	
Documents		📧 axis2-adb-codegen-1.5.1.jar	19.10.2009 11:01	Executable Jar File	165 KB	
🚽 Music	-	axis2-ant-plugin-1.5.1.jar	19.10.2009 11:42	Executable Jar File	17 KB	
Pictures		axis2-clustering-1.5.1.jar	19.10.2009 11:04	Executable Jar File	100 KB	
🗧 Videos		📧 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	
🜏 Homegroup		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	
📜 Computer	*	axis2-jaxbri-1.5.1.jar	19.10.2009 11:04	Executable Jar File	25 KB	
F	ile nar	ne: "XmlSchema-1.4.3.jar" "activation-1.1	.iar" "axiom-ani-1.2.8.iar" "ax	di 🔻 🔭 *.jar;*.zip		+

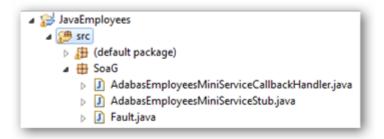
3. Open a command prompt (aka "DOS box"), change to the "JavaEmployees\src" directory and run the following command

```
wsdl2java -uri http://<yourserver>:<yourport>/adabas_EmployeesMini?WSDL -o ..\
-p SoaG
```



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



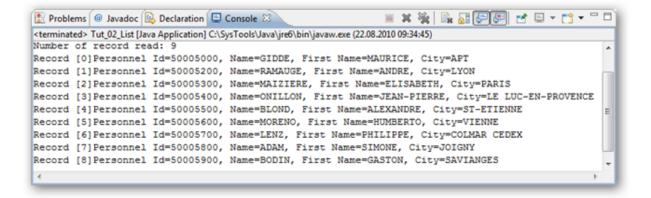
4. Add a new Java class named "Tut\_02\_List" to the project ( **File** -> **New** -> **Class** ), opt to create a "main" method, click "**Finish**".

New Java Class		
Java Class	efault package is discouraged.	C
Source folder:	JavaEmployees/src	Browse
Package:	(default)	Browse
Enclosing type:		Browse
Name:	Tut_02_List	
Modifiers:	public Odefault Oprivate Oprotected     abstract Infinal static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add
		Remove
Which method stu	bs would you like to create? public static void main(String[] args)	
	Constructors from superclass	
Do you want to add	Constructors from superclass  Inherited abstract methods comments as configured in the <u>properties</u> of the current pro Generate comments	oject?

- 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"

🔋 Package Exp	lorer	🕄 🍃 Hierarchy	- 0	🕼 Tut_02_List.java 🕸
			(⇒ ⇔ (⊚)   <b>≕ %</b> ▼	19 import SoaG.*;
a 😂 JavaEmj	olove	es	· · · · · · · · · · · · · · · · · · ·	2 import SoaG.AdabasEmployeesMiniServiceStub.*;
▲ 2 src ▲ 3 (default package)			3 5⊕ * The SOA Gateway LIST method will return all data matching the ∏	
				138/* Hit Ctrl+S to save the file.
	1 10	ut_02_List.java		14 * The IDE will automatically build the file, and display
- 🖽		New	•	15 * errors and warnings in the "Problems" view.
		Open	F3	16 * Now add the class definition and the main method. 17 */
Þ		Open With	•	18
▷ 🖽 :		Open Type Hierarchy	F4	19 public class Tut_02_List {
⊳ 🛋 JRE ⊳ 🛋 Refe		Show In	Alt+Shift+W ►	20 219 public static void main(String[] args) {
a buil		Сору	Ctrl+C	
	100		Carre	23 try (
		Copy Qualified Name		<pre>24 AdabasEmployeesMiniServiceStub stub = new AdabasEmployeesMiniServiceStub();</pre>
	6	Paste	Ctrl+V	25
	x	Delete	Delete	26 AdabasEmployeeKeyType keys = new AdabasEmployeeKeyType();
		Build Path	,	<pre>27 keys.setPersonnel_id("50005*");</pre>
				<pre>28 keys.setCity(""); 29 keys.setName("");</pre>
		Source	Alt+Shift+S ►	29 <b>keys.setName("");</b> 30
		Refactor	Alt+Shift+T >	31 AdabasEmployeeListElement listKey = new AdabasEmployeeListElement();
	2	Import		32 listKey.setAdabasEmployeeListElement(keys);
	_	Export		33
	<u> </u>	export		34 AdabasEmployeesMiniElement result = null;
		References	•	35
		Declarations	,	<pre>36 result = stub.list(listKey, null, null); 37</pre>
		Refresh	F5	3/ RR AdahaeFunlouaaeMiniFlamantTuma voot = vaeult vatAdahaeFunlouaaeMiniFlamant/).
	Ş		61	
		Assign Working Sets		🖹 Problems 🐵 Javadoc 🔞 Declaration 📮 Console 🕱
		Run As	•	1 Run on Server Alt+Shift+X, R C:\SysTools\Java\jre6\bin\javaw.exe (22.08.2010 01:33:51)
		Debug As	•	2 Java Application Alt+Shift+X, J
		Profile As	•	Conce Rive Dislam
		Validate		O Open Run Dialog

7. The output appears in the "Console" window:



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*");
```

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

# 3

# Accessing Adabas using the PHP SOAP extension

What is PHP ?	10
Installing the Eclipse PHP Development Tools	23
Accessing Adabas from PHP	10
PHP Examples	32

### 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<sup>™</sup> distributions, Windows<sup>®</sup>, Mac OS X, and iSeries<sup>™</sup>.

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.

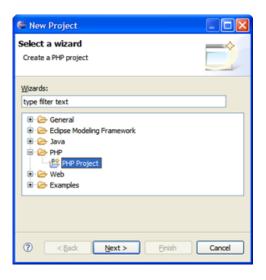
Please refer to the Eclipse PDT project pages for installation and configuration instructions.

## Accessing Adabas from PHP

1. First, create a new project within your workspace.

<u>ile E</u> dit <u>S</u> ource Refac <u>t</u> o	or <u>N</u> avigate Se <u>a</u> rch	Project Software <u>A</u> G <u>R</u> un <u>W</u> indov
New	Alt+Shift+N	▶ 📑 P <u>r</u> oject
Open File <u>.</u>		
⊆lose	Ctrl+W	Package
Close All	Ctrl+Shift+W	Class
Jave Save	Ctrl+S	🔹 👔 Interface
Save As		
🕞 Sav <u>e</u> All	Ctrl+Shift+S	Source Folder
Rever <u>t</u>		Ger Folder

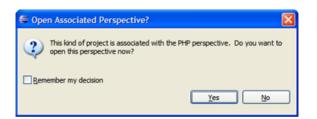
2. Opt to create a PHP Project, click Next



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** 

PHP Project	
New PHP Project Create a PHP project	P
Project name: PHPDemo Project contents: Use default Directory: E:\servers\Apache20\Apache2\	htdocs\asgdocs Browse
PHP Version  Enable project specific settings  PHP Version: PHP 5 or greater  Use ASP tags as PHP tags	Configure Workspace Settings
⑦ < <u>B</u> ack <u>N</u> ext >	Einish Cancel

If you are asked to switch to the PHP perspective, opt to do so



3. Create the PHP SOAP client

Create a PHP script file by right-clicking into the Eclipse Navigator area, select **New -> PHP** File

$\Diamond \Leftrightarrow \overline{\mathbb{Q}}$	
Demo PHPdemo .project	
Ne <u>w</u>	📑 P <u>r</u> oject 🗡
Go Into	
Open in <u>N</u> ew Window	PHP File
	😂 Folder
Paste	📑 File
💢 <u>D</u> elete	📑 E <u>x</u> ample
Mo <u>v</u> e	Ther Ctrl+N
Rename	

Specify a File name, click Finish

e		
PHP New This wizard	<b>File</b> I creates a new PHP file.	1
<u>C</u> ontainer:	/PHPdemo	Browse
<u>File name:</u>	PHPempMini.php	-
	Einish	Cancel

Opt to create a "Simple PHP File"

E New PHP File	
Select PHP Template	
Select a template as initial content in the	ie PHP page.
Use PHP Template	
Templates are "New PHP" templates for	und in the <u>PHP Templates</u> preference page.
Name	Description
New PHP file - HTML frameset	html 4.01 frameset
New simple PHP file	Simple php file
Preview	
php</td <td><u>^</u></td>	<u>^</u>
\${cursor}	_
?>	~
3	>
L	
(7) < Back	Next > Finish Cancel
- East	

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:



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:



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:

<ter< th=""><th>minated</th><th>d&gt; P⊢</th><th>PHPdemo [PHP Application] PHP D:\Servers\php512\php-win.exe : PHPempMini.php</th><th></th></ter<>	minated	d> P⊢	PHPdemo [PHP Application] PHP D:\Servers\php512\php-win.exe : PHPempMini.php	
Arı	ay			
(				
	[0]	=>	> adabasEmployeesElementType list(adabasEmployeeKeyType	<pre>\$adabasEmployeeListKey)</pre>
	[1]	=>	> adabasEmployeesElementType get(adabasEmployeePrimaryKe	yType \$adabasEmployeeGetKey)
	[2]	=>	> operationResult delete(adabasEmployeePrimaryKeyType \$a	dabasEmployeeDeleteKey)
	[3]	=>	> operationResult add(adabasEmployeesType \$adabasEmploye	es)
	F41	=>	> operationResult update(adabasEmployeesType \$adabasEmplo	oveesUndate)

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:

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'=>'', 'city'=>'');
```

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 "";
echo "echo ">th>Personnel IDNameCityAddress
foreach ($Adabasresponse->adabasEmployees->adabasEmployee as $Employee) (
    echo "",
"*tr>",
"$Employee->personnel_id",
"$Employee->firstname $Employee->name
r*td>?;
foreach ($Employee->city
foreach ($Employee->address_line as $adline)
    echo "$adline<br />";
echo "
```

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:

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
20000125	Momino	Dasher	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:

- First steps
- List Employees
- List Employees descending
- List Employees sorted
- List by Sub Descriptor
- List by Super Descriptor
- Add an Employee
- Get an Employee
- Delete an Employee
- A simple PHP form for accessing Adabas
- All-in-one PHP form accessing the Employees file



# Accessing a Portus Resource from a Ruby program

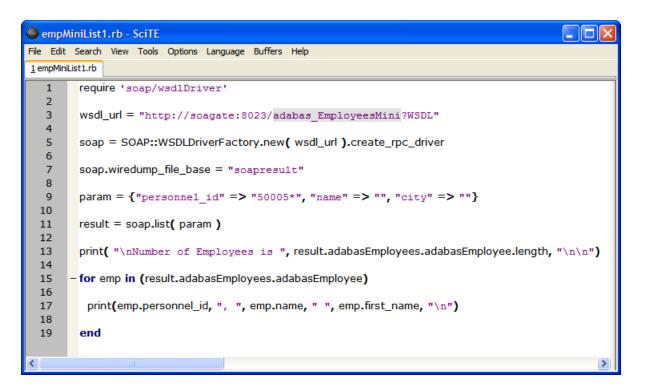
Running a Ruby pro	gram	. 20
--------------------	------	------

Ruby (**downloadable here**) is an Open Source object oriented language with a very simple, yet powerful SOAP interface.

### Running a Ruby program

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.



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

Semp <i>l</i>	NiniList1.rb - SciTE	
File Edit	Search View Tools Options Language Buffers Help	
<u>1</u> empMini	List1.rb	
1 2	require 'soap/wsdlDriver'	>ruby empMiniList1.rb
3 4	wsdl_url = "http://soagate:8023/adabas_EmployeesMini?WSI	Number of Employees is 9 50005000, GIDDE MAURICE
5	<pre>soap = SOAP::WSDLDriverFactory.new( wsdl_url ).create_rpc_dr</pre>	50005200, RAMAUGE ANDRE 50005300, MAIZIERE ELISABETH
7	<pre>soap.wiredump_file_base = "soapresult"</pre>	50005400, ONILLON JEAN-PIERRE 50005500, BLOND ALEXANDRE 50005600, MORENO HUMBERTO
8 9	<pre>param = {"personnel_id" =&gt; "50005*", "name" =&gt; "", "city</pre>	50005700, LENZ PHILIPPE 50005800, ADAM SIMONE
10 11 12	result = soap.list( param )	50005900, BODIN GASTON ignored element: {http://www.w3.org/20 >Exit code: 0
13 14	<pre>print( "\nNumber of Employees is ", result.adabasEmployees.</pre>	
15 16	- for emp in (result.adabasEmployees.adabasEmployee)	
10 17 18	<pre>print(emp.personnel_id, ", ", emp.name, " ", emp.first_name,</pre>	
18	end	
<		

# **5** Creating a sample C# application

### Tutorial: A sample C# application listing "Employees"

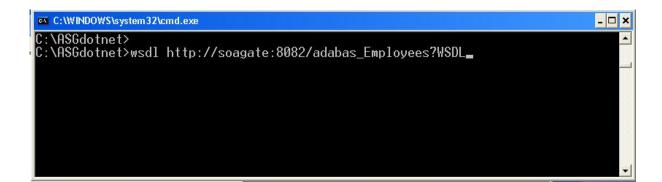
With a service description (WSDL), a proxy class can be created with the .NET Framework SDK Wsdl.exe tool. A XML Web service client can then invoke methods of the proxy class, which communicate with Portus over the network by processing the SOAP messages sent to and from the Portus server. The proxy class handles the work of mapping parameters to XML elements and then sending the SOAP message over the network.

Wsdl.exe 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#.

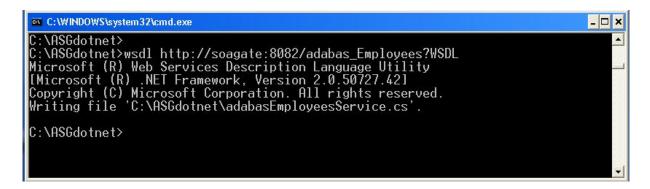
These are the steps required to generate the C# wrapper class using Wsdl.exe and create / run a program listing records from the Adabas demo file "Employees" using the generated proxy class:

1. From a command prompt, execute Wsdl.exe, specifying the URL / URI of the Portus DataSource to be exposed, append ?WSDL to instruct thePortus 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 ".



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



This file contains a proxy class exposing both synchronous and asynchronous methods for each SOAP operation provided by Portus for the DataSource. 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 the Portus server asynchronously.

For more information about asynchronous communication with a Web Service please refer to the .NET documentation.

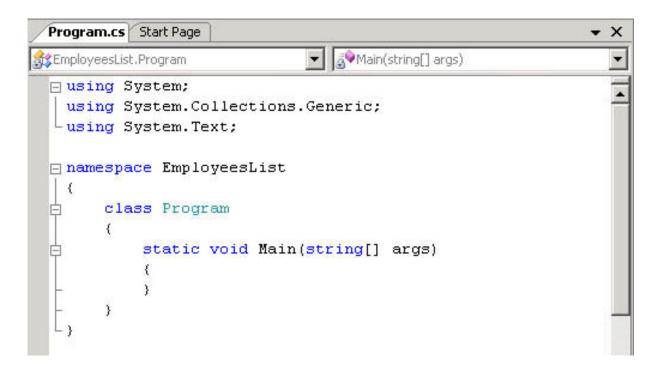
3. Start MS Visual Studio, create a new project with **File** -> **New - Project** (or the shortcut Ctrl+Shift+N):

File	<u>Edit View T</u> ools Te <u>s</u> t	W	<u>/indov</u>	v <u>⊂</u> ommunity	/ <u>H</u> elp
	New	•		Project	Ctrl+Shift+N
	<u>O</u> pen	•	1	Web Site	
	⊆lose		1	<u>F</u> ile	Ctrl+N
-	Close Solution			Project From [	Existing Tode
	Save Selected Items Ctrl+S		JU	uuiu	2005
	Save Selected Items As				
3	Save All Ctrl+Shift+S				Visual Studio Dev
	Export Template				The current news
1	Page Setup				Environment and
3	Print Ctrl+P				
	Recent Projects	•			
	E <u>x</u> it				
	Open: Project	W	eb Sit eb Sit		

Create a C# Console Application, assign a name to it, specify the storage location, click **OK** 

New Project					? ×
Project types:		Templates:		00	0-0- 0-0- 0-0-
<ul> <li>Pisual C#</li> <li>Windows</li> <li>Office</li> <li>Databas</li> <li>Starter k</li> <li>Test</li> <li>Other Langu</li> <li>Other Project</li> <li>Test Projects</li> </ul>	evite e Kits Hages tt Types	Visual Studio installed templates         Windows Application         Windows Control Library         Console Application         Empty Project         My Templates         Search Online Templates	e Class Library web Control Library 땡 Windows Service 교 Crystal Reports Application		
A project for cre	ating a command-line	application			
Location:	D:\ASGdotnet	-	•	Browse	
Solution Name:	EmployeesList	Creat	te directory for solution		
			ок	Cance	!

A skeleton class file has been generated into your project workspace, with the required class definition and an empty *Main* method



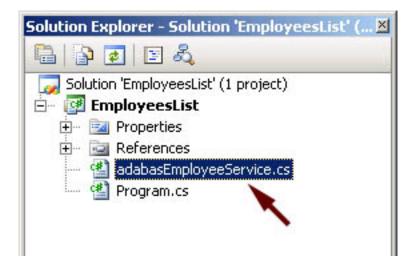
4. First of all, import the generated proxy into the project, right-click on the project name, select Add -> Existing Item

Solution Explorer - Em	ploy	eesList 🗵		
🔁 🔁 🛃 🕹				
Solution 'Employees		1 project)		
EmployeesLis		Build		
E References		R <u>e</u> build		
- 🥙 Program.cs		Clea <u>n</u>		
		Pu <u>b</u> lish		
		Run C <u>o</u> de Analysis		/
		A <u>d</u> d	-	New Item.
		Add <u>R</u> eference	:::	Existing Item
		Add Web Reference	-	New Fol <u>d</u> er
	Ł,	<u>V</u> iew Class Diagram		Windows Eorm
		Set as St <u>a</u> rtUp Project		User Control
		Debug +	Ð	Component
	*	Cu <u>t</u>	23	<u>⊂</u> lass
	8	Paste		

select the AdabasEmployeeService.cs proxy, click Add

P	dd Existing Ite	m - Employees	List			? ×
	Look in:	C ASGdotne	t	💽 🕝 - 🖄 🔍 🗡 📩	🛛 🔢 🕶 Tools	•
	Desktop	EmployeesL EmployeesL adabasEmpl				
	D My Projects		$\mathbf{N}$			
	My Computer					
		File <u>n</u> ame:			<u> </u>	<u>A</u> dd ▼
		Files of <u>type</u> :	Visual C# Files (*.cs	;;*.resx;*.settings;*.xsd;*.wsdl)	<b>_</b>	Cancel

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** 

Solution Explorer -	EmployeesList	×
🗟 🗿 🛃 🍇		
Solution 'Employees	eesList' (1 project) iList	
🗄 🖻 🖬 🔛	Build	
	R <u>e</u> build	
Pro 🔮	Clea <u>n</u>	
	Pu <u>b</u> lish	
	Run C <u>o</u> de Analysis	
	Add	•
	Add <u>R</u> eference	
	Add Web Reference	

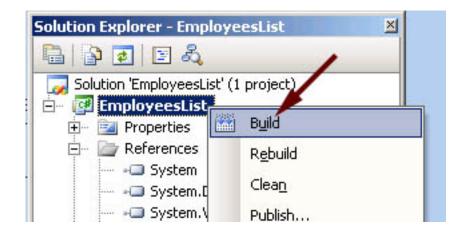
Scroll down to System.Web.Services, click to select it, click to select it, click **OK** to import the reference

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.Window 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 鷔	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.



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:

):\ASGdotnet\EmployeesList\Employ lumber of Employees returned: 10	eesList\bin\Debug>EmployeesList	
Record [0], Personnel_Id=30000001	. Name=Smith. First Name=Frank	
Record [1]. Personnel Id=30000007	. Name=Turner. First_Name=John	
Record [2], Personne1_Id=30000012	, Name=Winterton, First_Name=Robert	
Record [3], Personnel_Id=30000014	, Name=Singh, First_Name=Mumtaz	
Record [4], Personne1_Id=30000037		
Record [5], Personnel_Id=30000038		
Record [6], Personnel_Id=30000042		
Record [7], Personnel_Id=30000043		
Record [8], Personnel_Id=30000044		
Record LYJ, Personnel_1d=30000045	, Name=Garfield, First_Name=James	
••• •••• ••• ••• ••• •••••••••••••••••		
):\ASGdotnet\EmployeesList\Employ	eesList\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:

🔤 C:\WINDOWS\system32\cmd.exe	- O ×
D:\ASGdotnet\EmployeesList\EmployeesList\bin\Debug>EmployeesList Number of Employees returned: 3 Record [0], Personnel_Id=30000311, City=Derby, Name=Smith, First_Name=Gerald Record [1], Personnel_Id=30034001, City=Derby, Name=Smith, First_Name=Francis Record [2], Personnel_Id=30038013, City=Derby, Name=Smith, First_Name=Winston D:\ASGdotnet\EmployeesList\EmployeesList\bin\Debug>	
	• //.

## 6 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.

🛓 50	SoapUI SNAPSHOT						
Eile	Tools	<u>D</u> esktop	Help				
📴 Pr	ojects				\$		
		New WSDL I	Project	Strg-N			
T		Import Proj	ect Cre	eates a new ws	dl p	oject in this workspace	
		Save All Pro	jects	Strg+Alt-S			
	0	Online Help		F1			
	_						
						<b>v</b>	
Wor	kspace P	roperties				Wed Nov 08 17:24:32 CET 2006:INFO:Loading workspace from [C:\Dokumente und Einstellungen\A	
	Prope	rty		alue		Wed Nov 08 17:24:33 CET 2006:WARN:Failed to load settings [soapui-settings.xml (Das System ka	ann die a
File		C	\Dokume	ente und E			
					6		•
Det	ails W	/indows				soapui log http log	

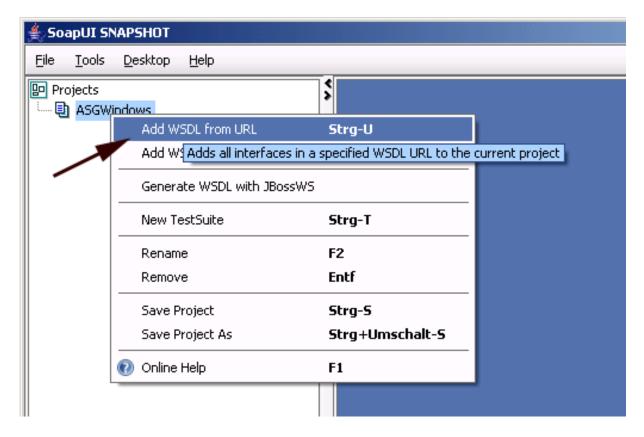
You will be prompted for a project name, enter one and click OK:

New WSD	L Project	×
?	Enter project name ASGWindows OK <u>A</u> bbrechen	

Specify a target location for the project files, click **Save**:

擒 Create Pro	oject		×
Spe <u>i</u> chern in:	🗀 soapui	•	🖻 🖄 🌁 🔡 🖿
Debeiserer	Accurate and a second second second		
Datei <u>n</u> ame:	ASGWindows-soapui-project.xml		
Da <u>t</u> eityp:	XML Files (*.xml)		•
			Save Cancel

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://<yourASGserverhost>:<yourASGserverport>/adabas\_Employees\_special?WSDL and click **OK** to import the webservice definitions from the resources / WSDL

Add WSI	DL from URL 🛛 🔀
?	Enter W5DL URL 122/adabas_Employees_special?W5DL

You will be asked if default requests for all operations are to be created, click Yes.

Import WSDL
Create default requests for all operations
Yes No Cancel

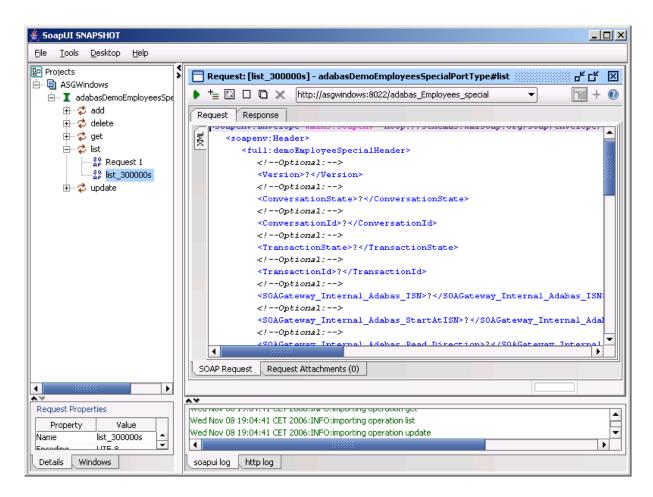
3. Default requests have been generated for all Portus operations for a webservice, undfold the "list" request by clicking the **plus** sign left of it, right-click the created **Request 1**, select **Clone Request**, this allows for unlimited duplication of the original request, which may be desirable when testing various options or "canning" requests.

🔛 Projects		
🖻 🖳 ASGWindow	IS	
🗄 🛛 🍸 adabasi	DemoEmployeesSpecialPor	rtType
🗄 🗘 ado	1	
🕂 🖓 dele	ete	
🕀 👶 get		
🚊 – 之 list		
	Request 1	
±¢ u	Open Request Editor	Eingabe
	<sup>+</sup> ≡ Add to TestCase	Strg+Alt-A
	🗋 Clone Request	F9
	Creates a copy of this r	equest
	Delete	Entf
	😰 Online Help	F1

Assign a name to the cloned request, click **OK** 

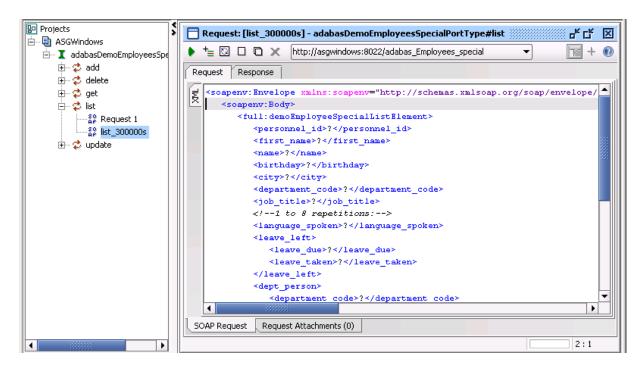
Clone Re	quest	×
?	Specify name of cloned request list_300000s <u>QK</u> <u>Abbrechen</u>	

4. soapUI now opens the request document.



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:



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:



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):

1	È 🖸		۵	×	http://asgwindows:8022/adabas_Employees_special	-	72 +
Req	uest	Res	spon:	se			
Į	<soap< td=""><td>env:</td><td>Ent</td><td>7elo</td><td>pe xmlns:full="urn:namespaces:com.risaris/xm:</td><td>iddle/qe/</td><td>resour</td></soap<>	env:	Ent	7elo	pe xmlns:full="urn:namespaces:com.risaris/xm:	iddle/qe/	resour
ì	≺s	oape			-		
		<rs< td=""><td>-</td><td></td><td>asDemoEmployeesSpecialElement xmlns:rsp="urn</td><td>namespac</td><td>es:com</td></rs<>	-		asDemoEmployeesSpecialElement xmlns:rsp="urn	namespac	es:com
			<ac< td=""><td></td><td>sDemoEmployeesSpecial&gt;</td><td></td><td></td></ac<>		sDemoEmployeesSpecial>		
				<de< td=""><td>moEmployeeSpecial&gt;</td><td></td><td></td></de<>	moEmployeeSpecial>		
					<pre><personnel_id>30000001</personnel_id></pre>		
					<first_name>FRANK</first_name>		
					<name>SMITH</name>		
					<middle_name>JOHN</middle_name>		
					<marital_status>M</marital_status> <sex>M</sex>		
					<sex>m</sex> <birthday>710031</birthday>		
					<address>12 PASTURES VIEW</address>		
					<address>BLIDWORTH</address>		
					<address>MANSFIELD</address>		
					<city>MANSFIELD</city>		
					<pre><zip code="">NG22 3PF</zip></pre>		
					<country>UK</country>		
					<pre><phone area="" code="">0623</phone></pre>		
					<pre><phone number="">345542</phone></pre>		
					<pre><department_code>FINA01</department_code></pre>		
					<job_title>ACCOUNTING MANAGER</job_title>		
			-		<income></income>		
					<currency_code>UKL</currency_code>		
					<annual_salary>000010360</annual_salary>		
					<annual_bonus>00000000</annual_bonus>		
					<pre><annual_bonus>00000000</annual_bonus></pre>		
					<pre><annual_bonus>00000000</annual_bonus></pre>		
					<annual_bonus>00000000</annual_bonus>		
			1		<annual_bonus>00000000</annual_bonus>		
				$\mathbf{i}$	<annual_bonus>00000000</annual_bonus>		
					<annual_bonus>00000000</annual_bonus>		
					<pre></pre>		
			-		<income></income>		
					<currency_code>UKL</currency_code>		
					<pre><annual_salary>000010050</annual_salary></pre>		
					<pre><annual_bonus>00000000</annual_bonus> </pre>		
					<pre><annual_bonus>00000000</annual_bonus> </pre>		
	•						
504	AP Res	ponse	•	Resp	onse Attachments (0)		

### 

### Preparing for the LOBs (Large OBjects) samples

Loading the LOB file into an OpenSystems database	42
Loading the LOB file into a Mainframe database	42

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</p>

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.
- Important: Note that 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.

### 8 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

This tutorial assumes the following

- Vou 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
- DataView : Select 'adabas\_photoblobs', which is one of the sample definitions delivered with Portus
- 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

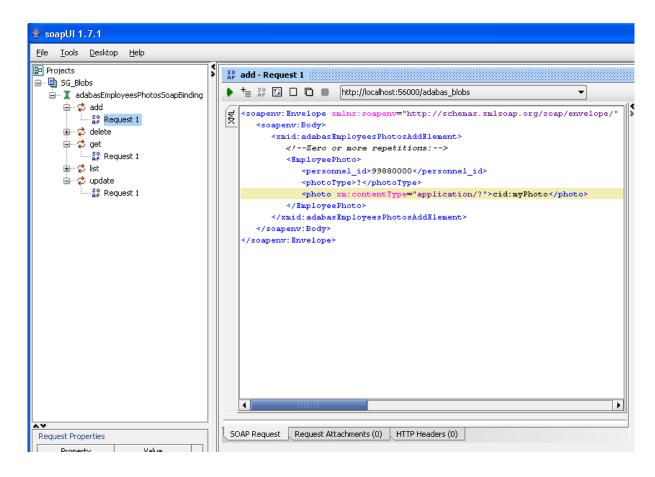
🎄 snapUl 1.7.1	
Ele Iools Desktop Help	
Projets       Projets     SG_Bobs       Image: State add as Employees Photos SoapBinding       Image: State add add add add add add add add add ad	
Project Properties	
Property Value	

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



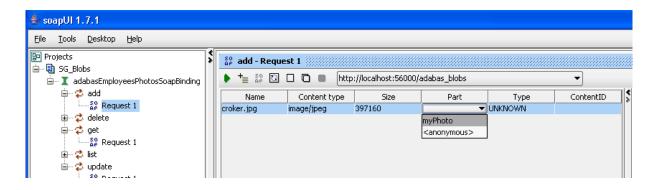
5. Click the Request Attachments tab, and click Add File

Select the required LOB file and click Open

👙 soapUl 1.7.1		
<u>Fi</u> le <u>T</u> ools <u>D</u> esktop <u>H</u> elp		
Projects	ន្តិ add - Request 1	
i ⊆ 🛃 SG_Blobs i ⊑ I adabasEmployeesPhotosSoapBinding	+ + S S □ □ □ http://localhost:56000/adabas_blobs	•
亩··· 😂 add 음을 Request 1	Name Content type Size Part Type	ContentID
🕀 🤣 delete		
금··· \$ get 1	🚖 Open	
ist ⊡ ≎ update	Look In: 🧰 My Pictures 🔹 💼 🟠	
Request 1		Modified
	Sample Pictures Shortcut 23/08/09 croker.jpg 387 KB JPEG Image 10/08/07	
	File Name: croker.jpg	
	Files of Type: All Files	
	Thes of Type. With thes	
	•	pen Cancel
		Open selected file
	Add file Remove selected Export selected Online Help	
Request Properties	SOAP Request Request Attachments (0) HTTP Headers (0)	

When asked to Cache Attachment select No

6. Click the Part column, and select the CID reference you changed earlier, for example myPhoto



7. In the Request Properties change Enable MTOM/Inline to true

1	🖢 soapUl 1.7.1								
	<u>File T</u> ools <u>D</u> esktop	Help							
	Ele       Tools       Desktop       Help         Projects       SG_Blobs         Image: SG_Blobs       Image: SG_Blobs		~		Image: Second state       Image: Secon			ht ent type	tp://localhost Siz 397160
	~*								
e H	Request Properties		]						
	Property	Value							
d	Name	Request 1							
	Encoding	UTF-8							
	Endpoint	http://localhost:560							
ł	Enable MTOM/Inline	false 🔻							
	Username	true							
	Password	false					A	ld file	Remove sele:
o	Domain								11011040 3010
	WSS-Password Type				SOAP Reque	est	Reque:	st Attach	iments (1)
	Inline Attachments	false							
	Expand MTOM Attac	talse	1	11					

8. Click the green arrow to send the request. The server's response will appear in the right pane

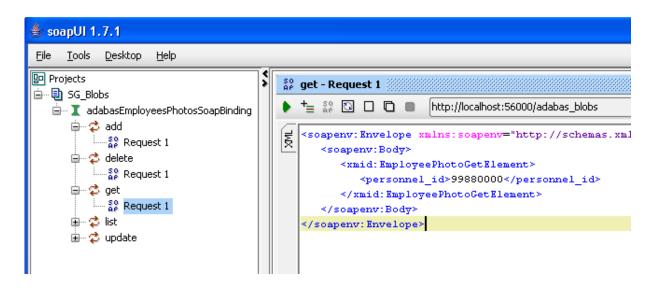
👙 soapUl 1.7.1	
<u>File T</u> ools <u>D</u> esktop <u>H</u> elp	
Projects SG_Blobs I adabasEmployeesPhotosSoapBinding </td <td><pre>     ## add - Request 1</pre></td>	<pre>     ## add - Request 1</pre>
Request Properties	
Property         Value           Name         Request 1           Encoding         UTF-8           Endpoint         http://localhost:560           Enable MTOM/Inline         true	
Username Password Domain	Add file
WS5-Password Type Inline Attachments false Expand MTOM Attac false	SOAP Request R SOAP Response Response Attachments (0) HTTP Headers (5) SSL Info (-) 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.

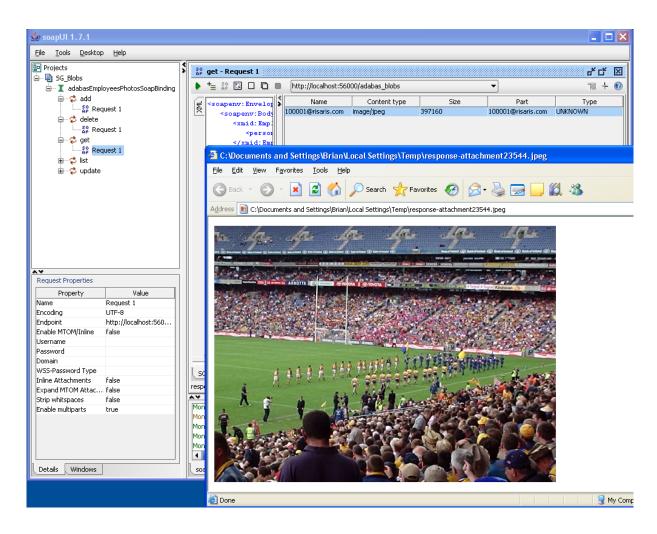
10. Add the personnel\_id of the record that you wish to retrieve, for example 99880000.



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** 



12 You may delete the record, including the LOB, by selecting the delete operation and entering the corresponding personnel\_id.

## 9 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:

wsf\_lobsGet: Get a record, save the LOB to a file

# **10** 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'll 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 Web Service.

Enter the following information

- Name: adabasTutorial\_blobs
- DataView : Select 'adabas\_photoblobs', which is one of the sample definitions delivered with Portus
- DatabaseId: 212 ( or the DBID relevant on your system )
- FileNumber: 90

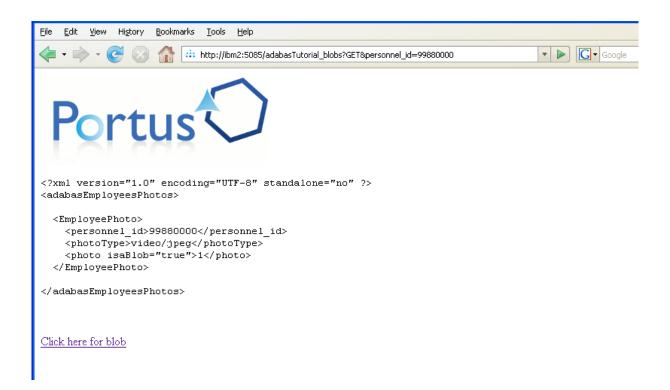
Publish your changes to the server.

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



- 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.



# 11 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**.

File	Edit View Insert Format	Tools Table Help
2	Fill Out a Form	2 🗉 🗷 🖄 🖉 📄
2	Design a Form	■ 結・1 油・油・油・
3	Open	Ctrl+O
	Close	
2	Save	Ctrl+S
	Save <u>A</u> s	
	Export To	•
	1 C:\Documents and Settings\Bria	n\D\updateRecord
	2 C:\Documents and Settings\Bria	n\D\deleteRecord
	<u>3</u> C:\Documents and Settings\Bria	n\Desk\getRecord
	4 C:\Documents and Settings\Bria	n\Des\listRecord
	Exit	
	*	

 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 Wiz	ard 🛛 🔀
	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: O Database (Microsoft SQL Server or Microsoft Office Access only) O Web service
	< Back Next > Cancel

Select Receive and submit data, click Next

Data Connection Wiz	ard 🛛 🔀
	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
	< <u>Back</u> <u>Next</u> Cancel

Enter the URL the WSDL for the "Employees" demo file is exposed as: *http://soagate:8885/ada-bas\_Employees?WSDL* 

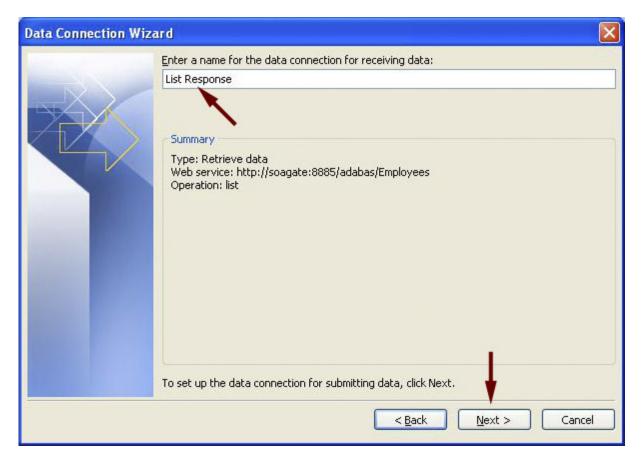
Click Next

Data Connection Wiz	ard 🛛 🔀
Data Connection Wiz	ard       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
	< <u>B</u> ack <u>N</u> ext > Cancel

Select the **list** method, click **Next** 

Data Connection Wi	zard	
	that your form receives. Select an <u>o</u> peration:	I has the following operations for providing the XML data
	update add delete get list	
		< Back Next > Cancel

Assign a name to the response document, click Next



Enter the same URL again, click Next

Data Connection Wiz	ard 🛛 🔀
	Web service details (submit data) Use these options to specify the Web service to which you submit data.
	http://soagate:8885/adabas_Employees?WSDL
	Example: http://www.contoso.com/Service.asmx?WSDL
	< <u>B</u> ack <u>N</u> ext > Cancel

Choose list and click Next

Data Connection Wiz	ard	
		ed has the following operations for submitting your XML data Description of operation:
		< Back Next > Cancel

Select "Entire form (XML document....", click Next

Data Connection Wize	ard		X
	fields or groups in you		s the following parameters. Specify which these parameters. If the Web service can specify that as well.
	Parameter	Туре	Element
	:personnel_id	string	1
	:name :city	string string	
	Parameter options Submit the following	data for the selected paran	neter:
Sector Sector	O Field or group:		
	Include:	Text and child elements or	ly 💽
	⊙ Entire form (XML	document, including proces	sing instructions)
	Submit <u>d</u> ata as a Note: Digitally si		d as a string to preserve white spaces.
		_	Back Next > Cancel

Assign a name to the Send Connection, click Finish

Data Connection Wize	ard	×
	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	
	< Back Finish Cancel	

3. You will now be presented with a form

😰 (Design) Template1 - Microsoft Office InfoPath 2003		
Elle Edit View Insert Format Icols Table Help	Type a question for	r help 🕞
🔛 🗃 🔒 I 🔍 Preview Form I 👍 🔍 ỡ I 🐰 😳 🕵 🛷 I 🤊 🕓 🕼 🗔 🗔 🕼 Design Tasks I 🛞 💂		
44 Verdana 🔹 10 🔹 Β Ζ 🖳 🧮 🗮 🗮 📜 🖓 τι 🖂 τ 📜 τ 📜 τ 🚝 👘 τ Λ τ 💂		
📝 Draw Table 😭 No border 🔹 T 🕫 🔫 🥥 -   🏠 -   Ingert + 🖄 🖼 🛠 🛄 🖓 💂		
	Data Source	▼ ×
Click to add a title	<b>O</b>   <b>O</b>   <b>A</b>	0.000
Click to add form content	E Layout	
	첫 Controls 웹 h Data Source	
Drag query fields here	Wews	
	Data source:	
	Main	~
	😑 汝 myFields	^
	😑 🥶 queryFields 😑 😂 asg:adabasEmployeeListElement	
Run Query	🔐 :personnel id	
	CP :name	
Drag data fields here	🔐 :dept	
	ijob_title	
	🖃 📴 asg:adabasEmployeesElement	
	😑 🔐 :adabasEmployees 🗃 🎯 adabasEmployee	
http://www.commence.com/commence.com/commence.com/commence.com/com/com/com/com/com/com/com/com/com/	🔐 personnel_id	
	🔐 first_name	
	ame	
	🔐 mar_stat	
	📑 sex	
	🔐 zip	
	P country	
	🔐 phone	
	📑 dept	
	P leave_due	
	📑 leave_taken	~
	Show details	
	Add	
	Help with the Data Source	

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"

Run Query											
Personnel Id First Nome	Middle Name	Name	Mar Stat	Sex City	Zip	Country	Area Code	Phone	Job Title	Leav c Due	Leave
Repeating Table											

4. Once you are happy with how your form looks like, select File -> Preview form -> Default

Edit View Insert Format					-						
Design a Form	2, 49   X Rad	2000		Design Tasks   🌚 薛   💇 • 🛕 •	-						
Preview Form 🕨 🔍 Defa	uit Ctrl+Shift+B	📒 🚍 🚛 🚛 -	恒・恒・課	谭 👻 • 🛓 •							
Glose With	Data <u>F</u> le	👻   Ingert 👻 🏢	1 頭 五 🗐	3							
	User <u>R</u> ole			_							
S											
st Employees											
ick to add form content											
2000											
Personnel Id:											
Name:											
City: Dept:											
Job Title:											
Job Hitle:											
1000 C											
Section											
action											
Run Query											
ersonnel Id First Name	Middle	Name	Mar S	ex City	Zip	Country		Phone	Job Title	Leav	
	Name		Stat		osue.		Code			e Due	
Repeating Table											

5. 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:	

6. Your table will now be populated with the data based on your request:

Personnel Id	First Name	Middle Name	Name	Mar Stat	Sex	City	Zip	Country	Area Code	Phone	Dept	Job Title		Leave Taken
40000001	HANS	ERIK	JENSEN	W	М	KOEBENHAVN	1800	<u>RK</u>	01	215161	ADMA	SEKRETAER	30	25
40000007	CLAUS	JESPER	JENSEN	S	М	YALBY	2500	RK	01	465800	SYSA	PROGRAMMOER	30	13
40000012	NIELS	CHRISTIAN	HANSEN	D	М	KOEBENHAVN	1364	<u>RK</u>	01	111122	ADMA	REVISOR	30	15
40000014	KARL		SOERENSEN	М	М	KOEBENHAVN	2200	<u>RK</u>	01	370252	SYSA	PROGRAMMOER	30	18
40000037	JONAS	KIM	ERIKSSEN	S	М	KOEBENHAVN	1019	<u>RK</u>	01	132526	SYSA	DB-ADMINISTRATOR	30	20
40000038	ANITA	KARINA	ANDERSEN	S	F	KOEBENHAVN	1850	<u>RK</u>	01	316510	SALG	SEKRETAER	30	18
40000042	KARSTEN		FREDERIKSEN	S	М	S@BORG	2860	<u>RK</u>	01	561519	SYSA	SYSTEMCHEE	30	22
40000043	ELIN	GRETE	GREGERSEN	М	F	ODENSE	5000	<u>RK</u>	09	101211	SALG	SAELGER	30	20
40000044	KARIN	MARIE	ANDERSEN	М	F	VANLOESE	2720	<u>RK</u>	01	743919	SYSA	PROJEKTSEKRETAER	30	19
40000045	HANS	OLE	MADSEN	D	М	KOEBENHAVN	1609	<u>RK</u>	01	938106	MASK	OPERATOER	30	22
40000100	HANNE	IRENE	HANSEN	М	F	YALBY.	2500	<u>RK</u>	01	463149	SYSA	PROGRAMMOER	30	20
40000107	JOHNNY		BENTSEN	М	М	BROENDBY STR	2660	<u>RK</u>	02	541019	MASK	OPERATOER	30	12
40000110	PREBEN		PETERSEN	S	М	HVIDOVRE	2650	<u>DK</u>	01	471719	ADMA	EFG-ELEY	30	14
40000112	FINN		HOLGERSEN	м	М	KOEBENHAVN	2300	<u>DK</u>	01	548959	SYSA	PROGRAMMOER	30	16
40000114	SUSANNE	BENTE	RASMUSSEN	S	F	FREDERIKSBERG	2000	<u>RK</u>	01	244950	SYSA	SEKRETAER	30	14
40000124	HANS	EMIL	PETERSEN	м	М	ROEDOVRE	2610	DK	02	910206	SALG	SAELGER	30	16

Run Query

# 12 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)

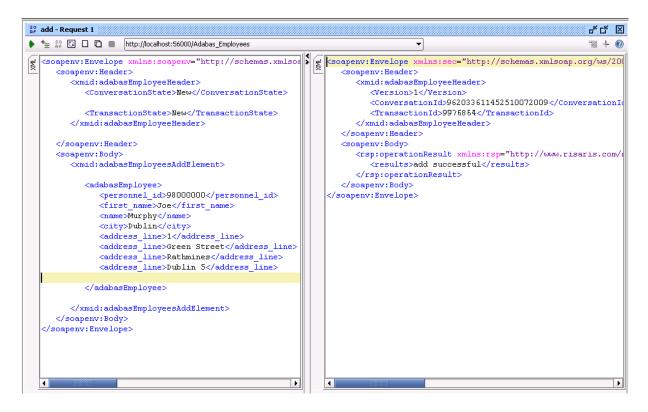
For this tutorial, we will use the Employees demo file (usually file number 11) that comes with Adabas, and the **adabas\_employees\_mini\_view** data view. It is assumed you have already created a web service for your Adabas file.



**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

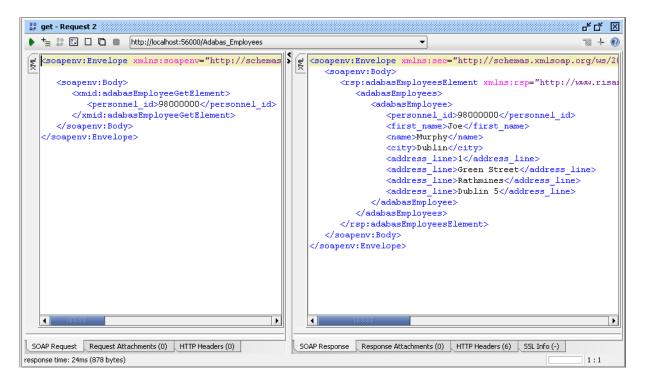


Portus has now created a new Conversation and a new Transaction for this request. The IDs for each of these are returned.

3. Verify that the record has been added successfully, (but not yet committed).

Choose a get request, remote the soapenv:Header element, and enter get the record.

E.g.



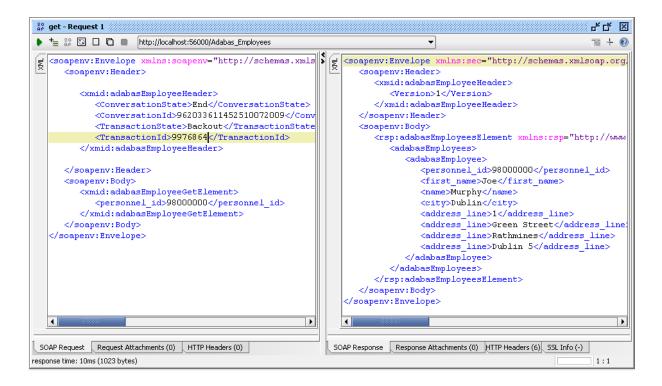
The record has been added, but not yet committed. Because Adabas' isolation level is "Read Uncommitted" (also known as "dirty read"), a request from a non-conversational request will still return the added, but-yet-uncommitted, record.

4. Now backout the transaction.

Choose a get request, and enter the following:

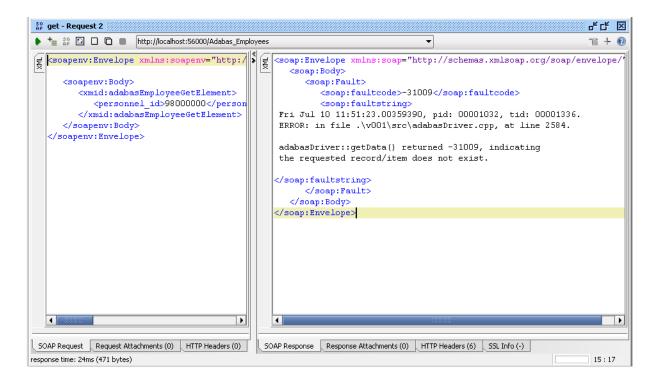
- ConversationState to End
- ConversationID to the value returned in the add response
- TransactionState to Backout
- TransactionID to the value returned in the add response

E.g



5. Now, if you re-run the request from Step 3, the item does not exist as the previous add has been backed out.

E.g.



# 

## Portus - Configuration Versioning with Eclipse and CVS

Introduction	76
Requirements	76
Example Setup	76
More Information	87

## Introduction

If Eclipse is used as the management interface to Portus then versioning of the ASG configuration files can be achieved easily with CVS. Eclipse provides CVS interfaces, so much of the following will be familiar to many users of Eclipse already.

Using CVS can also bring advantages including security, auditing control, redeployment facilitation and metadata management.

## Requirements

In order to take advantage of this option, you must have available :

- a server running CVS and to which you have access
- a CVS 'module' where you can add the ASG related files you wish to maintain

## **Example Setup**

- CVS Server Details
- CVS Module Selection
- Selecting a Location in Eclipse
- Adding Files to CVS
- Making CVS aware of file changes

Select File->Import from the Eclipse Menus and then choose "Checkout Projects from CVS".

Java - Eclipse SDK		_ <b>–</b> ×
<u>File</u> <u>E</u> dit Refac <u>t</u> or <u>S</u> ource <u>N</u> avigat	e Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
] 📬 🖩 👜 ] 🏇 🛛 🖉 🖨 Import 🎱	н н н	🗆 🗙 🖻 🖏 Java
Package Explorer S Select		Outline 🛛 🗖 🗖
Import a proj	ect from a CVS Repository.	n outline is not railable.
<u>S</u> elect an impo	rt source:	
ADABAS S	DA Gateway Configuration	
# ADABAS S	DA Gateway Resource Definition	
👰 Archive file		
🛒 Checkout P	ojects from CVS	
	jects into Workspace	
Sternal Fe		
	g-ins and Fragments	
📮 File system		
🗟 Team Proje	t Set	
	< Back Next > Einish Canc	el
	Revealed the second sec	₩ 券 ▽ - □
	0 errors, 0 warnings, 0 infos	
	Description Resource In Folder	Location

### **CVS Server Details**

Enter the details required for your CVS server.

🗣 Checkout from CVS 🎐	×
Enter Repository Location Information	
Define the location and protocol required to connect with an existing CVS repository.	
Location	٦
Host: cvsserver 🗸	
Repository path: /home/cvs/repository137	
Authentication	
User: cvsuser 🗸	
Password: ************************************	
Connection	
Connection type: pserver	
Ise default port	
○ Use por <u>t</u> :	
Save password	_
A Saved passwords are stored on your computer in a file that is difficult, but not	
impossible, for an intruder to read.	
Concel	1
< <u>B</u> ack <u>N</u> ext > Einish Cancel	J

### **CVS Module Selection**

Enter the name of the CVS module where you are to keep your ASG files, or select the module from a list.

🚔 Checkout from CVS 🎐	
Select Module	CVS
Select the module to be checked out from CVS	=0=
Use <u>specified</u> module name: AsgServer102	
$\bigcirc$ Use an existing module (this will allow you to browse the modules in the rep	ository)
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Car	ncel

### Selecting a Location in Eclipse

Select where you wish to check out the CVS module to. You may place it in various locations. If you use an existing Eclipse project, the module will appear as a folder within it. This example creates a new Simple Project.

😂 Check Out As 🍚	
Check Out As Select the method of check out	CVS
Choose how to check out folder 'AsgServer102'	
<ul> <li>Check out as a project <u>configured</u> using the New Project Wizard</li> <li>Check out as a project in the workspace</li> </ul>	
Project Name: AsgServer102	
<ul> <li>Check out into an existing project</li> </ul>	
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Ca	ancel

🗧 New Project 🎐	
Select a wizard	
Create a new project resource	
<u>W</u> izards:	
▷ 🗁 C++	
▷ 🗁 CVS	
Eclipse Modeling Framework	
🖻 🗁 Java	
Plug-in Development	222
▽ 🧁 Simple	
🖆 Project	•
	<u></u>
< <u>Back</u> <u>Next</u> > Einish Car	ncel
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Car	ncel

Check Out As	
🗧 New Project 🎐	
Project	
Create a new project resource.	
Project name: asgTest	
Project contents	
🗷 Use <u>d</u> efault	
Directory: /home/dm/eclipse/workspace/asgTest	Browse
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

The New Project is now shown in the Eclipse Package Explorer Window, the icon denoting it as a CVS item, and the server name shown at the end.

🗧 Java - Eclipse SDK 🍭				_	
<u>File E</u> dit Refac <u>t</u> or <u>S</u> ource <u>N</u> avigate	Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indo	w <u>H</u> elp			
] 📬 🖩 👜 ] 🏇 • 💽 • 🏊   🖄 🖶 🞯 • ]	🗠 جې چې 🖉 😵 🕲			🗈 🖏 Java	
I Package Expl × ×1 □□			- 8	🗄 Outline 🛿	
				An outline is not	
▶ ➡ >asgTest [cvsserver]				available.	
	olems 🛱 Javadoc Declaration	ı		₩ 눩 🏹	
	s, 0 warnings, 0 infos	1	1		
Des	cription Resource	In Folder	Location		
asgTest					Ğ

When you import items from ASG, you can now save them in this Project.

ᅌ Java - Eclipse S	DK					_	
Eile Edit Refactor		iaate Searc	ch Proiect Run Window H	lelp		)	
] 📬 🖩 👜 🖄 🆘 O						🗈 🖏 Java	
🛱 Package Expl	ADABAS SO	A Gateway	Configuration Import	ADF	abas	utline 🛿	
\$ \$ \$	Please sele	ect a valid De	estination Folder			utline is not lable.	
▶ 🛱 > asgTest [cvs	Host/IP:	asgServer1	02			lable.	
	Port:	80					
	Destination	?		Brows			
			🗧 Folder Selection 🍭				
			Select container to import S	SOA configuration to			
				i on contigui attori to			
			asgTest [cvsserver]				
					Cancel	₩ ‡ ▽	
	-					-	
asgTest			OK	Cancel			Ő

### Adding Files to CVS

Note that your imported files will need to be explicitly added to the CVS repository. Also note that you will have to select the file type when adding it to CVS. This would normally be ASCII TEXT and not the default of Binary.

🗧 Java - Eclipse SDK 🍭		_	
<u>File Edit Refactor Source Navigate Search Project Run Wir</u>	ndow <u>H</u> elp		
│ 🗂 • 🖩 👜 │ 🏇 • ⊘ • Q₄ • │ 🤔 🛱 ঔ • │ 🅭 🔗 │ 🎱 │ 🎨 ⇔ • ⇔ •		🗈 🖏 Java	
Hackage Explo x x □	- 8	Outline ន	
4 4 Q E 4 V		n outline is not	
▽ <sup>™</sup> <sub>6</sub> >asgTest [cvsserver]	av	ailable.	
asgServer102_80.xmc			
Ne <u>w</u>			
Open F3 Open With			
Eopy Ctrl+C     Paste Ctrl+V			
× Delete Delete			
Build Path			
Refac <u>t</u> or Shift+Alt+T	Synchronize with Repository		
i Import	<u>C</u> ommit		
La Export	Update		
🛷 Re <u>f</u> resh F5	Create <u>P</u> atch		
<u>R</u> un As	Apply Patch		
Debug As	Tag as Version		
Team			
Comp <u>a</u> re With Replace With	Switch to Another Branch or Version	× +	
Properties Alt+Enter	Show Annotation		
	Show Resource History		
Open As 🔸	Add to <u>V</u> ersion Control		
	Add to .cvsignore		
asgServer102_80.xmc - asgTest	Change ASCII/Binary Property		ć.

### Making CVS aware of file changes

Every time you make a change to your ASG related files, you should commit those changes to the CVS repository.

🗧 Java - Eclipse SDK 🎐					-	- 🗆 ×
<u>F</u> ile <u>E</u> dit Refac <u>t</u> or <u>S</u> ource <u>N</u> a	avigate Se <u>a</u> rch <u>P</u> ro	oject <u>R</u> un <u>W</u> indow	<u>H</u> elp			
] 🖸 • 📓 👜 ] 🏇 • 💽 • 🏊 ] 🖑 🗄	î G•   🍅 🛷   🥹	****		E 🖏	Java	
増 Package Explo ន 🔭 🗖			- 8	🗄 Outline 🛙	x	- 0
				An outline i available.	is not	
¬ □ asgTest [cvsserver]				avallable.		
>asgServer102_80.xmc	Ne <u>w</u>	•				
-						
	<u>O</u> pen Open Wit <u>h</u>	F3				
		Ctrl+C				
	Paste	Ctrl+V				
	X Delete	Delete				
	<u>B</u> uild Path	•				
	Refac <u>t</u> or	Shift+Alt+T 🔸				
	🔄 Import					
	🗳 Exp <u>o</u> rt					
	Re <u>f</u> resh	F5				
	<u>R</u> un As	•				
	<u>D</u> ebug As	•				
	T <u>e</u> am	•	Synchronize with Repository		≰ ≱	▼ ■ □
	Comp <u>a</u> re With Rep <u>l</u> ace With	•	<u>C</u> ommit Update		→1 <sup>2</sup>	
	Properties	Alt+Enter	Create Patch			
			Apply Patch			
	Open As	•	🐐 Tag as Version			
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )	1		☆ <u>B</u> ranch			
asgServer102_80.xmc -			☆ Merge Switch to Another Branch or `			ć

A useful comment should be added during the commit describing the change(s) made. Ticket Id's and other related information can be added here also.

🗢 Commit Files 🎐	
Commit	
Enter a comment for the commit operation.	CVS
Changes made to the Adabas SOA Gateway configuration. Added information the "Employees" file on the test DB.	for
	4
<choose a="" comment="" entered="" previously=""></choose>	•
Changes	E E 🔁
¬	
asgServer102_80.xmc (ASCII -kkv)	
<u> </u>	ncel

That should cover the basics required to use Eclipse in conjunction with CVS so that changes to your ASG configuration files are more controlled. Further information on the use of CVS with Eclipse, and how acquire and set-up a CVS server, can be found at the external sites listed below.

## **More Information**

More information related to CVS can be found at http://www.nongnu.org/cvs/

More information related to CVS use with Eclipse can be found in the Eclipse documentation at http://www.eclipse.org/

## **14** 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();
```

## 15 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 """
 * tags are required to format the object nicely.
  */
```

```
echo "";
print_r($soapfault);
echo "";
return;
/*
* Without any knowledge of, or looking at Adabas definitions, a "WebService ↔
programmer"
* can easily retrieve the signature of any exposed Adabas Portus Service and
* code based on it.
*
* First we use a method of the SoapClient class to retrieve the functions exposed
* by the specific "WebService", and print it:
*/
echo "<B>Functions:</B>\n\n";
print_r($soapclient->__getFunctions());
/*
* The function prototypes shown by the __getFunction() method also depict the \leftrightarrow
required
* parameters and the return values. Their definitions can be printed as well:
*/
echo ↩
                                             -----\n\n<B>Type ↔
"\n\n-----
                      Definitions:</B>\n\n":
print_r($soapclient->__getTypes());
echo "";
/*
* Proceed to ex02_SoaGatewayEmpList.php - List and format Employees data
*/
?>
```

# 16 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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
return;
/*
* Step 4: Format the Employee records nicely into a HTML table.
*/
echo "";
echo "Personnel IdNameFirst NameCity<th ↔
width=200>Address Line";
/*
* Loop through all "adabasEmployee" elements, creating a table row for every \leftrightarrow
single one
*/
if ( isset($listresponse->adabasEmployees->adabasEmployee) )
$Employees = $listresponse->adabasEmployees->adabasEmployee;
if (!is_array($Employees))
 $Employees = $listresponse->adabasEmployees;
foreach ($Employees as $Employee) {
 echo "$Employee->personnel_id$Employee->name",
  "$Employee->first_name$Employee->city";
 echo "";
/*
 * Format the "address_line" element (a MU(ltiple value field)) as a table within ↔
the table
 */
 if (!is_array($Employee->address_line)) {
  echo "$Employee->address_line";
 } else {
  foreach ($Employee->address_line as $addr) {
   echo "$addr";
  }
echo "";
echo "";
/*
   Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee record
```

### \*/ ?>

## 17 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 $soapfault) {
 echo "";
 print_r($soapfault);
 echo "";
 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' => '',
 \operatorname{'city'} \Rightarrow \operatorname{''};
/*
* 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 $soapfault) {
echo "";
 print_r($soapfault);
 echo "";
 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 $soapfault) {
 echo "";
 print_r($soapfault);
 echo "";
 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 "";
```

```
echo "Personnel IdNameFirst NameCity<th ↔
width=200>Address Line":
/*
* Loop through all "adabasEmployee" elements, creating a table row for every \leftrightarrow
single one
 */
if ( isset($listresponse->adabasEmployees->adabasEmployee) )
$Employees = $listresponse->adabasEmployees->adabasEmployee;
if (!is_array($Employees))
 $Employees = $listresponse->adabasEmployees;
foreach ($Employees as $Employee) {
 echo ↩
echo "";
 if (!is_array($Employee->address_line)) {
  echo "$Employee->address_line";
 } else {
  foreach ($Employee->address_line as $addr) {
  echo "$addr";
  }
 }
 echo "":
 echo "";
}
echo "";
}
/*
*
  Proceed to exO3_SoaGatewayEmpAdd.php - Add a new employee record
*/
?>
```

# 18

## 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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
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' => '',
 \operatorname{'city'} \Rightarrow \operatorname{''};
/*
* 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 $soapfault) {
echo "";
 print_r($soapfault);
 echo "";
 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 $soapfault) {
 echo "";
 print_r($soapfault);
 echo "";
 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 "";
echo "Personnel IdNameFirst NameCity<th ↔
width=200>Address Line";
/*
 * 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 $Employee) {
 echo ↩
"td>$Employee->personnel_idtd>$Employee->name$Employee->first_nametd>$Employee->citytd>*td>*;
 echo "";
 if (!is_array($Employee->address_line)) {
  echo "$Employee->address_line";
 } else {
  foreach ($Employee->address_line as $addr) {
   echo "$addr";
  }
 }
 echo "";
 echo "";
 }
echo "";
/*
* Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee record
*/
?>
```

# 19

## ex06\_SoaGatewaySpecial\_SubDescriptor.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 $soapfault) {
echo "";
 print_r($soapfault);
 echo "";
 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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
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 "";
echo "Personnel IdNameFirst ↔
NameCityDepartment";
/*
* Loop through all "adabasEmployee" elements, creating a table row for every ↔
single one
*/
foreach ($listresponse->adabasDemoEmployeesSpecial->demoEmployeeSpecial as $Employee) ↔
echo "$Employee->personnel_id$Employee->name",
"$Employee->first_name$Employee->city$Employee->department_code";
echo "";
/*
*
   Proceed to exO3_SoaGatewayEmpAdd.php - Add a new employee record
*/
?>
```

# 20

## ex06\_SoaGatewaySpecial\_SuperDescriptor.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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
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' \Rightarrow "".
 '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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
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 "":
echo "Personnel IdNameFirst ↔
NameCityDepartment";
/*
* Loop through all "adabasEmployee" elements, creating a table row for every \leftrightarrow
single one
*/
foreach ($listresponse->adabasDemoEmployeesSpecial->demoEmployeeSpecial as $Employee) ↔
echo "$Employee->personnel_id$Employee->name",
"$Employee->first_name$Employee->city$Employee->department_code";
echo "";
/*
*
   Proceed to exO3_SoaGatewayEmpAdd.php - Add a new employee record
*/
?>
```

## 21 ex03\_SoaGatewayEmpAdd.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 "result: $Adabasresponse->results";
/*
* The SOAPFault is handled in a function:
*/
function printSoapFault ($soapclient, $soapfault) {
echo "";
echo "\n\nSoap Fault occurred\n\nFaultCode..: ↩
".$soapfault->faultcode."\nFaultString: ".$soapfault->faultstring;
echo "";
}
/*
* Proceed to exO4_SoaGatewayEmpGet.php - Get and display the newlyadded employee \leftrightarrow
record
*/
?>
```

## 22 ex04\_SoaGatewayEmpGet.php

```
/*
* The "get" operation is even easier.
*/
try{
$soapclient = new SoapClient("http://localhost:8022/adabas_EmployeesMini?WSDL", ↔
array('trace' => 1));
} catch (SoapFault $soapfault) {
echo "";
print_r($soapfault);
echo "";
/*
* 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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
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 "";
print_r($Adabasresponse);
echo "";
/*
 * Proceed to ex05_SoaGatewayEmpDel.php - Delete the record added in ↔
ex03_SoaGatewayEmpAdd.php
 */
?>
```

## 23 ex05\_SoaGatewayEmpDel.php

```
/*
* Finally we delete the Employee record with personnel_id=999999999 again
*/
try{
$soapclient = new SoapClient("http://localhost:8022/adabas_EmployeesMini?WSDL", ↔
array('trace' => 1));
} catch (SoapFault $soapfault) {
echo "";
print_r($soapfault);
echo "";
/*
* 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 $soapfault) {
echo "";
print_r($soapfault);
echo "";
return;
/*
* Print the response
*/
```

```
echo "";
print_r($Adabasresponse);
echo "";
?>
```

## 24 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) \leftrightarrow
and format
```

```
* them into a HTML table. The code used here is, with the exception of the \leftrightarrow
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 "";
echo "Personnel IdNamefirst NameCity<td ↔
width=200>Address":
if ( isset($result->adabasEmployees->adabasEmployee) )
$Employees = $result->adabasEmployees->adabasEmployee;
if (!is_array($Employees))
 $Employees = $result->adabasEmployees;
foreach ($Employees as $Employee) {
 echo ↔
"td>$Employee->personnel_idtd>$Employee->nametd>$Employee->first_nametd>$Employee->citytd>td>";
 echo "";
 if (!is_array($Employee->address_line)) {
  echo "$Employee->address_line";
 } else {
  foreach ($Employee->address_line as $addr) {
   echo "$addr";
  }
 }
 echo "";
 echo "";
 }
echo "";
```

### } ?> </body></html>

## 25 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 "";
```

```
print_r($soapfault);
 echo "";
 return;
 }
}
$msg = "";
if ($_POST['submit'] == "Delete") {
try {
$Adabasresponse = $soapclient->delete(array('personnel_id' => $Persid));
} catch (SoapFault $soapfault) {
echo "";
print_r($soapfault);
echo "";
return;
}
$msg = $Adabasresponse->results;
}
if ($_POST['submit'] == "Get") {
try {
$Adabasresponse = $soapclient->get(array('personnel_id' => $Persid));
} catch (SoapFault $soapfault) {
echo "";
print_r($soapfault);
echo "";
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 $soapfault) {
  echo "";
  print_r($soapfault);
  echo "";
 }
 } else {
 trv {
  $Adabasresponse = $soapclient->update($adabasEmployees);
 } catch (SoapFault $soapfault) {
  echo "";
  print_r($soapfault);
  echo "";
 }
}
 }
} 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 />
      .....: <input type="text" size="20" ↔
maxlength="20" name="Addr[]" value="<? echo $Addr[1]; ?>"><br />
      .....: <input type="text" size="20" ↔
```

```
maxlength="20" name="Addr[]" value="<? echo $Addr[2]; ?>"><br />
      ....: <input type="text" size="20" ↔
maxlength="20" name="Addr[]" value="<? echo $Addr[3]; ?>"><br />
<br/>
<? echo $msq: ?><br><br>
<input type="submit" value="List" name="submit">
<input type="submit" value="Get" name="submit">
<input type="submit" value="Add" name="submit">
<input type="submit" value="Update" name="submit">
<input type="submit" value="Delete" name="submit">
<input type="submit" value="Reset" name="submit">
</form>
<?
if (isset($_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 $soapfault) {
 echo "";
 print_r($soapfault);
 echo "";
 return;
 }
echo "";
echo "Personnel IdNameFirst NameCity<th ↔
width=200>Address Line":
if ( isset($listresponse->adabasEmployees->adabasEmployee) )
 $Employees = $listresponse->adabasEmployees->adabasEmployee;
 if (!is_array($Employees))
  $Employees = $listresponse->adabasEmployees;
 foreach ($Employees as $Employee) {
  echo ↔
"td>$Employee->personnel_idtd>$Employee->nametd>$Employee->first_nametd>$Employee->citytd>td>";
  echo "";
  if (!is_array($Employee->address_line)) {
   echo "$Employee->address_line";
  } else {
```

```
foreach ($Employee->address_line as $addr) {
```

```
echo "$addr";
    }
    echo "";
    echo "";
    }
}
}
}
```

</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
```

## 27 ASGDemo.cs

```
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);
     }
}</pre>
```

## 28 MySQL\_City.cs

```
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);
```

}

}

# **29** FILE 90 (LOB demo file) FDT

ADACMP COMPRESS FILE=90 ADACMP FNDEF='1 , AA, 8 , A ,DE, UQ ' ADACMP FNDEF='1 , LM,32 , A ,NU ' ADACMP FNDEF='1 , LB, 0 , A ,LB, NV '

# FILE 90 (LOB demo file) load parameters

Sample load parameters for the 'base' file	134	4
Sample load parameters for the 'LOB' file	13	5

Loading a LOB file involves two steps

- 1. loading the 'base' file
- 2. loading the LOB file

### Sample load parameters for the 'base' file

//DDKARTE DD \* ADALOD LOAD FILE=90 ADALOD NAME=ASG-PHOTOS ADALOD LOBFILE=91 ADALOD MAXISN=100 ADALOD DSSIZE=10B ADALOD UISIZE=5B ADALOD NISIZE=10B ADALOD INDEXCOMPRESSION=YES ADALOD ISNREUSE=YES ADALOD LWP=1024K ADALOD SORTSIZE=<sortsize> ADALOD SORTDEV=<sortdev> ADALOD TEMPSIZE=<tempsize> ADALOD TEMPDEV=<tempdev> /\*

### Sample load parameters for the 'LOB' file

//DDKARTE DD \*

ADALOD LOAD FILE=91

ADALOD NAME=ASG-PHLOBS

ADALOD BASEFILE=90

ADALOD MAXISN=100

ADALOD DSSIZE=100B

ADALOD UISIZE=5B

ADALOD NISIZE=10B

ADALOD ISNREUSE=YES

ADALOD LWP=1024K

ADALOD SORTSIZE=<sortsize>

ADALOD SORTDEV=<sortdev>

ADALOD TEMPSIZE=<temspize>

ADALOD TEMPDEV=<tempdev>

/\*

## **31** FILE90.FDT (LOB demo file FDT)

- 1 , AA, 8, A, DE, UQ ; personnel\_id
- 1 , LM, 32, A, NU ; mime type for LOB
- 1 , LB, O, A, NB,NV,LB ; LOB data

## **32** FILE90.FDU (LOB demo file load parameters)

dbid = 212 file = 90 name = photos lobfile= 91 dssize = 50b nisize = 10b uisize = 5b maxisn = 100 reuse = (isn,ds)

## 33 wsf\_lobGet.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</pre>

<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, $f);
echo "File saved as ".$imageName."";
}
if(strcmp($contentType,"image/gif") ==0){
$imageName = "C:\\TEMP\\".$i."."gif";
file_put_contents($imageName, $f);
echo "File saved as ".$imageName."";
}
if(strcmp($contentType,"audio/mpeg") ==0){
$imageName = "C:\\TEMP\\".$i."."mp3";
file_put_contents($imageName, $f);
echo "File saved as ".$imageName."";
}
}
}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 "";
print_r($e);
print_r($reqMessage);
echo "";
}
```



| NSGNATI - Portus Natural interface driver | 146 |
|---|-----|
| ASGENVIN - Display natural environment    | 146 |
| ASGECHON - Echo input                     | 147 |
| ASGCALN - Simple calculator               | 147 |

The following Natural samples demonstrate the interface between Portus and the Natural programming language

- NSGNATI Portus Natural interface driver
- ASGENVIN Display natural environment
- ASGECHON Echo input
- ASGCALN Simple calculator

### **NSGNATI - Portus Natural interface driver**

RESET #CODE (B4) INPUT #CODE CALL 'XMIDNATH' #CODE END

### **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

### **ASGCALN - 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 O THEN
        COMPUTE #FUNCTION-RESULT = #OPERAND-1 / #OPERAND-2
      ELSE
       MOVE O TO #FUNCTION-RESULT
      END-IF
    VALUE '%'
      IF #OPERAND-2 NE O THEN
        DIVIDE #OPERAND-1 INTO #OPERAND-2 GIVING #WORK-RESULT
        REMAINDER #FUNCTION-RESULT
      ELSE
       MOVE O TO #FUNCTION-RESULT
      END-IF
     NONE VALUE
       MOVE O TO #FUNCTION-RESULT
   END-DECIDE
```

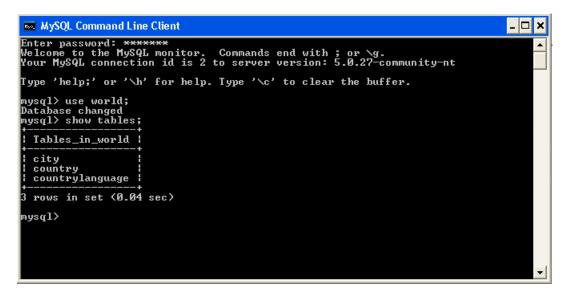
| Natural samples |  |  |  |
|-----------------|--|--|--|
|                 |  |  |  |
| *               |  |  |  |
| END             |  |  |  |

### 35 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. 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



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/ODB  | C 3.51.20 - Configure Data Sou      | rce Name 🔹 💽 🔀   |
|------------------|-------------------------------------|--|
| С                | onnector/ODBC                       | MySQL  |
| Login Connect O  | ptions Advanced                     | Connector/ODBC Configuration                             |
| Data Source Name | world_dsn                           | This dialog is used to edit a Data Source Name<br>(DSN). |
| Description      | World DB                            |  |
| Server           | localhost                           |  |
| User             | monty                               |  |
| Password         | •••••                               |  |
| Database         | world                               |  |
|                  | <u>T</u> est <u>D</u> iagnostics >> | Ok <u>C</u> ancel <u>H</u> elp                           |

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        | <pre>= /usr/lib/unixODBC/libmyodbc3.so</pre> |
| Setup         | <pre>= /usr/lib/unixODBC/libodbcmyS.so</pre> |
| 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.

### Legacy Perspective

| DataViews / XSDs / ) | KSLs] BusinessDataViews]  |   |   |  |
|----------------------|---------------------------|---|---|--|
| Driver               | Service                   | Vrs   | DataSource Id   | DataView   |
| MySQL_Dri            | world_dsn_city            | 1   | odbcDsn=world_dsn, tableName=city   | world_dsn_city_v1  |
| MySQL_Dri            | world_dsn_country         | 1   | odbcDsn=world_dsn, tableName=country  | world_dsn_country_v1   |
| MySQL_Dri            | world_dsn_countrylanguage | 1   | odbcDsn=world_dsn, tableName=countryla  | world_dsn_countrylanguage_v1   |
| ~                    | Driver<br>MySQL_Dri       | Driver Service Service MySQL_Dri world_dsn_city | Driver         Service         Vrs           MySQL_Dri         world_dsn_city         1           MySQL_Dri         world_dsn_country         1 | Driver         Service         Vrs         DataSource Id           MySQL_Dri         world_dsn_city         1         odbcDsn=world_dsn, tableName=city           MySQL_Dri         world_dsn_country         1         odbcDsn=world_dsn, tableName=country |

#### Administration Perspective

- 🔺 🚕 MySQL\_Driver
  - 🌆 world\_dsn\_city (v1)
  - 🌆 world\_dsn\_country (v1)

### Now select a tutorial

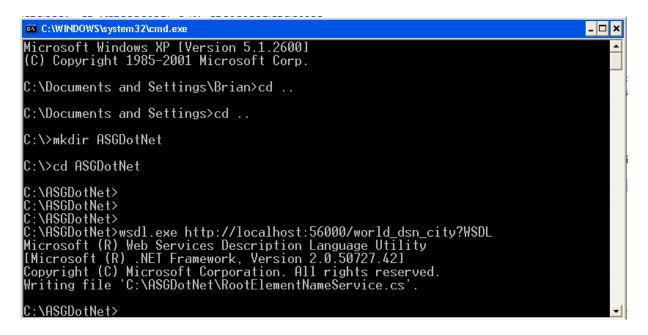
Using C#

# **36** 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 Portus Web Service to be exposed.



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 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):

| File | <u>Edit View T</u> ools Te <u>s</u> t | 1  | <u>W</u> indov | v <u>⊂</u> ommunit | y <u>H</u> elp     |
|------|---------------------------------------|----|----------------|--------------------|--------------------|
|      | New                                   | •  |                | Project            | Ctrl+Shift+N       |
|      | <u>O</u> pen                          | ∢  | ۵              | Web Site           |                    |
|      | ⊆lose                                 |    | 1              | <u>F</u> ile       | Ctrl+N             |
| ŕ    | Close Solution                        |    |                | Project From       | Existing Code      |
| a    | Save Selected Items Ctrl+9            | i. | JL             | uuiu               | 2005               |
|      | Save Selected Items As                |    |                |                    |                    |
| 7    | Save All Ctrl+Shift+S                 | 5  |                |                    | Visual Studio Deve |
|      | Export Template                       |    |                |                    | The current news   |
| 1    | Page Setyp                            |    | 1              |                    | Environment and    |
| 3    | Print Ctrl+P                          |    |                |                    |                    |
|      | Recent Projects                       | •  | 1              |                    |                    |
|      | E <u>x</u> it                         |    | 1              |                    |                    |
|      | Open: Project                         | 19 | Veb Sit        | -e                 |                    |

Create a C# Console Application, assign a name to it, specify the storage location, click **OK** 

| New Project   |                                       | and the second state of the later of the second state of the secon |  |        | ? ×                  |
|---|---------------------------------------|--|--|--------|----------------------|
| Project types:  |                                       | Templates:   |  |        | 8-0-<br>8-8-<br>8-8- |
| <ul> <li>→ Visual C#</li> <li>→ Windows</li> <li>→ Office</li> <li>→ Databas</li> <li>→ Starter k</li> <li>→ Test</li> <li>◆ Other Langu</li> <li>◆ Other Project</li> <li>↔ Test Projects</li> </ul> | sviče<br>e<br>(its<br>ages<br>t Types | Visual Studio installed templa         Image: Console Application         Image: Console Application <t< th=""><th>ites<br/>Class Library<br/>Web Control Library<br/>Windows Service<br/>Crystal Reports Application</th><th></th><th></th></t<>  | ites<br>Class Library<br>Web Control Library<br>Windows Service<br>Crystal Reports Application |        |                      |
| A project for cre<br>Name:  | ating a command-li                    |  |  |        |                      |
| Location:   | C:\ASGdotnet                          | -  | •  | Browse | )                    |
| Solution Name:  | CityList                              | ٦<br>آ   | Z Create directory for solution  |        | 10                   |
|   |                                       |  | ок   | Cance  | el                   |

A skeleton class file has been generated into your project workspace, with the required class definition and an empty Main method

| Progra  | am.cs   |   |
|---------|---|---|
| CityLis | t.Program   | ~ |
| 🖯 usi   | ing System;   |   |
| usi     | ing System.Collections.Generic;   |   |
| Lusi    | ing System.Text;  |   |
| ⊟ nan   | mespace CityList  |   |
| T,      | class Program   |   |
| F       |   |   |
|         | Contraction and Contraction of Co |   |
| 白       | <pre>static void Main(string[] args)</pre>  |   |
|         | {   |   |
| -       | )   |   |
|         | }   |   |
| 332     | 50 B  |   |

3. First of all, import the generated proxy into the project, right-click on the project name, select **Add Existing Item** 

| Calification (Chattel) (1 anniant)                     |  |
|--|--|
| Solution 'CityList' (1 project)                        |  |
| E Build  |  |
| ne Rebuild   |  |
| Publish  |  |
| Add  |  |
| Add Reference Existing Item                            |  |
| Add Web Reference 🕍 New Folder                         |  |
| Set as St <u>a</u> rtUp Project 🔄 Windows <u>F</u> orm |  |
| Debug 🕨 🔠 User Control                                 |  |
| 🔏 Cut 🍇 🖓  |  |
| Paste Daste  |  |
| × Remove   |  |
| Rename   |  |
| Properties   |  |

select the RootElementNameService.cs proxy file you created earlier, and click Add

| Add Existing I                        | tem - CityList     |                 |            |       |         |                    | ? 🗙   |
|---------------------------------------|--------------------|-----------------|------------|-------|---------|--------------------|-------|
| Look in:                              | 🛅 ASGDotNe         | t               | <b>v</b> Ø | - 🔰 🔍 | X 📬 🔟 • | Tools <del>-</del> |       |
| Desktop<br>My Projects<br>My Computer | CityList           | tNameService.cs |            |       |         |                    |       |
|                                       | File <u>n</u> ame: |                 |            |       | ~       | E                  | Add 🚽 |
| -                                     | Files of type:     | Visual C# Files |            |       | ~       | Ca                 | ancel |

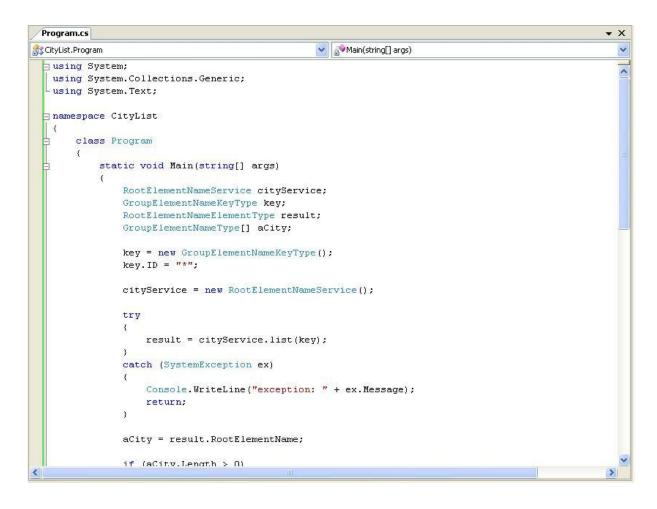
The proxy has been added to the project

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 | v2.0.501 |
| System.Web.RegularExpressions | 2.0.0.0 | v2.0.50  |
| System.Web.Services           | 2.0.0.0 | v2.0.501 |
| 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 | v2.0.501 |
| vjsvwaux                      | 2.0.0.0 | v2.0.50  |
| vjswfc                        | 2.0.0.0 | v2.0.50  |
| VjsWfcBrowserStubLib          | 2.0.0.0 | v2.0.50  |
| viswfrrw                      | 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\_CityDemo.cs to create your first C# program accessing MySQL Web Service via Portus.



5. Build the application. Right-click on the project name in the project explorer, click Build

| Solution Explorer -   | <b>-</b> ₽×   |   |
|-----------------------|---|---|
|                       |   |   |
| Carried Street Street | ist' (1 project)  |   |
| E (2 City)            | Build   |   |
| ⊡                     | R <u>e</u> build<br>Pu <u>b</u> lish                                |   |
|                       | A <u>d</u> d<br>Add <u>R</u> eference<br>Add W <u>e</u> b Reference | • |
|                       | Set as St <u>a</u> rtUp Project<br>Debu <u>g</u>                    | • |

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:\WINDOWS\system32\cmd.exe  |
|--|
| C:\ASGDotNet\CityList\CityList\bin\Release>  |
| C:\ASGDotNet\CityList\CityList\bin\Release><br>C:\ASGDotNet\CityList\CityList\bin\Release>                                     |
| C:\ASGDotNet\CityList\CityList\bin\Release>  |
| C:\ASGDotNet\CityList\CityList\bin\Release><br>C:\ASGDotNet\CityList\CityList\bin\Release>                                     |
| C:\ASGDotNet\CityList\CityList\bin\Release>  |
| C:\ASGDotNet\CityList\CityList\bin\Release><br>C:\ASGDotNet\CityList\CityList\bin\Release>                                     |
| C:\ASGDotNet\CityList\CityList\bin\Release>  |
| C:\ASGDotNet\CityList\CityList\bin\Release>CityList.exe  |
| Number of Citys returned: 11<br>City [0], ID=305 Name = Barueri CountryCode = BRA District = São Paulo Populatio<br>n = 208426 |
| City_ <u>[1]</u> ID=3050 Name = Malmö CountryCode = SWE District = Skåne län Population  |
| = 2595/9<br>City [2], ID=3051 Name = Uppsala CountryCode = SWE District = Uppsala län Popula<br>tion = 189569                  |
| City [3], ID=3052 Name = Linköping CountryCode = SWE District = East Götanmaan I<br>än Population = 133168                     |
| City [4], ID=3053 Name = Västerås CountryCode = SWE District = Västmanlands län<br>Population = 126328                         |
| City [5], ID=3054 Name = örebro CountryCode = SWE District = örebros län Populat<br>ion = 124207                               |
| City [6], ID=3055 Name = Norrköping CountryCode = SWE District = East Götanmaan<br>län Population = 122199                     |
| City [7], ID=3056 Name = Helsingborg CountryCode = SWE District = Skåne län Popu<br>lation = 117737                            |
| City [8], ID=3057 Name = Jönköping CountryCode = SWE District = Jönköpings län P<br>opulation = 117095                         |
| City [9], ID=3058 Name = Umeå CountryCode = SWE District = Västerbottens län Pop<br>ulation = 104512                           |
| City [10], ID=3059 Name = Lund CountryCode = SWE District = Skåne län Population<br>= 98948                                    |
| C:\ASGDotNet\CityList\CityList\bin\Release>  |

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 = "\*";

## **37** Usage Governance Tutorials

Usage Governance can be reported in 3 ways and written to 3 output types.

- **Using the local file system**
- Using another Portus
- Using MOM

## **38** 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.

| Portus Control Centre (Administration) - Eclipse SDK   |   | or some strong the out-that has been been   |  |
|--|---|---|--|
| File     Edit     Navigate     Search     Project     Bun     Window     Help       Image: Search     Image: Search     Image: Search     Image: Search     Image: Search     Image: Search       Image: Opentum 20     Image: Search     Image: Search     Image: Search     Image: Search       Image: Image: Search     Image: Search     Image: Search     Image: Search     Image: Search       Image: Image: Image: Search     Image: Search     Image: Search     Image: Search     Image: Search       Image: Image: Image: Image: Search     Image: Search     Image: Search     Image: Search     Image: Search       Image: Image: Image: Image: Image: Image: Search     Image: Image: Search     Image: Image: Search     Image: Search       Image: Imag | Properties  | ocalhost:56421) - v4.2.1.002000 ( Jul 12 2012 / 12:50:42 )  | 🖹 🔛 Portus (legac [ Portus Control ]   |
| DataYiews     BusinessDataViews  | General<br>Tracing<br>Global<br>Drivers<br>Governance<br>Statistics | Usage Governance settings      Basic  Information Max. File size  Output Data  Output Data  Governance Server Host Port | Messaging Use None PEntireX MQ EntireX MQSeries Broker Stub Broker ID Class Sener Senice |

- 3. Select the Save button.
- 4. Stop the server.See here on how to do this.
- 5. Start the server.
- 6. Issue a request to Portus e.g. a get request.
- 7. 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.

- 8. 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.
- 9. Open the file to check its contents.
- 10. There should be 1 entry for the get request plus all the governance data collected as per the options selected in 2.

### **39** Using another Portus

#### Tutorial: Writing Usage Governance data to another Portus

- 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.

| Portus Control Centre (Administration) - Eclipse SDK   |            | a search of the local sector of the second  |  |
|--|------------|---|--|
| File Edit Navigate Search Project Bun Window Help<br>File Edit Navigate Search Project Bun Window Help<br>Pottus Calibration Contents<br>Dotations<br>DataViews<br>BusinessDataViews | Properties | S3         ocalhost:56421) - v4.2.1.002000 ( Jul 12 2012 / 12:50:42 )         Usage Governance settings         • Basic         • Information       Max. File size         • Input Data         • Output Data         • Governance Server         Host         Port |  |

- 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    | Firefox *   |           |                |                         |                         |                 |                  |           | . II ×       |        |      |                         |       |                         |     |     |
|------------|---|-----------|----------------|-------------------------|-------------------------|-----------------|------------------|-----------|--------------|--------|------|-------------------------|-------|-------------------------|-----|-----|
| Risarís U  | td. SOA Gate  | eway Usag | ge Governanc   | +                       |                         |                 |                  |           |              |        |      |                         |       |                         |     | ÷   |
| <b>(+)</b> | localhost:56  | 5005/usag | egovernance?Ll | IST&UGGroup=*           |                         |                 |                  |           |              |        |      | ⊴ ≂ ୯ 🚼 - I             |       | ٩                       | ♠   |     |
|            | <b>Portus</b><br>Usage Governance Data  |           |                |                         |                         |                 |                  |           |              |        |      |                         |       |                         |     |     |
| Service    | Version   | Status    | Operation      | Start Time              | End Time                | Input<br>Length | Output<br>Length | Local IP  | Remote<br>IP | Method | User | Input Time              | Input | Output Time             | Out | put |
| MyCity     | 1   | Test      | list           | 2012-02-17T16:15:25.752 | 2012-02-17T16:15:25.752 | 27              | 8905             | 127.0.0.1 | 127.0.0.1    | GET    |      | 2012-02-17T16:15:25.752 | DATA  | 2012-02-17T16:15:25.752 | DA  | TA  |
| MyCity     | 1   | Test      | list           | 2012-02-17T16:15:36.297 | 2012-02-17T16:15:36.329 | 27              | 38935            | 127.0.0.1 | 127.0.0.1    | GET    |      | 2012-02-17T16:15:36.297 | DATA  | 2012-02-17T16:15:36.329 | DA  | TA  |
| MyCity     | 1   | Test      | list           | 2012-02-17T16:15:47.264 | 2012-02-17T16:15:47.295 | 27              | 41885            | 127.0.0.1 | 127.0.0.1    | GET    |      | 2012-02-17T16:15:47.264 | DATA  | 2012-02-17T16:15:47.295 | DA  | TA  |
|            | MyCity 1 Test list 2012-02-17T16:15:47.264 2012-02-17T16:15:47.295 27 41885 127.00.1   127.00.1   GET 2012-02-17T16:15:47.264 DATA 2012-02-17T16:15:47.295 DATA |           |                |                         |                         |                 |                  |           |              |        |      |                         |       |                         |     |     |

## 40 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.

| Portus Control Centre (Administration) - Eclipse SDK   | The Real and Article Provide Links   |   |
|--|--|---|
| Elle     Edit     Navigate     Search     Project     Bun     Window     Help       C     T     T     T     T     T     T     T     T       O     Portus     T     T     T     T     T     T | ⇔ ▼<br>■ Properties 🛙  | 🖹 🔝 Portus (legac 🚺 Portus Control  |
| ▲      ► localhost     →      ○ Services     □ DataViews     © BusinessDataViews   | localhost (localhost:56421) - v4.2.1.0020<br>General Usage Governance settin                                 |   |
|  | Tracing<br>Global<br>Drivers<br>Governance<br>Statistics<br>Governance<br>Governance<br>Fort<br>Host<br>Port | Messaging      Given and the size interval in the size interval interv |

- 4. Select the Save button.
- 5. Stop the server. See here on how to do this.
- 6. Start the server.
- 7. 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.

- 8. Issue a request to Portus e.g. a get request
- 9. 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.

10. 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 (Admini  | stration) - Eclip | se SDK  |  |
|---------------------------------------|-------------------|---|--|
| File Edit Navigate Search Project Run | Window Help       |   |  |
| 📬 • 🖫 🖷 🛆   🏊 •   😝 🔗                 | • ] 🖢 • 👌         | $* \diamondsuit \phi * \Rightarrow *$                     | 😭 🎯 Web  ြ Resource  |
| O Portus 🛛 🗖 🗖                        | Properties        | 3   |  |
| ✓ 48 8.4                              | local : Serv      | ce properties   |  |
| □ ► local<br>□ ♣ Services             | Properties        | ① Messaging properties for Service 'usagegovernance' (v1) | Switch back to service properties to update service definition |
| Adabas_Driver                         | Messaging         | ✓ Messaging   |  |
| DataViews     BusinessDataViews       |                   | Threads 1   |  |
| E. BusinessDataviews                  |                   | Use C None  C EntireX C MQ                                |  |
|                                       |                   | EntireX MQ Series   |  |
|                                       |                   | Broker Stub   |  |
|                                       |                   | Broker ID   |  |
|                                       |                   | Class   |  |
|                                       |                   | Server  |  |
|                                       |                   | Service   |  |
|                                       |                   |   |  |
|                                       |                   |   |  |
|                                       |                   |   |  |

- 11. 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.

- 12 As per the hint, switch back to Service properties and select the Save button when complete.
- 13. 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.

14. As soon as successful connections have been made, the input queue/service will be read and the message(s) processed.

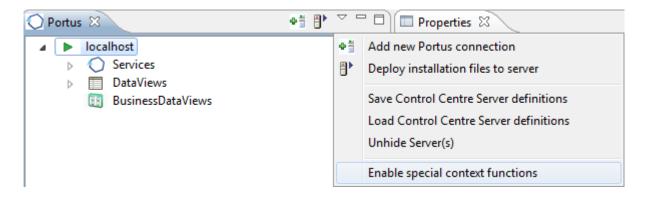
There are a couple of ways to check the status of this.

- 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.

# **41** Create the usagegovernance web service

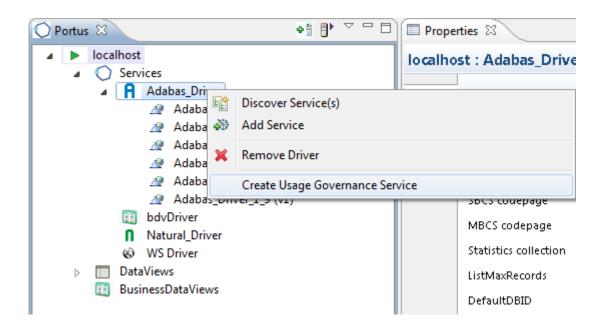
The usagegovernance web service is a bespoke service which is mandatory if usage governance data is being collected on a separate Portus. Here are the steps required to create it.

- 1. In the Control Centre select the server on which usage governance data will be stored.
- 2. Click on the View Menu icon and select the **Enable special context functions** entry.



3. Under Services select the driver which will support the new service. If one has not yet been created add one now. See here for details on this.

Right click on the driver entry and select the Create Usage Governance Service item.



4. Fill in the details appropriate to your installation.

Select the *Create database table now* check box.

Select Create.

| Please specify details for Usage Governance Service |                           |  |  |  |  |  |  |  |  |
|---|---------------------------|--|--|--|--|--|--|--|--|
|   | Create database table now |  |  |  |  |  |  |  |  |
| DatabaseId  |                           |  |  |  |  |  |  |  |  |
| FileNumber  |                           |  |  |  |  |  |  |  |  |
| FileName  |                           |  |  |  |  |  |  |  |  |
| MaxIsn  |                           |  |  |  |  |  |  |  |  |
| LobFileNumber                                       |                           |  |  |  |  |  |  |  |  |
| LobFileName   |                           |  |  |  |  |  |  |  |  |
|   | Create Cancel             |  |  |  |  |  |  |  |  |

5. The web service is now created and its properties displayed .

# 42 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.

| Portus Control Centre (Administration) - Eclipse SDK   | -   |   |
|--|---|---|
|  |   |   |
|  |   | 🖺 🔝 Portus (legac 🔘 P   |
| ) Portus 🛿 🔹 👘 🎽   | 🗖 🗖 Propertie   | s X   |
| <ul> <li>Iocalhost</li> <li>Services</li> <li>Adabas Driver</li> <li>bdvDriver</li> <li>www.MySQLDriver</li> <li>world.dsn.city (v1)</li> <li>Natural_Driver</li> <li>WS Driver</li> <li>DetaViews</li> <li>BusinessDataViews</li> </ul> | localhost<br>Properties<br>Messaging  | : Service properties<br>① Messaging properties for Service 'world_dsn_city' (v1) Switch back to service properties to update service definition         |
|  | <ul> <li>Edit Navigate Search Project Run Window He</li> <li>Contemportation of the search Project Run Run Run Run Run Run Run Run Run Run</li></ul> | Edit Navigate Search Project Bun Window Help  Portus  Portus  Adabas, Driver Body SQL Driver WySQL Driver WySQL Driver WySQL Driver WySDriver DataViews |

- 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 [SERV-ER\_INST]/Apache22/logs/error\_log replacing [SERVER\_INST] with the location in which you have installed Portus. Please correct and try again.

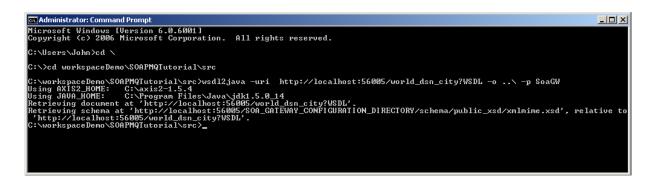
If successful then carry on to next step.



- 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. Right-click the "SOAPMQTutorial" project folder, select "Build Path", then "Add External Archives..."
- 7. Add all .jar files from the axis2 "lib" directory to the project's Build-Path.
- 8. Right-click the "SOAPMQTutorial" project folder, select "Build Path", then "Add External Archives..."

- 9. Add all the .jar files from your MQ lib directory e.g. ..\IBM\WebSphere MQ\java\lib
- 10. 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



- 11. 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.
- 12 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.
- 13. Get file mqTransports-0.1.jar and, when prompted, save it in a local directory.
- 14. Right-click on SOAPMQTutorial and select Build Path -> Add External Archives...

| Java - Eclipse SDK   |                            |                     |                                      |
|----------------------|----------------------------|---------------------|--------------------------------------|
| e Edit Source Refa   | ctor Navigate Search Proje | ect Run Window He   | lp                                   |
| 🔁 - 🖪 🖻 🛓            | ☆・○・♀・   🕸                 | 🞯 • ] 😕 🗁 🔗         | •】如 • 祠 • ← 今 • → •                  |
| 🖁 Package Explorer 🔀 |                            |                     |                                      |
| 🗉 🔂 SOAPMQTutorial   |                            |                     |                                      |
| 🚊 🔂 src              | New                        |                     | ▶                                    |
| 🖻 册 SoaGW            | Go Into                    |                     |                                      |
| 🕀 🕖 Fauli            | Open in New Window         |                     |                                      |
| 🕀 🔂 SOA              | Open Type Hierarchy        | F4                  |                                      |
| 🕀 🗹 🕀 🕀              | Show In                    | Alt+Shift+W         | • II                                 |
| 🗄 🛺 Worl             | SHOTTIN                    | Hier Shire Fit      | <u> </u>                             |
| 🗄 🛋 JRE System       | Сору                       | Ctrl+C              |                                      |
| 🗄 🛋 Referenced       | Copy Qualified Name        |                     |                                      |
| 🔤 🖹 build.xml        | 👕 Paste                    | Ctrl+V              |                                      |
|                      | 💢 Delete                   | Delete              |                                      |
|                      | € Remove from Context      | Ctrl+Alt+Shift+Down |                                      |
|                      | Build Path                 |                     | Link Source                          |
|                      | Source                     | Alt+Shift+S         | New Source Folder                    |
|                      | Refactor                   | Alt+Shift+T         |                                      |
|                      | Magnet                     |                     | — 🚝 Use as Source Folder             |
|                      | Export                     |                     | Add External Archives                |
|                      |                            |                     | Add Libraries                        |
|                      | 🔗 Refresh                  | F5                  | 🚵 Configure Build Path               |
|                      | Close Project              |                     |                                      |
|                      | Assign Working Sets        |                     |                                      |
|                      | Run As                     |                     | 🕨 🚺 Problems 🖉 Javadoc 😣 D           |
|                      | Debug As                   |                     | No consoles to display at this time. |
|                      | Profile As                 |                     | ►                                    |
|                      | Validate                   |                     |                                      |
|                      | Team                       |                     | •                                    |
|                      | Compare With               |                     | •                                    |
|                      | Restore from Local History | •                   |                                      |
|                      | Configure                  |                     |                                      |
|                      | Properties                 | Alt+Enter           |                                      |

Navigate to where mqTransports-0.1.jar was saved and select Open to add it the project.

15. Right-click on SOAPMQTut.java and select Run As -> Java Application:

| Java - SOAPMQTutorial/src/So  | aGW/SOAPMQTut.java - Ecl            | ipse SDK                   |  |
|---|-------------------------------------|----------------------------|--|
| File Edit Source Refactor Naviga  | ate Search Project Run V            | /indow Help                |  |
| ] 📫 • 🗒 🖻 🖻 🖄 • 🔘   |                                     |                            | ∦ □ □ ] 2 · 2 · 4 · 4 · 4  |
| 📕 Package Explorer 🛛  |                                     | 🚺 SOAPMQTut.java           | X  |
| SOAPMQTutorial  |                                     |                            | oaGW;<br>g.apache.axis2.description.Transpo<br>m.risaris.axis2.mq.transports.MQTr  |
| 🕀 🕖 World_dsn_cityRo  |                                     |                            | risaris.axis2.mq.transports.MQTr   |
| ⊞-ग्री World_dsn_cityRa<br>⊕-बो JRE System Library [Java<br>⊕-बो Referenced Libraries<br>श्री build.xml | LIDED                               | F3<br>F4<br>Alt+Shift+W    | <ul> <li>J.World_dsn_cityRootServiceStub;</li> <li>J.World_dsn_cityRootServiceStub.</li> <li>J.World_dsn_cityRootServiceStub.</li> <li>J.World_dsn_cityRootServiceStub.</li> </ul> |
|   | Copy                                | Ctrl+C                     | J.World_dsn_cityRootServiceStub.<br>J.World_dsn_cityRootServiceStub.   |
|   | 💼 Paste<br>💢 Delete                 | Ctrl+V<br>Delete           | <pre>s SOAPMQTut {     atic void main(String[] args) {</pre>   |
|   | ③ Remove from Context<br>Build Path | Ctrl+Alt+Shift+Down        | <pre>&gt; ioio ioia main(coring[] argo) ( &gt;</pre>   |
|   | Source<br>Refactor                  | Alt+Shift+S<br>Alt+Shift+T | <pre>ld_dsn_cityRootServiceStub stub</pre>   |
|   | ≥ Import<br>≧Export                 |                            | nsportOutDescription txOut = new   |
|   | References<br>Declarations          |                            | c  |
|   | 🔗 Refresh<br>Assign Working Sets    | F5                         |  |
|   | Run As<br>Debug As                  |                            | <ul> <li>I Run on Server</li> <li>2 Java Application Alt+Shift+X, J</li> </ul>   |
|   | Profile As<br>Validate              |                            | Run Configurations   |
|   | Team                                |                            | •  |

16. The output appears in the "Console" window:

