

**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 RoundcubePlugins plugin

Provides plugins for the Roundcube Webmail Suite.

## Requirements

- i-MSCP 1.5.x Serie (version  $\geq$  1.5.2)
- Roundcube  $\geq$  1.3.x

## Installation

1. Be sure that all requirements are met
2. Upload the plugin through the plugin management interface
3. Activate the plugin through the plugin management interface

## Update

1. Be sure that all requirements are met
2. Backup your plugin configuration file if needed
3. Upload the plugin through the plugin management interface
4. Restore your plugin configuration file if needed
5. Update the plugin list through the plugin management interface

## Configuration

See the [configuration](#) file for a fast overview.

When changing a configuration parameter in the plugin configuration file, you must not forget to trigger a plugin list update through the plugin management interface, else your changes won't be taken into account.

### Roundcube plugin definitions

This plugin make it possible to install any Roundcube plugin by adding plugin definitions in the plugin [configuration file](#).

#### Basic Roundcube plugin definition

A basic Roundcube plugin definition looks as follows:

```
'name' => [  
    'enabled' => true
```

]

Here we provide a definition for the name plugin and we tell that it must be enabled.

### Installing a Roundcube plugin through the PHP dependency manager (Composer)

Most of time, you will have to go one step further by telling how to get and install the plugin. This can be done by adding a composer section as follows:

```
'name' => [
    ...
    'composer' => [
        'require' => [
            'vendor/name' => '^1.0'
        ]
    ]
]
```

Basically, a composer section is what you would put in a [composer.json](#) file to make the PHP dependency manager (Composer) able to download and install a PHP dependency, here, a Roundcube plugin.

In the example above, we require the vendor/name composer package.

You can learn further about composer by looking at the official [documentation](#).

### Roundcube plugins not available through plugins.roundcube.net nor through packagist.org

Sometime, a plugin is not available through official [Roundcube plugin](#) repository, nor through [packagist.org](#) repository. In such a case and if the plugin provides a composer.json file, you can define your own repository as follows:

```
'name' => [
    ...
    'composer' => [
        'repositories' => [
            'type' => 'path',
            'url' => PERSISTENT_PATH .
'/plugins/RoundcubePlugins/name',
        ],
        ...
    ]
]
```

In the above example, we define a path repository in which composer will look for the name plugin. We assume here that the Roundcube plugin has been downloaded manually under the <PERSISTENT\_PATH>/plugins/RoundcubePlugins/name directory.

### Getting a Roundcube plugin from a Git repository

It is possible to grab a plugin from a Git repository automatically by adding a git section as follows:

```
'name' => [
    ...
    'git' => [
        'repository' => 'https://domain.tld/vendor/name.git'
    ],
    ...
]
```

By doing this, the repository will be automatically cloned under the  
 <PERSISTENT\_PATH>/plugins/RoundcubePlugins/name directory (eg:  
 /var/www/imsdp/gui/data/persistent/plugins/RoundcubePlugins/repository\_name)

However, note that usage of the git section above is only needed if the Git repository holds more than one plugin. Most of time, you can do the same thing through composer, using a [VCS](#) repository.

For instance:

```
'name' => [
    ...
    'composer' => [
        'repositories' => [
            'type' => 'vcs',
            'url' => "https://domain.tld/vendor/name.git"
        ],
        ...
    ]
]
```

## Roundcube plugin configuration

A last step is needed if you want to override default Roundcube plugin parameters. To do so, you can add a config section as follow:

```
'name' => [
    ...
    'config' => [
        'file' => 'config.inc.php',
        'include_file' => '/foo/bar/baz.php',
        'parameters' => [
            'param1' => 'value',
            'param2' => 'value'
        ]
    ]
]
```

where:

- `file` is an OPTIONAL parameter that allows to override default plugin configuration template filename
- `include_file` is an OPTIONAL parameter that allows to provide the path of a file that will be included in the Roundcube plugin configuration file. Note that the file must be readable by the

control panel user (eg: vu2000). A good place for that file could be the `<PERSISTENT_PATH>/plugins/RoundcubePlugins` directory.

- `parameters`: is an OPTIONAL parameter that allows to provide plugin parameters. Those parameters will be added at the end of the plugin configuration file

To resume, if you want to override a Roundcube plugin configuration parameter, you must not edit its configuration file directly. Instead, you must process as above. This is needed because the i-MSCP Roundcube package doesn't save your changes on reconfiguration/update.

### Configuration script

Sometime a plugin can require more configuration works. For that purpose, you can provide your own configuration script as follows

```
'name' => [
    ...
    'config' => [
        ...
        'script' => '/foo/bar/baz.pl'
    ]
]
```

In the above example, we provide the `/foo/bar/baz.pl` Perl script. That script will be executed automatically with the current action passed as first argument:

- On Roundcube plugin pre-configuration: `/foo/bar/baz.pl pre-configure`
- On Roundcube plugin configuration: `/foo/bar/baz.pl configure`
- On Roundcube pre-deconfiguration: `/foo/bar/baz.pl pre-deconfigure`
- On Roundcube deconfiguration: `/foo/bar/baz.pl deconfigure`

Note that the script must provides a correct [shebang](#).

For instance:

- `#!/usr/bin/env perl` for a Perl script
- `#!/usr/bin/env php` for a PHP script
- `#!/bin/sh` for a shell script
- ...

See [configure-managesieve.pl](#) for an example.

## List of default Roundcube plugin definitions

- `additional_message_headers`: Add additional headers to or remove them from outgoing messages.
- `archive`: Adds a button to move the selected messages to an archive folder.
- `calendar`: Provide calendaring features.
- `contextmenu`: Creates context menus for various parts of Roundcube using commands from the toolbars.
- `emoticons`: Adds emoticons support.
- `logon_page`: Logon screen additions.
- `managesieve`: Adds a possibility to manage Sieve scripts (incoming mail filters).

- newmail\_notifier: Provide notifications for new emails.
- odfviewer: Open Document Viewer plugin.
- pdfviewer: Inline PDF viewer plugin.
- password: Password Change for Roundcube.
- rcguard: Enforces reCAPTCHA for users that have too many failed logins
- tasklist: Task management plugin.
- vcard\_attachments: Detects vCard attachments and allows to add them to address book...
- zipdownload: Adds an option to download all attachments to a message in one zip file...

## License

i-MSCP - internet Multi Server Control Panel  
Copyright (C) 2017 Laurent Declercq <l.declercq@nuxwin.com>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

See [LICENSE](#)

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

Permanent link:  
<https://wiki.i-mscp.net/doku.php?id=plugins:roundcubeplugins&rev=1509240941>



Last update: **2017/10/29 01:35**