

Plugin Configuration

Plugin Main Configuration Files

Each plugin can provide a configuration file, which is a simple PHP file that return an array. This file allow the administrator to configure the plugin for its own needs.

When you want modify the configuration for a specific plugin, you must edit it configuration file and trigger a plugin update by clicking on the **Update Plugins** button which is available in the plugin management interface.

You must be aware that the main plugin configuration files are not persistent, meaning that each time you update a plugin to a new version, your changes will be lost. Therefore, to remediate to this problem, you must in order:

1. Make a backup of the plugin configuration file
2. Update the plugin by uploading the new version through the plugin interface
3. Restore your plugin configuration file (Don't forget to compare it with the new version first)
4. Trigger a plugin change by clicking on the **Update Plugins** button.

Plugin Local Configuration Files

Plugin local configuration files doesn't prevent you to check new main plugin configuration files when the plugins get updated. It's your responsibility to check for new parameters and deprecated parameters.

Introduction

Most plugins provide a standardized configuration file (see above) which is located at root of their directory. This file is not persistent, meaning that on each plugin update, you'll have to backup and restore it once the new plugin version will be uploaded.

Because such behavior is not really convenient, the plugin API provide a way to create plugin local configuration files, which are automatically merged with the main configuration files. Those files are persistent and never deleted automatically, even when i-MSCP get updated.

How It Works

When a local configuration file is found for a plugin, both, the main configuration file and the local configuration file are merged together.

Parameters defined in the local configuration files take precedence over those defined in the main configuration files.

Creating Plugin Local Configuration File

All plugin local configuration files must be stored in the **imscp/gui/data/persistent/plugins** directory.

Local configuration files ***MUST*** be named with plugin name followed by the php file extension. For instance, if the plugin for which you want create a local configuration file is named **JailKit**, the local configuration file ***MUST*** be named **JailKit.php**

Plugin Local Configuration File Sample

Below a small sample of a plugin local configuration file which can help you to understand how this works.

Here we want override the following plugin main configuration file:

```
<?php
return array(
    // Jailkit installation directory.
    // This path is used as value of the --prefix option (JailKit configure
    script).
    // IMPORTANT: You must never change this parameter while updating the
    plugin to a new version.
    'install_path' => '/usr/local', // (Recommended value)

    // Full path to the root jail directory which holds all jails. Be sure
    that the partition in which this directory is
    // living has enough space to host the jails.
    // IMPORTANT: You must never change this parameter while updating the
    plugin to a new version.
    'root_jail_dir' => '/home/imscp-jails',

    // See man shells
    // Don't change this value if you do not know what you are doing
    'shell' => '/bin/bash', // (Recommended value)

    // See man jk_init
    'jail_app_sections' => array(
        'imscp-base', // Include Pre-selected sections, users and groups
        'mysql-client'
    ),

    // See man jk_cp
    // Any file which is not installed on your system will be ignored
    'jail_additional_apps' => array(
        '/bin/hostname',
```

```
        '/usr/bin/basename',
        '/usr/bin/dircolors',
        '/usr/bin/dirname',
        '/usr/bin/clear_console',
        '/usr/bin/env',
        '/usr/bin/id',
        '/usr/bin/groups',
        '/usr/bin/lesspipe',
        '/usr/bin/tput',
        '/usr/bin/which'
    ),

    // See man jk_socketd
    'jail_socketd_base' => '512',
    'jail_socketd_peak' => '2048',
    'jail_socketd_interval' => '5.0'
);
```

We create the following local configuration file:

gui/data/persistent/plugins/JailKit.php file:

```
<?php
return array(
    // Override default jail roor directory
    'root_jail_dir' => '/var/www/imsdp-jails',

    // Append the git section to the jail_app_sections parameter
    'jail_app_sections' => array(
        'git'
    )
);
```

Important

Configuration items from main plugin configuration files are never removed automatically (this is by design). To remove an item, you must process as follow:

gui/data/persistent/plugins/JailKit.php file:

```
<?php
return array(
    '__OVERRIDE__' => array(
        // Override default jail roor directory
        'root_jail_dir' => '/var/www/imsdp-jails',

        // Append the git section to the jail_app_sections parameter
        'jail_app_sections' => array(
            'git'
        )
    ),
);
```

```
'__REMOVE__' => array(
    // Remove mysql-client section from the jail_app_section parameter
    'jail_apps_sections' => array(
        'mysql-client'
    ),

    // Remove hostname command from the jail_additional_apps parameter
    'jail_additional_apps' => array(
        '/bin/hostname'
    )
);
```

Here, the special array key **__OVERRIDE__** defines an array which contain elements to add/override, and the second special array key **__REMOVE__**, an array which contain elements to remove.

— [Nuxwin](#) 2014/02/24 06:00

From:

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

Permanent link:

<https://wiki.i-mscp.net/doku.php?id=plugins:configuration&rev=1393218330>



Last update: **2014/02/24 05:05**