

Connecting SAP and PHP

SAP is said to be the most comprehensive and widely-used Enterprise Resource Planning system ever. Due to its widespread use, SAP users and developers come across a variety of situations where they need to connect an installation of SAP to a different platform/programming language so that the data and data interpretations contained in relation to that installation can be accessed by that platform or programming language. Hence a myriad of SAP connectors for different programming languages have been born.

PHP, being the most widely used server-side programming language on the web, is one of the most probable languages that would need to connect to SAP at some point or the other. For this, the extremely handy [SAPRFC module for PHP](#) has been developed. This post will discuss an end-to-end implementation of the SAPRFC PHP module and explain how to connect SAP to the module as well.

The first step is to go to the above link and download the complete saprfc package. The easiest way to interface saprfc with PHP and to get it to work is to download [XAMPP 1.7.1 \(NOT the latest version\)](#). This contains PHP 5.28 which seems to be the only PHP version which works correctly with the SAPRFC module.

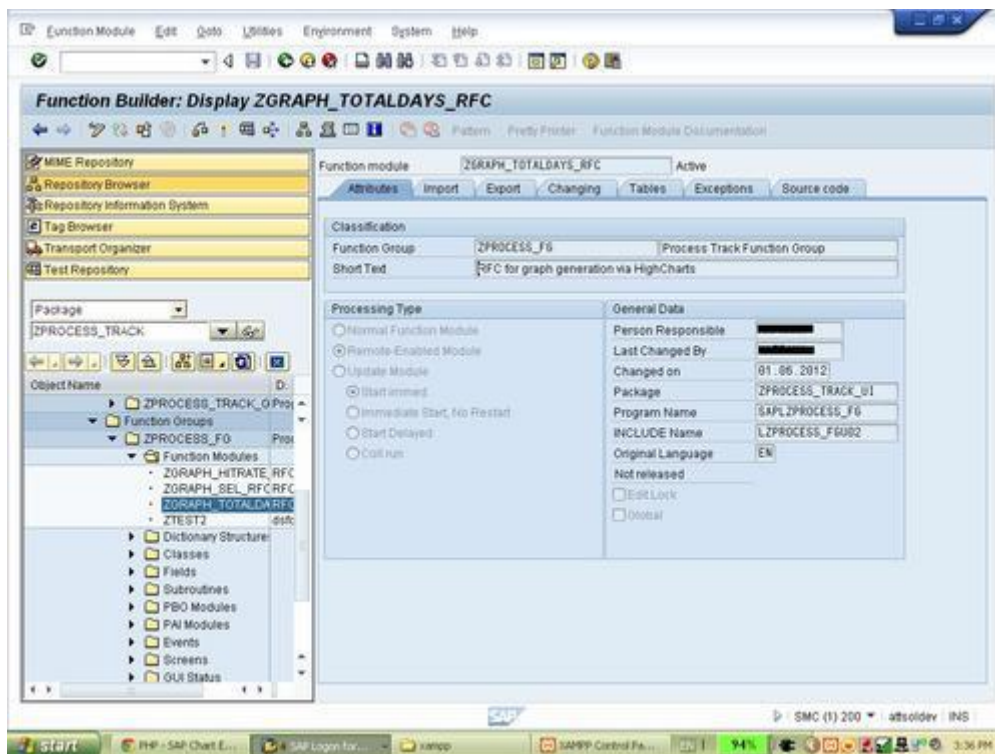
Unzip XAMPP and copy php_saprfc_528.dll from your saprfc/ext folder over to the /php/ext/ folder in your XAMPP directory. Locate the php.ini file in XAMPP and open it for editing. Here you will have to find the extensions section and add the line

```
extension=php_saprfc_528.dll
```

to this section. Now open the XAMPP Control Panel and start Apache. If you followed the above steps correctly, Apache should start now without any problems. IF it gives an error, go over the above steps again.

First of all, we'll see what we need to do on the SAP side to expose a Remote Function Module which can be accessed via RFC. All you need to do is to create a 'remote-enabled' Function Module in SAP.

This can be done by ticking 'Remote-enabled module' under the function module's attributes, as shown below:



You can write whatever program logic in ABAP in the Function Module. Pay special attention to the Importing and Exporting parameters of the Function Module as these will be what are passed to and from your PHP program.

Next, you need to copy over the saprfc.php library file over from the saprfc module folder to your PHP project directory. You can use the functions contained in this file to create a saprfc object which can login to SAP and access your remote-enabled Function Module.

The following code is essential for your PHP program to access SAP:

```

1  require_once('saprfc.php');
2  $submitted = $_GET['submitted'];
3  /**
4   * Login to SAP system
5   * @param String $user
6   * @param String $pwd
7   */
8  function login($user, $pwd) {
9      //Create SAPRFC instance
10     $sap = new saprfc(array(
11         "logindata" => array(
12             "ASHOST" => "HOSTNAME",

```

```

11             "SYSNR" =>; "SYSTEM NUMBER",
12             "CLIENT" =>; "CLIENT NUMBER",
13             "USER" =>; "USERNAME",
14             "PASSWD" =>; "PASSWORD"
15         ),
16         "show_errors"=>;true,
17         "debug" =>;false));
18
19     return $sap;
20 }
21 function logoff($sap) {
22     $sap->;logoff();
23 }
24 /**
25  * Function to call SAP RFC
26  * @param saprfc $sap
27  */
28 function callRFC($sap, $params) {
29     $cust_params = $params['cust_params'];
30     $task_params = $params['task_params'];
31     $proj_params = $params['proj_params'];
32     $result = $sap->;callFunction("ZGRAPH_TOTALDAYS_RFC",
33         array(
34             array("IMPORT", "CUST_PARAMS", $cust_params),
35             array("IMPORT", "TASK_PARAMS", $task_params),
36             array("IMPORT", "PROJ_PARAMS", $proj_params),
37             array("EXPORT", "CATEGORIES", array()),
38             array("EXPORT", "DATA_ACTUAL", array()),
39             array("EXPORT", "DATA_ESTIMATE", array())
40         )
41     );
42     return $result;
43 }
44
45

```

Note the callRFC function: Importing parameters on the SAP RFC are passed to the \$sap->callFunction function as “IMPORT” and Exporting parameters are defined as “EXPORT”. By leaving the “debug” value as ‘true’ on the login function, it is possible to get debug output for each SAP RFC call, as seen here:

```

localhost/charts/totaldays.php
Status OK
No errors detected.

Function module: ZPROCESS_WS (remote SAP R/3: 701 Unicode)

IMPORT

INPUT C 40 0 offset = 0

EXPORT

OUTPUT C 40 0 offset = 0

TABLE

Value of import (input) parameter INPUT (memory = 40):
**
Value of export (output) parameter OUTPUT (memory = 40):
*SAP RFC call from PHP successful :)*Array
(
    [OUTPUT] => SAP RFC call from PHP successful :)
)

```



Now you're ready to go out in to the world with your own PHP program that connects to SAP! Good Luck!

Here's a final look at what my program looks like. It connects to SAP to pull out project data for reporting using the JavaScript HighCharts library:

