



Doxygen

Issa Mahasneh
servit Open Source Solutions

Coding Standards

- Rules that computer programmers follow to help ensure that their source code is easy to **read** and **maintain**.



Standards make sure that your code is:

- 1- Readable
- 2- Maintainable

Why Should We Use Standards?

- Software Maintenance
 - Quality
 - Refactoring
 - Task Automation
- 80% of the lifetime cost of a piece of software goes to maintenance.
 - Hardly any software is maintained for its whole life by the original author.
 - Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly.
 - If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create.

Drupal Code Standards

- Based on PEAR Coding Standards (<http://pear.php.net/manual/>)

Indentation

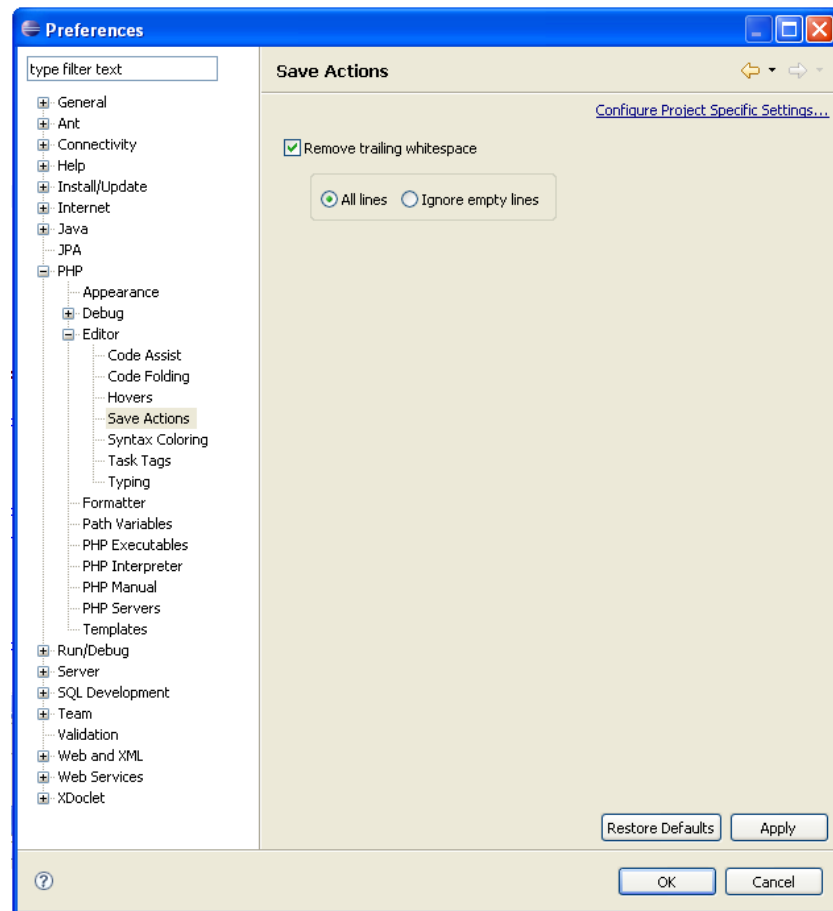
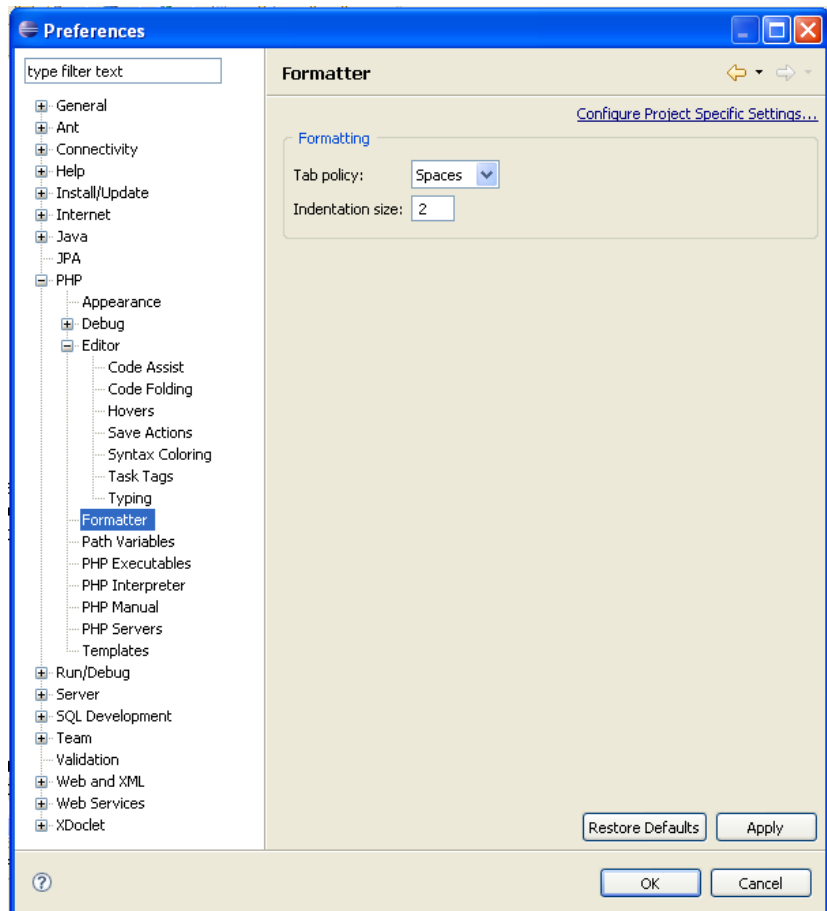
- Use an indent of 2 spaces, with **no tabs**.
- No trailing whitespace.

```
function mymodule_page() {  
switch (condition) {  
case 1:  
action1;  
break;  
case 2:  
action2;  
break;  
default:  
defaultaction();  
}  
}
```

```
function mymodule_page() {  
■ switch (condition) {  
■■■ case 1:  
■■■■■ action1;  
■■■■■ break;  
■■■ case 2:  
■■■■■ action2;  
■■■■■ break;  
■■■ default:  
■■■ defaultaction();  
■■ }  
}
```

Indentation (Eclipse)

- Window > Preferences



Control Structures

- Put a space between control structure and parenthesis.
- Use curly braces even if optional.

```
<?php
if ( (condition1) || (condition2) ) {
    action1;
} elseif ( (condition3) && (condition4) ) {
    action2;
} else {
    defaultaction;
}
?>
```

```
switch (condition) {
    case 1:
        action1;
        break;

    case 2:
        action2;
        break;

    default:
        defaultaction;
}
```

Function Calls

- No spaces – Spaces only between parameters and commas.
 - `$var = foo($bar, $baz, $quux); // CORRECT`
 - `$var = foo($bar, $baz, $quux); // WRONG`
- Block of Assignments:
 - `$short = foo($bar);`
 - `$long_variable = foo($baz);`

Function Declarations

- Arguments with default values go at the end of the argument list.
- Always attempt to return a meaningful value from a function if one is appropriate.

```
function man_said($name, $word = 'Hello!') {  
    return $name . ' says ' . $word;  
}
```

Arrays

- Arrays should be formatted with a space separating each element and assignment operator, if applicable:

```
$some_array = array('hello', 'world', 'foo' => 'bar');
```

Arrays

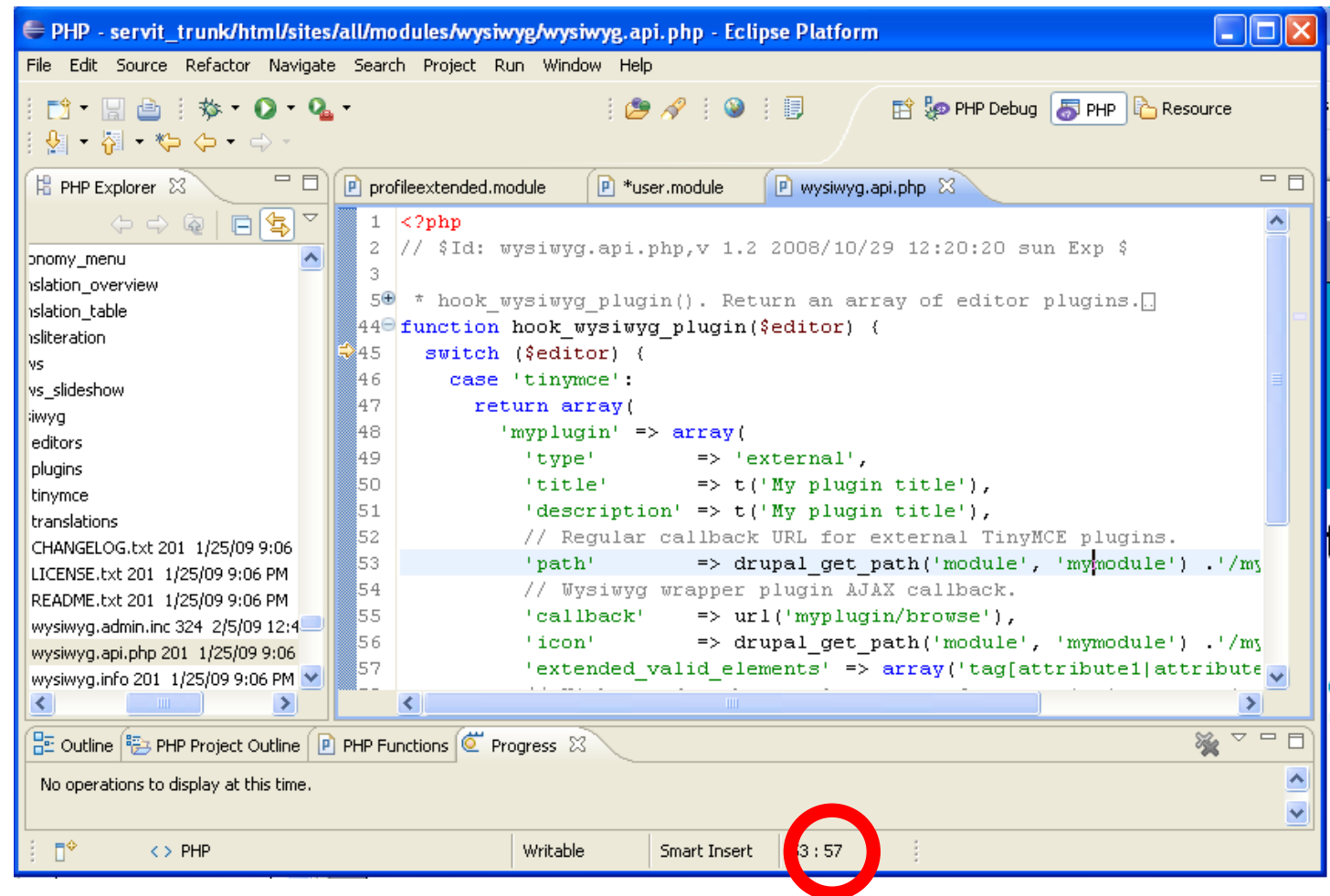
- If line length > 80 characters, break the elements:

01234567890123456789012345678901234567890123456789012345678901234567890123456789

```
#####  
$form['title'] = array(  
  '#type' => 'textfield',  
  '#title' => t('Title'),  
  '#size' => 60,  
  '#maxlength' => 128,  
  '#description' => t('The title of your node.'),  
);
```

Line Length (Eclipse)

- Press **End** button to move the cursor at the end of the line.



Quotes (" and ')

- No preferences.
- Single quotes faster (')
- Single quotes recommended
 - Exceptions:
 - Inline variables:
"Sorry. Name \$name is already taken" better than
'Sorry. Name' . \$name . 'is already taken.'
 - Translated strings with double quotes:
"He's my friend" better than
'He\'s my friend'

String Concatenation

- Use spaces between strings and dot operators.

```
$string = 'Foo' . $bar;  
$string = $bar . 'foo';  
$string = bar() . 'foo';  
$string = 'foo' . 'bar';
```

- Use double quotes with simple variables.

```
$string = "Foo $bar";
```

- Use spaces on sides of `. =` operator

```
$string . = 'Foo';  
$string . = $bar;  
$string . = baz();
```

Documentation

- Doxygen Formatting Conventions

- Block of code:

```
/**  
 * Documentation here.  
 */
```

- Start of Files:

```
<?php  
// $Id: theme.inc,v 1.202 2004/07/08 16:08:21 dries Exp $  
  
/**  
 * @file  
 * The theme system, which controls the output of Drupal.  
 *  
 * The theme system allows for nearly all output of the Drupal system  
 * to be customized by user themes.  
 */
```

Documentation

- **Functions:**

```
/**
 * Verify the syntax of the given e-mail address.
 *
 * Empty e-mail addresses are allowed. See RFC 2822 for
 * details.
 *
 * @param $mail
 *   A string containing an email address.
 * @return
 *   TRUE if the address is in a valid format.
 */
function valid_email_address($mail) {
```

- **“Easy” Functions:**

```
/**
 * Convert an associative array to an anonymous object.
 */
function array2object($array) {
```


Documentation

- Documentation conventions for:
 - Drupal hooks.
 - Theme templates.
 - Theme functions.
 - Forms.
- Available at:
<http://drupal.org/node/1354>

Comments

- Non-documentation comments encouraged.
- Non-documentation comments should use capitalized sentences with punctuation.
- Sentences should be separated by single spaces.
- All caps are used only when referencing constants, for example `TRUE`.
- Comments should be on a separate line immediately before the code.
- For example:

```
// Unselect all other contact categories. ■ Uses built-in method.  
db_query('UPDATE {contact} SET selected = 0');
```

PHP Code Tags

- Use `<?php ?>` not `<? ?>`
- Do not use `?>` at the end of the file.

Example URLs

- RFC2026 Asks to use:
<http://www.example.com>

Naming Conventions

- Functions:
 - Use lowercase
 - Use underscore
 - Private class members start with `_`
- Constants:
 - Uppercase: `FALSE`, `$PI`
- Global Variables:
 - `_mytheme_var_`

Naming Conventions

- Class Names:
`className`
- File Names:
 - `README.txt`

Coder Module

- Install the module just like others.

admin

Code review

• Reviews

[Converting 4.6.x modules to 4.7.x](#)

[Converting 4.7.x modules to 5.x](#)

[Converting 5.x modules to 6.x](#)

[Drupal Coding Standards](#) (every developer should use)

[Drupal Commenting Standards](#) (every developer should use)

[Drupal SQL Standards](#) (new review, so use with caution)

[Drupal Security Checks](#) (very basic, needs work, but what it finds is good)

apply the checked coding reviews

minor (most)

normal

critical (fewest)

show warnings at or above the severity warning level

Coder Module

▼ What to review

- active modules and themes
- core files (php, modules, and includes)
- include files (.inc and .php files)

▼ Select Specific Modules

| | |
|--|------------------------------------|
| <input type="checkbox"/> FixedDataDropdown | <input type="checkbox"/> geo_devel |
| <input type="checkbox"/> activeselect (active) | <input type="checkbox"/> geo_gmap |
| <input type="checkbox"/> aggregator (core) | <input type="checkbox"/> getid3 |
| <input checked="" type="checkbox"/> bio (active) | <input type="checkbox"/> gmap |



▼ sites/all/modules/bio/bio.install

bio.install

- Line 30: Use an indent of 2 spaces, with no tabs

```
drupal_set_message(t('You can edit Bio settings at ')) .
```
- Line 30: missing space after comma

```
drupal_set_message(t('You can edit Bio settings at ')) .
```