BRANTECT

Development Coding Conventions

2013/5/10

Daisuke Tsujino

The content below is summary of the Coding Conventions used in the development of BRANTECT. During the decision making process of conventions, it is aimed to protect high level code quality where multiple developers are involved. Please perform coding based on content of these conventions. However, if there is any exceptional case to occur during actual development, there will be update of conventions to satisfy the necessities.

1. **Naming Conventions**
2. File Name

Lower-case alphanumeric numbering and underscore are being used. Php extension is used. The format of class definition files are as [class.php].

1. Variable Name

Lower-case alphanumeric numbering and underscore are being used. Words are separated with underscore. However, it is preferred not to use numerical values.

Variable names are assigned with enough detail to easily understand the meaning. (In a realistic range)

$company\_name

However, for the variables that are used only in a short loops, declarations such as $i can be done. For a variable name that represents an array, beginning assigned as “is” and ending as “flg”.

1. Class Name

Upper-case alphanumeric numbering is used. However, we try not to use numbers as much as possible.

For class name, meaningful naming that easily expresses that class is used. For example for table definition, a name of that table is assigned.

class DeDesignHeader {

}

1. Method Name, Function Name

Upper-case alphanumeric numbering is used. First character is lowercase, for the words more than 2 words, first character is uppercase.

It is recommended that, you put one word related to what method does.

public function findByKeyword()

1. Constant Name

Upper-case alphanumeric numbering and underscore is used. Words are separated with underscore. For expressing the meaning, a detail named is given where it was possible.

define('TM\_CUSTOM\_SETTING\_TEXT', '01');

In coding, mainly define the constant without writing the number directly. Define the constant at its scope as definition location. As an example, please write the constant that is only used in file, at the beginning of that file; and for the ones that is in use at Trademark (tm) module at tm/lib/defines.php. And write at lib/system.ini for using at whole system.

1. **Coding Style**
2. Start Tag

“<?php” is used. Please avoid short tag usage.

1. Indentation

When adding to code block (for, function, etc.), there is definitely indentation.

1. Processing Blocks

Even if there is only one “if”, “else”, “while” etc. processing blocks, is enclosed with [{}].

1. Commenting

About the notation of comment, when there are few lines [//], when there are multiple lines [/\* … \*/] is used.

For variable definition, it is recommended that you put appropriate description of the process content. However, try to avoid comments that will make the code content to be immediately understood by just looking.

Definitely make use of comments for class, method, and function. For this comments use [/\*\* … \*/]. In comment please write, what it does, arguments, conditions such exceptional value, etc. (javadoc format)

/\*\*

\*Searches with client\_cd

\* @param string $client\_cd

\* @return returns the array of class

\*/

public function findByClientCd($client\_cd){

1. Length of 1 Line

Please use line break for lines that are too long. If you put line break, use the appropriate indentation.

public static function updateForSetData($de\_rights\_designs, $client\_cd, $sys\_ref, $upd\_user, $data\_set\_type, $db = null)

public static function updateForSetData(

$de\_rights\_designs,

$client\_cd,

$sys\_ref,

$upd\_user,

$data\_set\_type,

$db = null)

1. Switch part

Definitely set the [default] clause.

1. **Design**
2. DB: usage of ‘prepared statement’

When performing operation to DB, ‘prepared statement’ is used. With its usage, it is possible to automatically perform the escape of value that is passed.

Also, if you are issuing continuously to the same SQL, it is tried to create a ‘prepared statement’, and make use of it by design.

$sql = “SELECT \* FROM hoge WHERE id = ?”;

$stmt = $db->prepare($sql);

$res = $db->execute($stmt, array($id));

1. SQL Statement

The standard phrase of SQL is written in uppercase. Related to user defined functions, notation from the time function is defined being used.

For writing multiple lines, SELECT, FROM, WHERE, etc., the content is written in a new line with indentation.

SELECT

search\_id,search\_nm

FROM

de\_design\_search

WHERE

client\_cd = ?

AND delete\_flg = '0'

ORDER BY search\_id

1. Inspection of Condition

When performing inspections of the conditions for ‘if, while, etc.’ an inspection according to that variable type is performed. Explicit usage of isset(), is\_null() is considered. Especially, the differences between ‘false’, empty, and ‘0’ has been taken to account.

1. Functions
2. Global Functions

Try to avoid adding new one as much as possible. If necessary, prepare a class and define as static method. By doing so, you will be able to clarify the scope of the function is included in the semantic and the name space.

1. Local Functions

Please define at the end of the file where it is required. In that case, please put it emphasizes the comment later in the file so you can see the function definition.

1. Valid Function within the Action

For the function that is used in specific action, a filename that is used which has association with action is defined at another separate file.