Accessing Adabas using the PHP SOAP extension

This tutorial

- Demonstrates how to access SOA Gateway from a PHP script
- Provides a number of PHP examples.

What is PHP ?

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

The PHP interpreter is available as source code or as pre-compiled binaries for major platforms, including most Linux™ distributions, Windows®, Mac OS X, and iSeries™.

The latest release is PHP 5 and is seeing increasing adoption. PHP 5 introduces improvements to the object model; also, the underlying memory management has been redesigned with multi-threading and performance in mind.

For more information about PHP, or to download the software, please refer to the PHP homepage.

New in PHP 5 is a built-in SOAP extension. It is supplied as part of PHP.

For this tutorial to work, you should have PHP 5 up and running in your Web server, see the install.txt document in the PHP distribution library for details.

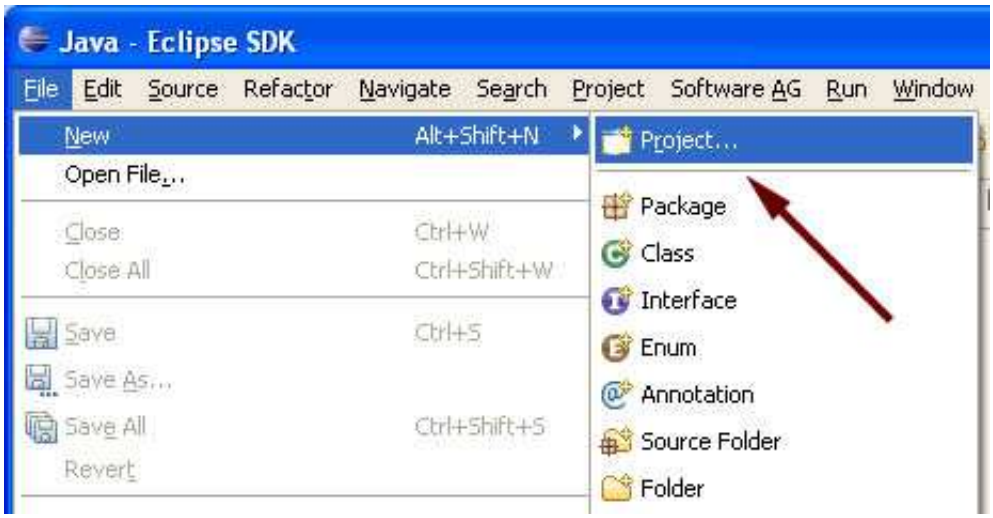
Installing the Eclipse PHP Development Tools

The Eclipse PHP Development Tools (PDT) are not absolutely necessary, but using Eclipse greatly simplifies the development process. This tutorial assumes the PDT to be installed.

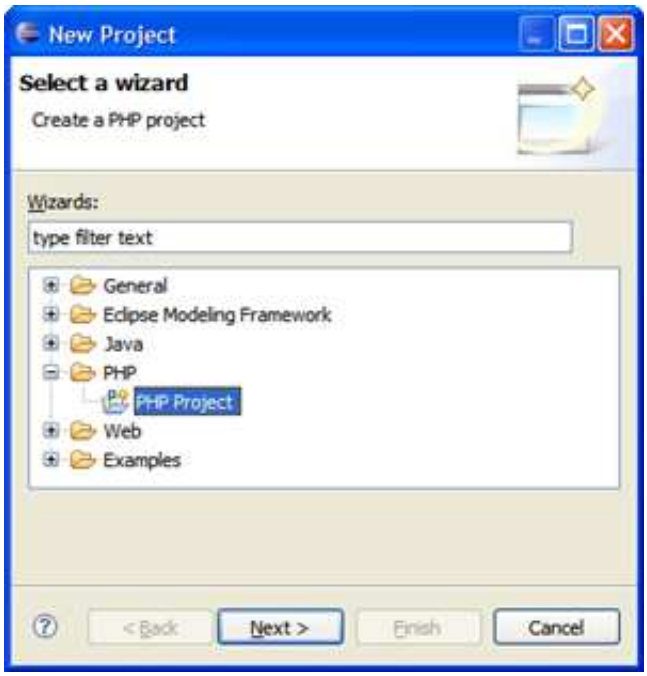
Please refer to the Eclipse PDT project pages for installation and configuration instructions.

Accessing Adabas from PHP

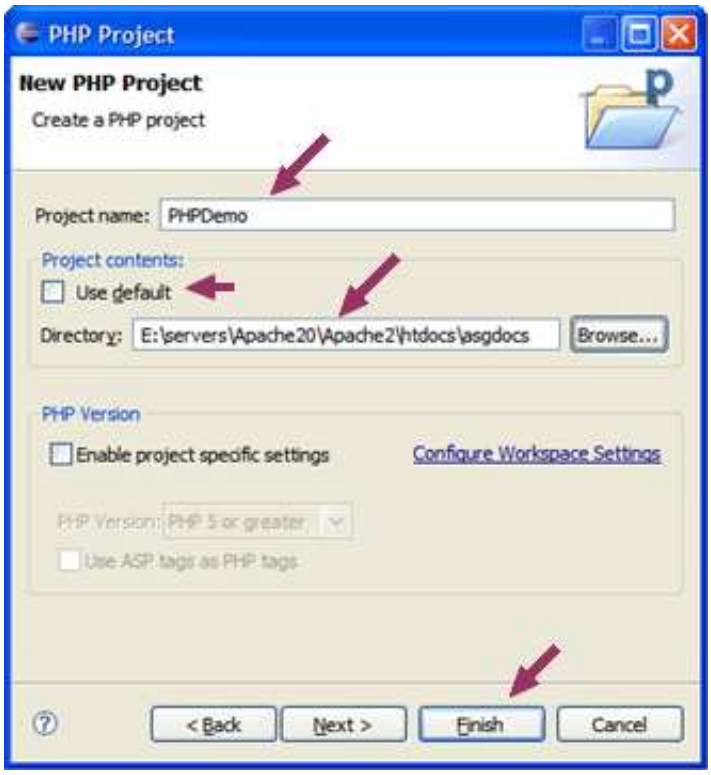
1. First, create a new project within your workspace.



2. Opt to create a PHP Project, click **Next**



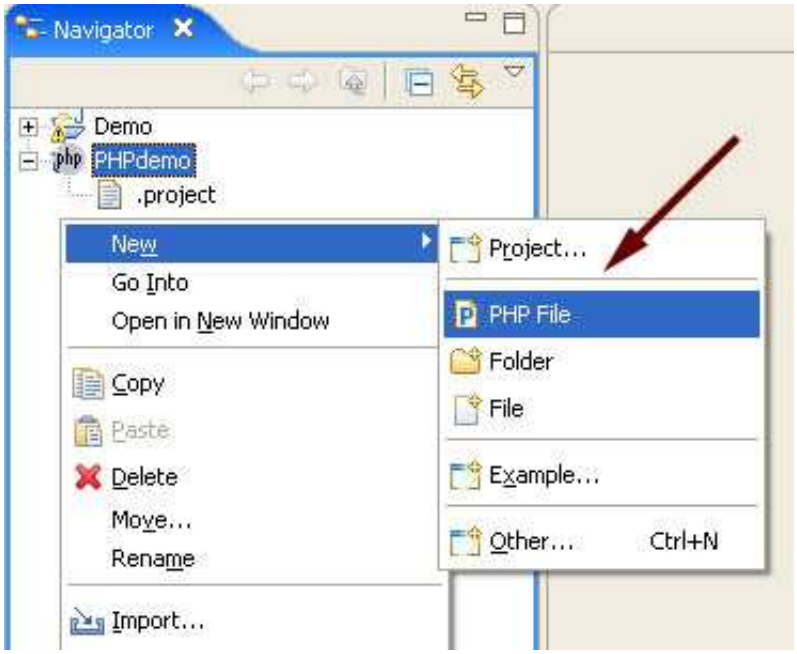
Give the project a name and unselect the "Use default" box. Browse to the default location of your html documents, in this case an Apache server document folder called "ASGdocs", click **Finish**



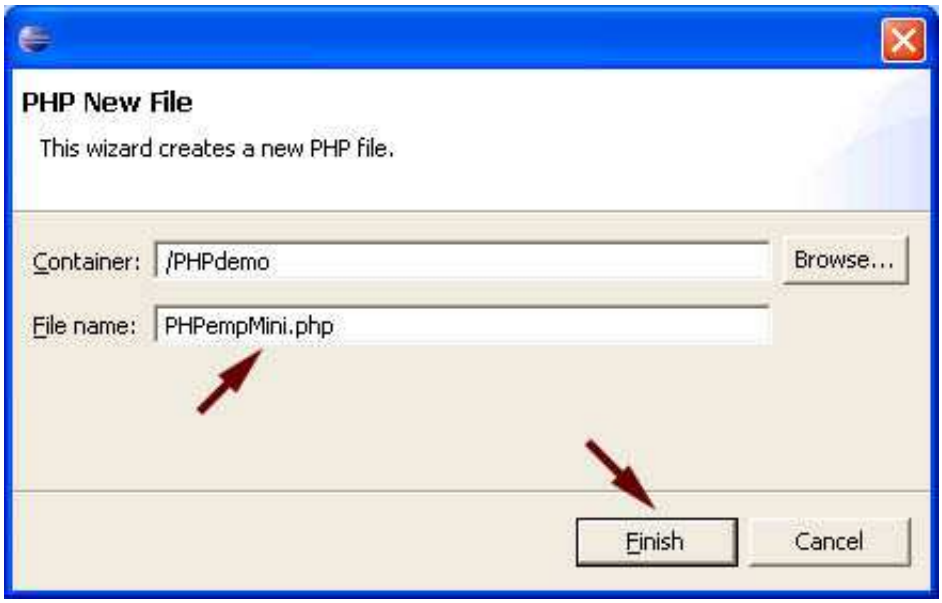
If you are asked to switch to the PHP perspective, opt to do so

3. Create the PHP SOAP client

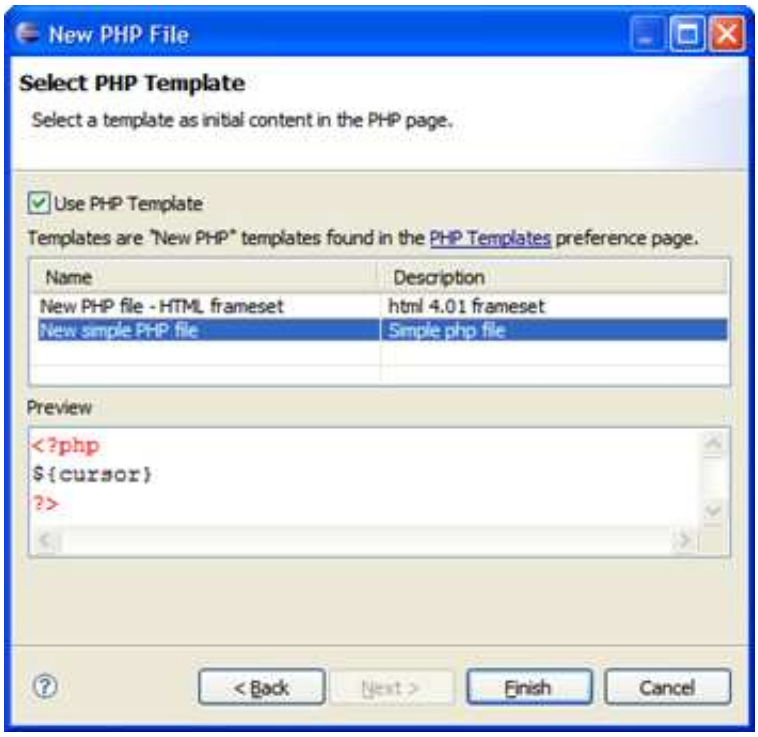
   Create a PHP script file by right-clicking into the Eclipse Navigator area, select **New** -> **PHP File**



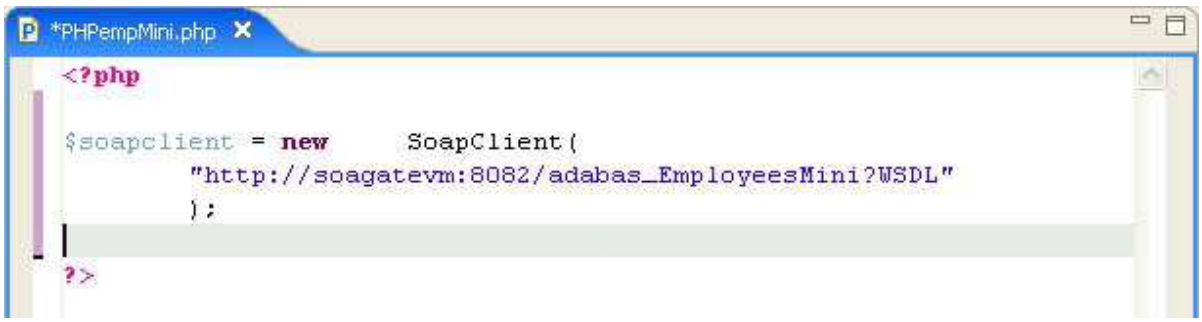
Specify a File name, click **Finish**



Opt to create a "Simple PHP File"



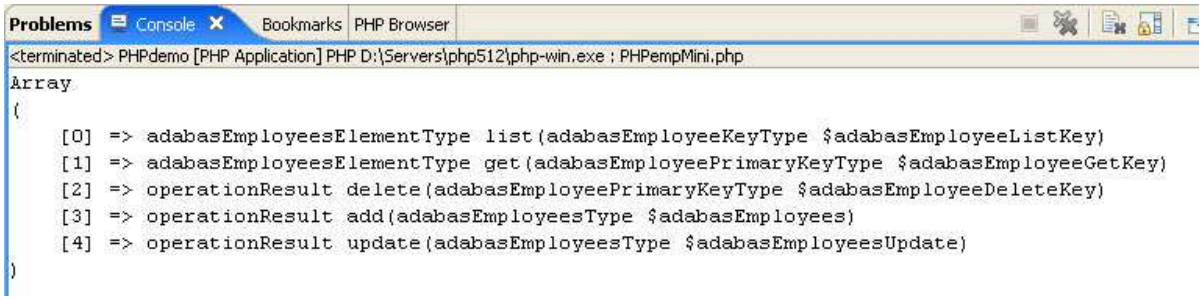
The PHP SOAP class to represent the Adabas Service is called SoapClient, the first step will be to instantiate SoapClient, passing the URL of an SOA Gateway WSDL as the parameter:



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:

```
$functions = $soapclient->__getFunctions();
print_r($functions);
```

If you run this as a console application (via "Run") the output is much better formatted than running it in the PHP browser. The console window will show the following:



This shows that the service described by the WSDL provides five operations: list, get, delete, add and update; It also lists the required parameters and the responses given.

A description of the input and output parameters can be retrieve by calling the __getTypes class:

```
$types = $soapclient->__getTypes();
print_r($types);
```

The output will look like this:

```
Array
(
    [0] => struct adabasEmployeesElementType {
 adabasEmployeesType adabasEmployees;
}
    [1] => struct adabasEmployeesType {
 adabasEmployeeType adabasEmployee;
}
    [2] => struct operationResult {
 string results;
}
    [3] => struct adabasEmployeeType {
 string personnel_id;
 string firstName;
 string name;
 string city;
 string address_line;
}
    [4] => struct adabasEmployeePrimaryKeyType {
 string personnel_id;
}
    [5] => struct adabasEmployeeKeyType {
 string personnel_id;
 string firstName;
 string name;
 string city;
}
    [6] => struct adabasEmployeeHeaderType {
 int Version;
 string ConversationState;
 string ConversationId;
 string TransactionState;
 string TransactionId;
}
    [7] => struct faultElementType {
 string Code;
 string Fault;
}
)
```

This information is sufficient to construct the first simple call to the "list" operation.

First an array of the required input parameters needs to be constructed:

```
$AdabasEmployeeListKey = array (
'personnel_id'=> '300001*', 'firstName'=>'', 'name'=>'', '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 "<table border=1 cellpadding=5>";
echo "<tr><th>Personnel ID</th><th>Name</th><th>City</th><th>Address</th></tr>";

foreach ($Adabasresponse->adabasEmployees->adabasEmployee as $Employee) {

    echo "<tr>",
    "<td>$Employee->personnel_id</td>",
    "<td>$Employee->firstname $Employee->name</td>",
    "<td>$Employee->city</td>",
    "<td>";

    foreach ($Employee->address_line as $adline)
        echo "$adline<br />";

    echo "</td></tr>";

}

echo "</table>";
```

Run this in the built in PHP Browser (activated with "**Window**" -> "**Show view**" and select "**PHP Browser**") or an external browser: http://<your_localhost_url>/PHPempMini.php:

Personnel ID	Name	City	Address
30000100	Lloyd	Ilkeston	276 Cotmanhay Road Ilkeston Derbyshire
30000107	O'Brien	Nottingham	25 Main Street Radcliffe-On-Trent Nottingham
30000110	Stilwell	Nottingham	11 Westwood Road Sneinton Nottingham
30000112	Finch	Stamford	13 Kings Road Stamford Lincs
30000114	King	Peterborough	14 Main Street Peterborough Northants
30000124	Grebby	Derby	23 The Paddock Kegworth Derby
30000125	Maguire	Derby	113 Derby Road Spondon

PHP Examples

The following PHP examples can be copied from here, moved to your web server's DocumentRoot (for example) and executed:

- 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