

PHP Development

Contents

- PHP Support
 - Creating a PHP Project
 - Creating a project using the new-project wizard
 - Creating a project using the import wizard
 - Editing
 - Syntax Coloring
 - Code Assist
 - Auto Indentation & Code Formatting
 - PHPDoc information pop-ups
 - Occurrences Marking
 - Attaching PHP Resources
 - External Libraries / Frameworks
 - Internal Project Resources
 - PHP API Level
- Previewing
 - Troubleshooting

Related Topics

- [PHP Script Debugging](#)
- [PHP Web Page Debugging](#)

PHP Support

Aptana Studio 3 is shipped with a built-in support for PHP editing. To add PHP editing for Titanium and Appcelerator Studio, see [Adding PHP, Ruby or Python Support](#).

Creating a PHP Project

Although it is possible to edit single PHP files, the full strength of the IDE will be available when the PHP files are in the Studio's workspace as PHP projects.

There are two ways to create a project:

Creating a project using the new-project wizard

The PHP project wizard can be accessed through multiple locations in the workspace.

A project can be created through the File->New->PHP Project, or by right-clicking the 'Project Explorer' or the 'App Explorer' and select New->PHP Project.



Creating a project using the import wizard

A project can be imported to the workspace using the 'Import Wizard' through File->Import and then selecting General->Existing Project Into Workspace. Same action exists when right-clicking in the 'Project Explorer' and the 'App Explorer'.

When in the 'Project Explorer' view, you can also navigate the 'Local System' node, right-click a directory and select 'Promote to Project'.

Once a project is imported, you may need to add the 'PHP Nature' order to enable some PHP-specific features.

To do so, right-click the project you have just imported and select 'Properties'. Navigate to the 'Project Natures' item and check the 'PHP' nature.

Editing

Paste in your files, import them or create new ones using the new file wizard.

When in the 'App Explorer', the new file wizard will offer you to create a file from a template as an option on the 'Next' page. These file-templates are contributed by the "PHP Ruble", and we discuss about the 'Ruble' support at the [Ruble Specification](#) page.

The PHP editor provides these features:

- Syntax Coloring according to the selected theme in the preferences.
- Code Assist
- Syntax error annotations

- Auto indentation and Code Formatting
- Hyper-linking to classes, functions and variables by hovering over elements and pressing the Ctrl key
- PHPDoc popups when hovering over items that have attached documentation
- Read and write Occurrences Markers when clicking on specific PHP elements

Syntax Coloring

Like with any other Studio-contributed editor, the syntax highlighting settings are controlled by the 'Theme' preference page under the Studio's preferences.

The supported keys is described in details at the [Current Theme Scope](#) page.

Code Assist

The editor provides smart code assist mechanism that displays suggestions by the context at the edit-location.

- Because of the nature of the PHP language, the code assist uses **PHPDoc** comments to provide accurate type assistance. It also incorporate some level of code analysis to provide assistance when the PHPDoc is missing. However, to maintain accurate proposals, make sure to document functions, classes, variables etc. with a valid PHPDoc comments.
- The code assist may be triggered manually by typing a letter, and then typing Ctrl+Space.
- When not bound to any type-context, the code assist proposals will show items from the PHP API, all of your project files and from any attached PHP Libraries or sibling projects. By default, the items are **not** strict to any import/require statement in the edited file, and it's up to the developer to update those when needed.
- In some context locations, the code assist may pop-up automatically to suggest available options. For example, an available list of functions, variables and constants will appear after typing a PHP arrow string '->'.
- When traversing the code assist suggestions, the editor may display the selected item's API, if available.
- When inserting a function call item, any expected parameters are pre-populated into the code (excluding any optional parameters). It's possible to traverse the inserted parameters by clicking the 'Tab' key. Clicking 'Return' will jump to the 'green-marker' at the end of the inserted code.



- When the caret is located inside the parameters area, clicking Ctrl+Shift+Space will pop-up a small tool-tip helper that will indicate the available parameters and will **bold** the current position.



Note: The content assist relies on an **indexing** process that is performed on the workspace and on the edited files. Inconsistent index state, like the one that may occur when turning off the 'Auto Building' option, may result in wrong or no accurate assistance.

Auto Indentation & Code Formatting

Auto indentation is done as you type and move to a new line. For example, when starting an *'if'* block and hitting the 'Return' key, the next edit position in the new line will be indented under that *'if'*.

Code formatting is an advanced feature that allows you to format your entire file as a post-editing action. The formatter is triggered by clicking Ctrl+Shif+F, or by selecting 'Source -> Format' in the editor's pop-up menu or top menu-bar.

The code formatter settings can be tweaked through the the Studio's preferences. You can create your own formatting profile and edit it to your needs.

The PHP code formatter provides control over the new-lines, indentation-levels, blank-lines, braces, spaces and comments.

As of version 3.0.4, the formatter also provides an ability to turn off and turn on formatting for a specific sections of code.

For example:

```
function foo() {
    $a = 5;
    // @formatter:off
    $arr = array(
        1,
        2,
        3
    );
    // @formatter:on
    $secondArr = array(4, 5, 6);
}
```

Note that the 'off' and 'on' tags are also customizable through the formatter preferences.

PHPDoc information pop-ups

When hovering over a PHP element, such as a function call, a class name etc., a pop-up will appear and display any PHPDoc information for that element. The content for that pop-up is parsed from the PHPDoc block that is on top of the element's definition. The header of the proposal will indicate the source file that was detected as the element's origin.



Occurrences Marking

When the 'Mark Occurrences' toggle button that appears at the toolbar is 'on', the PHP editor will highlight the 'read' and 'write' occurrences of the selected element.

Clicking on a variable element, for example, will highlight all the places in the edited code where the item is accessed (read-from or write-to as an assignment).

Controlling the elements and structures that will display the occurrences markings is done through the Studio's preferences, under the PHP item node.



Attaching PHP Resources

The PHP support allows attaching resources that will be indexed and used when proposing content-assistance. Those resources can be located out of the workspace as source directories, and inside the workspace as other projects.

External Libraries / Frameworks

External libraries can be attached to provide a global content-assist addition that will be available to all the projects in the workspace.

To attach one, go to the PHP preferences at Aptana -> Editors -> PHP -> Libraries and add the path to your external library/framework. An indexing process will start right after to build the content-assist database that will be used globally.

By default, every PHP project will show content-assist proposals from that library, unless indicated differently in the project's properties.

To specify a unique set of libraries for a specific project, right click the project, select 'Properties' and then select the '**Project Buildpath**' item. Click the '**Libraries**' tab and select the libraries you wish to see in your code-assist.



Through that same property page it's also possible to attach more external directories to be parsed for the content-assist proposals. To do so, click the '**External Directories**' tab and select a directory on your disk.

Internal Project Resources

Just like with the external resources, it's possible to add internal workspace resources to the content-assist support for a specific project. For example, if you wish to get assistance in project 'A' for classes that were defined in project 'B', you can add project 'B' to project's 'A' buildpath. Right click the project, select 'Properties' and then select the '**Project Buildpath**' item. Click the '**Workspace**' tab and select the projects (or directories from projects) you wish to see in your code-assist.

PHP API Level

The coloring and parsing of the PHP code in your workspace is determined by the PHP API level that was set on the project during the project creation (PHP4, PHP5 or PHP5.3).

You may change this setting after the project was created by accessing the project's properties.

Right click the project, select 'Properties' and then select the '**PHP Development**' item. Change the compatibility level to the desired API and hit OK. A rebuild process will start and will re-index your project's file.

Previewing

To preview PHP files, you will need to set up an external PHP server. A popular example is [XAMPP](#)

See [Setting up an external web server to preview PHP and other non-HTML pages](#)

Troubleshooting

- In case you don't get all the code-assist you expect for, and you imported a previously created project, make sure it has the Aptana PHP

'Nature' set for it. Right click the project and select 'Properties'. Select the 'Project Natures' item, check the 'PHP' nature and make it primary.

- In case something is not functioning as expected, please submit an issue through our ticketing system. The [Troubleshooting](#) section describes how to.
- In some unique cases, the indexing process may take long time to complete. This will happen when you have a very large number of files in your PHP projects. In case you feel it takes too long to complete, you may want to clean the workspace before submitting an issue. To do so, access the 'Project' menu, select 'Clean' and then 'Clean All...'. This operation will delete and recreate any cached index files, mending any corrupted indexes that can trigger such an issue.