NetBeans IDE for PHP Editor: Brief Overview

This document is a brief overview of the main features of the NetBeans IDE PHP Editor.

Contents

To follow this tutorial, you need the following software and resources.

Software or Resource	Version Required
NetBeans IDE [1]	PHP download bundle
A PHP engine	Version 5
A web server	Apache HTTP Server 2.2 [2] is recommended.
A PHP debugger	XDebug 2.0 or later [3]

Getting Ready

To successfully develop and deploy PHP applications in the NetBeans IDE for PHP, you need to have all the required software installed and configured for PHP development.

The NetBeans IDE for PHP Editor Layout

The editor contains the following windows:

- The Editor window shows the file you are editing. The Editor has a tab for every open file.

- The Projects window shows a list of currently existing projects
- The Files window shows the files for each project
- The Services window shows lists of currently available external services such as databases.

You can also open a Navigator window in the left-hand panel by pressing Ctrl + 7. The Navigator shows the structure of a PHP class if the class is open in the Editor. Non-class PHP files are not shown in the Navigator window.

To open help for one of these windows, select any element in the window and press F1.

Syntax Highlighting

The editor provides syntax highlighting for PHP, HTML, JavaScript, and CSS code blocks.

The following syntax elements are highlighted in the current version:

- PHP keywords and variables

```
14   class Bug
15   {
16       private $products = null;
17
18       public function assignToProduct($product)
```

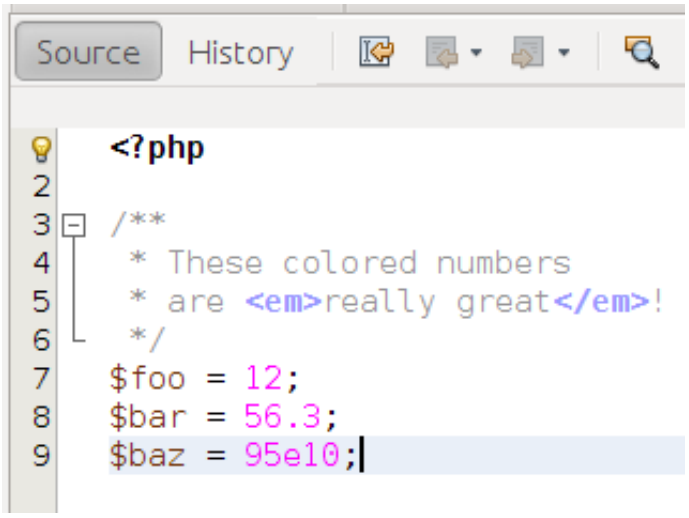
- HTML tags, including input form attributes

```
28           <a href="createNewWisher.php">Create now</a>
29           <form name="logon" action="index.php" method="POST" >
30               Username: <input type="text" name="user"/>
31               Password  <input type="password" name="userpassword"/>
32               <?php
```

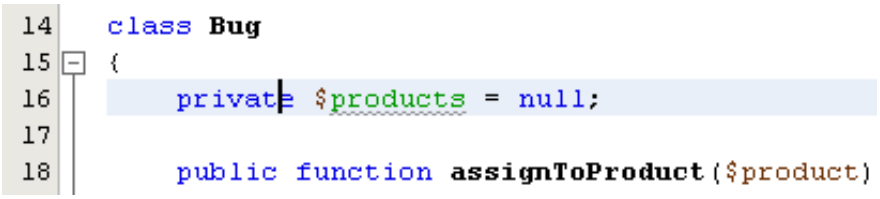
- Apache web server .htaccess or httpd.conf files

```
Source  History
1 Options +FollowSymLinks
2 # starting RewriteEngine
3 RewriteEngine On
4 RewriteBase /
5 RewriteCond %{REQUEST_URI} !^/robots\.txt$
6 RewriteCond %{HTTP_HOST} !^www\.example\.com$ [NC]
7 RewriteRule ^(.*)$ http://www.example.com/$1 [R=301,L]
```

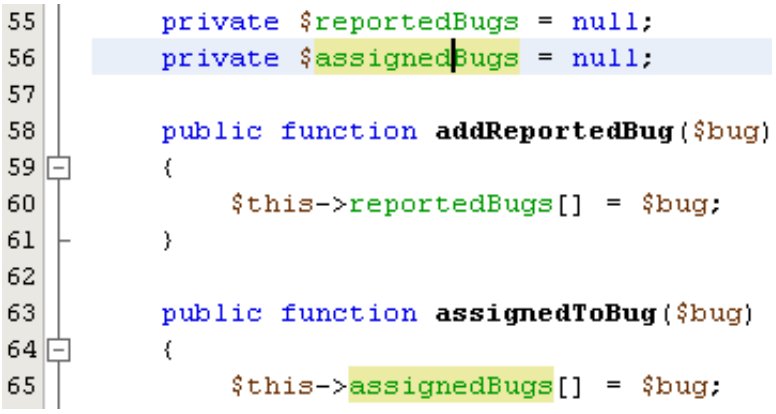
- PHP numbers
- HTML tags inside PHPDoc comment blocks



- The current line is displayed with a light-blue background.



- By placing the caret inside a PHP variable, a function, or a class, all the occurrences of the variable are highlighted.



To change the settings for highlighting, choose Tools > Options and switch to the Fonts&Colors tab. Specify the settings of your choice.

Go To Declaration

The Go To Declaration function navigates the user from an occurrence of a variable or function to the line where the variable or function is declared or initialized. To use this functionality, position the cursor on the relevant variable or function occurrence and choose Navigate > Go to Declaration from the context menu, or press Ctrl-B/⌘-B, or use Ctrl-click.

Navigator Window

The Navigator window displays code structures as elements of a list that shows the PHP structures of the currently active file. If the file contains any other text sources (for example, HTML) the list displays their structures too. By double-clicking a function or a variable in the list, the cursor points at the declaration of the clicked element.

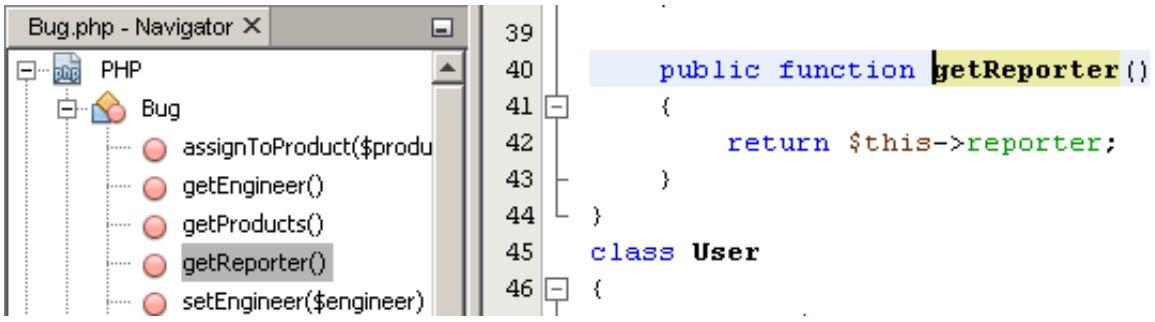
Caution: The Navigator window only shows PHP code for PHP classes. It cannot show PHP code outside of a class.

The Navigator list does not normally contain any embedded lists, but for complex statements it may contain embedded structures. For example, a PHP class declaration statement can be presented as a tree of declared class members.

The list is dynamically updated while the user types the code.

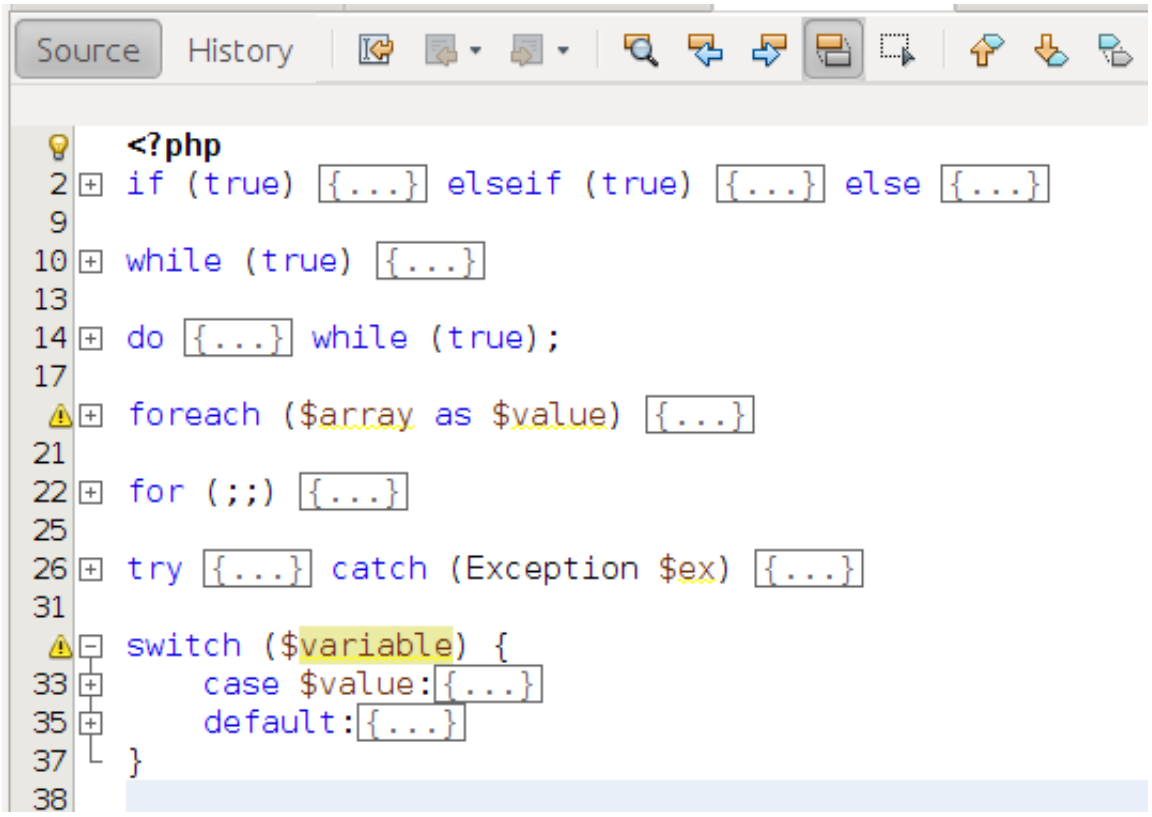
The Navigator window is shown in the bottom of the left-hand panel.

1. To open the Navigator window, press Ctrl + 7 or go to Window > Navigating > Navigator. The tree that displays the hierarchy of the HTML and PHP structures appears.
2. To move to the desired element, double click the relevant node in the list.



Code Folding

The editor enables the user to selectively hide and display modules, classes or functions of the currently edited file with PHP source code. To fold or display a piece of code, click the - or + icon to the left of the code, or press Ctrl+ or Ctrl-.



Smart Indent

The editor automatically formats the code while the user is typing.

Formatting

To format a piece of code to make it more understandable, perform the following steps:

1. Select the relevant code passage.
2. From the right-mouse context menu, choose Format or press Alt + Shift + F

Bracket Completion

The editor automatically adds and removes matching brackets and quotes while the user is typing the code.

- Paired single ' ' and double quotes "" "", braces (), and brackets [] are added when the user has typed the first symbol.
- At the end of a line, a semicolon is added in addition to the closing single or double quote.
- The closing curly brace is added after the user presses Enter, which also activates the Smart Indent function.

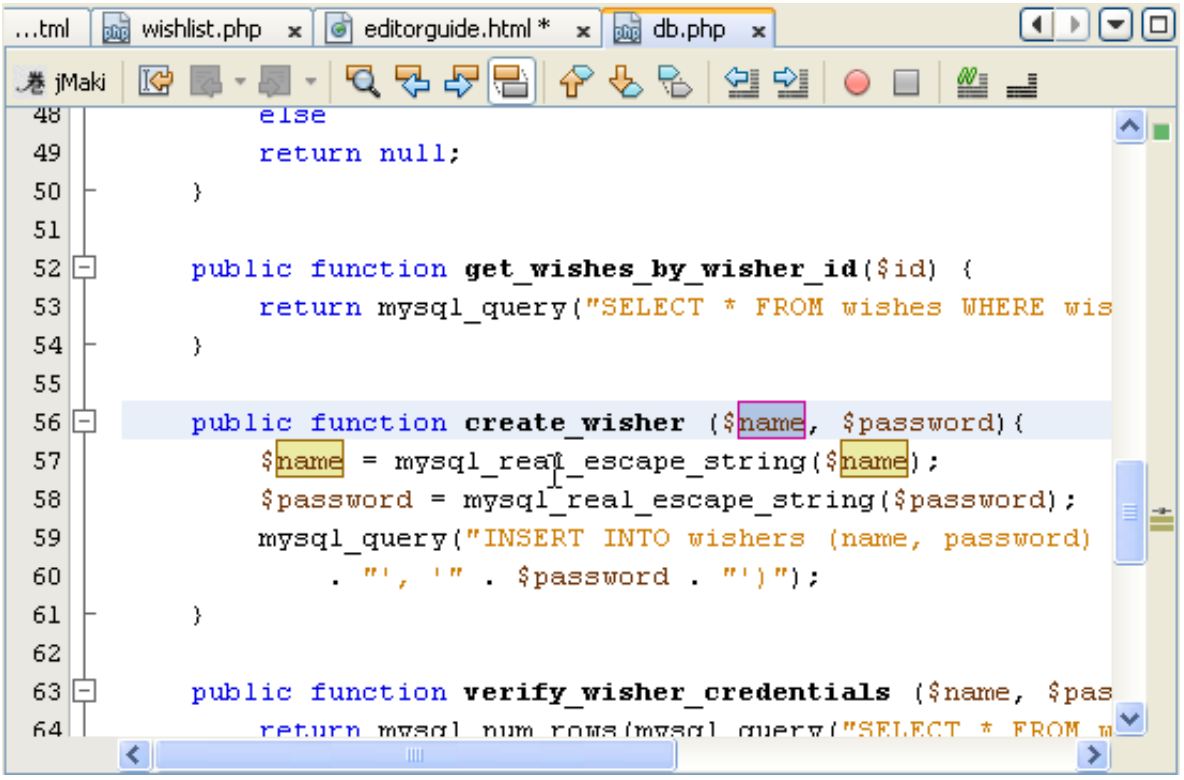
- Deleting the opening single or double quote, brace, or bracket causes automatic removal of the corresponding closing symbol but does not affect the line end semicolon.
- When the cursor points at a closing brace), a closing curly brace }, or a closing bracket], the paired opening symbol is highlighted in yellow.

Rename Refactoring and Instant Rename

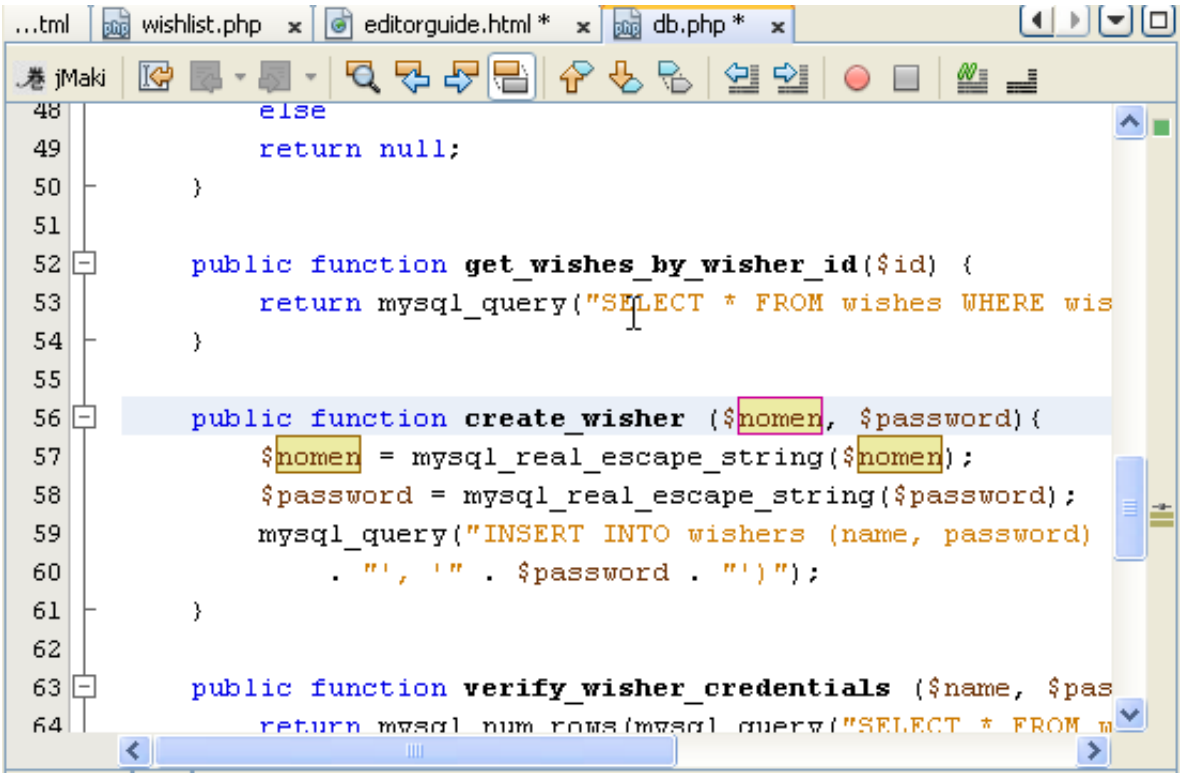
You can rename an element such as a class name across all files in a project. The feature forces you to preview your changes before you can make them. The preview window shows you every location of the element and lets you exclude individual occurrences of the element from being renamed.

Rename Refactoring is contrasted with the older feature, Instant Rename. Instant Rename is still available, but it only works in "non-public" contexts, such as renaming a variable inside a method, or renaming private variables and fields. Instant Rename lets you only rename an element within a file and does not provide a preview window.

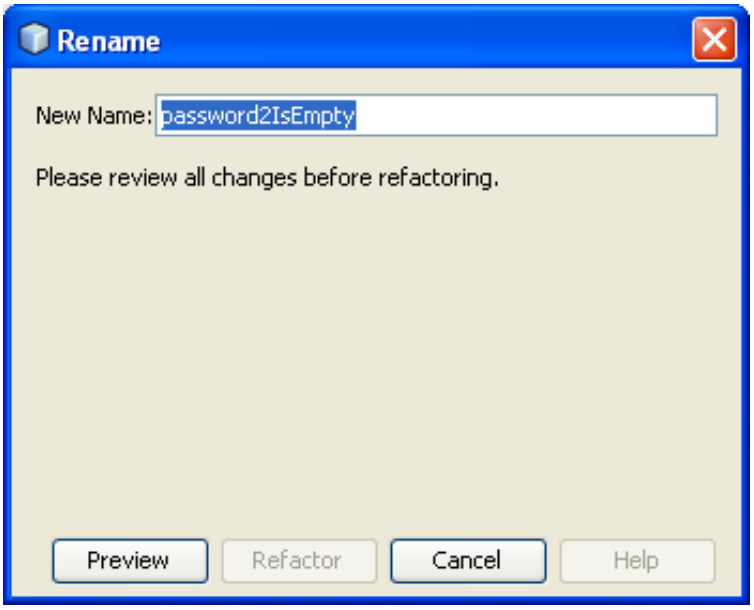
To use Instant Rename, place the cursor on a name you want to change and press Ctrl-R. If Instant Rename applies to that variable, all instances of that variable or function name are highlighted.



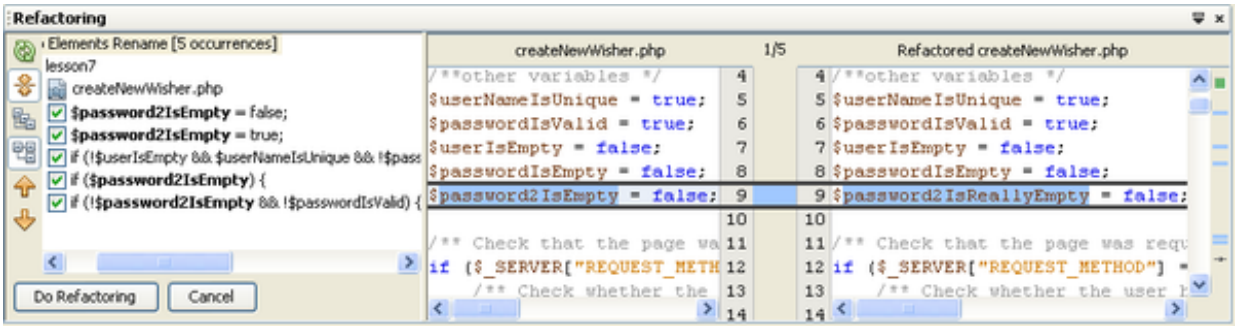
Change one instance of the name and all other instances in the file are changed simultaneously



To use Rename Refactoring, select the element you want to rename and either press Ctrl-R or right-click and select Refactor > Rename. A dialog opens for you to rename the element.



Rename the element and press Preview. The Refactoring window opens. In this window, you can find every instance of the element in your project and decide whether or not to rename it.



For more information about Rename Refactoring, see the screencast Rename Refactoring and Other Editor Improvements in NetBeans IDE 7.0 for PHP [4].

Code Completion

Code Completion is a common name for a set of features that speed up the process of coding.

The following types of code completion are distinguished:

Snippets

Snippets enable the user to generate code for various elements automatically.

1. Choose Tools > Palette > Code Clips. A palette containing various user interface elements appears in the right-hand panel.

2. Drag the required icon on the palette to the relevant position in the code. A dialog box for specifying the parameters of the corresponding elements appears. Fill in the data.
3. The code that displays the chosen element is generated and inserted in the chosen location.

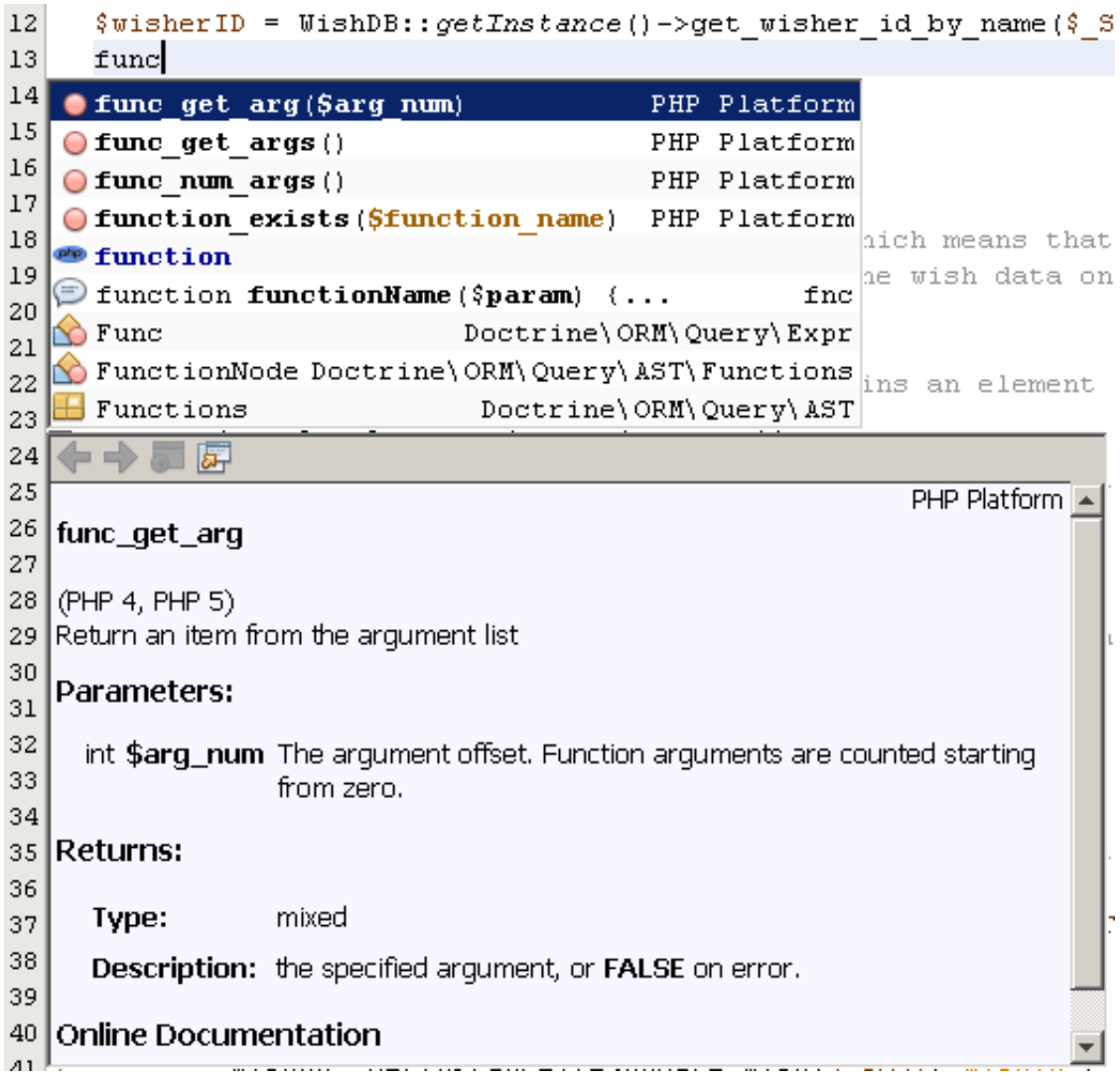
Context-Sensitive Proposals

The editor provides context-sensitive proposals for completing any number of starting symbols of:

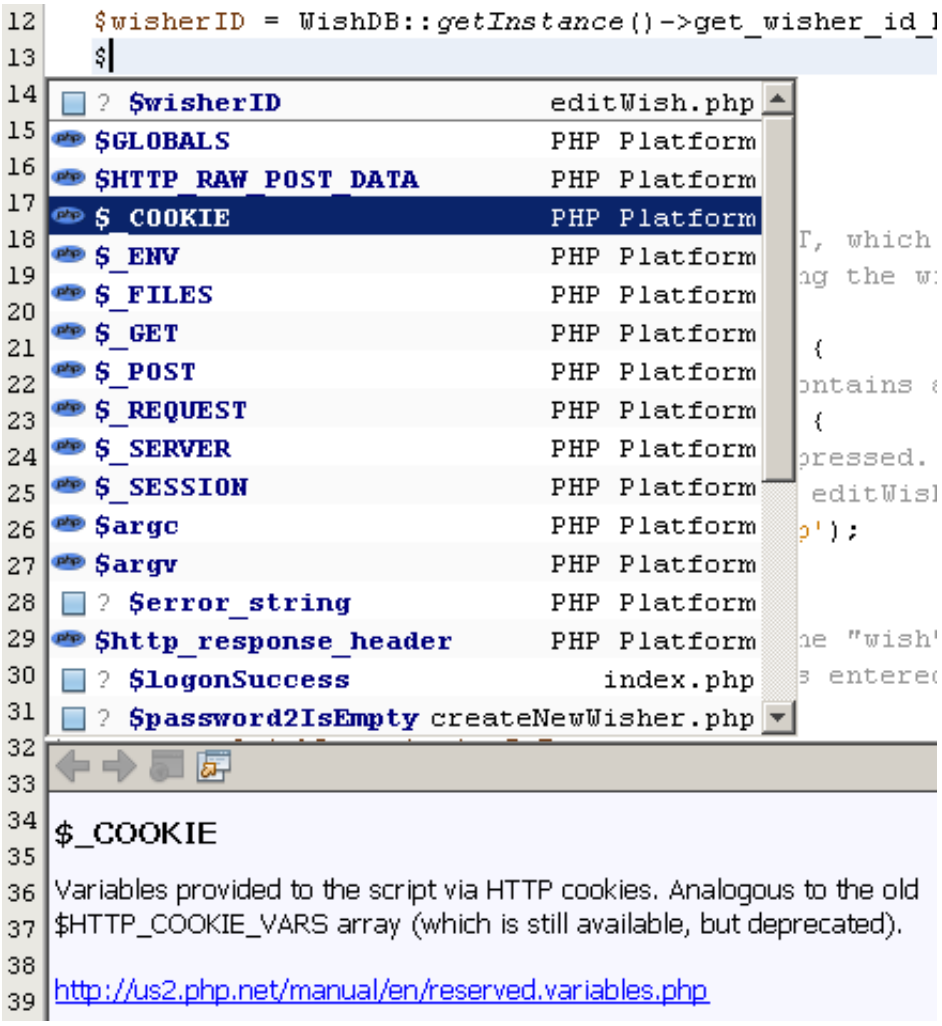
- A PHP keyword (for example, if, else, elseif, while, switch, function, and so on)
- A PHP built-in function (for example, substr, count, and so on)
- A pre-defined or user-defined variable

The editor not only suggests expansions but also provides parameter hints. To apply Code Completion:

1. Type the starting symbols of the required character string.
2. Press Ctrl + Space. A dropdown list shows the context-sensitive proposals. Each proposal is supplied with a description and parameter hints. The contents of the list change as you continue typing.

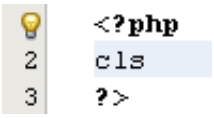


3. To obtain a list of the PHP key words that are available in the current context, press Ctrl + Space without any previous typing.
4. To obtain a hint on a variable, just type the dollar symbol "$". A list of all the currently available local and global variables appears.

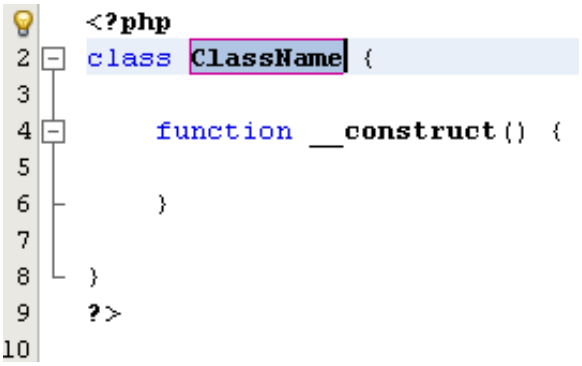


Code Templates and Abbreviations

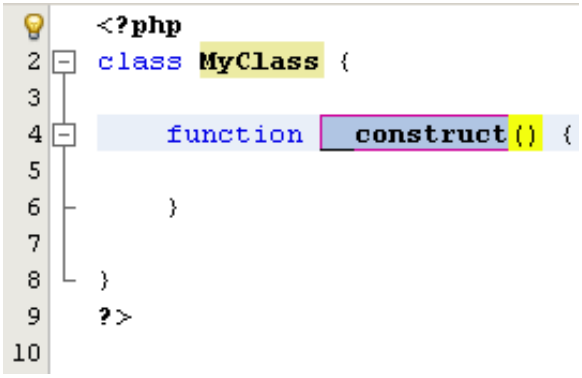
In the current context, the term ""abbreviations" refers to a set of predefined character strings that correspond to the key words used in a programming language. Each abbreviation is associated with an expanded text which contains the full key word and a code template for the key word with parameter hints. To apply this functionality, type an abbreviation and press Tab.



The abbreviation is replaced with the corresponding key word and the code template for the key word is provided.

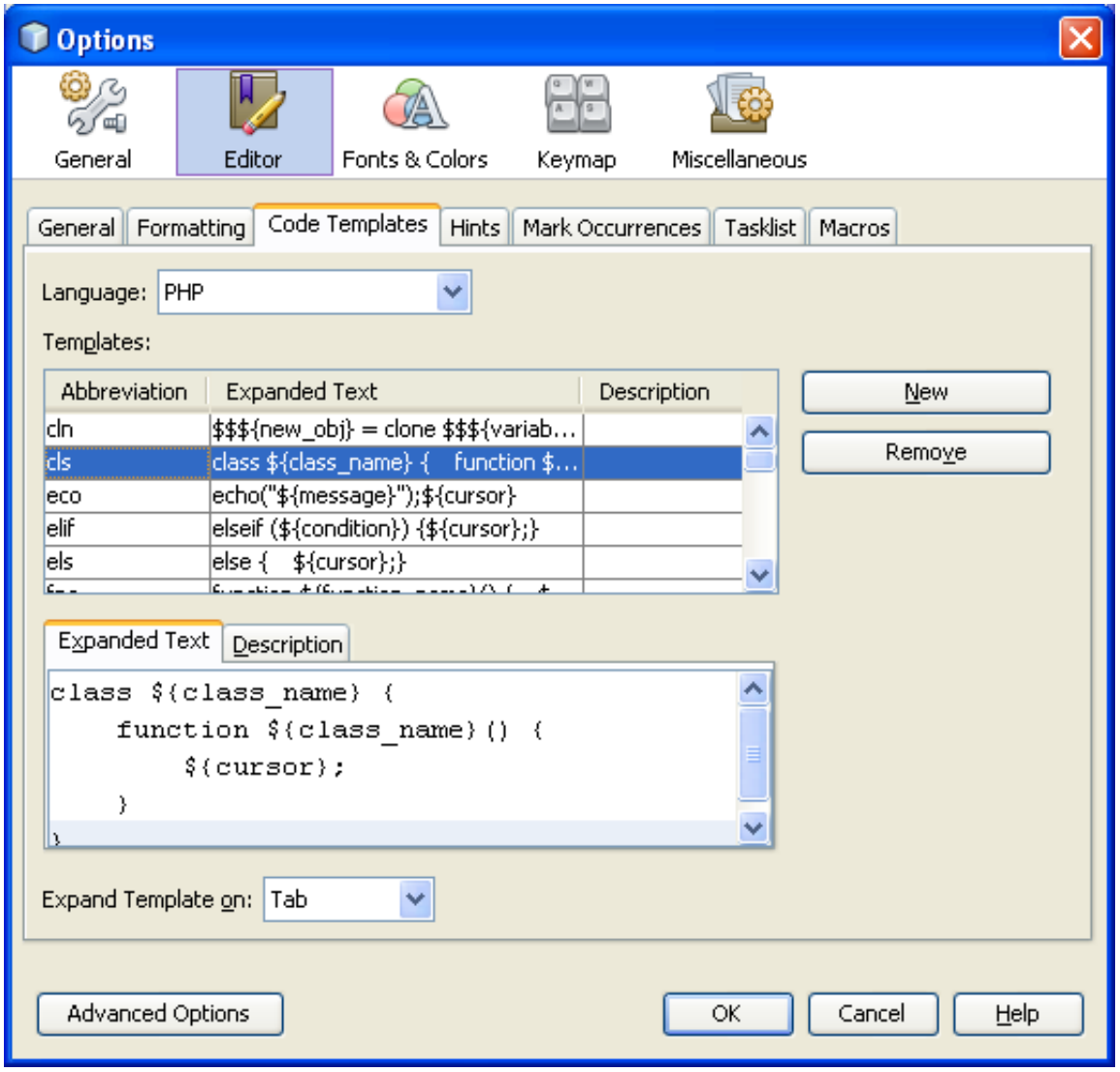


In the code template, if the cursor is in a blue box with a violet border, the cursor is in a field in the template. Type the field name or value. All instances of that field also change value. Press Enter when you are finished, and the cursor moves to the next field (or to the position of the ${cursor} variable, or outside the template if there are no remaining fields).



To view the list of defined abbreviation with code templates:

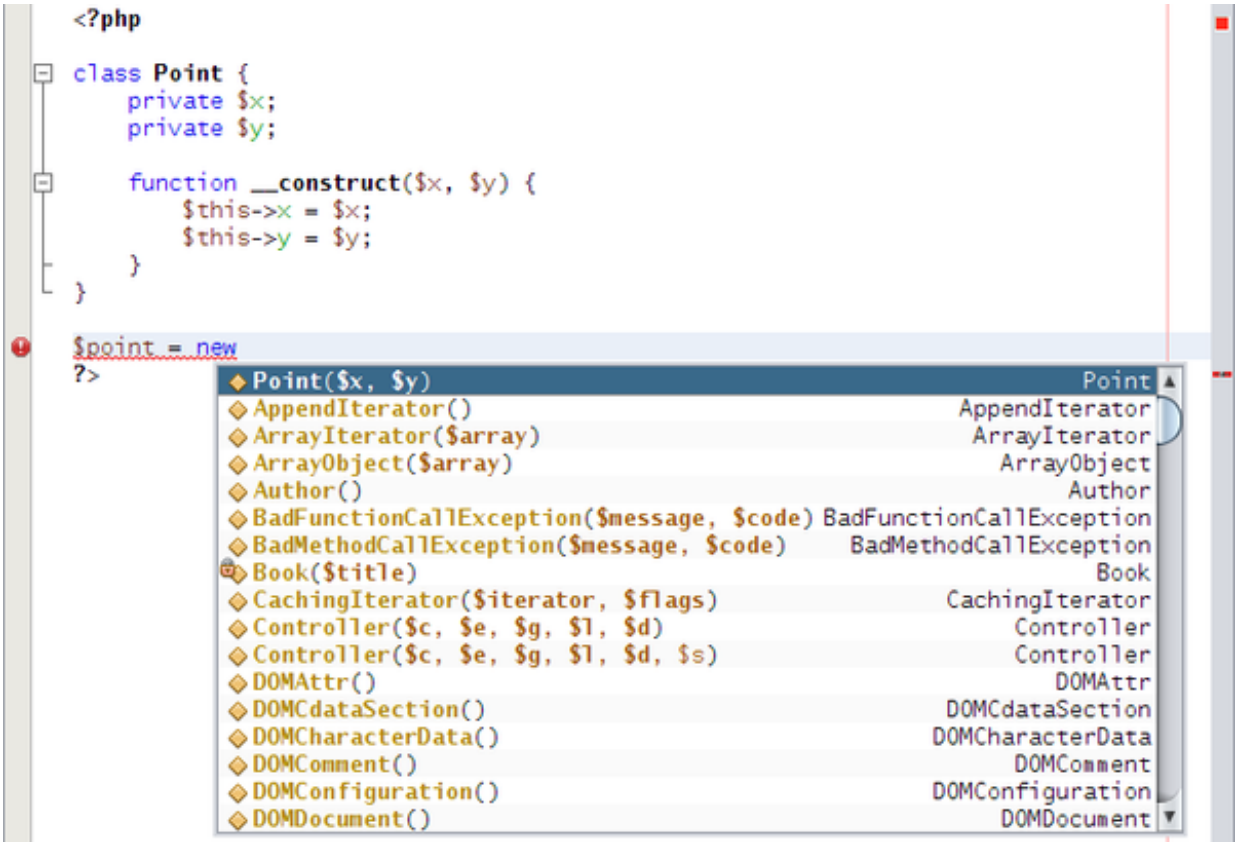
1. Choose Tools > Options > Editor > Code Templates.
2. From the Language drop down list, select PHP. The list of PHP abbreviations and code template defined for them is displayed.
3. To add or remove a definition from the list, use the New or Remove buttons respectively.
4. To edit a definition, select the relevant row and edit the text in the edit field below the list.



For more information, please see Code Templates in NetBeans IDE for PHP [5].

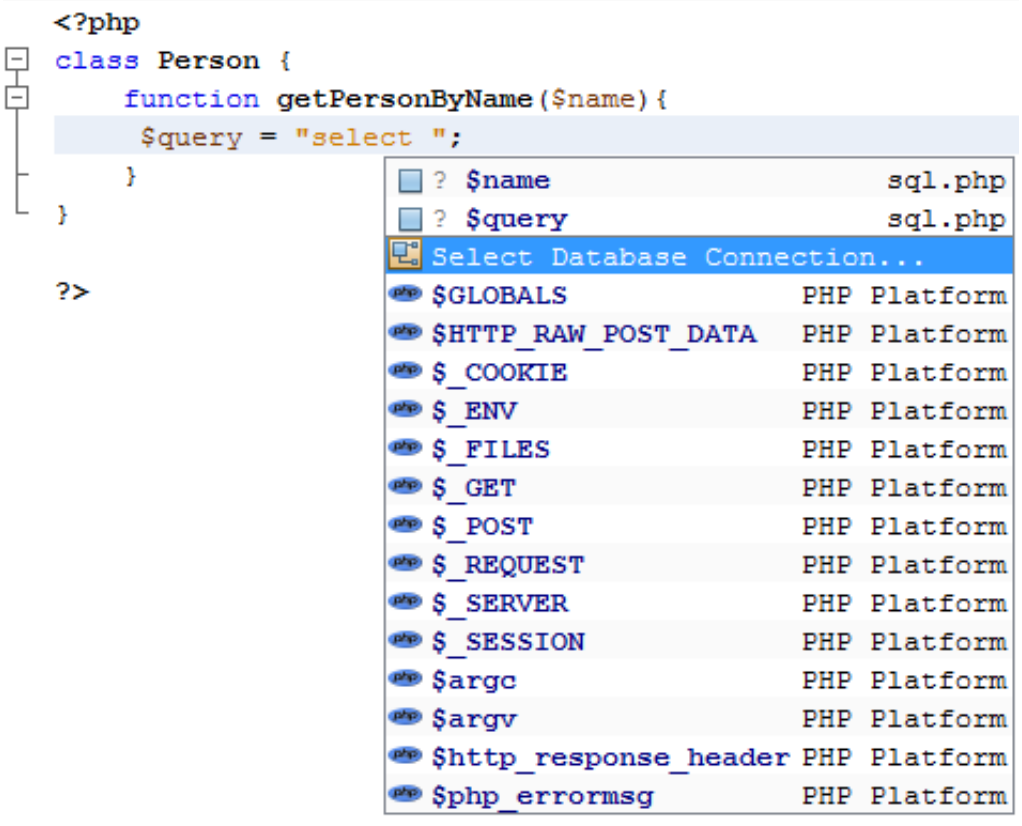
Code Completion for Constructors

After the new keyword, the code completion window is displayed with constructors and parameters for all available classes in the project.

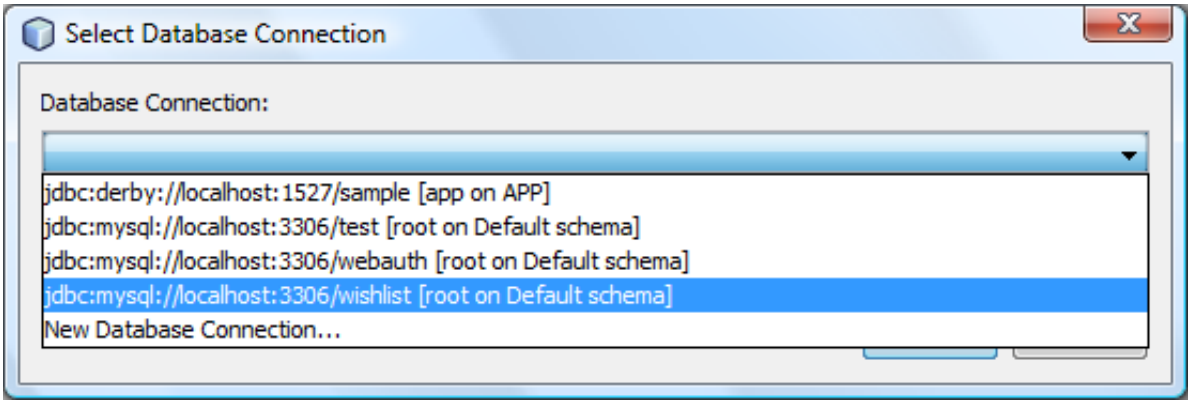


SQL Code Completion

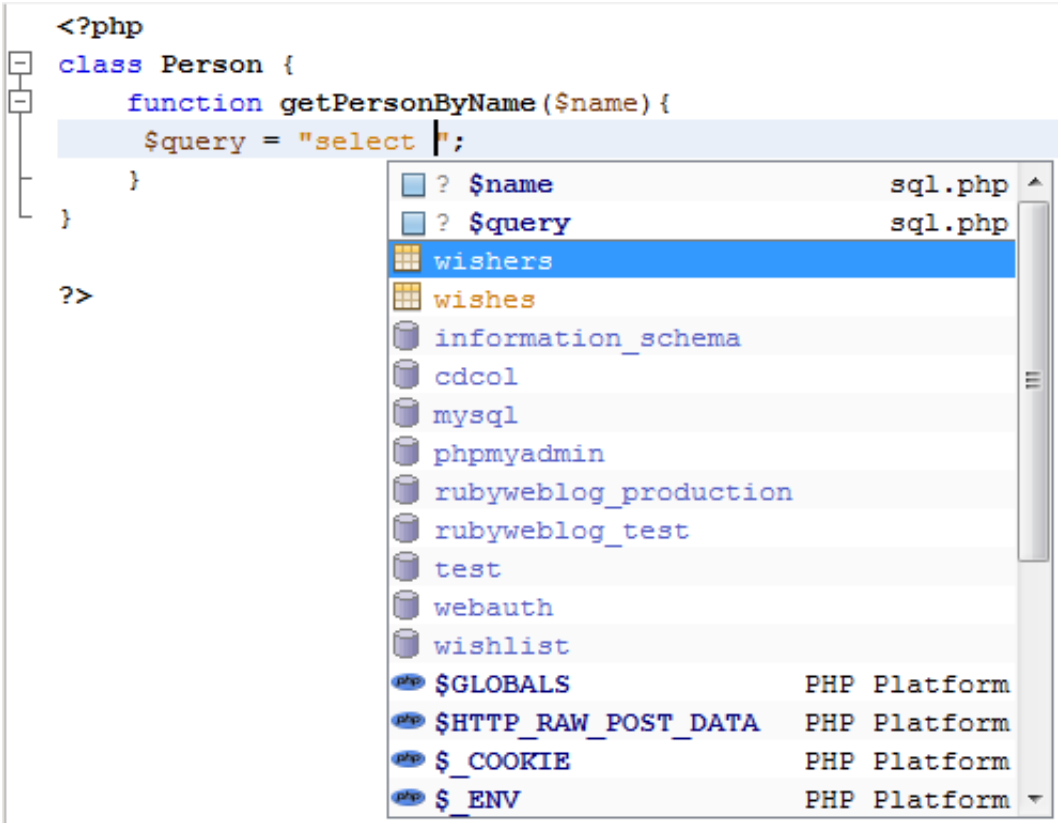
SQL code completion displays when a string begins with the SQL keyword "select." The first step is to select the database connection.



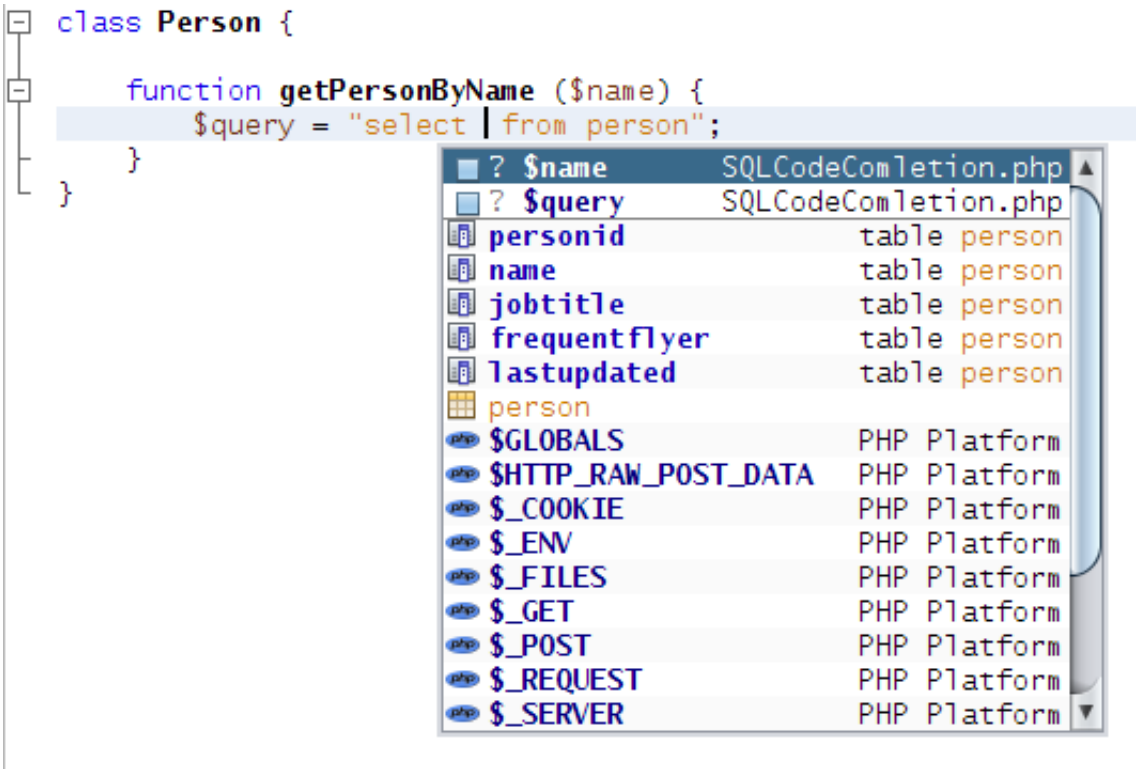
All database connections registered with the IDE are displayed.



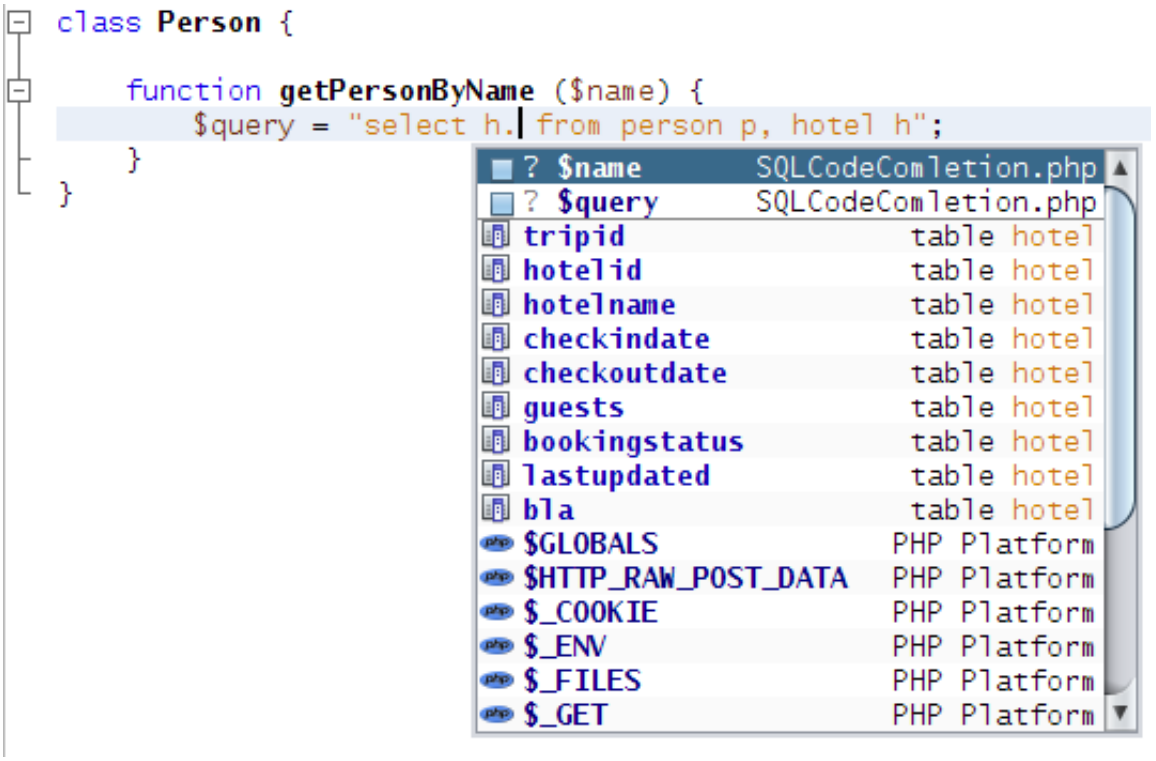
After you select the connection, SQL code completion offers all tables from that database connection.



If the table has columns, those are displayed as well.



SQL code completion also works with table aliases.



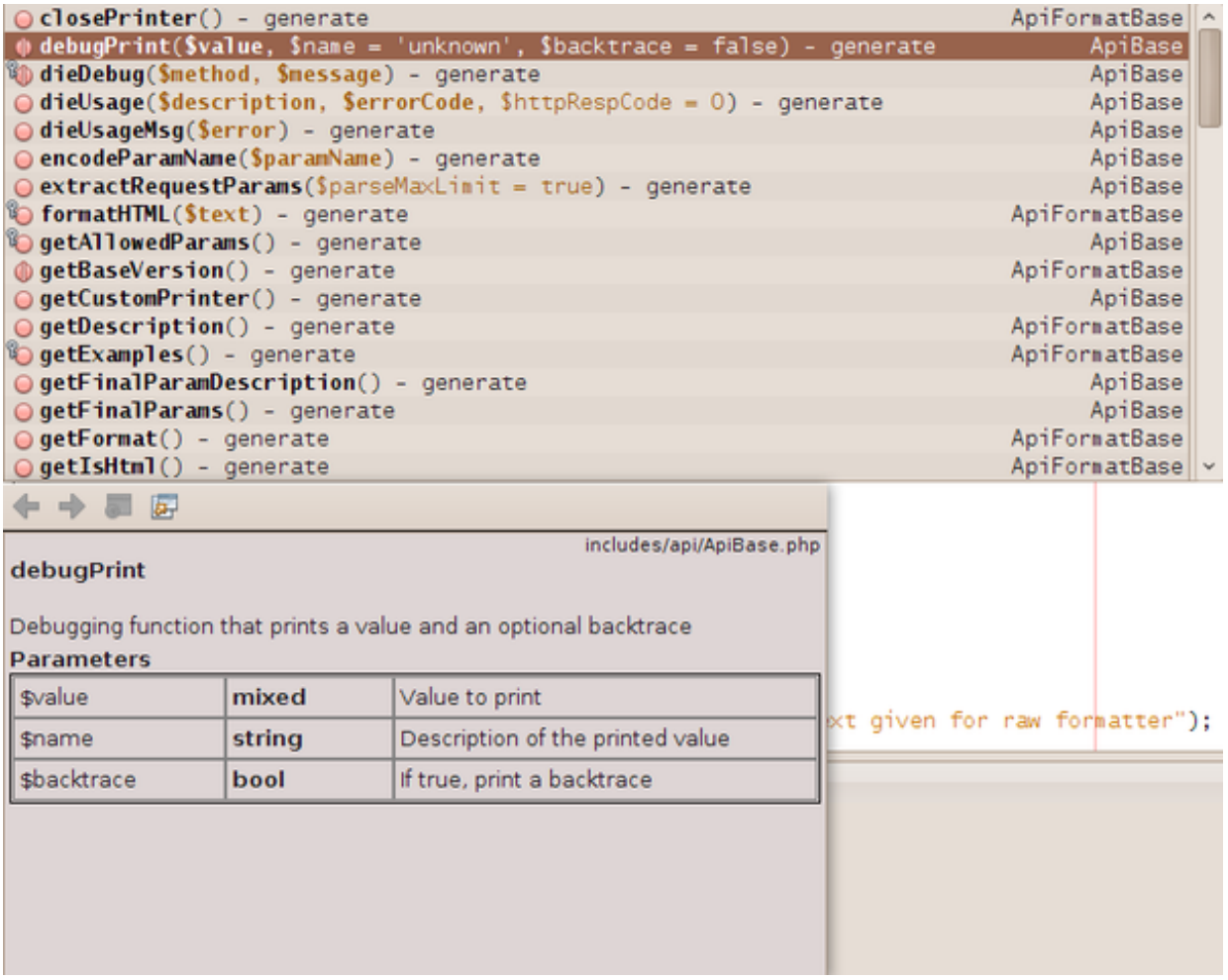
PHP 5.3 Namespaces

Code completion supports fully qualified, partially qualified, and unqualified namespace names from PHP 5.3. For more information, please see our screencast [6].

The IDE also helps you resolve missing namespace `use` statements. *Inside a namespace* where you want to fix missing `use` statements, either right-click and select Fix Uses..., or press Ctrl-Shift-I and go to Source > Fix Uses... A dialog opens offering fully qualified names for each needed use statement. For more information, see How to Fix Your Use Statements [7].

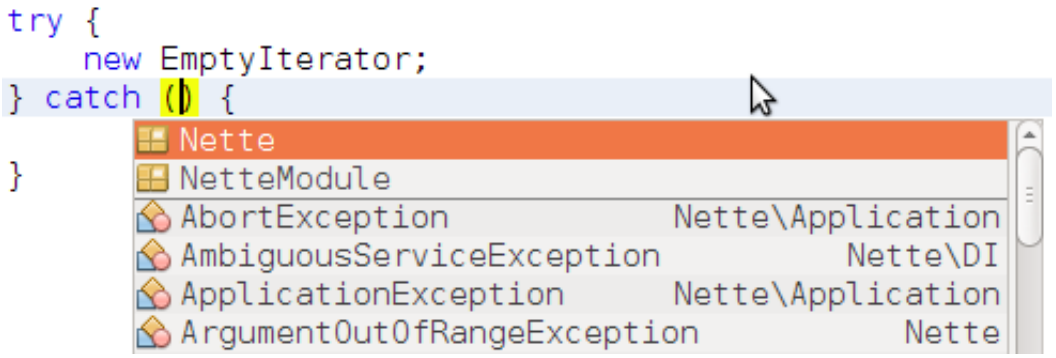
Overridden and Implemented Methods

Code completion between class members offers to override or implement methods.



Clever Try/Catch Code Completion

Starting in NetBeans IDE 7.1, code completion for `catch` statements includes only classes that inherit from `Exception`.



Annotations

NetBeans IDE code completion supports the following types of PHP annotations:

- ApiGen (legacy PHPDoc annotations)
- PHPUnit
- Doctrine 2 (ORM and ODM)
- Symfony 2

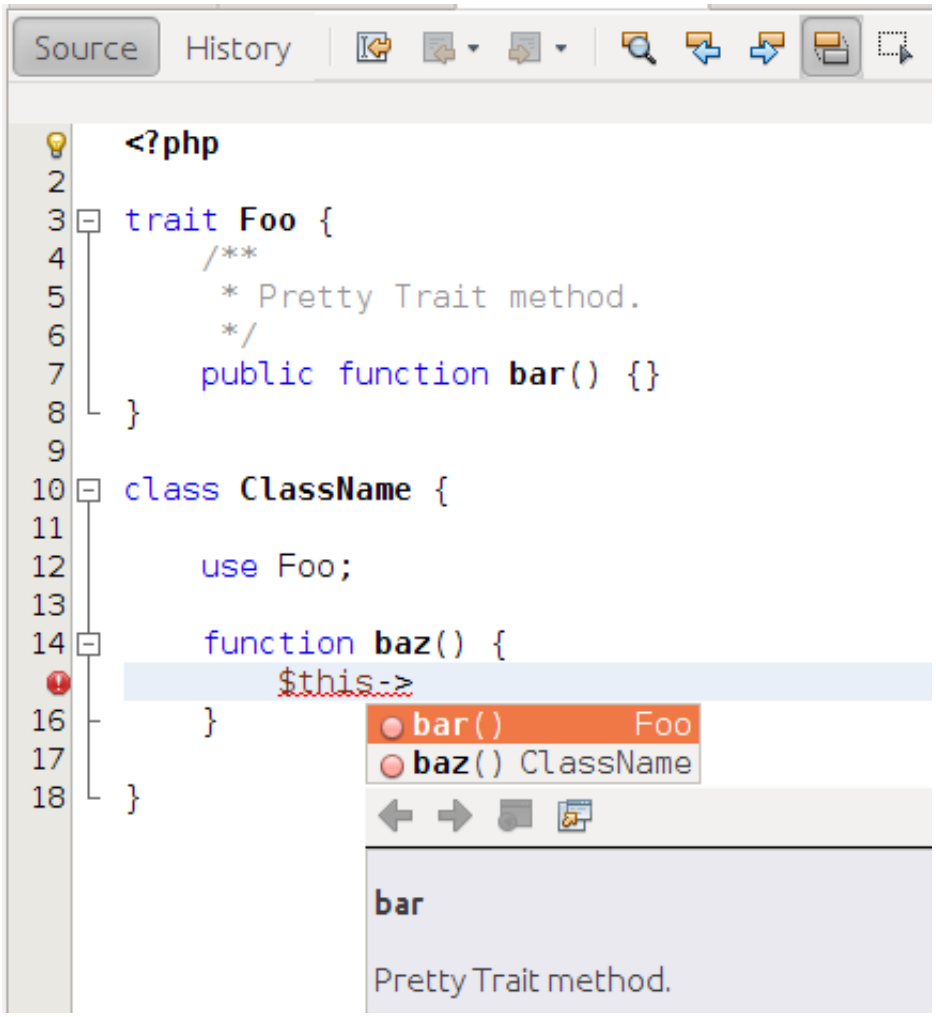
Every annotation can be associated with a context. NetBeans IDE recognizes four contexts:

- function
- class/interface (type)
- method
- field

You can add more annotations to code completion in Tools > Options > PHP > Annotations.

PHP 5.4 Traits

Traits are a mechanism of code reuse introduced in PHP 5.4.0. (For information about traits, see php.net documentation [8].) NetBeans IDE recognizes traits syntax and code completion includes methods, functions, and anything else inherited from or declared in a trait.



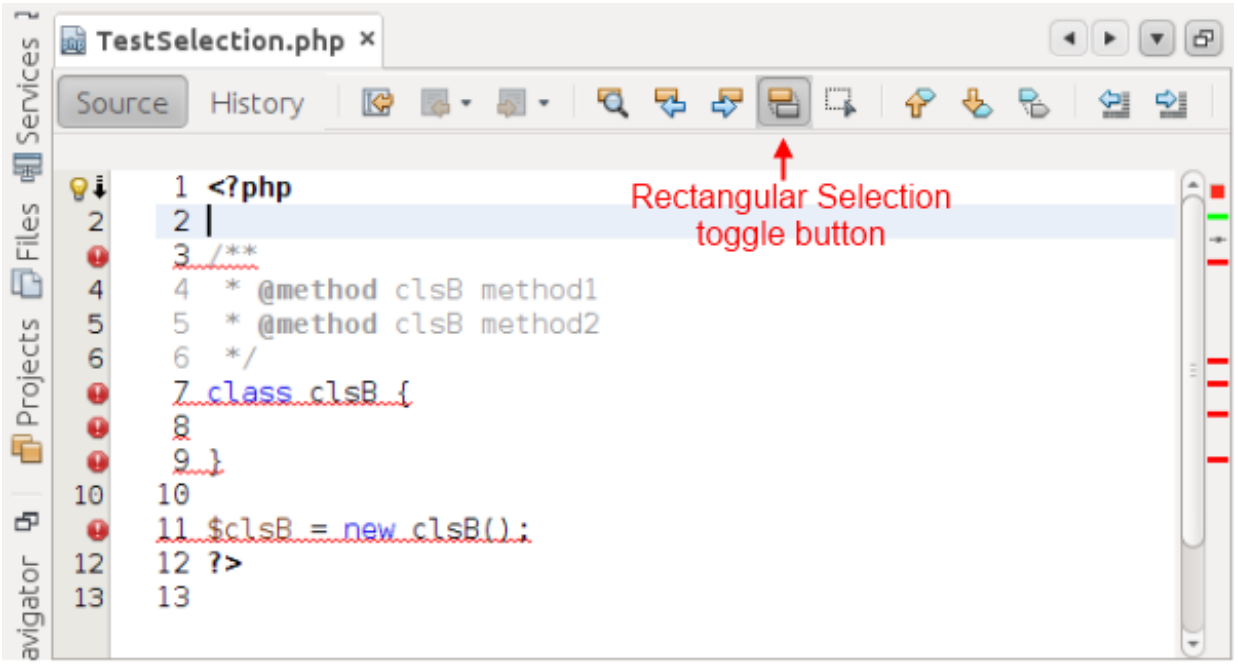
PHP 5.4 Support

NetBeans Editor for PHP offers a number of features specific to developing with PHP 5.4.

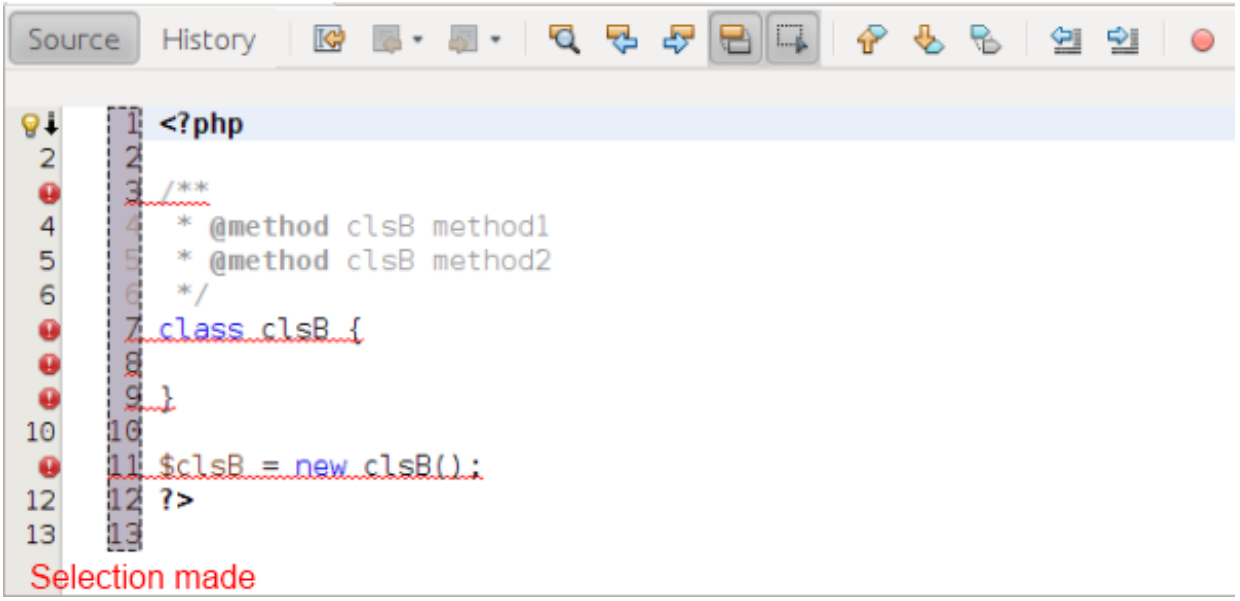
Rectangular Selection

NetBeans IDE 7.1 for PHP introduces the Rectangular Selection action. This editing function is useful in cases such as deleting line numbers in pasted code.

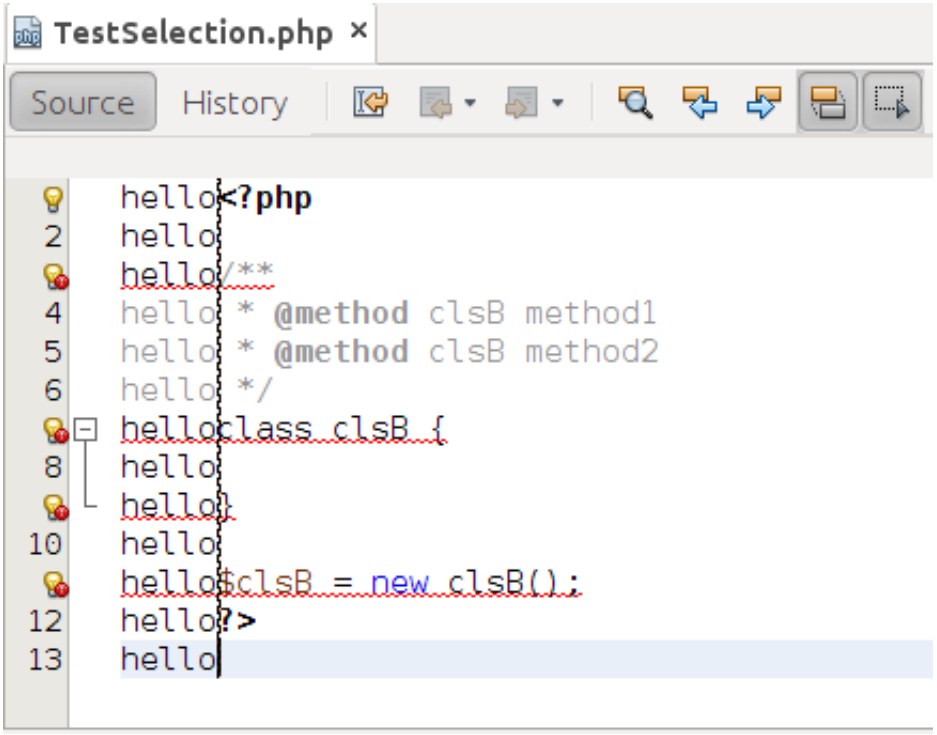
To enable rectangular selection, press the Rectangular Selection toggle button or Ctrl-Shift-R.



When Rectangular Selection is enabled, you can make the selection with mouse or keyboard. When you use the keyboard, place the caret on a corner, keep SHIFT pressed down, and select what you need with the arrows keys.



f you write text while an area is selected, the text is placed on every line in the selected area. This can be useful, for example, for simultaneously changing the access modifiers of multiple fields in a class.



ApiGen Support

Built-in support for ApiGen was added in NetBeans IDE 7.2. For more information, see the screencast Generating PHP Documentation With NetBeans IDE [9].

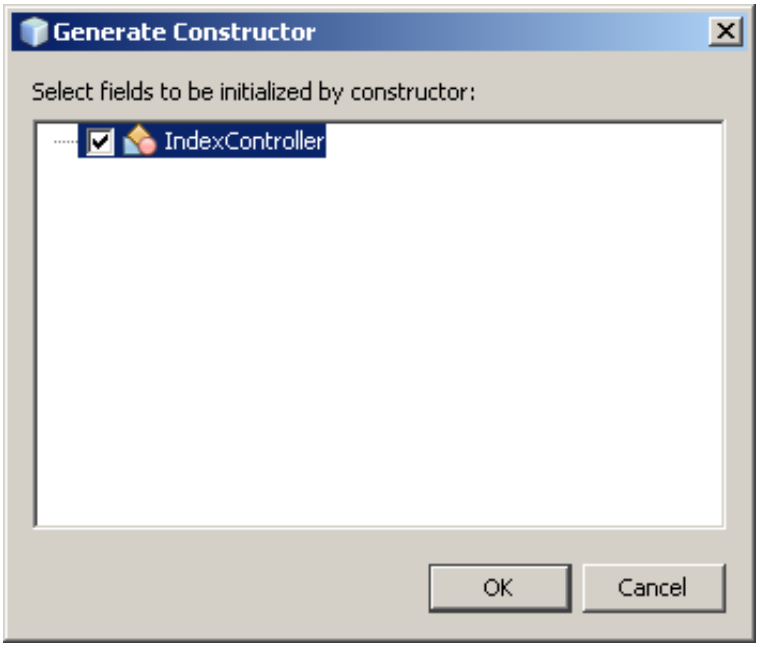
Warning: PHPDocumentor does not fully support PHP 5.3. Namespaces and some other 5.3 features are not documented by PHP Documentor.

Code Generators

When you press the combination Alt-Insert (Ctrl-I on Mac), a menu opens with all possible code generators. The list of generators is context sensitive. It depends on the position of the caret in the code when the key combination is pressed. Depending on your position in the code, you can generate a database connection, database tables, lorum ipsum text, and several others. This section describes only the following code generators:

Constructors

You can generate constructors by pressing Alt-Insert (Ctrl-I on Mac) when the caret is inside a class body, but not inside any functions contained in that body. When you select Generate... Constructor, a dialog opens llisting the fields you can initialize by the constructor. The field names are used as parameters of the constructor.



You can decide not to select any fields. In this case, the IDE generates an empty constructor, with no parameters. If the field is a class with properties, you can either select individual properties, or you can select the class, in which case all the class' properties are selected automatically.

For more information, see the NetBeans PHP blog post [10].

Getters and Setters

You can generate getters and setters by pressing Alt-Insert (Ctrl-I on Mac) when the caret is inside a class body and selecting Getter, Setter, or Getters and Setters. Only the possible functions are displayed. For example, if you already have setters for the available properties, only the getter option appears.

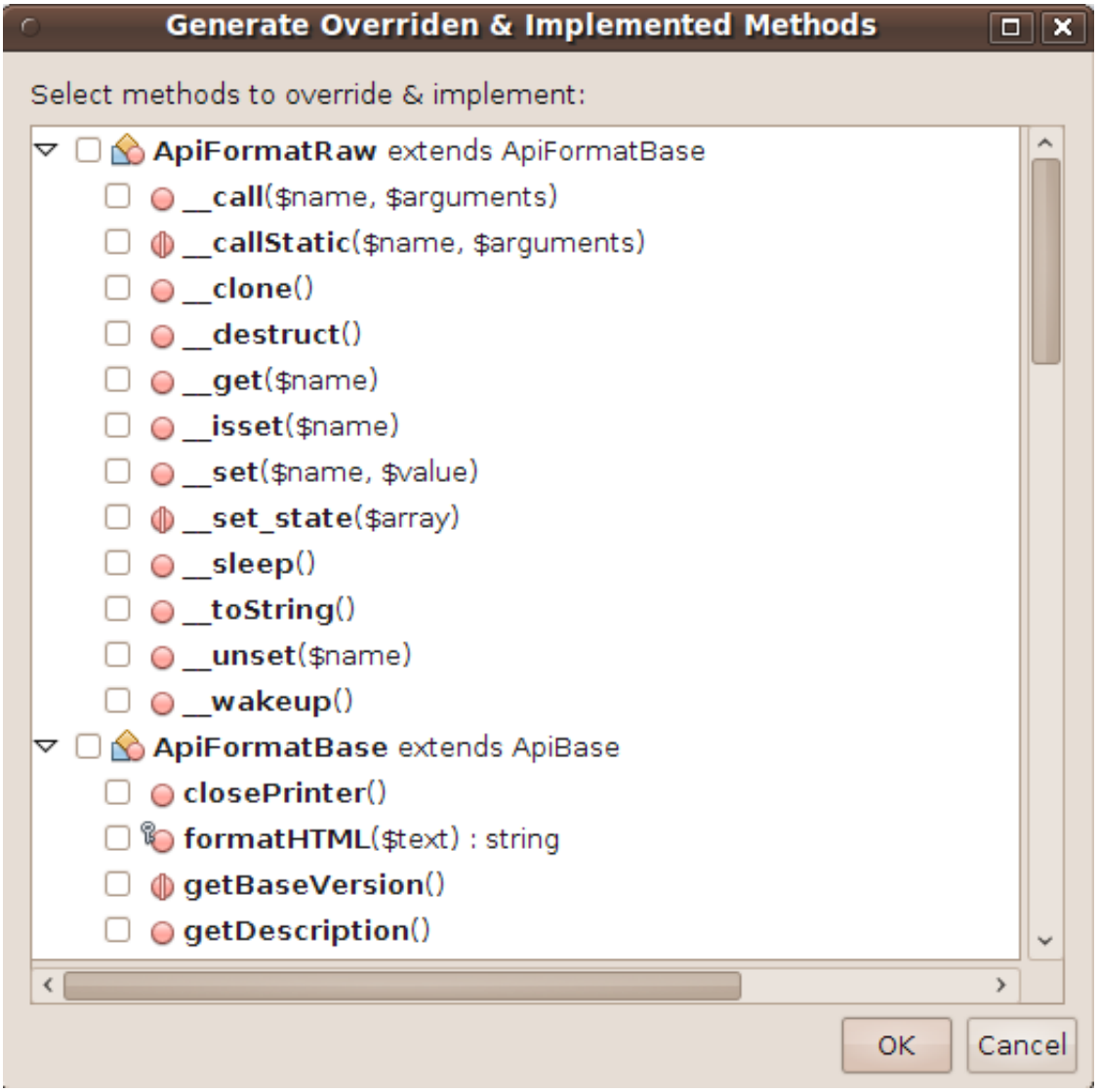
When you select Generate... Getter/Setter/Getter and Setter, a dialog appears with the properties for which you can generate a getter or setter. The properties are displayed in a tree. If you select a parent class, you automatically select all that class' properties.

You can name a getter or setter according to either the convention getName or the convention get_name.

For more information, see the original NetBeans PHP blog post [11] and the post on improved getter and setter generation [12].

Overridden and Implemented Methods

You can generate overridden or implemented methods by pressing Alt-Insert (Ctrl-I on Mac) when the caret is inside a class declaration and there are multiple class members. A dialog opens showing the methods you can insert and indicating whether they are overridden or implemented.



This feature complements the "Implement all abstract methods" hint and code completion for overridden and implemented methods.

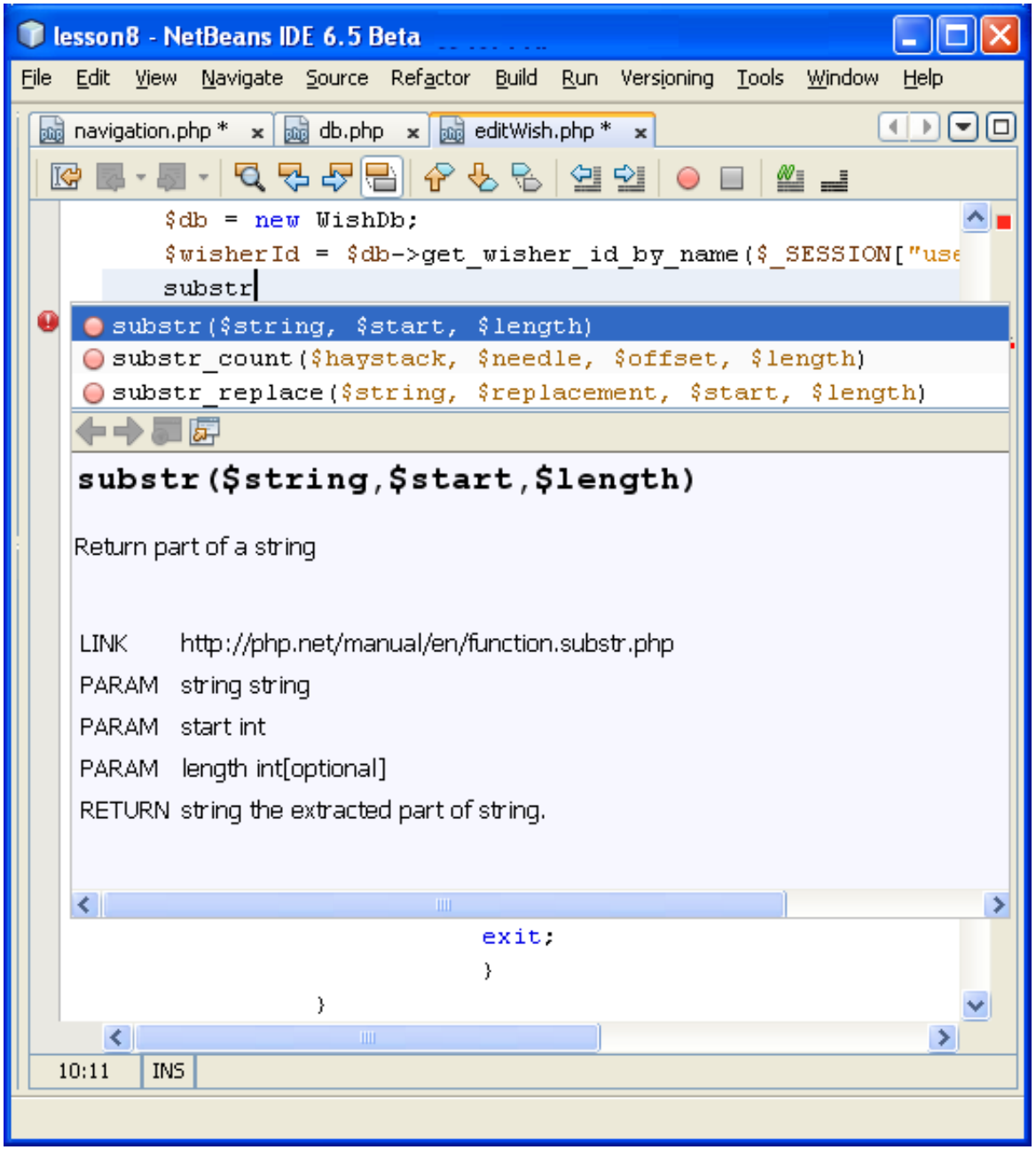
For more information, see the NetBeans PHP blog post [13].

Parameter Hints

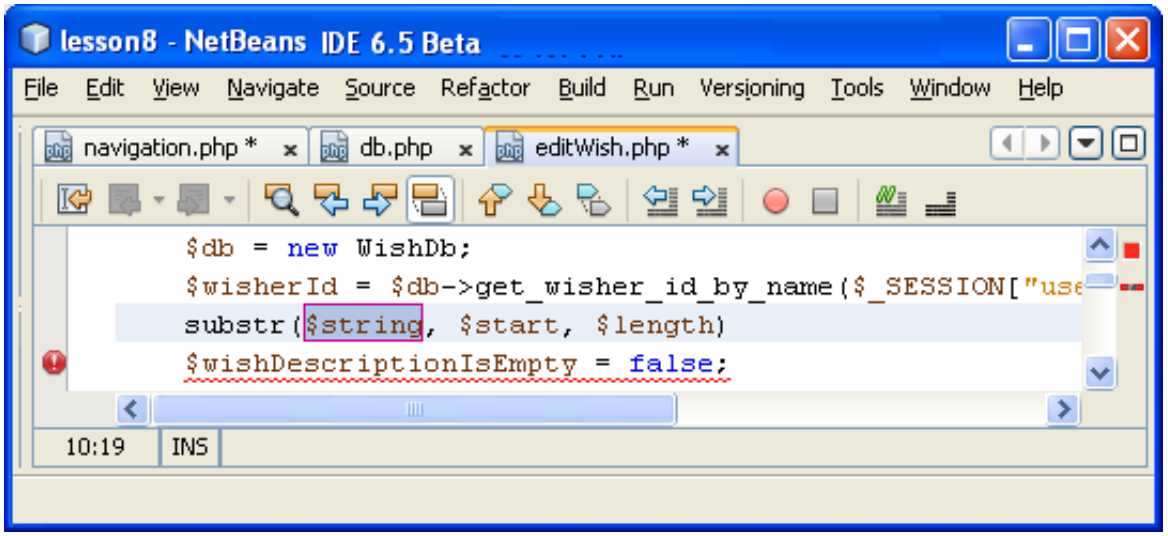
The editor prompts the user regarding the formal parameters of a function or a method in the context where the function or method is called.

1. Type the starting characters of the function you want to call.

2. Press Ctrl + Space. A dropdown list shows the context-sensitive proposals with the formal parameters for each proposal.

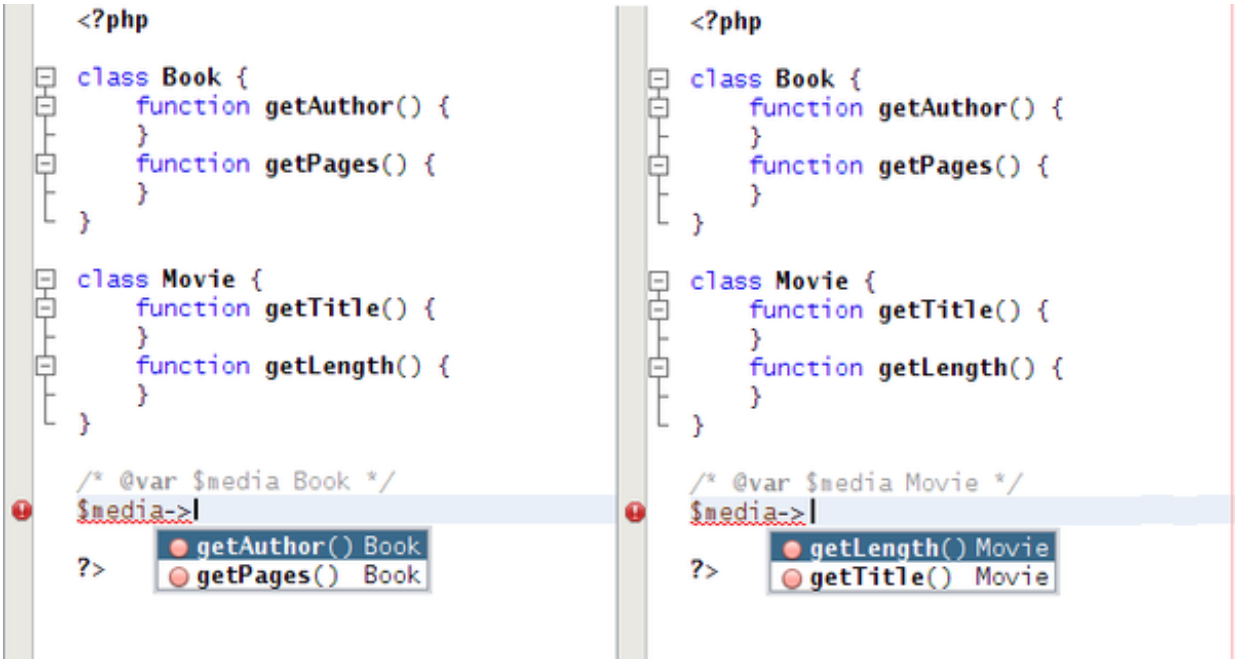


3. Choose the relevant proposal and press Enter. The name of the chosen function is inserted in the code and a template for entering the parameters is shown in brackets.



Defining Variable Type in Comments

You can define a variable and its type in a comment. The comment has to be in the format */* **@*var*** *$variable type */* . If the comment is written correctly, the ***var*** tag is in bold font.



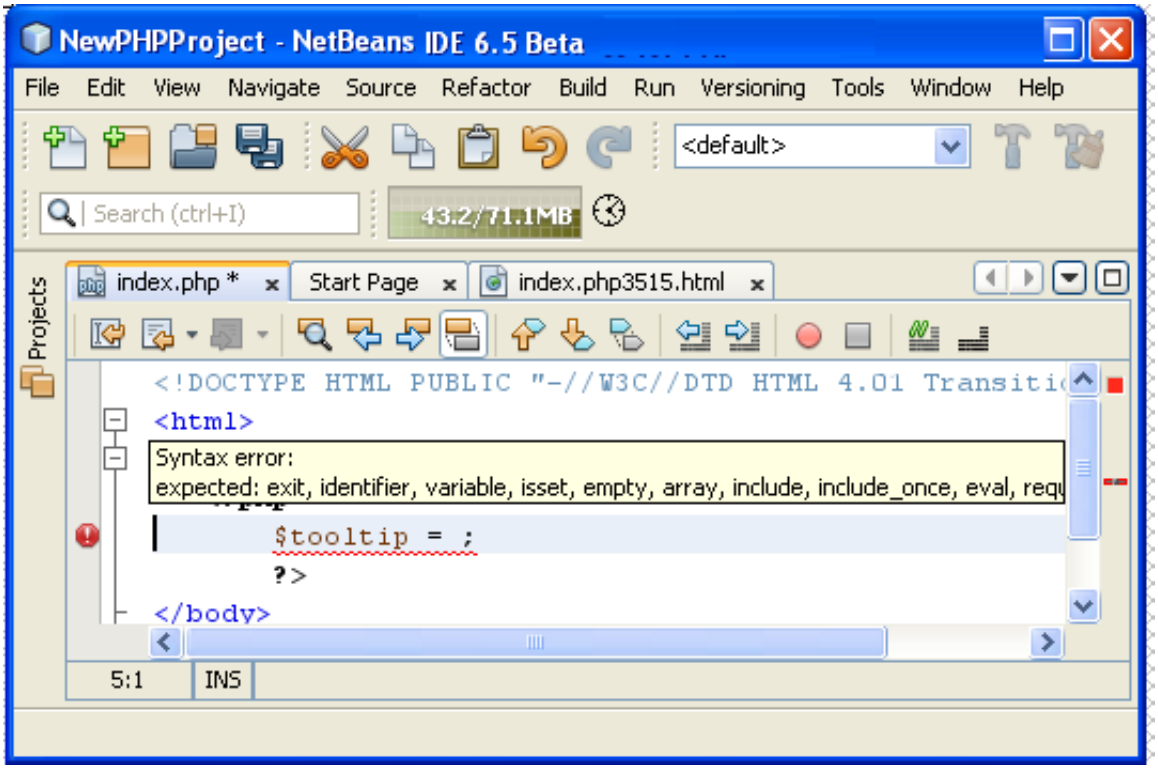
You can use the code template vdoc, followed by Tab, to generate a comment that defines a variable. The variable name is selected, and you can change it. Then press Tab again, and the type is selected.

The code template automatically sets the variable name and type. If a variable is used after the place where you insert the template, then that following variable name is suggested by default. If there is not any variable used after the place where you insert the template, but there is a variable used above the template, then that preceeding variable name is suggested as default. If NetBeans IDE is not able locate any variable near where you use the template, then the default name is *variable*. The variable type is set automatically according to the same rules.

Error Messages

The editor analyzes the syntax of the code while you type, and marks syntax errors by underlining the erroneous lines and setting a red bullet next to them. To get an

explanation of the error, mouse over the erroneous line or click the red bullet. A tooltip appears with a short explanation of the error.



PHPUnit and Selenium Tests

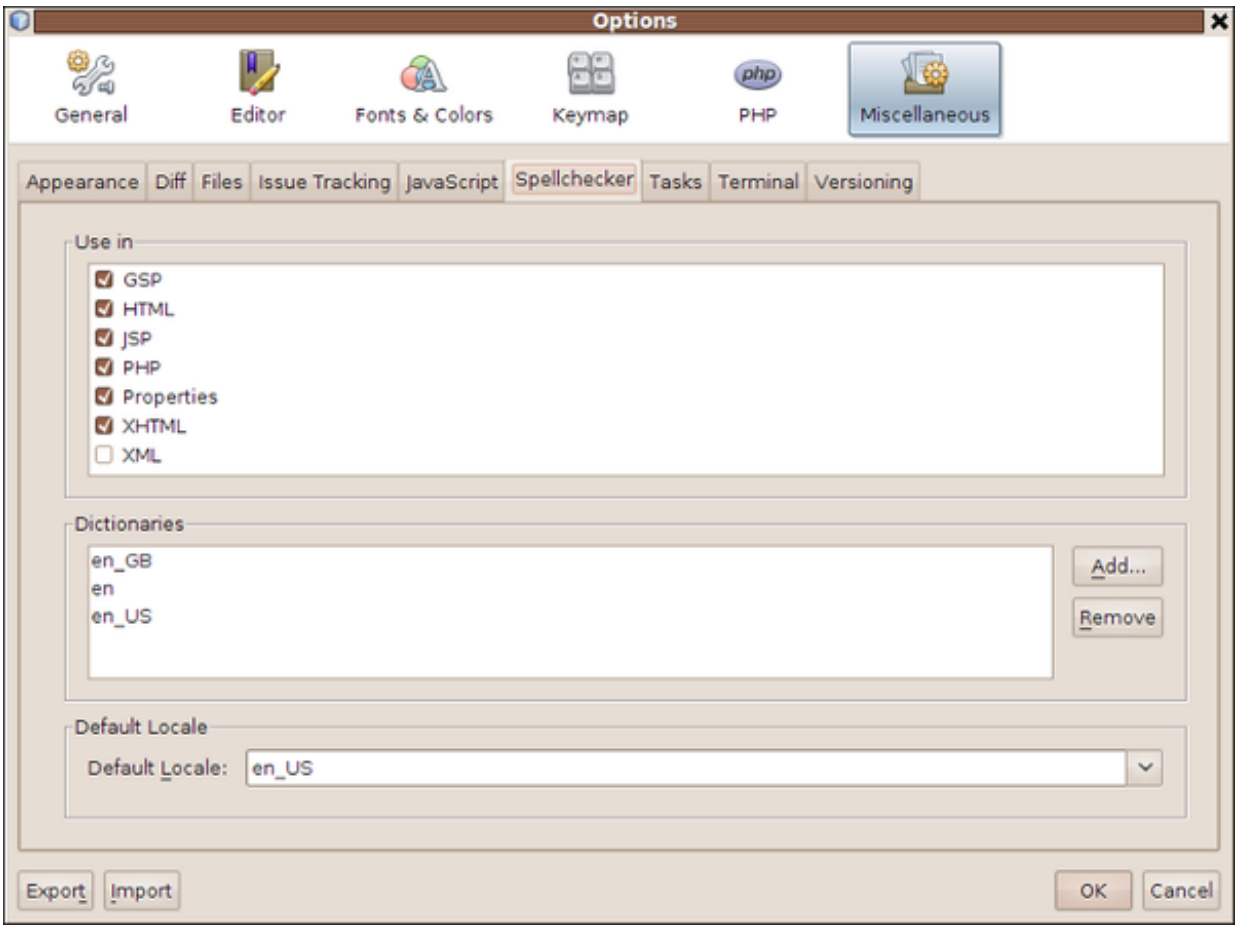
NetBeans IDE for PHP supports PHPUnit [14] automated tests, including code coverage. You can use test groups in your code.

NetBeans IDE also supports the Selenium portable test framework, in combination with PHPUnit. A Selenium plug-in is available from the Update Center. Installing this plugin adds a Selenium server to the IDE's registered servers and adds Selenium test options to the PHP menus.

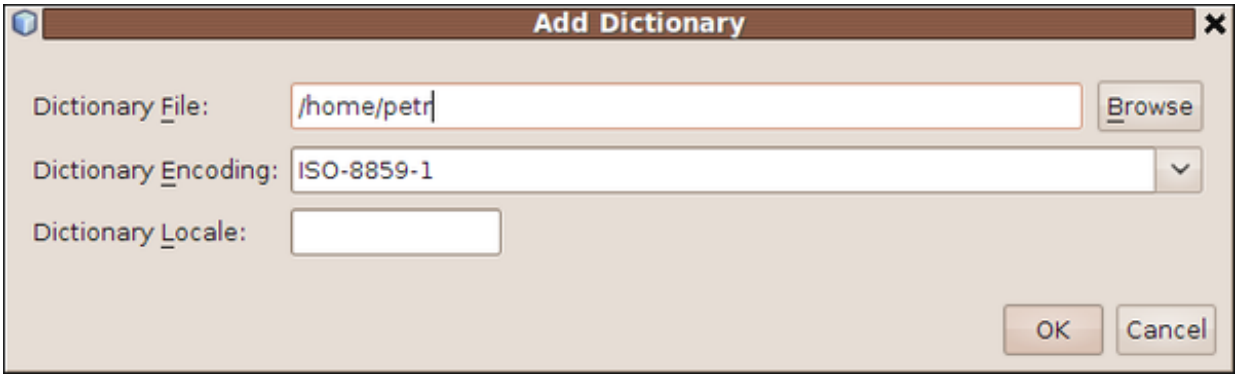
See Testing with PHPUnit and Selenium [15] for more information.

Spellchecker

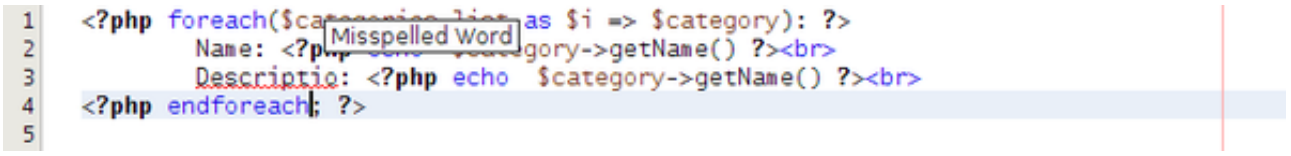
A spellchecker is available. In the Options dialog, you can switch spellchecking on or off for different languages.



In the Options dialog, you can add and remove new dictionaries and set the locale. When you want to add a new dictionary, you have to have a file with the words. The format of the dictionary file is a list of words with every word on a new line. You have to tell NetBeans the encoding of the dictionary file and for which locale the file should be used.



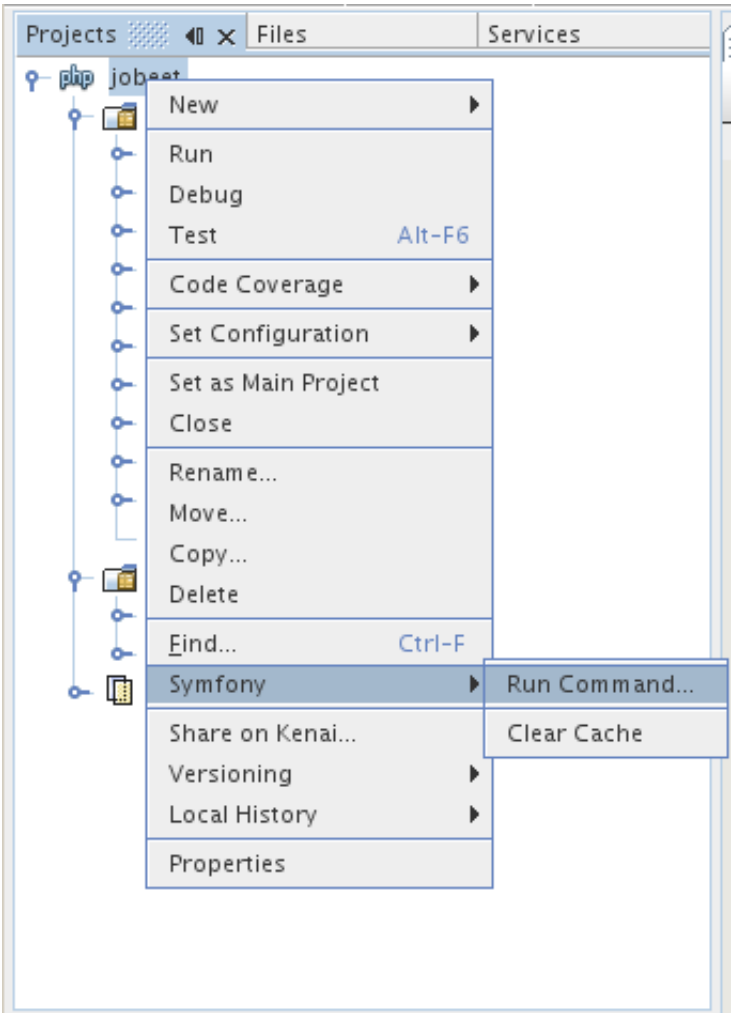
When the spellchecker finds a word that is not in its dictionary, it underlines that word in red. The spellchecker also adds a tooltip to that word that says "Misspelled word." Note that HTML text is checked in PHP files.



Symfony, Symfony2, and Zend Framework Support

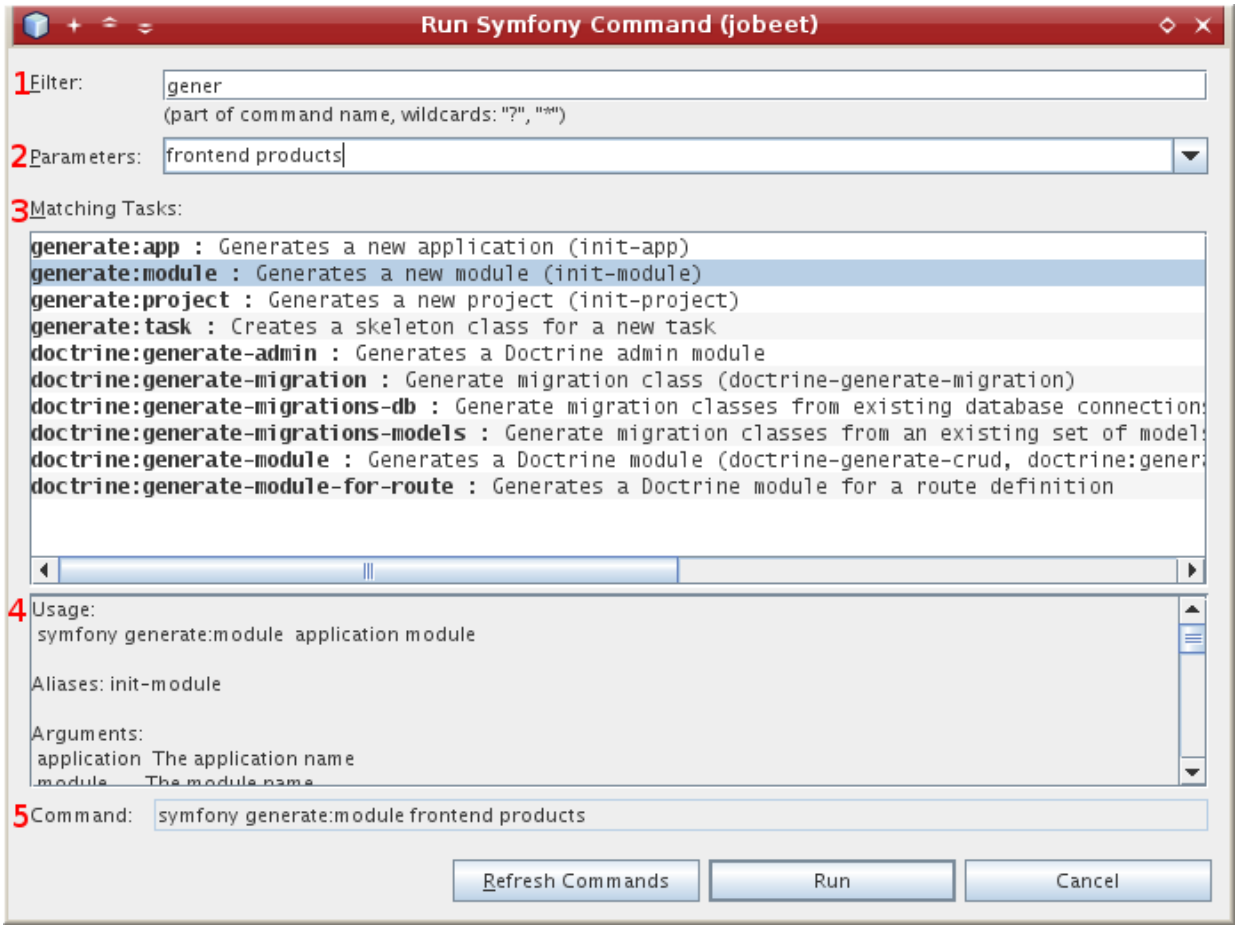
Symfony, Symfony2, and Zend framework support are built-in to the PHP editor. It recognizes existing Symfony and Zend frameworks, and you can add the framework to new projects in the last panel of the new PHP project wizard.

In a PHP project with Symfony, Symfony2, or Zend support, you can run Symfony or Zend commands from the project's context menu.



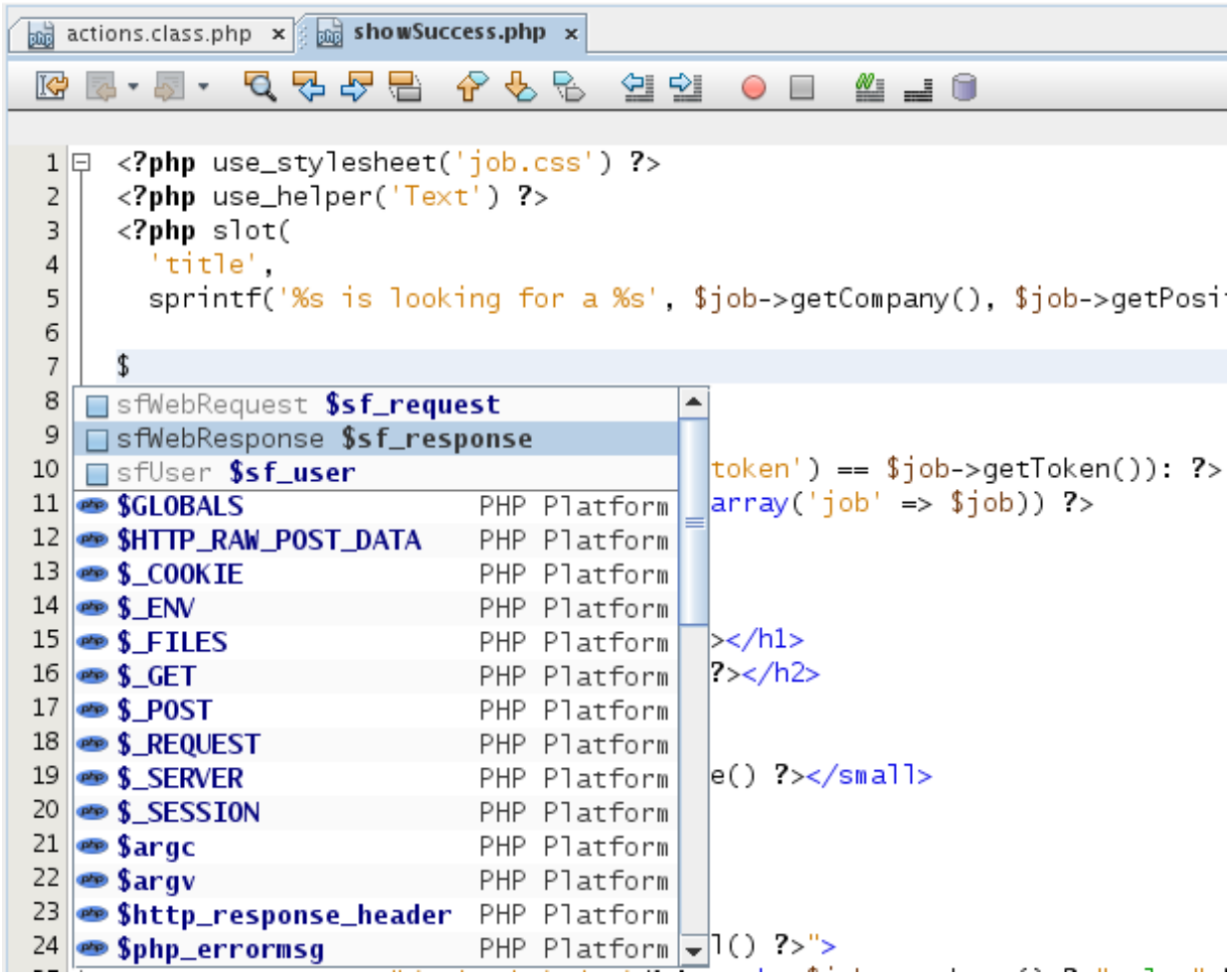
Use the UI for the commands to:

- Filter the Symfony/Zend commands (1)
- Specify command parameters (2)
- Select a command (3)
- See the command's context help (4)
- Preview the entire command that will be run (5)



The list of commands is project specific. This means that if one uses e.g. Doctrine instead of Propel, no Propel commands are listed, just the Doctrine ones. Also, all the commands from the project's Symfony plugins are available.

Code completion includes Symfony variables in Symfony views, Symfony2 variables in Symfony2 views, and Zend variables in Zend views.



The IDE also provides navigation shortcuts between Symfony or Zend views and actions. These shortcuts were improved in NetBeans 6.9. In addition, code completion in *view* templates/files now includes variables that are declared in *controller/action* files. See the NetBeans PHP blog post [16] for details.

For more about our Symfony framework support, please see the Symfony screencast [17].

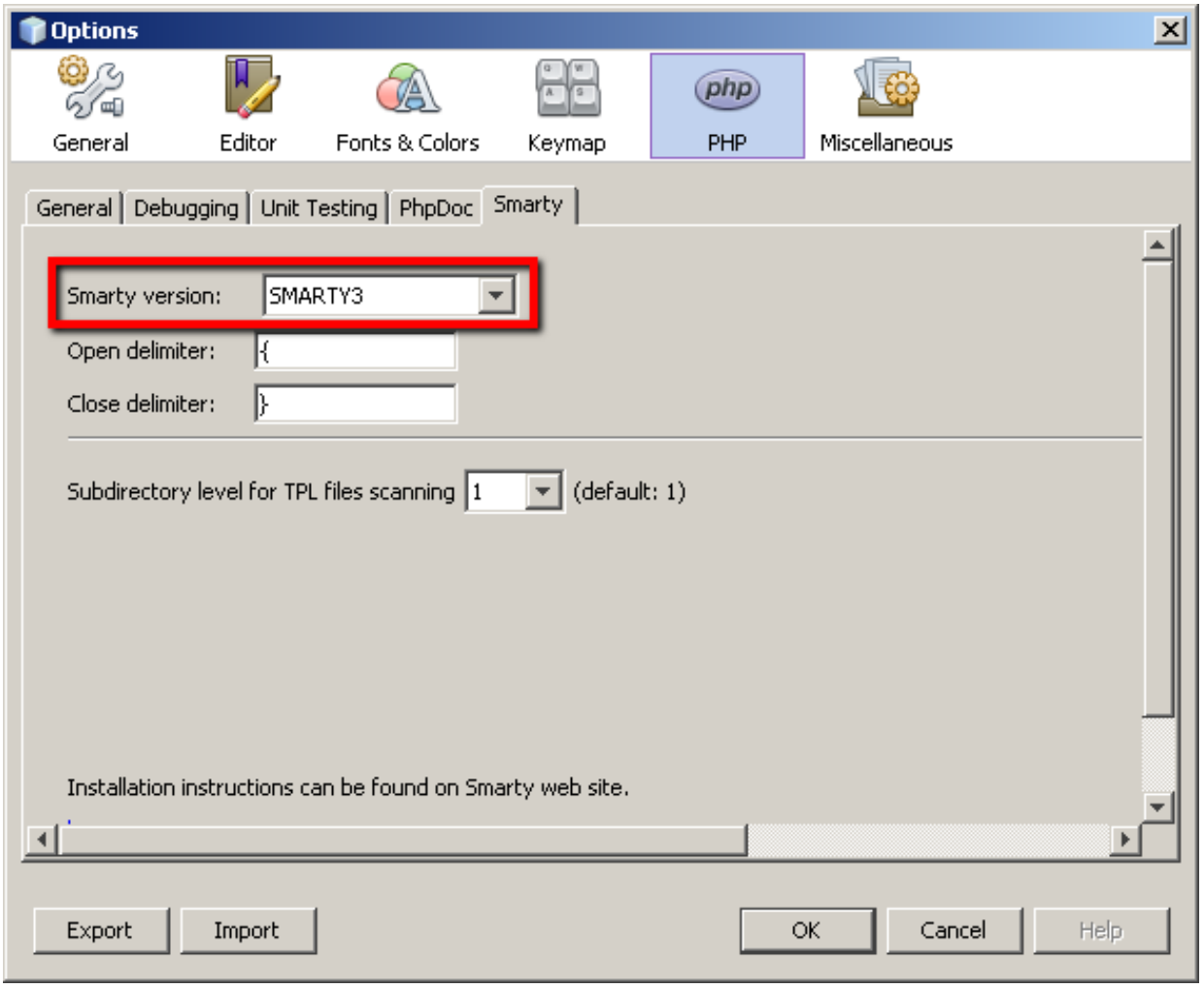
For more about our Zend framework support, please see the Zend screencast [18].

Smarty Support

NetBeans IDE includes a plugin for Smarty framework support. This support includes basic Smarty 3 features, namely Smarty 3 code completion and delimiters.

```
{if ($includeMyFunction)}
    <script type="text/javascript">
        function myFunction() {
            alert("Hello World!");
        }
    </script>
{/if}
```

Toggle between Smarty 3 and Smarty 2 behavior in the Options window. Go to Tools > Options (NetBeans Preferences on MacOS), select PHP, and open the Smarty tab. The Smarty framework plugin must be installed.



Doctrine2 Support

If you have Doctrine2 ORM and its requisite packages installed, you can use NetBeans IDE's code completion with Doctrine2 functions and you can run Doctrine2 commands from the IDE. You need to set the path to the Doctrine2 run script in the IDE's Options, and your PHP project has to have Doctrine2 enabled. See Doctrine2 Support Added [19] in the NetBeans PHP blog for details.

Send Feedback on This Tutorial [20]

To send comments and suggestions, get support, and keep informed on the latest developments on the NetBeans IDE PHP development features, join the users@php.netbeans.org mailing list [21].

Back to the PHP Learning Trail [22]

1. http://www.netbeans.org/downloads/index.html
2. http://httpd.apache.org/download.cgi
3. http://www.xdebug.org/
4. https://netbeans.org/kb/docs/php/screencast-rename-refactoring.html
5. https://netbeans.org/kb/docs/php/code-templates.html
6. https://netbeans.org/kb/docs/php/namespace-code-completion-screencast.html
7. https://blogs.oracle.com/netbeansphp/entry/how_to_fix_your_use
8. http://php.net/manual/en/language.oop5.traits.php
9. https://netbeans.org/kb/docs/php/screencast-apigen.html
10. http://blogs.oracle.com/netbeansphp/entry/generate_constructor_getters_and_setters
11. http://blogs.oracle.com/netbeansphp/entry/generate_constructor_getters_and_setters
12. http://blogs.oracle.com/netbeansphp/entry/generating_getters_and_setters_improved
13. http://blogs.oracle.com/netbeansphp/entry/generate_overriden_implemented_methods
14. http://www.phpunit.de/
15. https://netbeans.org/kb/docs/php/phpunit.html
16. http://blogs.oracle.com/netbeansphp/entry/framework_support_improvements
17. https://netbeans.org/kb/docs/php/symfony-screencast.html
18. https://netbeans.org/kb/docs/php/zend-framework-screencast.html
19. https://blogs.oracle.com/netbeansphp/entry/doctrine2_support_added
20. https://netbeans.org/about/contact_form.html?to=3&subject=Feedback:%20PHP%20Editor%20Guide
21. https://netbeans.org/community/lists/top.html
22. https://netbeans.org/kb/trails/php.html