SAP and PHP: a practical example

Within this weblog I would like to share my first successful attempt to put SAP and PHP working together using SAPRFC.

It is assumed that a properly configured environment (i.e. Apache, MySQL, PHP and SAPRFC) is available. By the way, the one I am working with is composed by: Apache 2.0.50, MySQL 4.0.15, PHP 4.3.8, SAPRFC 1.4-5.0.4, on a MS Windows XP machine. SAP system is release 4.6B.

Let first address the question I was facing, a good excuse to start experimenting a SAP connection from PHP: I needed to make a search among SD sales orders header texts to find out those dealing with a particular customer specification.

In system customizing, header texts appear in the sales order like this:



while their determination sequence is as follows:



Needing to look inside the ‘Additional specification’ text above, a SAP Remote Function Call is available: RFC\_READ\_TEXT.

Testing it under SE37, the following appears:



Just supplying info about mandant (100), TDOBJECT (VBBK), TDNAME (sales orders number), TDID (Z003) and TDSPRAS (EN), we will get a properly filled-in TDLINE by return.



As far as the MySQL part is concerned, a simple database has been created with two tables, as follows:

database ‘sap’
table ‘sap\_import’


table ‘sap\_export’


Table ‘sap\_import’ is collecting all sales orders numbers that shall be passed to the RFC, while table ‘sap\_export’ will receive the relevant texts.

Now, let’s talk about the PHP script: we are going to connect to SAP using SAPRFC extension for PHP, telling it we want to use RFC\_READ\_TEXT.

We will then supply the RFC with input data (sales orders are read from MySQL table ‘sap\_import’).

After calling the RFC, returned data are inserted in the ‘sap\_export’ table. Here is the script (rfc\_read\_text.php):



# **SAPRFC-Class: Get Sales Orders Header Texts from SAP-System**

change it acc. to your settings $login = array ( “ASHOST”=>””, // your host address here “SYSNR”=>”00”, “CLIENT”=>”100”, “USER”=>”CIOTOLA”, // your username here “PASSWD”=>””, // your logon password here “LANG”=>”EN”, “CODEPAGE”=>”1100”); $rfc = saprfc\_open ($login ); if (! $rfc ) { echo “RFC connection failed”; exit; } //Discover interface for function module RFC\_READ\_TEXT $fce = saprfc\_function\_discover($rfc,”RFC\_READ\_TEXT”); if (! $fce ) { echo “Discovering interface of function module failed”; exit; } //Set import parameters. You can use function saprfc\_optional() to mark parameter as optional. //Fill internal tables saprfc\_table\_init ($fce,”MESSAGES”); saprfc\_table\_init ($fce,”OBJECTLINKS”); saprfc\_table\_init ($fce,”TEXT\_LINES”); $sql = “SELECT import\_tdname FROM sap\_import ORDER BY import\_tdname”; $qry = mysql\_query($sql) or die(“Problem with the query: “.$sql); while($row = mysql\_fetch\_array($qry)){ $order\_nr = $row[‘import\_tdname’]; saprfc\_table\_append ($fce,”TEXT\_LINES”, array (“MANDT”=>”100″,”TDOBJECT”=>”VBBK”,”TDNAME”=>$order\_nr,”TDID”=>”Z003″,”TDSPRAS”=>”EN”,”COUNTER”=>””,”TDFORMAT”=>””,”TDLINE”=>””)); } //Do RFC call of function RFC\_READ\_TEXT, for handling exceptions use saprfc\_exception() $rfc\_rc = saprfc\_call\_and\_receive ($fce); if ($rfc\_rc != SAPRFC\_OK) { if ($rfc == SAPRFC\_EXCEPTION ) echo (“Exception raised: “.saprfc\_exception($fce)); else echo (saprfc\_error($fce)); exit; } $rows = saprfc\_table\_rows ($fce,”TEXT\_LINES”); for ($i=1;$i<=$rows;$i++) $TEXT\_LINES[] = saprfc\_table\_read ($fce,”TEXT\_LINES”,$i); //Debug info // saprfc\_function\_debug\_info($fce); saprfc\_function\_free($fce); saprfc\_close($rfc); $qry = “TRUNCATE TABLE sap\_export”; mysql\_query($qry) or die(“Problem with the query: “.$qry); $insert\_head = “INSERT INTO sap\_export (export\_tdname, export\_tdline) VALUES “; ?> “; $insert\_body .= “(‘$tdname’, ‘$tdline’), “; } $insert .= $insert\_head.substr($insert\_body, 0, strlen($insert\_body)- 2); mysql\_query($insert) or die(‘Error performing query.
‘ . $insert . ‘
‘ . mysql\_error()); ?>

|  |  |
| --- | --- |
| Run successful | OK. OK. |
| “. $tdname.” | “.$tdline.” | “.” |

As for the config.php script included in the code above, it is just the connection to the MySQL database, something like this:



Running the main script, the result will be shown on the screen (see screenshot below) and inserted into the MySQL table sap\_export.

With this data available, we are able to build whatever kind of searches or reporting we need.



**Conclusion**:
This is a very simple but effective example about reading SAP data via a remote function call from an open source environment.