Bear in mind that this documentation is for the last released version. If you use an older version, you must refer to the README.md file inside the plugin archive.

i-MSCP PhpSwitcher plugin

Provides additional PHP versions for customers.

Requirements

- i-MSCP Serie ≥ 1.4.x
- Either the Fcgid or the PHP-FPM Apache2 httpd server implementation

You can switch between HTTPD server implementations by executing the following commands:

```
cd <path_to_your_imscp_archive>
perl imscp-autoinstall -dar httpd
```

Note that when asked, it is recommended to choose the `per_site` PHP configuration level. This allow your customers to choose a different PHP versions for any of sites.

Installation

1. Be sure that all requirements as stated in the requirements section are met
2. Upload and install the plugin through the plugin management interface
3. Compile and install your PHP versions (see below to know how)
4. Setup your PHP versions (see below to know how)

Update

1. Read the UPDATE file inside the plugin archive
2. Be sure that all requirements as stated in the requirements section are met
3. Upload the plugin through the plugin management interface
4. Re-compile your PHP versions if needed

Glossary

In the documentation below, the following variables are used:

- %version% : One of available PHP version such as 4.4, 5.2, 5.4 ...
- %ymd% : The date of the day such as 20170306
- %installdir% : Installation directory such as /opt/phpswitcher/20170306

Compiling and installing PHP versions

This plugins comes with a Perl script) that is able to downloads, configures, compiles and installs additional PHP versions in one step, without any human interaction. It can also automatically setup PHP versions.

For instance, if you want to install all available PHP versions, you can execute the following command:

```
cd /var/www/imscp/gui/plugins/PhpSwitcher/PhpCompiler
perl php_compiler.pl all
```

or if you want to install one or many specific PHP versions, you can execute the following command:

```
cd /var/www/imscp/gui/plugins/PhpSwitcher/PhpCompiler
perl php_compiler.pl %version% %version%
```

You can also ask the compiler to automatically setup the PHP versions with the `--auto-setup` command line option. For instance:

```
cd /var/www/imscp/gui/plugins/PhpSwitcher/PhpCompiler
perl php_compiler.pl --auto-setup %version% %version%
```

This will automatically configure, build, install and setup the specified PHP versions. Doing this avoid the need to setup all PHP versions manually through the PhpSwitcher administration interface.

To get list of all available command line options, you can execute the following command:

```
cd /var/www/imscp/gui/plugins/PhpSwitcher/PhpCompiler
perl php_compiler.pl --help
```

Supported PHP versions

Supported PHP versions depend on the i-MSCP httpd server implementation in use.

If you use the Fcgid Apache2 i-MSCP httpd server implementation, the following PHP versions are supported:

```
4.4, 5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1
```

If you use the PHP-FPM Apache2 i-MSCP httpd server implementation, the following PHP versions are supported:

```
5.3, 5.4, 5.5, 5.6, 7.0, 7.1
```

Note that the supported PHP versions are the last that were available when this plugin version has been released. This means that by default, the versions provided by the PHP compiler can be lower than the last that were released on the PHP site. In such case, you can ask the PHP compiler to download the last available PHP version by adding the `--force-last` command line option.

However, you must be aware that some patches could not longer apply on these versions. If so, you should report us the failure message by creating a new issue on our bug tracker.

Build and installation directories

By default the new PHP versions are build into the `/usr/local/src/phpswitcher/php%version%` directory, and installed in the `/opt/phpswitcher/%ymd%/php%version%` directory but you can change this behavior with the `--install-dir` command line option.

SSL support for PHP 4.4

PHP 4.4 is built against a bundled OpenSSL (openssl-0.9.8).

SSL support for PHP ≥ 4.4 but < 5.6

For systems on which OpenSSL library is greater or equal to 1.1.0, PHP versions greater than 4.4 but older than 5.6 are built against a bundled OpenSSL (1.0.2l).

Changes made on PHP versions

PHP versions provided by the PHP compiler are almost identical to those that are provided by the Debian team. For each PHP version, a set of patches is applied on PHP upstream source before compiling them. The patches include the following changes:

- Multiarch support
- Usage of system libtool instead of the bundled version (whenever possible)
- Any patch that fix a bug, a security issue or a FTBFS issue

Most of the patches were pulled from the Debian PHP source packages and adjusted when needed, while some other were created to resolve FTBFS issues or adjust configuration. Patches which were not pulled from Debian PHP source packages have the `nxw_` prefix in their names.

Available PHP extensions

Available PHP extensions are the same that are enabled in PHP versions provided by Debian/Ubuntu team. However, it must be noted that some of them can have been disabled for some PHP versions due to incompatibilities with system libraries that are too recent.

Most of PHP extensions are compiled as shared module.

Build dependencies

The PHP compiler installs the build dependencies for you. However, if a package isn't available, it will go ahead and the PHP configuration process will fail.

Parallel Execution (GNU make)

For faster compilation, the parallel execution feature provided by GNU make is enabled whenever possible. This feature allows to execute many recipes simultaneously. By default, the number of

parallel jobs is set according the number of CPU core available on your system. You can still set the value manually with the `--parallel-jobs` command line option as follow:

```
cd /var/www/imscp/gui/plugins/PhpSwitcher/PhpCompiler
perl php_compiler.pl --parallel-jobs 1 %version%
```

Here, only one recipe will be run at time.

See https://www.gnu.org/software/make/manual/html_node/Parallel.html for further details about this feature.

Installation layout

First of all, you must note that in all cases, we build the following PHP SAPIs: `cli`, `cgi` and `fpm`. This allows switching on other i-MSCP httpd server implementation without having to rebuild all PHP versions. This is also needed to build static phpinfo files. Indeed, we always use the PHP FastCGI binary to build them.

For all PHP versions provided by the PHP compiler, the following installation layout applies:

Common to all PHP SAPIs

- Main php.ini file: /opt/phpswitcher/%ymd%/php%version%/etc/php/php.ini
- Additional *.ini files: /opt/phpswitcher/%ymd%/php%version%/etc/php/conf.d

PHP cli SAPI

- PHP CLI binary: /opt/phpswitcher/%ymd%/php%version%/bin/php

PHP cgi SAPI

- PHP FastCGI binary: /opt/phpswitcher/%ymd%/php%version%/bin/php-cgi
- User php.ini files: /var/www/fcgi/%domain%/php%version%/php.ini

PHP fpm SAPI (only when available)

- PHP-FPM binary: /opt/phpswitcher/%ymd%/php%version%/sbin/psw%version%-fpm
- PHP-FPM configuration file: /opt/phpswitcher/%ymd%/php%Version%/etc/php-fpm.conf
- PHP-FPM checkconf script: /usr/local/lib/phpswitcher/psw%version%-fpm-checkconf
- PHP-FPM instance pid file: /var/run/phpswitcher/psw%version%-fpm.pid
- PHP-FPM log file: /var/log/phpswitcher/psw%version%-fpm.log
- PHP-FPM pool configuration files: /opt/phpswitcher/%ymd%/php%Version%/etc/php-fpm.d/%domain%.conf
- PHP-FPM reopenlogs script: /usr/local/lib/phpswitcher/psw%version%-fpm-reopenlogs
- PHP-FPM sysvinit script: /etc/init.d/psw%version%-fpm
- PHP-FPM tmpfile: /etc/tmpfiles.d/psw%version%-fpm.conf

As you can see, we use a specific naming convention for PHP-FPM. This is needed to avoid conflict with PHP versions that are provided by your distribution.

Note that the `psw` prefix stands for `PhpSwitcher`.

PHP configuration

Before going further it is important to note that it is useless to edit any PHP configuration file located under the `/etc/php%version%` or `/etc/php/%version%` directories. Indeed, the files located under these directories are only relevant for the PHP versions provided by your distribution.

For the same reasons, it is useless to try to enable/disable a PHP module using the command line tools phpenmod/phpdismod that are provided by your distribution. Those tools only operate on the INI files that are provided by your distribution.

PEAR packages

Each PHP version comes with its own PEAR distribution that is installed under the `/opt/phpswitcher/%ymd%/php%version%/share/pear` directory. Please don't try to include PEAR library from another location without knowning what you are doing. Doing this could lead to serious problems.

If you want to install additional PEAR packages, you must install them using the `pear` command provided by the PHP version for which you want act. For instance:

```
cd /opt/phpswitcher/%ymd%/php%Version%/bin
./pear install Net_SMTP
```

PHP extensions

For convenience, most of PHP extensions are compiled as shared modules. When installing a new PHP version, the PHP compiler create a specific INI file that enable most of available PHP extensions. By default, this file is located at `/opt/phpswitcher/%ymd%/php%version%/etc/php/conf.d/modules.ini`.

Here, a single INI file is used for ease. This is not like with your distribution PHP packages where an INI file is created for each PHP extension.

To enable/disable a specific PHP extension, you must just edit the INI file and then, restart the Web server or PHP-FPM instance, depending on the i-MSCP httpd server implementation in use.

Pecl extensions

If you want install a Pecl extension, you must not use the `pecl` or `phpize` commands that are shipped with your distribution. Instead, you must use the commands that are provided in the directory of the PHP version for which you want act.

For instance, if you want install the `ImageMagick` Pecl extension you must do as follow:

```
apt-get update && apt-get -y install libmagickwand-dev imagemagick
cd /opt/phpswitcher/%ymd%/php%version%/bin
./pecl install imagick
echo 'extension = imagick.so' >>
/opt/phpswitcher/%ymd%/php%version%/etc/php/conf.d/modules.ini
```

then, you must restart the Web server or PHP-FPM instance, depending on the i-MSCP httpd server implementation in use.

APCu extension

APCu is a userland cache stripped of opcode caching that comes in replacement of the APC extension. It is a good complement for PHP versions ≥ PHP 5.5 that already provide an opcode cache through the OPcache extension.

If you want to install this extension for one of your PHP version, you can do as follow

```
cd /opt/phpswitcher/%ymd%/php%version%/bin
./pecl install channel://pecl.php.net/apcu-%apcu_version%
PHP_EXT_DIR=$(/opt/phpswitcher/%ymd%/php%version%/bin/php-config --
extension-dir)
echo "extension = $PHP_EXT_DIR/apcu.so" >
/opt/phpswitcher/%ymd%/php%version%/etc/php/conf.d/02_apcu.ini
```

then, you must restart the Web server or PHP-FPM instance, depending on the i-MSCP httpd server implementation in use.

Note that the APcu version to install depends on your PHP version

- 4.0.x for PHP 5.5.x and 5.6.x
- 5.1.x for PHP 7.0.x and PHP 7.1.x

IonCube loader

If you want to install this extension for one of your PHP version, you can do as follow (example for the x86-64 version):

```
cd /usr/local/src
wget
http://downloads3.ioncube.com/loader_downloads/ioncube_loaders_lin_x86-64.ta
r.gz
tar -xzf ioncube_loaders_lin_x86-64.tar.gz
cd ioncube
PHP_EXT_DIR=$(/opt/phpswitcher/%ymd%/php%version%/bin/php-config --
extension-dir)
cp ioncube_loader_lin_%version%.so $PHP_EXT_DIR/ioncube.so
echo "zend_extension = $PHP_EXT_DIR/ioncube.so" >
/opt/phpswitcher/%ymd%/php%version%/etc/php/conf.d/01_ioncube.ini
```

then, you must restart the Web server or PHP-FPM instance, depending on the i-MSCP httpd server implementation in use.

phpinfo files

For each PHP version that is setup through the PhpSwitcher administration interface (see below), a static phpinfo file is generated, which allows your customers to get PHP information via their own PhpSwitcher interface. This feature, if not desired, can be disabled by editing the plugin configuration

file and by triggering a plugin list update through the plugin management interface.

Be aware that because the phpinfo files are static, you must re-generate them each time you made a configuration change for a specific PHP version (e.g. when you enable or disable a PHP/Pecl extension). This task can be done through the PhpSwitcher administration interface. Be also aware that because those files are static, some information such as memory could not matches with those that you have set for your customers (e.g. using the PHP editor provided by i-MSCP).

Setup of new PHP versions

Once you have installed new PHP versions, you need to setup them into the PhpSwitcher plugin to make them available for your customers.

Automatic setup

Simply execute the following command:

```
cd var/www/imscp/gui/plugins/PhpSwitcher/PhpCompiler
perl php_compiler.pl --auto-setup --install-dir %installdir%
```

Note that doing this is only required if you do not have used the `--auto-setup` command line option while compiling your PHP versions.

Manual setup

1. Login into the panel as administrator and go to the PhpSwitcher administration interface (settings section)
2. Setup your new PHP versions as follow:

Common settings (Required)

- Path to PEAR directory: /opt/phpswitcher/%ymd%/php%version%/share/pear

PHP FastCGI settings (Required)

- Path to binary: /opt/phpswitcher/%ymd%/php%version%/bin/php-cgi

PHP-FPM settings (Only required if you use the PHP-FPM httpd server implementation)

- Path to binary: /opt/phpswitcher/%ymd%/php%version%/sbin/psw%version%-fpm
- Path to configuration file: /opt/phpswitcher/%ymd%/php%version%/etc/php-fpm.conf
- Path to pool directory: /opt/phpswitcher/%ymd%/php%version%/etc/php-fpm.d

Plugin translation

You can translate this plugin using a gettext translation editor such as `poedit`. Translation files are located under the `./l10n` directory inside of this plugin archive. Once translated you can send us your translation file (po file) for integration in future release.

Note that if no translation file exists for your localization in the `./l10n/po` directory, you must create it first from the l10n/PhpSwitcher.pot file. Be aware that your file must be UTF-8, else, it won't be accepted.

License

```
i-MSCP PhpSwitcher plugin
© 2014-2017 Laurent Declercq <l.declercq@nuxwin.com>
i-MSCP License <https://www.i-mscp.net/license-agreement.html>
```

See the LICENSE file inside the archive for further details.

From:
https://wiki.i-mscp.net/ - **i-MSCP Documentation**

Permanent link:
https://wiki.i-mscp.net/doku.php?id=plugins:phpswitcher&rev=1530965762

Last update: **2018/07/07 13:16**