



eSolutions

---

ICT Volume 3 : Application Standards

## ICT 3.2.1.1-2014 PHP Coding Standards

---

### **Abstract**

This document defines the standards applicable to PHP coding.

*All rights reserved. No part of this work covered by Deakin University's copyright may be reproduced or copied in any form or by any means (graphic, electronic or mechanical, including photocopying, recording, taping or information retrieval systems) without the written permission of Deakin University.*

#### Document Control

<b>Document Title</b>	ICT 3.2.1.1-2014 PHP Coding Standards
<b>Version</b>	2014
<b>Controlled Copy (Electronic Reference)</b>	

#### Document History

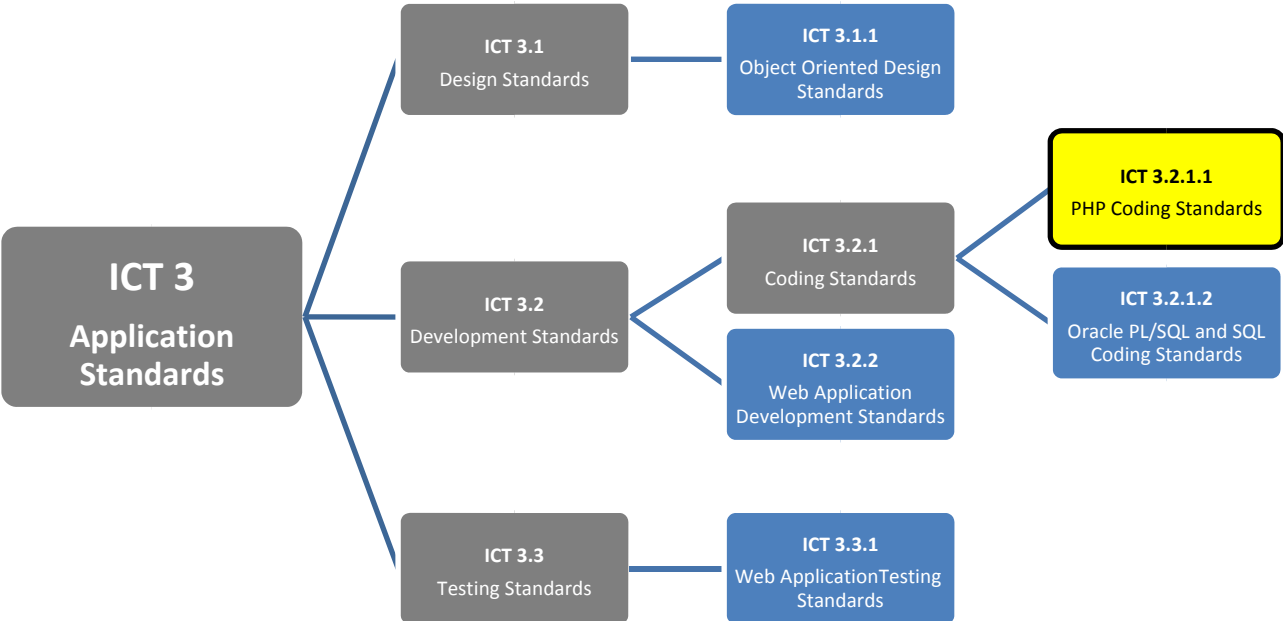
<b>Ver.</b>	<b>Primary Author(s)</b>	<b>Description of Version</b>	<b>Date Completed</b>
0.01	Steven George	Initial draft	26-11-2008
0.02	Steven George	Standards compliance	25-02-2009
0.03	Steven George	Standards compliance	18-03-2009
0.04	Steven George	Update from Melb PHP User Group Feedback	27-03-2009
0.05	Steven George	Corrected numbering error	07-04-2009
0.06	Steven George	Update from feedback	17-04-2009
1.0	Steven George	Annual review	05-07-2010
2.0	Steven George	2011 Update	03-11-2010
2.1	Michael Heley	Standards Compliance Review	11-09-2014

---

**Table of Contents**

<b>1 CODING STANDARDS</b> .....	<b>6</b>
1.1 THE <i>ZEND FRAMEWORK CODING STANDARD FOR PHP</i> SHALL BE ADOPTED.....	6
1.2 EXCEPTIONS .....	6
1.2.1 Section B.4.2.4 - String Concatenation .....	6
1.2.2 Section B.4.4.1. Class Declaration .....	6
1.2.3 Section B.4.6 – Control Statements .....	6
1.2.4 Section B.4.6.1. If/Else/Elseif .....	6
1.2.5 Section B.4.6.2 - Switch .....	7
1.2.6 B.4.7.4 Functions.....	7
1.2.7 Indentation shall be via tab characters.....	7
1.3 ADDITIONS .....	7
1.3.1 Output of HTML via PHP shall be avoided where practical.....	7
1.3.2 "Magic numbers" shall not be used .....	7
<b>2 APPENDIX A</b> .....	<b>8</b>

# ICT Volume 3 : Application Standards



## **Standards Brief**

This document serves to outline standards that shall apply within Deakin University.

## **Standard Document Access**

All Deakin University staff and authorised/approved contracted personnel are provided access to this document.

## **Policy**

These standards must be used in conjunction with all other referenced standards, and when considered in isolation from the referenced standards may not constitute adequate conformance.

## **Conflict of Information or Clarification**

Whenever a conflict of information occurs or clarification of instruction is required, all queries shall be made to Deakin University eSolutions (DeS).

# 1 Coding standards

## 1.1 The *Zend Framework Coding Standard for PHP* shall be adopted

All conventions outlined in the Zend Framework 2 Coding Standards B.4, "Coding Style" shall be adopted with exception to items identified under section 1.2 herein.

<http://framework.zend.com/manual/2.3/en/ref/coding.standard.html#coding-style>

## 1.2 Exceptions

Exceptions to standard 1.1 for Deakin University coding standards.

Unless otherwise stated, a "section" refers to the relevant section within the *Zend Framework Coding Standard for PHP* document referenced in 1.1 above.

### 1.2.1 Section B.4.2.4 - String Concatenation

A space is not required before or after the concatenation operator.

### 1.2.2 Section B.4.4.1. Class Declaration

Classes must be named according to Deakin's class naming conventions

### 1.2.3 Section B.4.6 – Control Statements

All braces in program code shall appear on a new line. This includes control statements, as class and function definitions. Example:

```
if ( )
{
    // Code here
}
```

### 1.2.4 Section B.4.6.1. If/Else/Elseif

Braces may be omitted in the instance where the body of the control structure consists of only one line. Note that indentation rules shall always apply. Example: `if ( )`

```
    // Code here
else
    //Code here
```

The alternate syntax for views (<http://au2.php.net/manual/en/controlstructures.alternative-syntax.php>) may be used.

### 1.2.5 Indentation shall be via tab characters

Code shall be indented via 4 spaces or TAB, TAB is preferred.

### 1.2.6 Section B.4.6.2 - Switch

The `break` directive should be included; however, documentation is mandatory if `break` is to be excluded.

Indentation can be either 4 spaces or tabs as long as it's in line with the entire application.

### 1.2.7 B.4.7.4 Functions

The `@param` tag must be included, along with the type of parameter that is expected. If two types are accepted, they should both be specified, eg. `string|boolean`. If more than two types are accepted, then the parameter type should be `mixed`.

## 1.3 Additions

The following additions shall apply to the *Zend Framework Coding Standard for PHP* document referenced in 1.1 above.

### 1.3.1 Output of HTML via PHP shall be avoided where practical

Any HTML output that is required to be returned to the client must be emitted outside of php delimiter tags, rather than via php functions such as `echo` or `print` where practical.

### 1.3.2 "Magic numbers" shall not be used

Constants shall be employed to represent any constant value. Configuration files shall be employed to represent any configurable value. Magic numbers shall not be used within any program code.

Where applicable, class constants must be used in preference over global constants (per <http://framework.zend.com/manual/2.3/en/ref/coding.standard.html#naming-conventions>)

## 2 Appendix A

### Definitions

Term/Abbreviation.	Definition
Class	The basic building block of software in the <i>object-oriented programming</i> paradigm.
Zend Framework	An open source, object-oriented web application framework implemented in PHP 5 and licensed under the New BSD License