


PHP coding standard for eZ publish



27th July 2001

 **eZ systems** The double squares and eZ are trademarks belonging to eZ systems of Norway, registration number NO 981 601 564 (<http://www.brreg.no/oppslag/enhet/detalj.ssc?orgnr=981601564>). eZ publish is a software package released under the GPL licence (<http://www.gnu.org/copyleft/gpl.html>), its primary point of distribution and information is <http://developer.ez.no/>

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 2 | Files | 4 |
| 2.1 | Headers | 4 |
| 3 | PHP | 5 |
| 3.1 | Indenting | 5 |
| 3.2 | Placement of brackets | 5 |
| 3.3 | Parenthesis, brackets and comma | 5 |
| 3.4 | If and while statements | 6 |
| 3.5 | Naming | 6 |
| 3.5.1 | Template naming | 7 |
| 3.6 | Functions | 7 |
| 3.7 | Classes | 7 |
| 3.8 | Commenting | 8 |
| 4 | SQL | 9 |
| 4.1 | Syntax | 9 |
| 4.2 | Tables | 9 |
| 5 | HTML / XHTML | 10 |
| 5.1 | Syntax | 10 |

Introduction

This coding standard defines how the code in eZ publish should be formatted. The code in eZ publish must follow this standard to ensure a coherent looking code base.

To easier comply with this coding standard we recommend using emacs with our configuration files which handles proper indenting and other nice features. You can get it from: <http://developer.ez.no/filemanager/list/31/>

Files

All files uses lower case. PHP files are named .php, SQL dump files are named .sql, template files are named .tpl and language/setting files are named .ini.

2.1 Headers

All files should

PHP

3.1 Indenting

Indenting is 4 characters, real tab characters should not be used. Spaces are used for indenting.

3.2 Placement of brackets

Start and end brackets should be placed on the same column.

Examples:

```
multiply( $x, $y )
{
    // function body
}
if ( x == y )
{
    ..
}
else if ( x > y )
{
    ..
}
else
{
    ..
}
class foo
{
    function foo( )
    {
        ..
    }
}
```

3.3 Parenthesis, brackets and comma

It should always be a space after a start parenthesis and one space before an end parenthesis. One space is placed after every comma. Square brackets [] should not have space after the start bracket and before the end bracket. It should be a space between each type.

Empty lines should be used to group variables and code lines which have a connection. Variable names which is grouped should be indented to the same level.

Empty lines should not only appear empty, but they should also be empty, all tab and space characters should be removed.

Use more parenthesis than necessary if you're not sure on the sequence the expression is executed.

3.4 If and while statements

Nested if statements should always use {}. If you have a lot of if, else statements these should probably be replaced with case statements, or it should be redesign to use object oriented methods. If and while statements should be constructed of logic statements, no assignment or changing of data should occur.

3.5 Naming

All names should be English and grammatically correct. Names like "foo" and "tmp" should be avoided since they do not describe anything. Names should not contain numbers, numbers should be described with letters unless there is a good reason for it.

Example:

```
$ValueOne
```

Function names can be constructed by several words, all words should have lower case unless the first letter which should be a capital letter. Exception for this is the first letter which always should be in lower case. If the first letter is part of an abbreviation the whole abbreviation should be in lower case.

Example:

```
setText(. . );  
id();  
setID();  
getTheValuesForThis();
```

Constants should be constructed by only capital letters, _ should be used to separate words.

Example:

```
define( "MY_CONSTANT", 42 );
```

Member variables can be constructed of several words, every word is spelled in lower case with capital first letter. It should not be too many words, 1-3 is normal, they can be abbreviated as long as it's understandable.

Example:

```
$Len;  
$X, $Y;  
$FirstName
```

Local variables can also consist of several words with lower case, the first character is a capital letter except the first word. The number of words should be as few as possible and the words can be abbreviated.

```
$index, $x, $y, i;  
$xAdd;  
$firstName, $lastName;
```

Parameters are names like local variables.

Class names can consist of several words, every word should be in lower case with the first letter in capital. All class names should begin with eZ.

Example:

```
eZUser
eZArticleGenerator
```

Global variables are not used, place them in a class and make them static.

3.5.1 Template naming

eZ publish uses templates to define the XHTML which will be presented as a webpage. These templates contain variables. There are two types of variables, POST variables and template variables.

The syntax for template variables are as follows:

```
{value_one}
{first_name}
{long_variable_with_good_description}
{category_id}
```

POST variables should have the following syntax (since they will be global variables in the php receiver script):

```
FirstName
ValueOne
CategoryID
LongVariable
```

3.6 Functions

All functions should be placed in classes unless there is a good reason not to.

Functions should be short and do one thing. The length of a function depends on the complexity of the function, the more complex the function is the shorter should the function be. If it gets too complex then divide the function up into helper functions.

The number of local variables in a function should not be larger than 7, 7 is a known limit to the number of things people can control at one time. Split up the function if it gets too large.

Default parameters should be used with care.

You should let the parameter line go over several lines with grouping of the similar parameters, if it gets long.

3.7 Classes

Classes should have one responsibility. All member variables are private¹. Help functions should also be private². Functions which does not access member variables should be static³.

¹Since PHP does not have private variables it is up to the programmer to follow these rules.

²PHP does not allow private member functions, but this should be commented in the function header with the `\private` tag.

³PHP does not differentiate between static and non static functions so this should be commented in the function header with the `\static` tag. A static function doesn't use the `$this` variable.

3.8 Commenting

The comments used should be written to comply with the syntax of the eZ phpdoc tool. This tool is used to generate class reference documentation which is available at <http://doc.ez.no>.

```
    ///! Module Name
    ///! A brief description of the class
    /*!
        This is a more complete description of the class.
        Here is an code example:
        \code
        $myObj = new MyClass();
        \endcode
        See also theese classes:
        \sa eZUser, eZArticle
    */
    /*!TODO
        First thing to do.
        Second thing to do.
        Yet, another thing to do.
    */
class MyClass
{
    /*!
        This is the constructor of MyClass. It handles....
    */
    function MyClass()
    {
        ...
    }
    /*!
        \private
        This is a private helper function.
    */
    function doSomething()
    {
        ...
    }
    ///! This is a variable description.
    var MyVar;
}
```

SQL

4.1 Syntax

SQL commands are written in all capital letters.

```
SELECT * FROM Person
WHERE ID= '$personID'
HAVING Age>'42'
LIMIT 0,20;
```

4.2 Tables

Table names should be in English. The names should be grammatically correct. Table names with several words should be separated with capital letters. The tables should be prefixed with the module name, e.g. eZUser.

Example:

```
eZUser_User
eZArticle_Category
eZTrade_ProductCategoryLink
```

Row names should be named with capital first letters for each word.

Example:

```
eZUser_User
{
  ID int,
  FirstName char,
  LastName char,
  Login char
}
```

HTML / XHTML

All HTML should be written in XHTML 1.0 (or higher). It should also be written with no browser specific tags, this is to ensure that the application works in as many browsers as possible. Java/Ecma script should be avoided. Frames is not recommended, eZ publish is designed to work without frames.

Reasons for not using frames are:

- Hard to bookmark
- Hard to print
- Does not work correctly in all browsers

5.1 Syntax

HTML code should be written in lower case.