SAP Mobile Infrastructure WebConsole using PHP Script

Applies to:

This documentation is applicable to those, who are familiar with Mobile Infrastructure WebConsole for mobile device administration and those PHP programmers with basic idea of MI WebConsole.

Summary

The purpose of this document is to give a clear idea about functionalities of pages in this PHP application. This PHP application can be used to simulate the functionalities provided by the MI WebConsole. The functionalities like, delete the installed application from the mobile device, assign particular application to particular mobile device, mobile user and on all devices, create the mobile component descriptor, delete the mobile component descriptor, start role synchronization, display the client installation logs, and filter the mobile devices based on mobile device user , mobile component descriptor, application installation status and the device id are simulated using this PHP application. Creation of mobile component descriptor is not exactly like that in the standard MI WebConsole.

Author(s): Kishor Gopinathan

Company: IVL India

Created on: 30 September 2006

Author Bio

Kishor Gopinathan is currently working as an SAP Development Consultant for IVL India.

© 2006 SAP AG

Application Overview

This PHP application can be used to simulate the functionalities provided by the SAP Mobile Infrastructure WebConsole. The MI WebConsole is to administer the mobile devices attached to the MI Server. The main functionalities provided by the WebConsole are given below,

1. Assign new applications to Mobile Devices
2. Delete the assigned applications of the Mobile Device
3. Create Mobile Component Descriptor (MCD)
4. Delete the Mobile Component Descriptor
5. Display installation logs and traces
6. Start role synchronization
7. Filter the mobile devices based on user, mcd, status of installation and device id.

In this PHP application, I have simulated these functionalities. Creation of MCD is not exactly the same as that one using the standard MI WebConsole. In the coming sections, I am discussing the functionalities of the different pages used in this application.

The code used in this article can be found in this zip file.

Assign new applications to Mobile Devices

The PHP page used to provide this functionality in this application is **addApplications.php** file. Here the end user should have to fill the screen inputs like Mobile Device User, Mobile Component Descriptor and Version, and the particular mobile device or to all mobile devices. The end user can select the mobile users from the search help against each input field. On selecting this search help, the user can view the list of users present on the MI Server. The end user can select the particular mobile user by clicking on the mobile user, which is displayed as hyperlink. The PHP page to provide the search help is **F4.php**. This is a generic search help page for displaying all the mobile device users, mobile component descriptors and the device ids attached to the MI Server.

Delete the assigned applications of the Mobile Device

The PHP page used to provide this functionality is **displayEntries.php**. By default, this page will display all the application installation of the MI Server. Here we have the provision to filter the application installations based on the Mobile Device user, Mobile component descriptor, installed application status and the device id. This provides one more functionality, that is not available in the standard MI WebConsole. In the standard webconsole, if we search with the status of application installation, the search criteria is not persisting after the page refresh. But this is handled in this PHP application. Here we can delete the particular application installation by clicking on the delete button provided against each row. After deletion, the status of the application installation will be changed to “Deletion with next synchronization”.

© 2006 SAP AG

Create Mobile Component Descriptor (MCD)

The PHP page to provide this functionality is **addComponent.php**. All the functionalities present in the standards MI WebConsole is not present in this application. Using this application, we can create MCDs. But we should have to upload the WAR files and the ZIP files for this MCD explicitly through the standard MI WebConsole since the file upload needs proper authentication.

Delete the Mobile Component Descriptor

The PHP page to provide this functionality is **mobileComponent.php**. Here we can see the list of MCDs present in the MI Server. From this, we can select and delete the particular MCD. If there are valid installations for the same MCD, then we cannot delete the MCD. This part is also working perfectly as that one in the standard MI WebConsole.

Display the installation logs and traces

The PHP page to provide this functionality is **displayLog.php**. Through this, we can view the installation logs and the device trace. This is also similar to standard MI WebConsole. The standard function module used to view the installation logs and traces is **“MEMGMT_GET_DEVICE_LOGS”**. But this is not remote enabled function module. So we have to copy this standard function module to **"ZMEMGMT_GET_DEVICE_LOGS”** and make it remote enabled. So if you also need this functionality, you would have to follow the same.

Start role synchronization

Using this, you can start the role synchronization. This is also similar to that in standard MI WebConsole.

Filter the mobile devices

Using this functionality, we can filter the device installations. User name, mobile component descriptor, status of application installation and the device id are the filtering options available here. We can select the value for the filtering criteria from the search help attached to each input.

Function Modules Used

In this section, we can see the list of RFC enabled function modules used in this PHP application.

Assign new applications to Mobile Devices

MEMGMT_DEVICE_ASSIGN_APPL	- with ACTION import parameter value as R.

© 2006 SAP AG

Delete the assigned applications of the Mobile Device

MEMGMT_DEVICE_ASSIGN_APPL - with ACTION import parameter value as U.

Create Mobile Component Descriptor (MCD)

MEMGMT_MCD_UPDATE - this can be used for adding new MCDs and to update the existing ones.

Delete the Mobile Component Descriptor

ME_JSP_SMOMO_MSD_DELETE

Display the installation logs and traces

MEMGMT_GET_DEVICE_LOGS - this is the standard function module for displaying the installation logs and traces. This one is not remote enabled. So we will have to copy this function module to our name space and made it remote enabled. Set the pass by value attribute for all import parameters.

Start role synchronization

WAF_DEVICECONFIG_UPD_BY_ROLES - this can be used to start role synchronization.

Filter the mobile devices

WAF_DEVICECONFIG_GET - used to get the list of application installations. We will have to fill the import parameters appropriately to filter the list of application installation.

How to set up the application

To configure this application in your local machine, you will have to follow like this.

Prerequisites

1. Apache HTTP Server should be installed and configured properly on your local machine
2. PHP should be installed and configured properly on your local machine
3. System should be configured properly to support the **PHPRFC**(This provides an RFC interface to SAP function modules in PHP).
4. The MI server user used to login, should have the appropriate authorizations to administrate the mobile devices and to execute the RFC enabled function modules related with MI.

5 To work with the “Display the installation logs and traces”, you should have to copy the standard Function module “**MEMGMT_GET_DEVICE_LOGS**” to your own name space in the name “**ZMEMGMT_GET_DEVICE_LOGS**” and make it remote enabled.

© 2006 SAP AG

In this zip file, we can see the list of PHP pages, java script in the script directory, and the style sheet in the style directory.

Unzip the zip and copy the PHP pages, script directory, and style sheet directory to the **htdocs** directory of apache server installation folder.

After starting the apache server, through the browser access the login page of the application by typing URL as **http://localhost:<port of the apache server>/login.php**(in my case).

In this login page, we have to enter the MI server login credentials like, MI Server Name or IP address, System Number, Client, Server User and Password. After successful login, the user will be redirected to the main page. From this page, the end user can navigate to various pages.

Useful links

- How to setup your system for SAPRFC? (Special Thanks to : Craig Cmehil)
- Apache HTTP Server
- PHP version 4 or 5
- SAPRFC for PHP
- List of SAP / PHP Blogs

© 2006 SAP AG

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

© 2006 SAP AG