

# How to customize SlapOS

Note : as of the 2011/09/05, this is work in progress.

by [SlapOS Team](#).

▼ Details

You can view this tutorial in [presentation mode](#).

## Agenda

- SlapOS Internals
- How to customize a SlapOS component
- How to customize the LAMP recipe
- How to debug an instance

## Requirements

- Good understanding of Buildout (via tutorials on [buildout.org](#))
- SlapOS Tutorials (day 1)
- SlapOS PHPMyAdmin Tutorial
- Git knowledge

## SlapOS internals

Same as Buildout

- Profile = Software Release (software.cfg)
- Recipe = slapos.cookbook

▼ Details

## Profiles

SlapOS uses Buildout to install software and deploy instances.

In SlapOS, a Software Release is defined by its URL. It is in fact a Buildout Profile. When SlapOS installs a Software Release, it launches Buildout with this URL.

A Software Release file, named software.cfg, can reuse other Buildout profiles using the extends mechanism. It allows you to create a software.cfg in a few lines.

Many different profiles exist in SlapOS. You can use components that are utilities and/or binaries like cURL, PHP, Apache, Java. You can also use profiles installing libraries like libxml, to fulfil requirements.

If you need, you can also use what we call "stacks". A stack is a group of components working together, in a reusable way. For example, when creating a PHP Software Release, you will extend the "LAMP" stack, containing all what is needed to run PHP applications (Apache, MySQL, etc) so that you can define only what is specific to your software in a few lines.

## Recipes

In order to deploy Software Instances, SlapOS uses custom recipes. If you have read the Buildout tutorial, you know that recipes are Python Eggs. Recipes achieving high quality level are included in a single Egg : [slapos.cookbook](#). You can of course use your own egg, usually mycompanyname.recipe.myinstancename.

## Git repository architecture

- /component
- /stack
- /software
- /slapos/recipe

▼ Details

The official git repository is separated in several directories. Each category defined in the previous slide can be found here.

## How to customize a SlapOS component

## Use cases

- I need to enable XXX fonctionnality in PHP at compile time
- My Software Release needs a dependency that is not in SlapOS to work

## Check extends

From webrunner : modify the extend list to use your git repository.

### ▼ Details

The "lamp-template" Software Release is ready to use from webrunner during development. It will extend everything from official git repository, git.erp5.org.

If you need to modify or add components (or stack), you will have to change the extends URL to reference your git repository at gitorious.org. When you modify, commit and push some changes in your repository, changes will apply when you rerun the software.cfg on your webrunner instance.

## Modify, check changes

### ▼ Details

## Use webrunner as a sandbox

When you want to introduce a change in a component, say a new configure parameter in apache-php, you can test it directly by copying the interesting part (here [apache-php]) from the profile (here component/apache-php/buildout.cfg) to the software.cfg in webrunner. This way you can test every change without having to commit.

When you are happy with the changes, just cut the [apache-php] from webrunner's software.cfg.

Paste, commit and push the changes in component/apache-php/buildout.cfg in you local git repository. It will be available in gitorious.org, and by side effect in the software.cfg from webrunner also (from extends). Run the software from webrunner and see if it still works.

Of course, it can apply to every buildout profile you want.

## How to customize the LAMP recipe to fit your needs

### Use cases

- I need to create several configuration files for my PHP software
- I need to customize the PHP configuration file
- I need to remove/add/change files at runtime, after installation

## Use recipe from git repository

Add the following code in webrunner software.cfg, under the last "extends" line:

[buildout.cfg](#)

```
slaposcookbook_recipe_location = ${:parts-directory}/slaposcookbook

develop =
    ${:slaposcookbook_recipe_location}

[slaposcookbook]
recipe = plone.recipe.command
repository = http://git.erp5.org/repos/slapos.git
git_param = --quiet
git = ${:git:location}/bin/git
location = ${buildout:parts-directory}/${:_buildout_section_name_}
stop-on-error = true
command = ${:git} clone ${:git_param} ${:repository} ${:location} || exit 1;
update-command = cd ${:location} && ${:git:location}/bin/git pull --quiet
```

### ▼ Details

When a software.cfg references a module from SlapOS.cookbook, all the egg will be downloaded from standards location such as pypi.python.org. It means that only the last known stable version will be downloaded and used.

When modifying and developping a recipe, you want Buildout to ignore standard location and use your gitorious repository instead, so that you will be able to check your changes.

We use a dirty hack to do that. You just need to specify the "clone" URL of your repository in the "repository" parameter of the [slaposcookbook] part.

## Modify recipe, check changes

- Modify your recipe
- Check syntax with pyflakes
- git commit, git push
- On webrunner : run software, run instance
- Test changes

## Debug your SlapOS instance

Agenda:

- TODO
- Only from computer where you can login
- Check supervisord status
- Check logs
- Run it manually