

# User manual

## Structure101g PHP flavor

Version 1.0 (Rev. 2)

## Table of contents

<b>Introduction</b>	<b>3</b>
<b>1. Installation</b>	<b>4</b>
1.1. Installing Structure101g	4
1.2. Installing the PHP-Flavor	5
<b>2. Using the PHP-Flavor</b>	<b>6</b>
2.1.1. Configuring the static PHP-Parser	6
2.1.2. Configuring the XDebug-Parser	7

## Introduction

The Structure 101 PHP-flavor is a plugin for Structure101g. It adds the capability to analyze the architecture of PHP-code to s101g. To collect the needed data, it uses a combination of static and dynamic analysis. The dynamic part makes use of the well-known PHP-debugger XDebug, which provides information about the runtime behavior of a PHP-project. Although the flavor may produce an output with static analysis only, it is highly recommended to include the dynamic analysis for optimal results.

# 1. Installation

## 1.1. Installing Structure101g

To use the PHP-flavor, Structure101g needs to be installed on your system. You can download it at the Headway website <http://www.headwaysoftware.com>

After installing, you need to place the license files you received via email in your Structure101g user directory. The directory can be found at

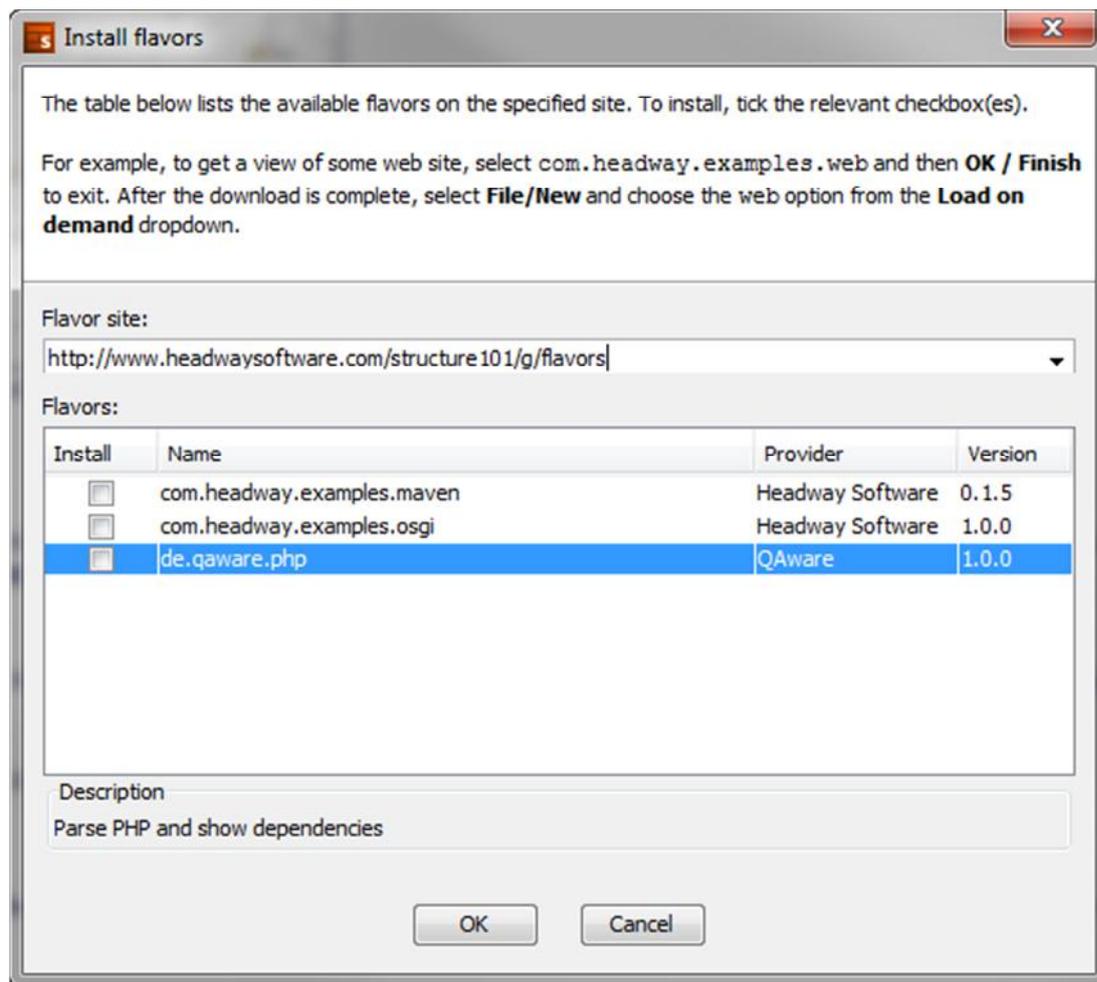
<USER>\Structure101\gen
-------------------------

**NOTE:** Older versions of Structure101 may need the licenses in their installation directory instead of the user directory.

Additional licenses may be bought at <http://www.headwaysoftware.com/products/purchase.php>

## 1.2. Installing the PHP-Flavor

To install the s101 PHP-flavor, simply start Structure 101g and on the menu bar click on Flavours->Install...



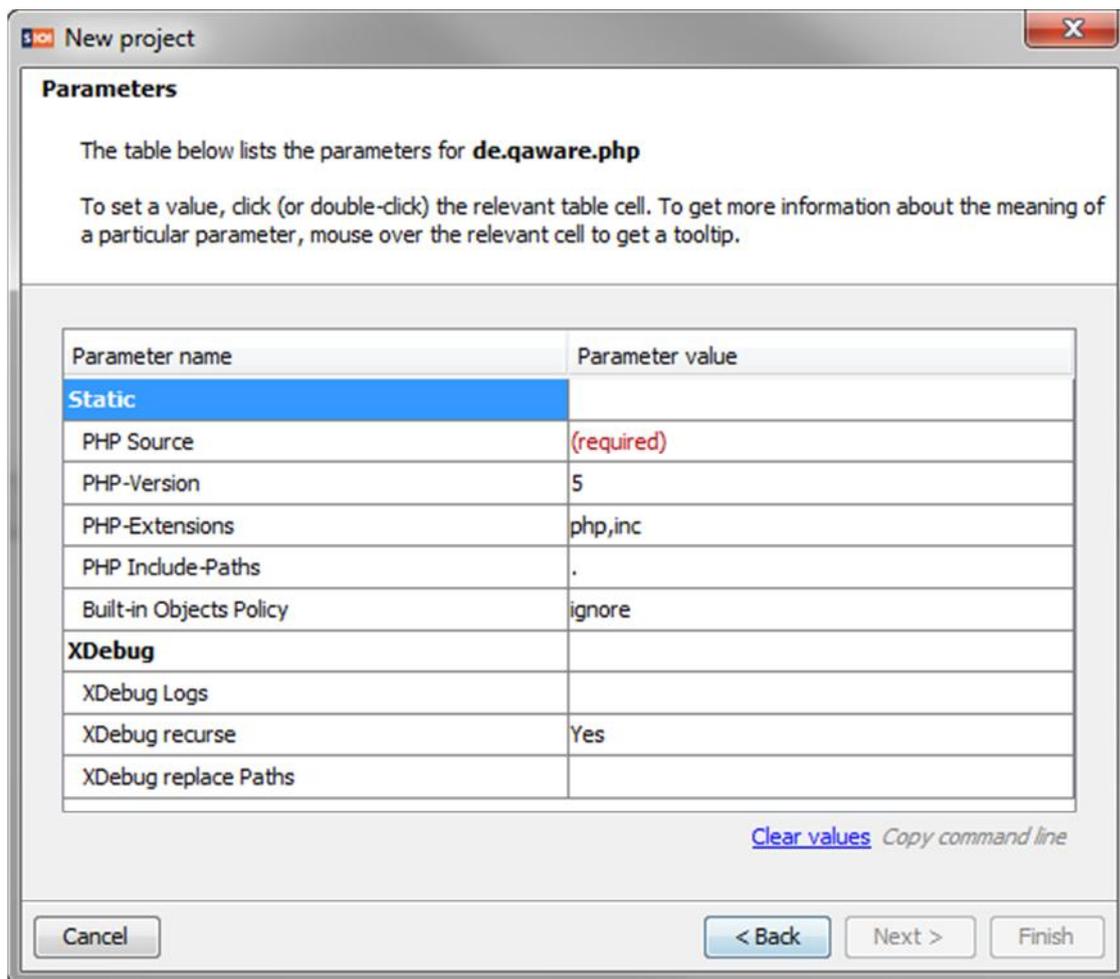
In the Menu, select the `de.qaware.php` flavor and press the OK button. If the php flavor is not in the list, try changing the Flavor site to <http://www.headwaysoftware.com/structure101g/flavors>

To verify the installation, start structure101g and click on *File->New*. You should be able to select the `de.qaware.php` flavor from the list of available “Load on demand” flavors.

## 2. Using the PHP-Flavor

To start an Analysis, click on *File->New* and select the *de.qaware.php*-Flavor from the list of “load on demand” flavors.

After pressing next, you are presented with various parameters to configure. The following section describes the function of each parameter.



### 2.1.1. Configuring the static PHP-Parser

This section describes the parameters for the static analysis:

#### PHP Source

This parameter selects the root-directory of the PHP-project to analyze. The static PHP-Parser will scan all files in this and all subdirectories. Please note, that hidden files and directories are always ignored by the static PHP-parser. This parameter is **required**.

## PHP Version

This parameter describes the PHP-Version of the source files being analyzed. Possible values are “4”, “5” and “5.3”. Defaults to “5”.

## PHP Extensions

A comma separated list of file extensions which should be analyzed. The PHP-Parser will only analyze files with one of the given extensions. Defaults to “php,inc”.

## PHP Include-Paths

A comma separated list of include-paths relative to the path given in the PHP Source parameter. This parameter is necessary to resolve include and require operations in the PHP source code. Defaults to “.”.

## Built-in Objects Policy

Determines how references to built-in PHP objects are handled. Valid values are:

- **Ignore:** Ignore any references to (or from) built-in objects.
- **Detailed:** Built-in objects as well as any references with built-in objects are written to the output file as if they were user defined objects.
- **SingleTarget:** Built-in objects as well as any references with built-in objects are written to the output file as if they were one single object/function.

### 2.1.2. Configuring the XDebug-Parser

This section describes the parameters for the dynamic analysis:

#### XDebug Logs

A File or Folder containing the XDebug-logs to analyze. If not set, the dynamic analysis is omitted.

#### XDebug recurse

If the XDebug Logs parameter is set to a folder and this parameter is set, the folder and all its subfolders will be parsed. If this option is set to false, only the main folder will be scanned for XDebug logs.

#### XDebug Replace Paths

Gives a list of replacement paths in the form 'path1=>replacement1,path2=>replacement2...'. Any string 'path' that occurs at the beginning of a file path in an XDebug log is replaced by the corresponding 'replacement' prior to resolving the file. You may need this, since XDebug profile logs contain absolute paths or are generated on another machine, or the PHP source files are parsed at another location than the execution directory.