Test suite: Create a new Php Project

Purpose: Check the Wizard for creation empty Php Project.
Setup: PHP engine and Apache 2.2 are configured. Apache 2.2 is installed to default location.

1. **Create an empty Project with default settings.**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP. In the Projects area, choose PHP Application to create an empty project and click Next.
    3. Accept default values for Name and Location step. Press Finish button.
    4. EXPECTED RESULT: Created Php project appears on Project tab. index.php opens in the source view. Project Sources are located in htdocs directory of Apache.
    5. ACTUAL RESULT: Luckily Project Sources are not located in htdocs of Apache. Practice to place the PHP application in the web root is dangerous. At least there should be web folder and index.php in it. There could be two nodes in the project view: Web Root and Source Files.

1. **Create empty Project with the given name**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP, in the Projects area, choose PHP Application to create an empty project and click Next.
    3. Type Project Name myPhpProject. the rest fields use default values for Name and Location step, press Finish button.
    4. Repeat above steps for creating project with the same name.
    5. EXPECTED RESULT: There should be the warning message: Source directory is already NetBeans project.
    6. OK. The message is: Sources directory is already NetBeans project (maybe only in memory). http://ubuntuone.com/p/d1y/

2. **Create empty Project with incorrect name**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP, in the Projects area, choose PHP Application to create an empty project and click Next.
    3. For Name and Location step type Project Name myPhpProject*. Accept default values for the rest fields, press Cancel button.
    4. EXPECTED RESULT: There should be warning message: Source directory is not a valid folder name.
    5. OK

3. **Create empty Project and put metadata into a separate directory**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP, in the Projects area, choose PHP Application to create an empty project and click Next.

    3. Check "Put NetBeans metadata into a separate directory" checkbox. Accept default values for the rest fields, press Finish button.
    4. EXPECTED RESULT: Created Php project appears on Project tab. Index.php opens in source view. Project Sources are located in the Source Folder and metadata are located in a separate directory.
    5. OK

1. **Create empty Project with renamed index.php file**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP, in the Projects area, choose PHP Application to create an empty project and click Next.
    3. Accept default values for the rest fields, press Finish button.
    4. Rename index.php Index File to myindex.php.
    5. Invoke Properties for this PHP Application. Slect Run Configuration on Properties window. Press "Browse..." button for index file. Select myondex.php file. Press Select File button. Press OK.

- EXPECTED RESULT: Created Php project appears on Project tab. Index file has new name myindex.php opens
- OK

2. **Create empty Project and choose copy location.**

- In the Projects window, choose File > New Project.
- In the Categories list, select PHP, in the Projects area, choose PHP Application to create an empty project and click Next.
- Accept default values for the rest fields, press Next button.
- Set checkbox 'Copy files from Sources Folder to another location' on. Press Browse button and select path for copying. Press Finish button.
- EXPECTED RESULT: Created Php project appears on Project tab. index.php opens in source view. Project Sources are located in the Sources Folder and have a copy in Copy Location.
- OK

1. **Create empty Project and set copy location to the doc root.**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP, in the Projects area, choose PHP Application to create an empty project and click Next.
    3. Select NetBeansProjects directory for the Sources Folder combobox. Accept default values for the rest fields, press Next button.
    4. Set checkbox 'Copy files from Sources Folder to another location' on. Select htdocs\PhpProjectN for the Copy to Folder combobox. Press Finish button.
    5. EXPECTED RESULT: Created Php project appears on Project tab. index.php opens in source view. Project Sources are located in the Project Folder and have a copy in htdocs

directory of Apache.

6. ACTUALL RESULT: Sources cannot be copied to the Apache default web root since by default only owner, www-data system user, has write access to the folder /var/www. However, the message *Target Folder cannot be created.* will be displayed and the user can select writable folder. http://ubuntuone.com/p/d2A/

Test suite: Create Air Alliance Sample Application

Purpose: Create the sample and work with it.
Setup: PHP engine, Apache 2.2 and MySQL data base are configured. Apache 2.2 is installed to default location.

1. **Create Air Alliance Sample**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select Samples>PHP. In the Projects area. Choose Air Alliance Sample Application to create the sample. Press Next.
    3. Name of Project is AirAllianceN. Press Finish.
    4. Follow the instruction in the appeared readme.html.
    5. EXPECTED RESULT: Created Air Alliance Sample appears on Project tab. There should be opened readme.html in default browser. Finally, we can run the project.

Test suite: Check Properties of PHP Project

Purpose: Check Properties of PHP Project
Setup: PHP engine, Apache 2.2 and MySQL data base are configured. Apache 2.2 is installed to default location.

1. **Check Properties of php Project with default settings.**
    1. In the Projects window, choose File > New Project.
    2. In the Categories list, select PHP. In the Projects area, choose PHP Project to create an empty project and click Next.
    3. Accept default values for Name and Location step, press Finish button.
    4. Created Php project appears on Project tab. Right click on the project and select Properties.
    5. EXPECTED RESULT: Project Properties window appears. There are six Categories: Sources,Run Configuration, PHP Include Path, JavaScript Libraries, Debug, Formatting.
    6. OK
2. **Check Properties of php Project with default settings.**
    1. Create a PHP Project with default settings. (Source folder is htdocs folder of Apache, Copy folder is not specified.)
    2. Right click on the project and select Properties. Select Sources node in the Categories tree.
    3. Set check box 'Copy files from Sources Folder to another location' on. Expand combobox and select htdocs\PhpProjectN+1 folder. Press Ok button.

4. EXPECTED RESULT: Project has a copy of sources in Copy Location.
5. ACTUALL RESULT: Sources cannot be copied to the Apache default web root since by default only owner, www-data system user, has write access to the folder /var/www. However, the message *Target Folder cannot be created.* will be displayed and the user can select writable folder. http://ubuntuone.com/p/d2A/

Test suite: Code Completion

Purpose: Check that code completion works properly in PHP code and HTML for PHP source and PHP web page.
Prerequisites: a PHP project is created in the Netbeans IDE.

1. **Create a PHP source file.**
    1. In the main menu, choose File > New File...
    2. In the Categories list, select Scripting. In the File Types list, choose PHP File. Click Next.
    3. Accept default values by pressing Finish button.
    4. EXPECTED RESULT: A new PHP web page is created and opened in the editor.
    5. OK, except that PHP page file is created and not PHP web page, which is OK, of course.
2. **Verify automatic code completion invocation.**
    1. In PHP source, type $.
    2. EXPECTED RESULT: Code completion is invoked automatically - a list of available variables appears.
    3. OK
3. **Verify local variable code completion.**
    1. Create a PHP function.
    2. Declare a variable in the function.
    3. Type several first symbols of the variable name and invoke code completion.
    4. EXPECTED RESULT: Code completion appears and the variable name is among suggestions.
    5. OK
4. **Verify global variable code completion.**
    1. Create a PHP function.
    2. Declare a global variable.
    3. Type several first symbols of the variable name and invoke code completion.
    4. EXPECTED RESULT: Code completion appears and the variable name is among suggestions.
    5. OK
5. **Verify variable from included file code completion.**
    1. Include an other PHP file into PHP source.
    2. Open that PHP source and create a global variable there.

    3. Create a PHP function in the first source.
    4. Inside that function, type several first symbols of the variable name and invoke code completion.
    5. Choose that variable in the list and press Enter.
    6. EXPECTED RESULT: Code is completed properly. No require or include directive is auto-generated.

6. **Verify variable from required file code completion.**
    1. Add an other PHP file into PHP source using the require directive.
    2. Open that PHP source and create a global variable there.
    3. Create a PHP function in the first source.
    4. Inside that function, type several first symbols of the variable name and invoke code completion.
    5. Choose that variable in the list and press Enter.
    6. EXPECTED RESULT: Code is completed properly. No require or include directive is auto-generated.
    7. OK
    8. ACTUAL RESULT: Variable declared in some other PHP file is present in the code completion regardless of that if the file where variable is defined is added using require/include directive. When working with legacy PHP code, having polluted global scope, it will make sense to display only variables from added files. However, current behavior is consistent with modern usage and standards: Instead using require/include auto-loader will be used and modern PHP applications tend to keep glocal scope clear.
7. **Verify code completion inside the identifier.**
    1. Place the caret inside of existing PHP identifier.
    2. Invoke code completion.
    3. EXPECTED RESULT: Code completion appears and the list of suggestions corresponds to the beginning of the identifier.
    4. OK
8. **Verify documentation hints for built-in identifiers.**
    1. Invoke code completion for a whitespace.
    2. Select a built-in identifier in the list (like $GLOBALS, $_ENV)
    3. EXPECTED RESULT: In the details code completion view, a description of the identifier appears.
    4. OK
9. **Verify documentation hints for keywords.**
    1. Invoke code completion for a whitespace.
    2. Select a keyword in the list (like extends, if).
    3. EXPECTED RESULT: In the details code completion view, a description of the keyword

appears.

4. OK

10.Verify keywords code completion.

1. Type the beginning of a keyword in a situation appropriate for its usage.
2. Invoke code completion.
3. EXPECTED RESULT: Code completion appears and the keyword is among suggestions.
4. OK

11.Verify code completion with a single option.

1. Type the beginning of a identifier so that there is only one option for code completion.
2. Invoke code completion.
3. EXPECTED RESULT: Identifier is instantly completed without any popup windows.
4. OK

12.Verify JavaDoc window.

1. Create a PHP function.
2. Add some JavaDoc to that function. Don't save the source.
3. In some other place, type the beginning of that function name and invoke code completion.
4. Select that function in the popup window.
5. EXPECTED RESULT: Just entered JavaDoc is shown in the JavaDoc window of code completion.
6. OK
7. JavaDoc obviously should be replaced with PHPDoc

13.Verify code completion after EXTENDS.

1. Define Foo class.
2. Type "class MyClass extends F".
3. Invoke code completion.
4. EXPECTED RESULT: Code completion appears and Foo class is shown. No any other identifiers are shown.
5. OK

14.Verify that require directive is automatically added.

1. Create a function in PHP file.
2. In another PHP file, invoke code completion so that that function is available.
3. Select that function and press Enter.
4. EXPECTED RESULT: code is completed with that function name, and require directive is added for the corresponding class.
5. OK

   6. It is not clear what is meant by “and require directive is added for the corresponding class.”

15.Verify code completion in // comments.

   1. Create // style comment .
   2. Invoke code completion at that line after //.
   3. EXPECTED RESULT: Code completion doesn't appear. It shouldn't be available in comments.
   4. OK

16.Verify code completion in /* */ comments.

   1. Create /* */ style comment .
   2. Invoke code completion inside of that comment.
   3. EXPECTED RESULT: Code completion doesn't appear. It shouldn't be available in comments.
   4. ACTUAL RESULT: It is possible to invoke code completion inside /* */ style comment. See http://ubuntuone.com/p/d5C/

17.Verify code completion in / */ comments.**

   1. Create /** */ comment .
   2. Invoke code completion inside of that comment.
   3. EXPECTED RESULT: Code completion shows a list of JavaDoc tags (like @abstract).
   4. OK
   5. Code copletion is not available on blank line but only after typing @

Test suite: Code Folding

Purpose: Check that PHP code and HTML is folded properly for PHP source and PHP web page.
Prerequisites: a PHP project is created in the Netbeans IDE.

1. **Create a PHP web page.**
   1. In the main menu, choose File > New File...
   2. In the Categories list, select Scripting. In the File Types list, choose PHP Web Page. Click Next.
   3. Accept default values by pressing Finish button.
   4. EXPECTED RESULT: A new PHP web page is created and opened in the editor.
   5. OK
2. **Verify code fold tooltip.**
   1. Collapse a code fold.
   2. Move mouse cursor onto {...} folding symbols.
   3. EXPECTED RESULT: Full content of folded code should appear as a tooltip.
   4. OK
3. **Expand code fold by double-click.**

   1. Collapse a code fold.
   2. Double-click {...} folding symbols.
   3. EXPECTED RESULT: Code fold is expanded.
   4. OK

4. **Recurrent code folding.**
   1. Collapse several code folds.
   2. Collapse a code fold containing all of them.
   3. Expand that code fold back.
   4. Expand all other code folds.
   5. Collapse that main code fold again.
   6. Expand that code fold back.
   7. EXPECTED RESULT: When expanding the main code fold back, other code fold should be in the same states as they were before collapsing the main code fold.
   8. OK
5. **Repeat all previous testcases for HTML folding.**
   1. EXPECTED RESULT: Same as in previous testcases.
   2. OK
6. **PHP functions code folding.**
   1. Add a PHP code with function definition to the page.
   2. Collapse a code fold with this function.
   3. Expand that code fold back.
   4. EXPECTED RESULT: The code fold should exist and work properly. The function title should be displayed outside of code fold.
   5. OK
7. **PHP /**/ comments code folding.**
   1. Add a PHP code with /**/ style comment to the page.
   2. Collapse a code fold with this comment.
   3. Expand that code fold back.
   4. EXPECTED RESULT: The code fold should exist and work properly.
   5. OK
8. **PHP // comments code folding.**
   1. Add a PHP code with several lines of // style comments to the page.
   2. EXPECTED RESULT: No code fold should exist for that comments.
   3. OK
9. **Copy/Paste code folds.**
   1. Add a PHP function with the body.

    2. Collapse its fold.
    3. Copy the fold (mark it and press Ctrl-C).
    4. Paste code in some other place.
    5. EXPECTED RESULT: Code is pasted properly and is unfolded.
    6. OK

10. **Verify code folding from the main menu.**
    1. In the IDE main menu, choose View -> Code Folds.
    2. Try each item of this submenu: Collapse Fold, Expand Fold, Collapse All, Expand All.
    3. EXPECTED RESULT: Code folds behave correspondingly.
    4. OK
11. **Verify code folding via keyboard shortcuts.**
    1. In the IDE main menu, choose View -> Code Folds.
    2. Behold keyboard shortcuts for each item of this submenu.
    3. Press shortcut for each item of this submenu: Collapse Fold, Expand Fold, Collapse All, Expand All.
    4. EXPECTED RESULT: Code folds behave correspondingly.
    5. ACTUAL RESULT: Shortcuts are not available Code Folds menu nor in Keyboard Shortcuts Card

Test suite: Mark Occurrences

Purpose: Check that occurrences of PHP identifier highlighted when of the identifiers is selected.
Prerequisites: a PHP project is created in the Netbeans IDE.

1. **Create a PHP web page.**
    1. In the main menu, choose File > New File...
    2. In the Categories list, select Scripting. In the File Types list, choose PHP Web Page. Click Next.
    3. Accept default values by pressing Finish button.
    4. Copy/paste a large PHP code peace into the editor.
    5. EXPECTED RESULT: A new PHP web page is created and opened in the editor. The code is added there.
    6. OK
2. **Verify local variable occurrences marking.**
    1. Create several PHP functions.
    2. Introduce a global variable.
    3. Introduce a variable with the same name in each of these functions.
    4. Add operators which use each of these variables.
    5. Place the cursor onto usage one of these variables.
    6. EXPECTED RESULT: Definition and usages of this variable are highlighted. Definition

    and usages of other variables aren't.
    7. OK

3. **Verify function occurrences marking.**
    1. Create a PHP function.
    2. Add several calls of that function, both inside and outside of its code.
    3. Place the cursor onto one of these calls.
    4. EXPECTED RESULT: Definition and calls of this function are highlighted.
    5. OK
4. **Verify parameter occurrences marking.**
    1. Create a PHP function with a parameter.
    2. Use that parameter in the function body.
    3. Place the cursor onto one of these usages.
    4. EXPECTED RESULT: Definition and usages of this parameter are highlighted.
    5. OK

Test suite: HTML Palette
Purpose: Check adding the HTML Palette components to PHP source.
Prerequisites: a PHP project is created in the Netbeans IDE.

1. **Create a PHP web page.**
    1. In the main menu, choose File > New File...
    2. In the Categories list, select Scripting. In the File Types list, choose PHP Web Page. Click Next.
    3. Accept default values by pressing Finish button.
    4. EXPECTED RESULT: A new PHP web page is created and opened in the editor. Palette window on the right is opened and displays HTML and HTML Forms groups.
    5. OK
    6. Palette will not open by default. Once opened HTML and HTML Forms groups are displayed
2. **DnD Table HTML component to HTML code.**
    1. DnD Table HTML component to the editor, not into PHP code.
    2. In the "Insert Table" dialog, press OK.
    3. EXPECTED RESULT: HTML tags added to DnD position. They correspond to visual representation of Table HTML component in the "Insert Table" dialog.
    4. OK
    5. Created table has error: The "border" attribute on the "table" element is obsolete. Use CSS instead.
3. **DnD Table HTML component to PHP code.**

   1. DnD Table HTML component to the editor, into PHP code.
   2. In the "Insert Table" dialog, press OK.
   3. EXPECTED RESULT: HTML tags added to DnD position. They correspond to visual representation of Table HTML component in the "Insert Table" dialog. Editor lines are marked with red, and tooltip reads "Syntax error".
   4. OK

4. **DnD Table HTML component outside of editor view.**
   1. DnD Table HTML component to the Projects window.
   2. DnD Table HTML component to the Navigator window.
   3. EXPECTED RESULT: In both cases, mouse cursor indicates that DnD is impossible.
   4. OK
5. **DnD big Table HTML component to HTML code.**
   1. DnD Table HTML component to the editor.
   2. In the "Insert Table" dialog, specify large values for rows and columns (like 1000), change values of other fields and press OK.
   3. EXPECTED RESULT: After reasonable time (few seconds), HTML tags added to DnD position. They correspond to the visual representation of Table HTML component in the "Insert Table" dialog (with made changes).
   4. OK
   5. The 2000 x 5 table having 14k lines was created in about two minutes. On the good side operation did not take all computer resources and other tasks could be performed in parallel.
6. **DnD Table HTML component with given parameters to HTML code.**
   1. DnD Table HTML component to the editor.
   2. In the "Insert Table" dialog, change values of each field and press OK.
   3. EXPECTED RESULT: HTML tags added to DnD position. They correspond to the visual representation of Table HTML component in the "Insert Table" dialog (with made changes).
   4. OK
   5. However, there where a bunch of errors:
      1. The "border" attribute on the "table" element is obsolete. Use CSS instead.
      2. The "width" attribute on the "table" element is obsolete. Use CSS instead.
      3. The "cellspacing" attribute on the "table" element is obsolete. Use CSS instead.
      4. The "cellpadding" attribute on the "table" element is obsolete. Use CSS instead.
7. **Add Table HTML component with given parameters to HTML code using double-click.**
   1. Place the cursor to some position in editor.
   2. Double-click Table HTML component.

    3. In the "Insert Table" dialog, change values of each field and press OK.
    4. EXPECTED RESULT: HTML tags added to DnD position. They correspond to the visual representation of Table HTML component in the "Insert Table" dialog (with made changes).
    5. OK

8. **Undo adding Table HTML component to HTML code.**
    1. DnD Table HTML component to the editor.
    2. In the "Insert Table" dialog, change values of each field and press OK.
    3. In the main menu, choose Edit -> Undo.
    4. EXPECTED RESULT: HTML tags added to DnD position, but reverted after Undo.
    5. OK
9. **DnD Form HTML component with given parameters to HTML code.**
    1. DnD Form HTML component to the editor.
    2. In "Insert Table" dialog, change values of each field and press OK.
    3. EXPECTED RESULT: HTML tags added to DnD position. They correspond to the visual representation of Form HTML component in the "Insert Form" dialog (with made changes).
10. **DnD other HTML components with given parameters to HTML code.**
    1. Same as previous, for other components in HTML Palette.
    2. EXPECTED RESULT: HTML tags added to DnD position. They correspond to the visual representation of an HTML component in the insert dialog (with made changes).
    3. OK
    4. In all examples usage of obsolete HTML elements is present.

Test suite: Formatting

Purpose: Check Format code feature for PHP and HTML.
Prerequisites: a PHP project is created in the Netbeans IDE.

1. **Create a PHP web page.**
    1. In the main menu, choose File -> New File...
    2. In the Categories list, select Scripting. In the File Types list, choose PHP Web Page. Click Next.
    3. Accept default values by pressing Finish button.
    4. EXPECTED RESULT: A new PHP web page is created and opened in the editor.
    5. OK
2. **Format default code of PHP web page.**
    1. Invoke Format from the popup menu in the editor.
    2. EXPECTED RESULT: The page source isn't changed, since it's already formatted

    properly.
    3. OK

3. **Undo Formatting of PHP web page.**
    1. Invoke Format from the popup menu in the editor.
    2. Choose Edit->Undo in the main menu.
    3. EXPECTED RESULT: All formatting changes are reverted.
    4. OK
4. **Create a PHP file.**
    1. In the main menu, choose File -> New File...
    2. In the Categories list, select Scripting. In the File Types list, choose PHP File. Click Next.
    3. Accept default values by pressing Finish button.
    4. EXPECTED RESULT: A new PHP file is created and opened in the editor.
5. **Format default code of PHP file.**
    1. Invoke Format from the popup menu in the editor.
    2. EXPECTED RESULT: The source isn't changed, since it's already formatted properly.
    3. OK
6. **Undo Formatting of PHP file.**
    1. Invoke Format from the popup menu in the editor.
    2. Choose Edit->Undo in the main menu.
    3. EXPECTED RESULT: All formatting changes are reverted.
    4. OK
7. **Formatting of folded code.**
    1. Add code with code foldings, collapse several folds.
    2. Invoke Format from the popup menu in the editor.
    3. Expand those folds back.
    4. EXPECTED RESULT: Folded code is formatted as well as unfolded.
    5. OK

Test suite: Navigation

Purpose: Check Navigation in PHP code.
Prerequisites: a PHP project is created in the Netbeans IDE.

1. **Create a PHP web page.**
    1. In the main menu, choose File -> New File...
    2. In the Categories list, select Scripting. In the File Types list, choose PHP Web Page. Click Next.

    3. Accept default values by pressing Finish button.
    4. EXPECTED RESULT: A new PHP web page is created and opened in the editor.
    5. OK

2. **Navigate to a specified line.**
    1. In the editor popup menu, choose Navigation -> Line...
    2. Type a valid line number and press GoTo.
    3. EXPECTED RESULT: The caret is moved to the specified line.
    4. OK
3. **Navigate to an invalid line.**
    1. In the editor popup menu, choose Navigation -> Line...
    2. Type a line number which is greater that number of lines in the file and press GoTo.
    3. EXPECTED RESULT: The caret is moved to the end of file.
    4. OK

Approvals

Reviewer Name & Title	Date	Status
Goran Miskvoic	February, 13th 2011	pass

Product Version: NetBeans IDE 7.0 Beta 2 (Build 201102111519)

Java: 1.6.0_22; Java HotSpot(TM) 64-Bit Server VM 17.1-b03

System: Linux version 2.6.35-25-generic running on amd64; UTF-8; en_US (nb)

Document can be downloaded in following formats:

Open Office Writer:: http://ubuntuone.com/p/d6C/

PDF:: http://ubuntuone.com/p/d6E/

MS Word:: http://ubuntuone.com/p/d6D/