



Switch Web Services Module

PHP Sample documentation

Authors

Enfocus

Contents

Switch Web Services Module	1
PHP Sample documentation	1
1. Introduction	4
1.1. Summary	4
1.2. Requirements	4
1.3. Copyrights	4
1.4. Basic switch Server Set-up	5
2. PHP	7
2.1. HTML and PHP	7
2.2. Required knowledge	7
2.3. Installing PHP	7
Xampp and Mamp	8
Installing Xampp on Windows	8
Xampp Trouble shooting	9
The Xampp Control Panel	9
Configuring Xampp	11
Installing Mamp on Mac	11
The Mamp Control Panel	11
Start the Apache server	12
Test if PHP is running correctly	13
3. The sample	13
3.1. Installing the sample	13
Locate your localhost folder	13
Copying the sample	14
Setting the Root path and running the Sample for the first time	16
Giving the public folder write permissions	17
Finding and editing your PHP configuration file (php.ini)	19
3.2. The structure and how to customize the sample	21
3.2.1. Two main pages	21
3.2.2. Settings	23
3.2.3. PHP's to include	23

3.2.4.	CSS.....	23
3.2.5.	Images.....	24
3.3.	Getting started with the sample.....	24
3.3.1.	Settings	24
3.3.2.	Two main pages	24
3.3.3.	PHP's to include.....	25
3.3.4.	CSS.....	25
3.4.	The sample manual.....	25
3.5.	Customizing the sample.....	26
3.5.1.	Placing the login window on the start page.....	26
3.5.2.	Disabling submit points, checkpoints or messages	27
3.5.3.	Determining the widths of the submit points and checkpoints lists	27
3.5.4.	Changing the distance between the submit points and their metadata.....	27
3.5.5.	Customizing strings	27
3.5.6.	Working with a fixed Switch user	27
3.5.7.	Working with users from another system	28
3.5.8.	Working within a web portal with its own login	28
3.5.9.	Give the metadata area a different fixed height	28
3.5.10.	Using a fixed submit point or checkpoint	28
3.5.11.	Using a different icon for the refresh and download buttons.....	29
3.5.12.	Changing the overall color scheme	29
3.5.13.	Changing the overall font scheme	29
3.5.14.	Changing the color or font of an individual element	29
3.6.	Known limitations.....	29
4.	Troubleshooting.....	30

1. Introduction

1.1. Summary

This is an example of how Switch Client functionality can be implemented in a website. It shows some basic features and is meant as a first introduction for Switch users to the module. It should be clear that the actual SDK offers much more than what is implemented in this sample website. Everything that is possible with the Switch Client application is also possible to implement in a website or application using the Web Services module.

The purpose of a sample is to give guidance to people who want to build their own system, it is not meant to be a finished product. We have attempted, however, to make the sample in such a way that it is relatively easy to set up and customize into a usable product within hours and without an in-depth knowledge of PHP.

This manual will not explain much about the actual Web Services SDK, the Web Services folder contains a different document specifically about the SDK. This manual will explain how the PHP sample is built up and where, when and how we communicate with the Switch Server.

1.2. Requirements

- Minimum Switch Server 11 with a web service license/trial
- A web server with PHP, this can be set up locally
- A code editor

Recommended freeware code-editors are:

for Windows : [notepad++](#)

for Mac OS X : [TextWrangler](#)

1.3. Copyrights

All information contained herein is the property of Enfocus BVBA. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Enfocus BVBA.

This publication and the information herein are furnished AS IS, are subject to change without notice, and should not be construed as a commitment by Enfocus BVBA. Enfocus BVBA assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and no infringement of third party rights.

Enfocus Switch, Enfocus SwitchProxy, and Enfocus SwitchClient are product names of Enfocus BVBA.

- Apache is a product of The Apache Software Foundation.
- TextWrangler is a product of Bare Bones Software, Inc.
- Notepad++ is a product of Don Ho.
- Microsoft Windows is a product of Microsoft Corporation.
- Mac OS X is a product of Apple Inc.
- XAMPP and Apache Friends are registered trademarks of Kai Seidler.
- Mamp is a registered trademark of appsolute GmbH

Other names may be trademarks of their respective owners.

Copyright ©2000-2014, Enfocus BVBA.

1.4. Basic switch Server Set-up

If you are a developer in charge of writing a web service with the Switch Web Services Module and you don't have any Switch experience it is recommended to quickly read through the Switch [documentation](#) found on the Enfocus website.

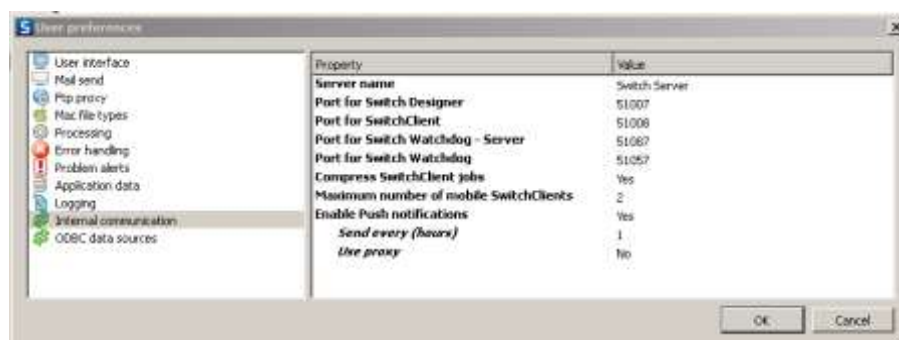
To use this Sample website you need access to a Switch Server, either locally or somewhere in a LAN or WAN.

To access the Switch Server you'll need the following:

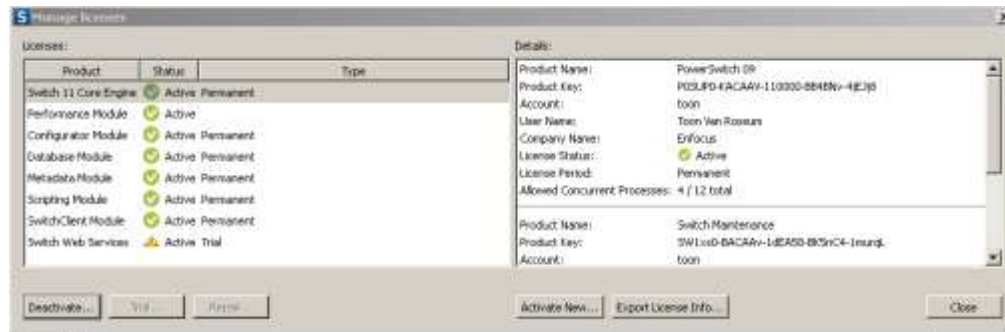
(See below for information on how to find the necessary settings)

- The IP-address or DNS of the system running Switch Server
- The Port-number for client connections.
- An active Web Services Module
- Switch Server-User-credentials
- One or more test flows

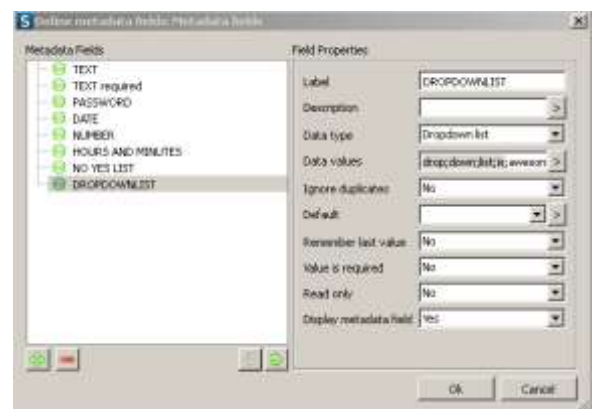
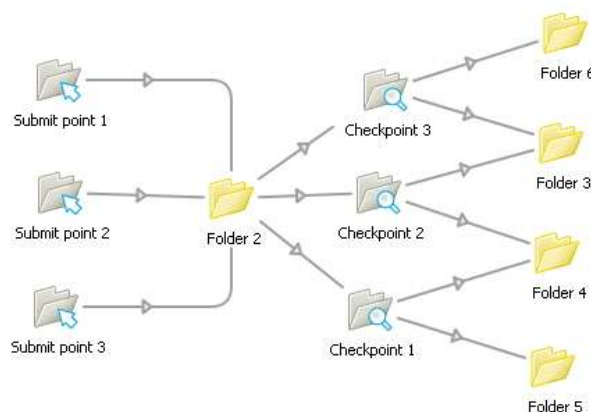
The port for SwitchClient can be found in the user preferences of Switch under 'internal communication'. By default this is 51008.



The Switch Web Services License: in Switch Server click help>manage licenses. If **you don't have** a valid key for the Switch Web Services module, you can activate a trial by selecting the module and click Trial.



A test flow: to try out this sample website It is sufficient to have a flow with one Submit Point and one checkpoint. Adding metadata entries will make it possible to experiment with metadata forms.



To be able to connect to a Switch Server you will need Switch-user-credentials, these can be found under the 'Users' tab in Switch.

Note: Switch users don't always use a password.



2. PHP

2.1. HTML and PHP

The difference between HTML and PHP is that HTML is static, whereas PHP is dynamic. Or to put it more accurately, PHP can create content in a dynamic way: a browser will ultimately always display HTML, never PHP, but the PHP server code creates the HTML client code on the fly. PHP files therefore contain a mix of PHP and HTML, but there is range: you will find PHP files that contain only PHP code to perform certain background operations like accessing databases, reading XML files, communicating with Switch over SOAP, etc., but you will also find PHP files with one or two minor PHP constructs lost in a heap of HTML.

2.2. Required knowledge

To run the sample, no PHP knowledge is needed. Also to customize the sample no real PHP knowledge is required, but you should feel comfortable in making small and documented changes to PHP files, HTML files and CSS files.

To write your own web services using the module you will need SOAP experience in combination with a language that supports SOAP Calls (PHP, ASP, .NET).

2.3. Installing PHP

To be able to run PHP webpages on a local system it is necessary to have PHP installed as a service. Without this service your web browser won't be able to translate PHP pages to html, and it won't be able to display the pages to the user.

There are multiple solutions to get PHP running. One way is to actually set up a (local) webserver and include a PHP component. This might be too complex for novice users and advanced users might already know how to do this or where to find the necessary information. If you already have PHP running or you have a webserver available, you can skip to the next chapter "3. The sample".

For first-time users we will look at Xampp and **Mamp**. These are two user-friendly applications that offer a User Interface to start and stop certain services, including PHP.

If you choose not to install Xampp or Mamp or are determined on finding out how to run a webserver the following links might be a good starting point:

[Installing IIS on Windows](#) & [Installing PHP on IIS](#) OR

[Installing Apache on Windows](#) & [Installing PHP on Apache](#)

Mac OSX includes an apache webserver with PHP included:

[Enabling php on macosx](#)

In general <http://www.php.net> should offer an answer to all questions relating to PHP installations.

Xampp and Mamp

Like mentioned before, Xampp and Mamp are 2 applications that offer a simple user interface to start and stop certain services on your system. One of the services they both offer is to run PHP.

Xampp is a cross-platform application meaning you can run it on both Mac and Windows. Mamp is a Mac only application. Because of some minor differences between the Xampp installation on Mac and Windows it seems like Mac users prefer Mamp. Therefore in this manual we will look at Xampp on Windows and Mamp on Mac.

Installing Xampp on Windows

You can find a lot of information about XAMPP on the internet. In this chapter we will explain the most basic setup needed to get the Switch webservices running.

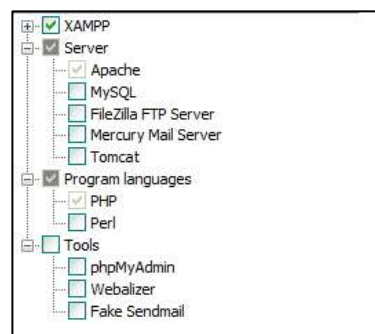
Download link

<http://www.apachefriends.org>

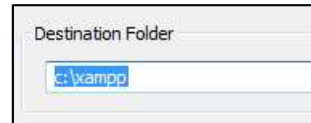
From the website, download and run the installer, it's a straight forward process that does not need a lot of user interaction.



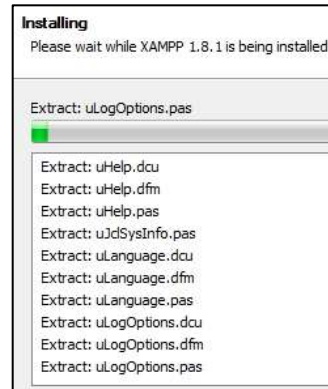
To run the Switch webservices sample, only a minimal installation is necessary. Although including all modules might be fun if you're interested in experimenting with more services.



It is recommended to pick the default installation folder. Further steps in this manual will be based on this.



Depending on the services you selected the installation may take up to 20min.



pick 'yes' to start the Xampp Control Panel for the first time.



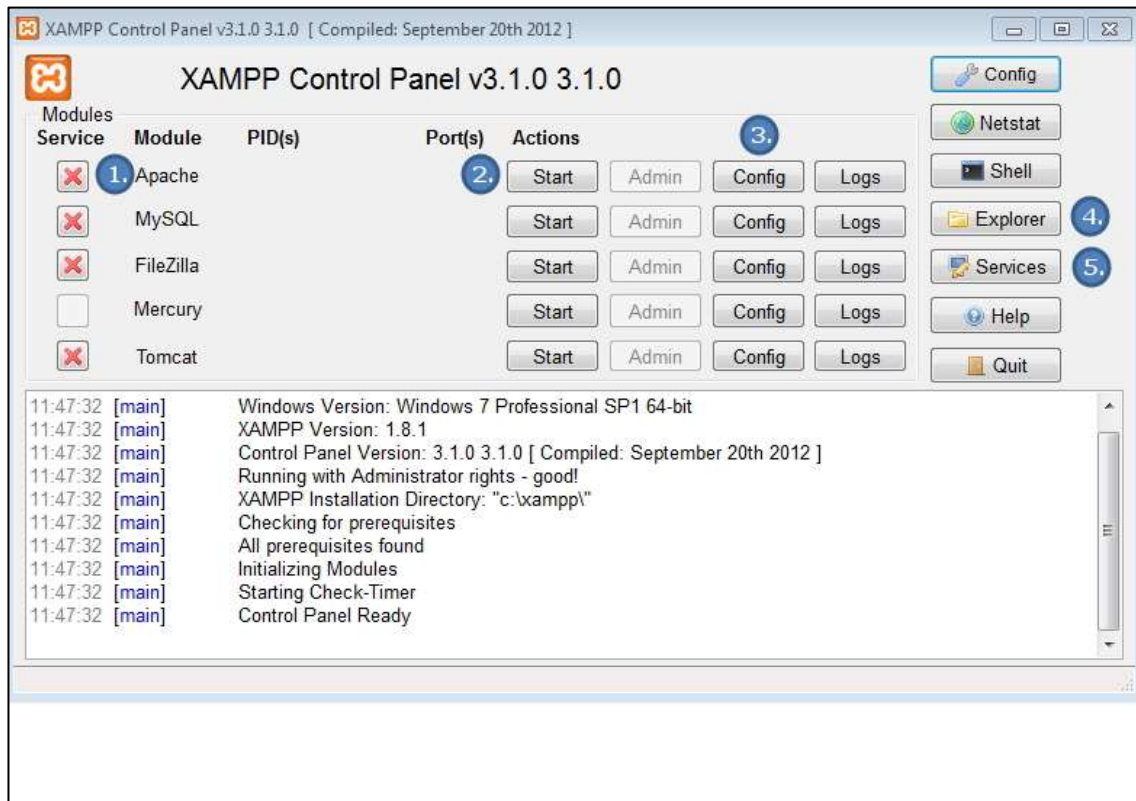
Xampp Trouble shooting

Skype and Xampp

There is a known issue when running both Skype and XAMPP on the same system. This is related to both applications using Port 80 on your system. A simple workaround is to first start Xampp and then Skype. You can also change the following setting in Skype to fix this: Topbar > Extra>settings > Advanced > connections > disable 'Use port 80 and 443 as alternatives for incoming connections.

The Xampp Control Panel

At first sight it might look a bit overwhelming, but that's only because the extended range of features offered by Xampp. You won't need any of the advanced features to run the sample.



1. Install as service.

A red cross means that the module is not set up to run as a service. If you enable it, an alert panel will be shown saying 'click yes to install the <Module name > as a service.

Installing a module as a service means that it will start to run as soon as you start windows and you won't need the control panel anymore.

2. Start/Stop modules

Depending on the status of the module this button will either show Start or Stop. In combination with 'install as service' this means that even after quitting Xampp the service will stay running.

3. Shortcut to config files

Clicking this button will show a pop-up menu with shortcuts to configuration files specific to each module.

4. Shortcut to the Xampp root in explorer.

This will open a windows explorer window in the root of the Xampp installation.

5. Services.

A shortcut to Microsoft services.mcs, useful for problem solving, troubleshooting,...

Configuring Xampp

To be able to run the Switch webservices sample, Xampp does not need any further changes. To install the Sample please continue reading chapter '3.The sample'.

Installing Mamp on Mac

Just like Xampp there is a lot of information available on the internet about Mamp, for the Switch webservices sample we will only look at what is needed to get the sample working.

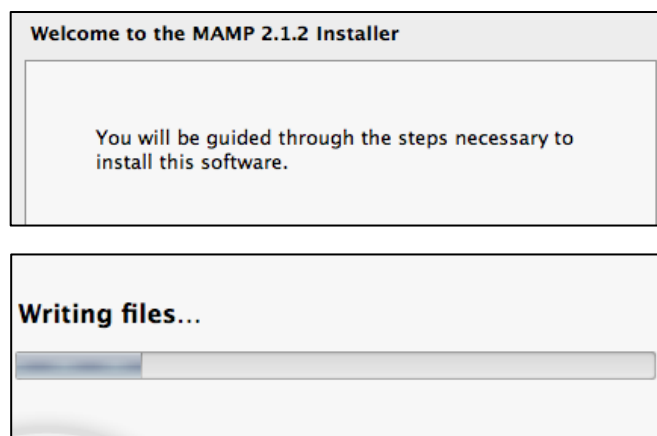
Mamp is available both as a free version and a PRO version, for this sample the free version has all the needed features.

Download link

<http://www.mamp.info>

Install Mamp.

During the installation the user only needs to provide the Mac OSX user-password and agree on the terms of agreement.



The Mamp Control Panel

Compared to the Xampp Control panel, Mamp offers a lot less features. Still, everything needed to run the sample is available.



1. Start/ Stop Servers
Depending on the status of the servers this button will either make the servers stop or start. It's not possible to only start one of the available servers
2. Preferences
This button opens the preferences panel.
Note: The default installation does not need any adjustments to run the Switch webservices sample.

Start the Apache server

After installing either Xampp or Mamp you're almost ready to run the PHP sample. The first step is to activate the needed modules. In Xampp this can be done by pressing the Start button for the Apache module, in Mamp you press the Start Servers button.

Test if PHP is running correctly

In Mamp: Click Open Start page. A browser window should open showing a 'Welcome to Mamp' Page.



In Xampp: Click the 'Admin' button next to the Apache server module. A browser page should open



3. The sample

3.1. Installing the sample

Locate your localhost folder

Your localhost folder is the folder on your system where your browser will look for content when directed to localhost. This is the folder where we want to copy our Webservices sample website.

If you decided to install Xampp or Mamp (on the default location) the localhost folder is located:

- Xampp : C:\xampp\htdocs
- Mamp : /Applications/MAMP/htdocs

If you decided not to use Xampp or Mamp please consult the documentation of your webserver.

Copying the sample

After locating the localhost folder you can either:

- delete everything in it and use it as the main folder for the Switch webservices PHP sample. For Xampp and Mamp this will remove all Xampp or Mamp help files.
- Create a new sub-folder to use as the root for this sample.

The next steps will assume that you picked the second option and created a subfolder named 'Switch'.

For a default Xampp and Mamp installation this will result in the following folder structure:

Xampp: C:\xampp\htdocs\Switch\

Mamp: /Applications/MAMP/htdocs/Switch/

Now copy everything from the sample folder from the webservices PHP folder (*ISamples\PHPI\Sample*) to this newly created folder.

This should result in the following file structure:

Path to webserver localhost /Switch/

```
| index stripped down.php
| index.php
| phpinfo.php
| switchworkspace stripped down.php
| switchworkspace.php
|
+---CSS
|   index.css
|   switchworkspace.css
|   webservicesample.css.php
|
+---images
|   ajax-loader.gif
|   banner930x100.jpg
|   close.gif
|   download.png
|   refresh.png
|   refresh_hover.png
|   smart-resources-banner.png
```

```
|
| +---include
| | alertjobslist.php
| | checkloginfromswitch.php
| | checkpoint.push.php
| | client.process.php
| | debug.php
| | errorhandling.php
| | includes.php
| | infobox.php
| | loginwindow.php
| | logout.php
| | messages.php
| | submit.metadata.php
| | submitpointlist.php
| | web.session.php
| | workspace_usermenu.php
| |
| +---lib
| | | butterfly.v3.2.wsdl
| | | SwitchClientService_v4.0.php
| | |
| | \---soap-patterns
| | | download-pattern.xml
| | | submit-pattern-arguments-ref.xml
| | | submit-pattern-booleans-ref.xml
| | | submit-pattern-multi-ref.xml
| | | submit-pattern.xml
| |
| +---public
| | \---scripts
| | | alertjobs.js.php
| | | infoboxmessages.js
| | | jquery-1.7.2.min.js
| | | submitpoints.js.php
| | | workspace.js.php
| |
| \---settings
| | settings.php
```

Setting the Root path and running the Sample for the first time

If you copied the sample into the subfolder 'switch' of your localhost folder you can access the website by directing your browser to localhost/Switch



As this is the first time you run the sample you will run into some errors. The first error that will be shown is:

*please adjust the preferences stored in the settings folder
- check the ROOT setting and adjust if necessary*

What this error actually means is that because we copied the files to the Switch folder and not in the root of the webserver, the sample is not able to find certain files.

To fix this we need to change a settings of the sample, you can find the settings file at the following location:

Path to webserver localhost /Switch/settings/settings.php

Open this file in your code editor or text editor. As this is the first time you're opening the settings file it's advised to read the first lines that offer some information and a short introduction to the file. Later in Chapter '3.2 The Structure' we take an more in depth look at this file.

Search for the line:

```
define("ROOT", $_SERVER["DOCUMENT_ROOT"]. " ");
```

And change it to

```
define("ROOT", $_SERVER["DOCUMENT_ROOT"]. " /Switch/");
```

By doing so you made sure that the sample will look for all files in the subfolder 'Switch' of the webserver.

Again: direct your browser to localhost/Switch. Now the first page of the Php Sample should be shown.



Welcome to the Enfocub Switch Webservices sample

This is an example of a customized start page into which you can integrate a login window for the Switch Web Services.

You can build this page any way you like, but it is a good idea to divide it into several div sections that are positioned and sized using css.

On this page there are four divs: one for the banner, one for this text you are reading, one for the Smart Automation banner, and one for the footer.

The place and size of the div elements on this page are determined by the order of appearance in the body of the page and by the *index.css* file.

The *index.css* sets a fixed container width for the page of 960 pixels. When the width of the browser window is smaller than 960 pixels a horizontal scroller will appear.

The idea is that you modify this page by adding a couple of lines of PHP code that you copy from the *index.naked.php* file. Insert the copied lines in the right location so the login window appears on the top right part of the screen under the banner. You can then change the position of the login panel in *index.css* and, if so desired, the position and look inside the panel in *webservicesample.css.php*.

Save the modified file as *index.php*.

Switch Web Services Login

Name

Password

Login

SMART
AUTOMATION
RESOURCES
TO GET YOU STARTED
Get them for free!

Trademark information :: Online Privacy Policy :: Contact :: Enfocub.com

Although the page is loaded correctly some functionality will still not work. Therefore we need to adjust some more settings.

Giving the public folder write permissions

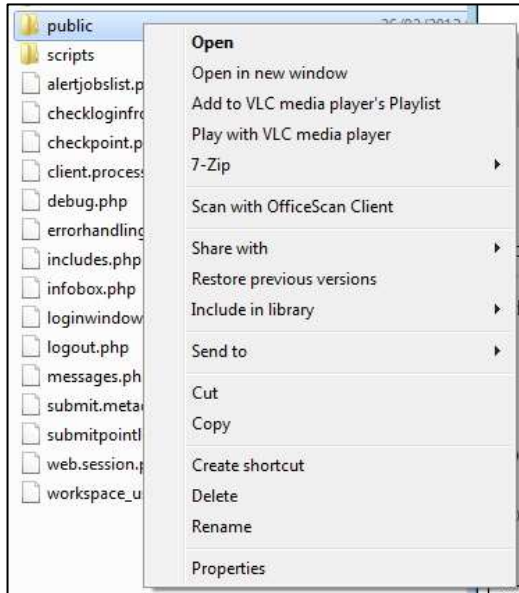
And important part of setting up the sample is to make sure that the webserver has permission to read and write in the public folder of the sample. This is important because the folder is used to temporarily store submitted jobs, thumbnails,...

The folder is located:

Path to webserver localhost /Switch/include/public

The procedure to give this permission differs from Mac OSX and Windows

Windows

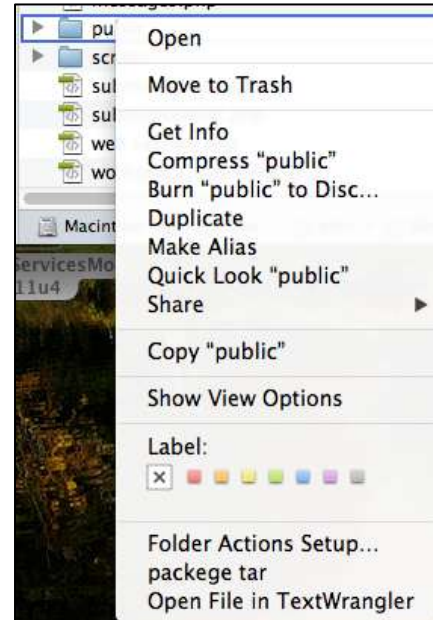


Right click on the folder.
Choose 'Properties'



1. Disable Read-only
2. Click Apply
3. Make sure that the changes are applied on subfolders as well

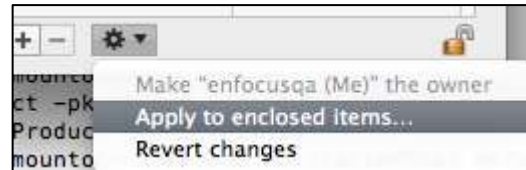
Mac OSX



Right click on the public folder.
Choose 'Get Info'.



1. Unlock to edit
2. Give 'Read & Write' permission for user 'everyone'



Make sure to select the 'Apply to enclosed items' option

Finding and editing your PHP configuration file (php.ini)

PHP has a function to display all PHP settings, this function is called `phpinfo()`. We need this information to check if certain PHP extensions are enabled or disabled. The sample depends on some specific extensions that might be disabled by default.

To do so we need to know where our configuration file is stored on our system. Both Xampp and Mamp offer a link to a page showing this information.

Xampp ;

Click on the Admin button next to the Apache module.

On your welcome page, click on the `phpinfo()` link on the left of the screen



Mamp :

Click on the Open start page button. On the welcome page is a link to a `phpinfo` page.



If you didn't choose to run Xampp or Mamp the easiest way to see the `phpinfo` output is to create a text file in you localhost directory (for example name it `phpinfo`). Then change the extension to `.php` and enter the following lines into the file :

```
<?php phpinfo(); ?>
```

Then direct your browser to `localhost/phpinfo.php`.

The `phpinfo` page

PHP Version 5.3.0 	
System	Windows XP SP3 (x86) (5.0.5.1) (i386) Windows XP Professional Service Pack 3, 32-bit
Build Date	Aug 23 2010 17:40:20
Compiler	MSVC9.0 (Visual C++ 9.0)
Architecture	x86
Configure Command	configure --enable-ipv6 --enable-openssl --enable-sockets --enable-zlib --enable-bz2 --enable-gd --enable-gd-native-ttf --enable-gd-jpeg --enable-gd-tiff --enable-gd-fontconfig --enable-gd-librsvg --enable-gd-xpm --enable-gd-webp --enable-gd-freetype --enable-gd-icu --enable-gd-xml --enable-gd-ldap --enable-gd-mcrypt --enable-gd-mhash --enable-gd-mcrypt --enable-gd-mhash --enable-gd-mcrypt --enable-gd-mhash
System Root	Windows 2.0 (x86)
Virtual Directory Support	enabled
Configuration File	php.ini (C:\WINDOWS\system32\php.ini)
Loaded Configuration File	C:\WINDOWS\system32\php.ini

To find the loaded configuration file search for this line:

Loaded Configuration File	C:\xampp\php\php.ini
---------------------------	----------------------

Now open this file in your code editor. (see 1.2 Requirements for freeware code editors).

The file contains all the settings and loaded extensions. A disabled extension will have a semicolon in front of it.

For example:

```
;extension=php_openssl.dll
```

An enabled extension will not have a semicolon in front of it.

For example :

```
extension=php_curl.dll
```

search for the following extension and make sure that they are enabled :

php_curl.dll

php_openssl.dll

php_soap.dll

IMPORTANT! When you save the `php.ini` file you will also have to restart the webserver. For Xampp and Mamp you can stop the server first en then start it again from within the control panel.

3.2. The structure and how to customize the sample

To customize the sample you need to understand what the role is of the different constituent elements. Although the sample is based on PHP and you have to manipulate some PHP no knowledge of PHP is required.

3.2.1. Two main pages

There are two pages at the very top of the sample “site”:

- A start page where the login window is shown (*index.php*).
- A workspace page where a user menu is shown with access to submit points, checkpoints and messages. Selecting a submit point or a checkpoint will show the metadata (*switchworkspace.php*).

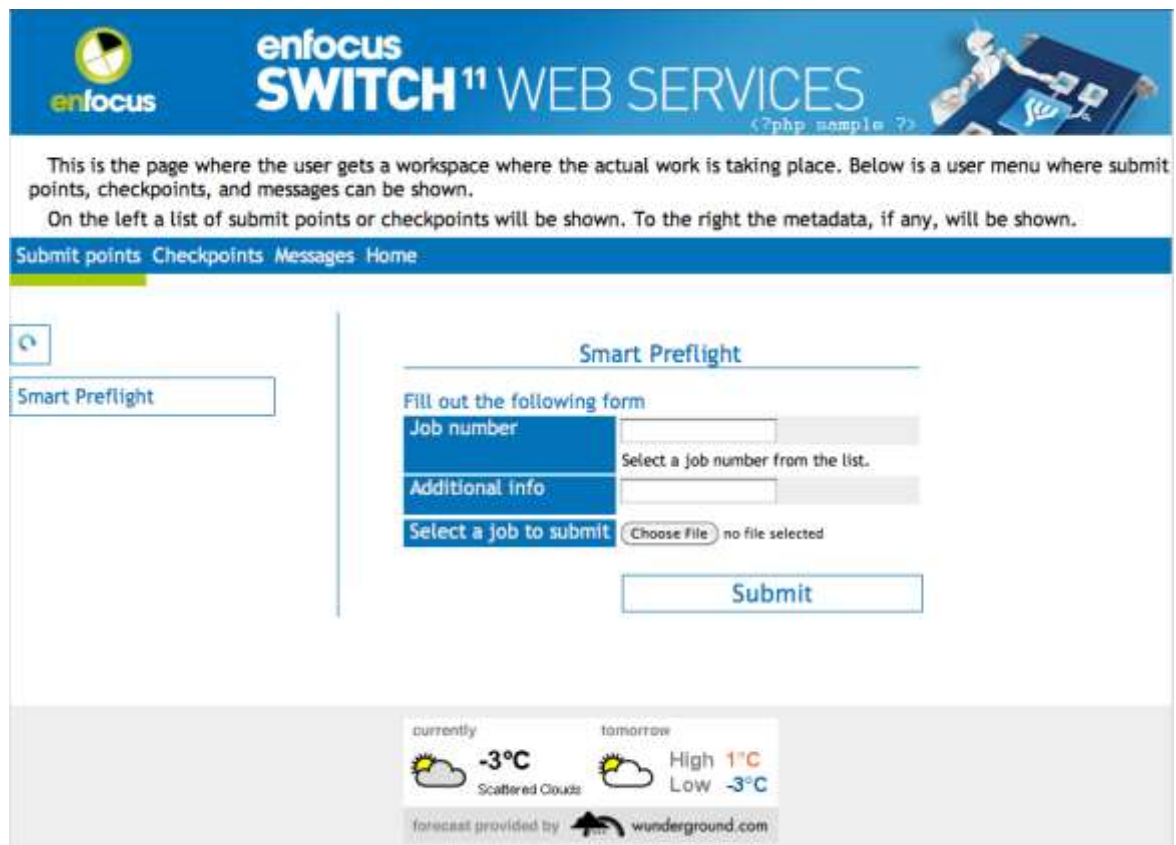
This is what the start page looks like:



The screenshot shows the start page of the Enfocus Switch Web Services. The header features the Enfocus logo and the title "enfocus SWITCH¹¹ WEB SERVICES". Below the header, there is a welcome message and a login window. The login window has fields for "Name" and "Password", and a "Login" button. To the right of the login window is a banner for "SMART AUTOMATION RESOURCES TO GET YOU STARTED". The footer contains links for "Trademark information", "Online Privacy Policy", "Contact", and "Enfocus.com".

You will have to build your own content of course and include the login window on your start page.

The same goes for the workspace page.



The screenshot shows the workspace page of the Enfocus Switch Web Services. The header is the same as the start page. Below the header, there is a message about the workspace and a user menu. The user menu has links for "Submit points", "Checkpoints", "Messages", and "Home". The main content area is titled "Smart Preflight" and contains a form for submitting a job. The form has fields for "Job number", "Additional info", and "Select a job to submit". There is a "Submit" button at the bottom of the form. At the bottom of the page, there is a weather forecast widget showing "currently" and "tomorrow" conditions, and a link to "forecast provided by wunderground.com".

The part in the middle with the menu, the submit point list and the metadata is generated by the sample, the rest is up to you.

3.2.2. Settings

In the *settings* folder there is a *settings.php* file. You will have to edit this file to provide some essential configuration information, but you can also edit it to customize the sample. It has three important parts:

- Some general configuration settings and the settings to connect to the Switch server.
- All the strings used in the sample for easy customization and translation.
- The main colors, the font and the font sizes used in the sample.

3.2.3. PHP's to include

In the *include* folder are all the PHP files that generate the HTML code for the different parts of the sample that are specific for the Switch Web Services: the login window, the submit points overview, the metadata input, etc. The PHP files in this folder (and below) do not place anything on the two main pages outside of the login window and the workspace window.

It is not necessary to change these PHP files to get a working environment. You could of course do that, but then you are knee deep into PHP and that falls outside the scope of this document.

The *lib*, *public*, and *scripts* subfolders contain the low-level parts of the sample. Here resides the code to communicate with Switch Server. There is really no reason to change any of these files. Doing so requires a very in-depth knowledge of PHP and JavaScript, and falls even further outside the scope of this document.

3.2.4. CSS

CSS stands for cascading style sheets. They describe how to render the HTML elements of the page in terms of the colors and the fonts that are used, the position of the elements, etc. These files are found in the *css* folder. There are three files in there:

- *index.css*
This css is only there to show *index.php* correctly. When you make your own *index.php*, you may have your own css attached to it in which case you will either have to link to this css additionally or copy parts of this css into the existing one.

- *switchworkspace.css*
This css is only there to show *switchworkspace.php* correctly, and the same remark applies as the one above.
- *webservicessample.css.php*.
This CSS file is set up as a PHP, because it uses constants from the settings file. It is the css for all the elements inside the login window and the workspace window.

3.2.5. Images

The *images* folder contains a few images that are used in the sample. There are two images to display banners on the start and workspace pages outside of the scope of the web services.

3.3. Getting started with the sample

3.3.1. Settings

Edit *settings/settings.php* with a text editor and modify the following lines:

```
define("ROOT", $_SERVER["DOCUMENT_ROOT"]."/foldername/");  
define("STARTPAGE", "index.php");  
define("SWITCHWORKSPACE", "switchworkspace.php");  
define("SWITCHSERVER", "localhost");  
define("SWITCHPORT", "51008");
```

The ROOT constant must refer to the folder where you installed the sample. The STARTPAGE constant refers to the page where the login window is placed. This is also where the sample will jump to when logging out. The SWITCHWORKSPACE constant must refer to the second main page that describes the workspace. The main start page is either called in the URL provided by the user, or it is linked from another page of the site.

The constants above are the only ones you have to change to get the sample to work. There are other constants that you will certainly want to change, but those are described in the section on how to customize the site.

There is also more explanation available in the comments of the file.

3.3.2. Two main pages

In the sample there is an *index.php*. You will want to make your own start page, but keep this one for reference! The PHP file with your start page can have any name, but then the user has to specify it as part of the URL so we recommend calling it *index.php*.

Some authoring applications, Dreamweaver being one of them, allow you to insert PHP code directly, but we will assume only the availability of a text editor. The principle to make your start page is as follows:

- Create/copy/edit the HTML of the start page on which you want to show a login window.
- Name the file *index.php* and not *index.html*. A file containing PHP code must have *.php* as an extension; an HTML file containing PHP code will generate errors. Make sure there is no *index.html* file, as the web server will take HTML before PHP.

Of course this does not apply when you are integrating the Switch Web Services in an existing environment, but make sure that you do not break any links if you have to rename an existing file.

- Add the PHP code to kick-start the Switch environment. In *index.php* it is documented what lines you need to copy. Look for the comment lines with *//SWITCH WEB SERVICES*. There are four places to copy from.

The same principle applies to the Switch workspace page. Do not forget to change the settings file if you are using a different file name than the one from the sample:

```
define("SWITCHWORKSPACE", "myownworkspacename.php");
```

3.3.3. PHP's to include

As mentioned in the overview of the structure of the sample you do not have to change any of these files to have a working system.

3.3.4. CSS

Here too, no changes to the CSS files are necessary to have a working system, but you will have to edit them if you want to change the visual appearance of the sample.

3.4. The sample manual

You are now ready to test the sample. When *index.php* is your start page, then all you have to type for the URL is: <http://www.mysite.com>.

You should now see the start page with the login window. It may not be in the correct location yet and it may not have the look you want yet, but it should be there and you should be able to log on. When no connection can be established you will see the error message "Connection failed" in the login window. The user credentials are checked

against Switch directly. When you have not specified valid Switch credentials you will get the error message "Login failed".

On the workspace page you should see a menu where you can choose between "Submit points", "Checkpoints", "Messages", and "Logout". Check to see if you get a reaction to all the buttons. Again, not everything may be in the right place yet.

The submit point and checkpoint lists are cached. That makes switching submit points and checkpoints a lot quicker, and that is why there is a refresh button at the top of the list.

How to make sure everything is in the right place, has the right size, etc. is explained in the chapter on how to customize the sample.

3.5. Customizing the sample

You will most probably not want to use the site as it is. You may want a different color scheme; you may want to change some of the strings that are displayed. Not all the elements may be in the right place yet, or have the right size.

Some of what follows is based on css. This is not a course in css, but fortunately quite a lot is self-explanatory. With some trial and error even the novice css user can get quite far. Unfortunately, however, css changes can have unintuitive results especially when you are dealing with widths. Also note that browsers sometimes interpret css definitions in different ways.

There are three css files in the sample. In *index.css* and in *workspace.css* you find how you can size and position the different div sections on the start and workspace pages. When incorporating the sample in another environment you will have to copy and paste some lines (read the comments) into already existing css files or link to those css files additionally. In *webservicesample.css.php* you find everything related to the content of the div sections where Switch-specific elements are shown.

Here is a non-exhaustive list of changes you may need or want to make and how to do that.

3.5.1. Placing the login window on the start page

Where the `<div id="divforloginwindow">` section is in the `<body>` of your start page will largely determine where the login window is shown on the screen, but also the css plays a role. Read the comments in *css/index.css* for more explanation.

3.5.2. Disabling submit points, checkpoints or messages

By default the workspace page will show submit points, checkpoints and messages. You can disable them by making the value string empty in the settings file:

```
define("SUBMITPOINTS","");  
define("CHECKPOINTS","Check points");  
define("MESSAGES","");
```

When there is only one submit point or one checkpoint the sample will immediately display the metadata without requiring the user to make a choice from the list.

3.5.3. Determining the widths of the submit points and checkpoints lists

On the workspace page the submit points and checkpoints are shown in a list on the left-hand side. The widths are defined in the settings file:

```
define("SUBMITPOINTLISTWIDTH","200px");  
define("CHECKPOINTLISTWIDTH","500px");
```

The metadata is displayed to the right of the list. The reason the widths are different is that the width of the names of the submit points is different from that of the jobs in the checkpoints. Also when you start modifying font sizes you may need to tweak these values to get a visually acceptable result.

3.5.4. Changing the distance between the submit points and their metadata

The metadata is placed to the right of the list. Changing the margin and padding properties of the metadata table in *css/switchwebsample.css.php* creates a distance between the list and the metadata. Edit the css file and look for the part that describes the properties of the metadata table.

The comments of the file explain where to find what.

3.5.5. Customizing strings

All the strings used by the sample are defined as constants in the settings file. Simply change the values. Take care not to change the names of the constants.

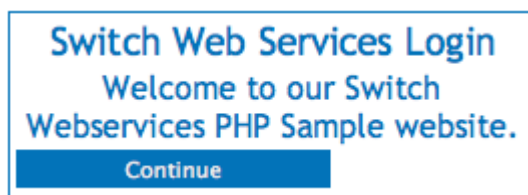
3.5.6. Working with a fixed Switch user

It is conceivable that the customer does not want to manage all the users inside Switch and that he wants all users to simply use the same credentials.

When you fill in values for the following constants in the settings file:

```
define("USER","SwitchUser");  
define("PASS","SwitchPassword");
```

the start page will not show a login window, but a window with a message and a link to jump to the workspace page.



3.5.7. Working with users from another system

When the Web Services site is not integrated in another web site, but you want to check the credentials against another system, e.g. an MIS system, then you will have to work with a fixed user in Switch, and you will have to write your own PHP code to do the checking.

3.5.8. Working within a web portal with its own login

When the portal into which you are integrating already has login functionality you should also define a fixed user, but instead of using the start page you immediately jump to the workspace page.

3.5.9. Give the metadata area a different fixed height

In the css of the workspace the sample defines a minimum height for #submit_or_checkpoint_metadata. When you take that away anything under the metadata will go up and down depending on how big the metadata area is for the selected submit point. Depending on how big your metadata areas are you may want to adjust this value.

3.5.10. Using a fixed submit point or checkpoint

In Switch you can define what submit points and checkpoints are visible by what group of users. If you want to use a fixed submit point or checkpoint then select that one in the Users pane of Switch Designer.

3.5.11. Using a different icon for the refresh and download buttons

Replace the file *images/refresh.png* or *download.png* with a new one. As the refresh button has a fixed size in *css/webservicesample.css.php* you may have to modify it there (search for "refresh"). The position of the download image (to the bottom right of the preview of the selected job) is such that no css definitions are required.

3.5.12. Changing the overall color scheme

The sample uses a color scheme with three colors, defined in the settings file as:

```
define("MAINCOLOR", "#0273b9");  
define("SUPPORTCOLOR1", "#eee");  
define("SUPPORTCOLOR2", "#abca0c");
```

The main color is the Switch blue, the first supporting color is a light grey, the second supporting color the Enfocus green.

The *webservicesample.css.php* file uses these constants. It is therefore very easy to change the color scheme of the Switch Web Services part by changing only the constants in the settings file.

3.5.13. Changing the overall font scheme

The sample uses the same font everywhere in three different sizes. You can change them in the settings file:

```
define("FONT", "Trebuchet MS");  
define("LARGEFONTSIZE", "1.2em");  
define("NORMALFONTSIZE", "1em");  
define("SMALLFONTSIZE", "0.8em");
```

3.5.14. Changing the color or font of an individual element

Changing the color or font of an individual element means you have to edit *css/webservicesampe.css.php*. There is more explanation in the comments of the file itself.

3.6. Known limitations

In this version of the sample there is no validation of the metadata yet and there is no support yet for dependent metadata.

4. Troubleshooting

The most likely mistakes or oversights:

- Switch Server is not running.
- The user credentials are incorrect. Check them using Switch Client and if necessary change them in Switch Designer.
- The value of ROOT is incorrect in the *settings/settings.php*. Beware of forward and backward slashes and the PHP method of concatenating strings by means of a period.
- The values of STARTPAGE and SWITCHWORKSPACE are incorrect.
- The permissions of the *include/public* folder are not set to read and write for everyone.
- The sample
- Check the php.ini file to see if the following setting have the correct values :

Setting	Value	Remarks
allow_call_time_pass_reference	Off	
always_populate_raw_post_data	Off	
arg_separator.input	&	
arg_separator.output	&	
asp_tags	Off	
file_uploads	On	
upload_max_filesize	2M	Change this if you need to upload bigger files
post_max_size	8M	Change this if you need to upload bigger files
memory_limit	128M	
log_errors	On	
magic_quotes_gpc	Off	
magic_quotes_runtime	Off	
magic_quotes_sybase	Off	
max_file_uploads	20	
max_input_nesting_level	64	
max_input_time	60	
register_argc_argv	Off	
register_globals	Off	
register_long_arrays	Off	
request_order	GP	
variables_order	GPCS	

Setting	Value	Remarks
safe_mode	Off	
short_open_tag	Off	