

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

PHP Development Tool (PHP IDE)

Software Requirements Document

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

Software Architecture Document

1. Introduction

1.1 Purpose

This document is designed to specify the requirements needed for developing the PHP IDE – PHP Development Tool, an Eclipse.org project.

2. General Description

2.1 Product Functions

PHP IDE is a PHP Integrated Development Environment, which is designed to answer PHP developing needs and to provide a framework for development tools of PHP.

PHP IDE is a collection of an Eclipse plugins and Eclipse features, designed for use in collaboration with existing Eclipse products or as a stand-alone product. PHP IDE will be based on WTP project.

2.2 User Objectives

PHP IDE's main user objective is to answer all PHP developer needs.

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

3. Functional Requirements

3.1 PHP Project (PHP Setting)

- **General Info**

A PHP project contains source code and related files for maintaining a PHP Application. A PHP project can be organized in two different ways

- Using the project as the source container.
- Using source folders inside the project as the source container.

A wizard is available for creating a PHP project. The wizard can be accessed by the Project Wizard in Eclipse (File>>New>>Project). In the Project Wizard there will be a project type named "PHP Project" that will launch the create PHP project wizard.

When in the PHP perspective a Create PHP project wizard icon will be on the tool bar and a File>>New>>PHP Project will exist on the menu.

Note:

Generally Eclipse doesn't have Create Project wizards under the Project menu. This is because there are so many different projects that can be created.

- **PHP Project Properties**

- **PHP Version** – The user should be able to choose between PHP4 and PHP5 and by that change all related components: code assist, syntax coloring and PHP functions.
- **ASP tags support** - support ASP tags as PHP tags for all PHP actions and parsing. The following tags will be supported.

```
<% @ ... %>
<% ... %>
<%= ... %>
```

Note:

ASP tags support should be on the project level and also on the PHP preferences level.

- **Default File Encoding** – The project will support a default encoding that will be used for all files created in the project.
- **Location** – The location of the project. The default location when a project is created is the Eclipse workspace. When a location is specified, any directory/file at that location will be part of the project. Only the developer's data files will reside at the location. Eclipse internal project files will reside in an Eclipse internal directory.

The PHP project will support a properties menu item to display the properties of a project.

- **Adding files to a project**

Files can be added to a project by the following methods.

- Import from File System
- Drag and Drop (according to Eclipse behavior)

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

Copy/Paste
Using the Project Create Wizard
New PHP file Wizard

Any file that is not added to a project using the “New PHP file Wizard” will be given the default file encoding of the project.

○ **Team Support**

A project can be saved, retrieved, and synchronized with a CVS repository. All the properties of the project and files in the project (encoding, version, etc) will be saved, retrieved, and synchronized with the repository.

○ **Project Menu Items**

▪ **Project Menu Items**

These are the menu items that affect a PHP project. There may be other items on the Project Menu because eclipse supports other types of projects. The Project Menu items vary based on the project selected.

Open Project
Close Project
Properties
Build All
Build Project
Build Working Set
Build Automatically

During typing the PHP inspector is run and highlighting of errors (a squiggly line under the code) occurs for the file. Errors detected during typing are not displayed in the problems view. The build options in the project menu determine when the eclipse builder is run. The eclipse builder is potentially a long running operation. Depending on the changes made, every file in the workspace could be rebuilt. Eclipse attempts to determine the files that have changed based on the project dependencies. The eclipse builder is run as a background task so that the developer can continue to work. Problems detected during the build phase will show up in the problems view and will be highlighted using a squiggly line under the code.

When a project is closed, it can no longer be changed and its resources no longer appear in the Package Explorer, Project Outline View, etc. To re-open the project, select the project in the PHP Explorer View and click on Open Project. Eclipse doesn't have a concept of .prj files.

▪ **Right Click Menu Items**

Following are the right click menu items for a project. Some of the menu items may be handled by the low level eclipse infrastructure and not require implementation by the PHP project. Other are open issues to be resolved, but are listed here since they will show up in the Menu unless some specific action is taken to turn them off.

Copy
Paste
Move
Rename
Import – Project
Eclipse PHP Project

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

- Export - Project
 - Eclipse PHP Project
- Refresh
- Close project
- Run As
- Debug As
- Source – Format document
- Team
- Restore from local history
- Properties

- **New Project Wizard** – contains all the information and initial preferences for building a project.
 - Project name
 - Project location or Workspace (default)
 - When a location is specified, any directory/file at that location will be part of the project. Only the developer's data files will reside at the location. Eclipse internal project files will reside in a eclipse internal directory.
 - PHP version
 - Include path values

3.2 PHP file

The ability to create a PHP file with the following properties will be supported.

- **New PHP File Wizard** – Creates a PHP file in the project.
 - File Name
 - Location – relative to the project
 - File Template – Template to use.
 - Advanced Options
 - Link to file in the file system (All the file create wizards in eclipse seem to have this option.)

When in the PHP perspective, a New PHP File wizard icon will be on the tool bar and a File>>New>>PHP File will exist on the menu.
- **Options and Menu items on a PHP file**
 - Open
 - Open With – default the PHP editor
 - Copy
 - Paste
 - Delete
 - Source – Format File
 - Refactor - Rename/Move (Will only rename or move the file. Will not do any fix up of dependencies).
 - Refresh
 - Team Options
 - Comparing resources with local history or CVS
 - Replacing a resource with local history or CVS
 - Restoring deleted resources from local history or CVS
 - Supporting local history preferences (Preference wizard is supported by the eclipse Workbench not the editor).
 - Properties – Properties of the file

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

A Local history of a file is maintained when you create or modify a file. Each time you edit and save a file, a copy of it is saved. This allows you to compare your current file state to a previous state, or replace the file with a previous state. Each state in the local history is identified by the date and time the file was saved.

If the project is being maintained in CVS, the option to compare, replace, restore, etc the file from CVS will be supported. Anytime a PHP file is synchronized with a CVS repository, the properties of a file are transferred (encoding).

3.3 PHP Inspector / Code Builder

The PHP Inspector is responsible for analyzing all PHP code files in an Eclipse Platform. The PHP inspector should inspect the following:

- **PHP Elements:**
 - PHP classes – including the following data:
 - File name
 - Positions:
 - Class start position (offset + line number)
 - Class end position (offset + line number)
 - Class name position (offset + line number)
 - Modifier: Class / Interface / Abstract
 - Abstract / interface classes data that the current class implements / extends from
 - Internal variables data
 - Internal functions data
 - PHP functions – including the following data:
 - File name
 - Positions:
 - Function start position (offset + line number)
 - Function end position (offset + line number)
 - Function name position (offset + line number)
 - Function parameters names and types
 - Function return value (from the PHPDoc inspection)
 - PHP constants - including the following data:
 - File name
 - Positions:
 - Constant start position (offset + line number)
 - Constant end position (offset + line number)
 - Constant value
 - PHP include statements - including the following data:
 - File name
 - Positions:
 - Include statement start position (offset + line number)
 - Include statement end position (offset + line number)
 - Included file name value
- **PHPDoc**
 - The Inspector should also inspect PHPDoc comments for Variables, parameter types and function return values (See PHPDoc support for more details).
- **PHP warnings / errors**
 - Syntax Errors – The parser should create a list of all syntax error in the code. The errors shouldn't influence or interrupt the process of the parser for further syntax

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

error occurrences. Syntax errors examples:

- “Syntax error: expecting ;”
- “Syntax error: Unexpected end of line”
- “Syntax error: unexpected \$a, expecting ;”

Note:

The Inspector should support both PHP 4 and PHP 5 (see PHP Project – PHP Version).

Note:

The Inspector implementation should support frequent changes in the PHP Engine parser and scanner.

3.4 PHP Functions – PHP Language Data

This is the source of all PHP language classes, functions and constants. This data should come in the format of PHP files that can be editable.

The file should be located as part of the plugin/installation directory in a lib directory.

Two files should be included:

- phpFunctions4.php – includes all PHP4 language data
- phpFunctions5.php – includes all PHP5 language data
- Any change in these files will effect the following in all projects:
 - Code assist
 - PHP functions view

3.5 PHP Editor

PHP Editor should extend the Web Tools text editor and should inherit all its capabilities and actions.

The PHP Editor should allow working on more then one programming language at the same time (PHP & HTML).

- **Syntax Coloring** – PHP Editor should color several languages at the same time. The Syntax Coloring should be customizable through the Editor preferences.
 - PHP
 - Should support both PHP 4 and PHP 5 (see PHP Project – PHP Version)
 - Should color: default text, keywords, variables, numbers, strings, herdocs, Single-line comments, Multi-line comments, PHP tags, function names.
 - PHPDoc: tags, variables...
 - HTML, XML, JavaScript, CSS – inherits from Web Tools editor?
- **Code Assist** – the Code assist should be customizable through the Editor preferences.
 - PHP & PHPDoc
 - Insertion options:
 - Completion insertion / override
 - Insert single proposals automatically
 - Supports both PHP 4 and PHP 5 (see PHP Project – PHP Version)
 - Support Auto Activate if possible and suitable for Eclipse (with size limit and time delay)
 - Suggests and completes classes, functions, variables, constants and keywords, PHPDoc tags, PHPDoc variables
 - Shows function signature while calling the function
 - Additional PHP options:
 - Show variables from other files.
 - Determine object type from other files
 - Filters:
 - Show / hide constants assist

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

- Case sensitive for constants
 - Show / hide class names in Global Completion list
- HTML
 - Code Assist for HTML in a PHP file will be supported.
- **Parameter Hints**
 - Support Auto Activate after ‘(’, ‘,’ and after relevant code completion
 - Option to show function signature while calling the function
- **Templates**
 - Inherits capabilities from default Editor
 - Supported contexts: PHP, PHPDoc, HTML, JavaScript and CSS.
 - Support import and export
 - Support the following Templates variables:
 - YEAR – current year
 - WORD_SELECTION – selected word
 - USER – user name
 - TIME – current time
 - FILE – current file
 - ENCLOSING_PROJECT – current project
 - ENCLOSING_CLASS – current class
 - ENCLOSING_METHOD_ARGUMENTS – list of method arguments names
 - ENCLOSING_METHOD – current method
 - DATE – current date
 - CURSOR – cursor position
 - COLLECTION – collection’s name
 - ARRAY_LYPE – array’s type name
 - ARRAY_ELEMENT – current array’s element
 - ARRAY – array’s name
 - And more...
 - Appear as part of the code assist with its name in the list. The available templates should be according to the context of the completion
 - The optional templates context are: PHP, PHPDoc, New PHP.
- **File Templates**
 - File templates which were defined in the templates table as “New PHP” will appear in the “New PHP File Wizard”.
- **Code Folding**
 - Set initialize folding options in Preferences should include
 - Fold all non PHP Code
 - Fold PHP code
 - Fold PHPDoc comments
 - Fold Classes
 - Fold methods
 - Support the following actions from menus or key bindings:
 - Fold all
 - Expand all
- **Code Hovers** – as in Eclipse JDT, supports combined hover, variables value, PHPDoc and source. Hover will support

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

- Simple text hover
- Debug variable hover
- Syntax error hover
- Got to source hover and link
- Extension point for additional hovers

○ **Typing**

- Automatically close text
 - PHP:
 - Strings ' and "
 - Parentheses, squares, and brackets (, <, [
 - Brackets {
 - Block Comments
 - PHPDocs (add PHPDoc tags)
 - HTML, CSS and JavaScript: inherits capabilities from Web Tools Editor
- Indentation
- When pasting - Adjust indentation

○ **Matching Brackets** – The matching brackets should be customizable through the Editor preferences, should support:

- PHP
- HTML HTML, XML, JavaScript, CSS – inherits from Web Tools editor?

○ **Comment / Uncomment Code** – context add or remove comments, should support:

- PHP
 - Single-line comments
 - Multi-line comments
- HTML and JavaScript

Note:

The operation should fail if the beginning and the end are in different contexts (start: PHP, end: HTML)

○ **Report problems as you type** - link to inspection -> errors / warnings

○ **Opening Files in the Editor**

A file can be opened in an editor by the following methods.

1. Clicking the file in a view (project, search, etc.)
2. Right-clicking the file in the Navigator view and then selecting Open from the pop-up menu
3. Dragging and dropping the file from a view to the editor area
4. Doubling-clicking a bookmark that is associated with that file.

○ **Search Options**

The PHP editor will have a wizard that will support the following search options

- Classes
- Functions
- Constants
- Variables

The scope of files searched can be limited to Workspace, Selected Resources, Enclosing Projects, and Working Set.

When the search is completed the PHP will open a search view with the results. Files

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

when opened in the editor will be annotated with the search result.

- **Annotation**

Annotation is a piece of information attached to a certain region of a text document. Annotations are shown in a vertical ruler on the left side of the text, an overview ruler on the right side of the text, as a highlighted word or as colored squiggles underneath text. Annotations are largely dependent on the kind of document being edited. The following annotations will be supported:

- Errors
- Warnings
- Info
- Search Results
- Tasks
- Bookmarks
- Diff changes/additions
- Breakpoints

The editor will support moving to the next/previous annotations by icons on the Tool Bar and menu items. The editor will support the preferences set in annotation preference wizard supported by the Workbench.

- **Quick Diff**

Quick Diff provides color-coded change indication while you are typing. When the mouse cursor is placed over a change in the vertical ruler, a hover displays the original content, which can be restored using the ruler's context menu. Quick Diff will support both local history and CVS.

Note:

All the editors in eclipse seem to support Quick Diff.

- **Key Bindings**

The PHP Editor will support the standard eclipse Key Bindings. Some examples are below. Look at the preference page in eclipse to see the complete list.

- Ctrl+J - Edit >> Incremental Find Next
- Ctrl+Shift+J - Edit >> Incremental Find Previous
- Ctrl+Q - Navigate >> Go to Last Edit Location
- Alt+Arrow Up - move lines up
- Alt+Arrow Down - move lines down
- Ctrl+Alt+Arrow Up – copy selected lines to above the current line
- Ctrl+Alt+Arrow Down – copy selected lines to below the current line
- Ctrl+Shift+Enter - insert a new line above
- Shift+Enter - insert a new line below
- Ctrl+Shift+Y - convert to lowercase
- Ctrl+Shift+X - convert to uppercase

3.6 Outline View

The Outline view displays a Tree representation of the contents of the editor that is currently visible in the editor area of the workbench. For PHP files, the Outline should display elements like classes, functions, variables, included files, and constants. It should optionally also display the content outside the PHP areas (ex. HTML, XML, CSS, JavaScript).

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

NOTE: The Outline view for JSP and HTML editors do not render embedded CSS and JavaScript. This support will have to be implemented.

NOTE: There is also the issue of PHP elements that span multiple PHP scriptlets. How can these be rendered as Tree nodes where there is the possibility of interleave foreign content (i.e. HTML).

- **Global Menus:** Collapse all, Expand all, Show PHP/HTML
- **Context Menus:**
 - PHP nodes: Add/Delete/Rename class, function, variable, included file, constant (in the long future)
 - Other content: Provide default behavior that is available on the other editors. For example, HTML content nodes should have the same context menu when in an HTML file.
- **Selection:** Selecting the Tree node in the Outline view highlights the represented element on the editor. This should also work in the other direction, meaning that the Outline view should highlight the node represented by the area under the editor's caret.
- **Sorting:**
 - Alphabetically order
 - Original order of the data in the file
- **Grouping** (option to group by)
 - Classes
 - Functions
 - Constants
 - Include files
- **Filtering** – buttons to filter the content (should be added in the future)
 - Classes
 - Functions
 - Constants
 - Include files
 - Variables

3.7 Project Outline View

The Project Outline View is a new View to the Eclipse Platform. This View is very similar to the Outline View except that it should display the elements of the project correspondent to the opened file.

All other functionality should be identical to the Outline View.

This view is basically gathering of all Outline Views data of the entire project files.

3.8 PHP Explorer View

Similar to the Package Explorer from the JDT, the PHP Explorer shows the PHP element hierarchy of the PHP projects in the workbench. It provides a PHP-specific view of the resources shown in the Navigator. The element hierarchy is derived from the project's build paths.

The PHP Explorer View will support using Working Set to determine the projects that are shown.

The actions (right click menus) of the Explorer view are derived from the Project actions (appears in section 3.1 under Right Click Menu Items).

Additional View actions:

- Collapse all
- Link with Editor
- Layout view

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

- Filters

3.9 Problems View

The Problems view should display all PHP errors and warnings.

Should include “include files” problem (in the future).

All errors and warnings will have a description, resource, folder, and location associated with them.

- Resource refers to the php file containing the problem.
- Folder refers to the directory containing the php file.
- Location will refer to the line number containing the problem.
- When a user clicks on a problem in the problems view, the php file containing the error or warning will be displayed in the editor.
- When a user clicks on a missing include file, a file wizard will appear to add the file to the project.
- Problems view will also display the total number of errors and warnings

3.10 PHPDoc support

- **Syntax Coloring** – customized syntax highlighting for PHPDoc tags and their values (see Editor – Syntax Coloring).
- **Inspection** – The value of the following tags should be inspected: abstract, access, author, copyright, deprecated, example, exception, final, global, ignore, internal, link, magic, name, return, package, param, see, since, static, staticvar,subpackage, throws, todo, var and version.
Note: These tags are affecting the Outline View, Code Assist and Project Outline View.
- **Code Assist** – for the list of tags appears above there should be code assist as well.
- **Adding description**
 - Outline view – right click action to add description to the selected data
 - Project Outline View - right click action to add description to the selected data
 - Code Assist – Button and shortcut to add description to the selected data in the list (if possible, will be added in future versions)

3.11 Code Formatter

In a document with PHP as well as other content (i.e. HTML), each content should have a Formatting strategy. The strategy for foreign content (i.e. HTML) should be the same as those provided by standalone editors. This section emphasized on the Formatting options for the PHP content and Doc areas.

- **Invocation:** Formatting is initiated by either “ctrl-shift F” key combination of through menu. There should be both a global menu item and a context menu item.

3.12 Encoding

- Default encoding for each project – will open file from the Explorer view with this encoding
- File chooser will include a list of all available encoding options with the default encoding selected.
- Saving file with a specific encoding or the default one

3.13 Open PHP Resources

This is a Toolbar and Menu option similar to the JDT’s Open Type. The JDT’s version of Open Type is a dialog that allows the user to search for Classes and Interfaces and subsequently open the file that defines it. For PHP, this behavior will be extended for Classes, Functions, and Constants.

- **Features:**
 - Display and narrow search results as the user types
 - Allow for wildcards – (“?” = any character, “*” = any string)

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

- Search the entire workspace (all PHP projects that are open)
- Multiple name matches need to be further qualified with:
 - Workspace location, includes project, folder, and file name
 - Type of element, such as Class, Function, Constants

3.14 PHP Functions View – PHP Language Data

View with all the PHP functions available in the language according to the PHP version defined in the edited file's project (see PHP Functions – PHP Language Data).

- **Open manual** - with an explanation about the selected PHP function from the list.
- **Short description** - in a tool tip about the function and its arguments
- **Adding** - the function to the edited file.

3.15 PHP Manual View

This view is an offline view of the PHP Manual (<http://www.php.net/download-docs.php>). The PHP IDE should be customizable to work with any downloaded version of the PHP Manual (should support different languages). The manual control will be supported by extension point to add new manuals from different sites.

The view should support:

- Quick link from the editor depends on the caret location and the relevant function.
- Link from the PHP functions list.

3.16 PHP Perspective

In eclipse, Perspectives define the initial set and layout of views in the Workbench window. A Perspective provides a set of functionality aimed at accomplishing a specific type of task or working with specific types of resources

The PHP perspective will include the following views

- PHP Explorer View
- Outline View
- Editor View
- Project Outline View
- Problems View
- PHP Manual View (if possible with embedded browser problem)
- PHP Functions View
- Search View (after the first search)

The PHP perspective will save state (views, files opened, etc) when closed. And reopen to the same state.

The PHP perspective will support customization through the Customize Perspective wizard. The customized perspective can be saved under a user specified named. It is possible for a developer to have multiple PHP perspectives with different customizations each with its own name. The PHP perspective will also support resetting the perspective back to the default settings.

- **New Menus**
The PHP perspective will add the following new menu and menu items to the menu bar.
- **Existing Menus**

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

File>>New
 New PHP Project
 New PHP File

Note:

Need to review all the menus for complete content.

o **Tool Bar**

The following items will be on the Tool Bar

- New PHP Project/File
- Debug
- Run
- Search

3.17 PHP Debug/Run Configuration

As in JDT, the user can define the debug/run configuration from the Run/Debug configuration dialogs.

This dialog will contain 2 nodes:

- PHP Executable – will include the php used for this configuration. There is a default option and an alternate option. New executable locations can be defined from the PHP preferences.
- PHP Web Server
 - o Default web server as defined in the project/workspace. This web server doesn't include any publishing.
 - o Customized web server which includes addition of new web server with ability to provide the publishing directory and its url.
- Each of the configuration provides the ability to define the project and the debugged file with a context root (if necessary)

3.18 Debug Perspective

The Debug Perspective will present the user with common debugging tasks. There will be toolbar buttons to stop, pause, and start execution. The perspective will contain the following views;

- The Outline View of the current file in the debugger
- Console View
- Task View
- Debug Output View
- Browser Output View
- Editor containing file with the current breakpoint.
- Variable View showing runtime values of selected variables.
- Breakpoints View
- Debug View. The Debug View will show the runtime call stack up to the current breakpoint. The user will be able to click on the function name in the stack and the editor will display the file at the appropriate line.
- Server View (should consider whether it's suitable)
- Watches View

3.19 TODO / TASKS

Gather all TODO tasks from the entire PHP project files to a complete list (as it is today). The List should appear next to the Problems View. TODO task should start with "TODO".

The TODO information should be collected from:

- Regular comments (// or /* */)

PHP Development Tool (PDT)	Version: 1.0
Software Requirements Document	Date: 23/August/2006

- PHPDoc blocks (as @todo)

The list should include the following information:

- File name
- Folder name
- Line number
- Description of the TODO task
- Tool tip with a complete TODO task information
- Double click should get you the TODO location in the document