

Using PHP Code Sniffer Tool

In addition to built-in coding assistance, IntelliJ IDEA provides code style check through integration with the [PHP Code Sniffer](#) tool, which validates your code for consistency with a coding standard of your choice. You can appoint one of the predefined [coding standards](#) or use [your own previously defined coding standard](#) with the root directory outside the default PHP Code Sniffer's Standards directory. Moreover, you can share your custom coding style with your team fellows.

To use PHP Code Sniffer right from IntelliJ IDEA instead of from a command line, you need to register it in IntelliJ IDEA and configure it as a IntelliJ IDEA [code inspection](#). Once installed and enabled in IntelliJ IDEA, the tool is available in any opened PHP file, so no steps are required from your side to launch it. The on-the-fly code check is activated upon every update in the file thus making it extremely easy to get rid of problems reported by PHP Code Sniffer.

Errors and warnings reported by PHP Code Sniffer on-the-fly are displayed as pop-up messages. When the tool is run in the batch mode, the errors and warnings are displayed in the **Inspection** tool window. Anyway, each message has the `phpcs` prefix to distinguish it from IntelliJ IDEA internal inspections.

On this page:


- [Prerequisites](#)
- [Registering PHP Code Sniffer in IntelliJ IDEA](#)
- [Specifying advanced PHP Code Sniffer options](#)
- [Configuring PHP Code Sniffer as a IntelliJ IDEA inspection](#)
- [Appointing a custom coding style to check your code against](#)
- [Sharing a custom coding style with the team](#)
- [Running Code Sniffer in the batch mode](#)
- [Excluding files from inspection](#)

Prerequisites

1. The [PEAR](#) package manager is installed on your machine.
2. [PHP Code Sniffer](#) is installed on your machine. To check it, switch to the directory with the `pear.bat` file and run the following command: `phpcs --version`

If the tool is available, you will get a message in the following format: `PHP_CodeSniffer version <version> (stable) by Squiz Pty Ltd. (http://www.squiz.net)`
3. To have code checked against your own custom coding standard, [create it](#). Store the rules and the `ruleset.xml` file that points to them in the coding standard root directory.

To register PHP Code Sniffer in IntelliJ IDEA

1. [Open the project settings](#), and click **Code Sniffer** under the **PHP** node.
2. In the **PHP Code Sniffer path** text box, specify the location of the Code Sniffer executable `phpcs.bat`. Type the path manually or click the **Browse** button  and select the path in the dialog box, that opens.

To check that the specified path to `phpcs.bat` ensures interaction between IntelliJ IDEA and Code Sniffer, that is, the tool can be launched from IntelliJ IDEA and IntelliJ IDEA will receive problem reports from it, click the **Validate** button. This validation is equal to running the `phpcs --version` command. If validation passes successfully, IntelliJ IDEA displays the information on the detected Code Sniffer version.

To specify advanced PHP Code Sniffer options

IntelliJ IDEA provides the ability to specify advanced PHP Code Sniffer options and thus fine tune the PHP Code Sniffer process behavior depending on the configuration of your computer and the rule sets used.

1. [Open the project settings](#), and click **Code Sniffer** under the **PHP** node.
2. In the **Maximum number of messages per file** text box, set the upper limit for the total number of messages to be reported for a file. All the messages above this limit will be rejected and IntelliJ IDEA will display the following warning right in the code:

Too many PHP Code Sniffer messages

3. In the **Tool process timeout** text box, specify how long you want IntelliJ IDEA to wait for a result from PHP Code Sniffer, whereupon the process is killed to prevent excessive CPU and memory usage.



To configure PHP Code Sniffer as a IntelliJ IDEA inspection

1. [Open the project settings](#), and click **Inspections**.
2. On the [Inspections](#) page that opens, select the **PHP Code Sniffer validation** check box under the **PHP** node.
3. On the right-hand pane of the page, configure the PHP Code Sniffer tool using the controls in the **Options** area:
 1. From the **Severity** drop-down list, choose the [severity degree](#) for the Code Sniffer inspection. The selected value determines how serious the detected discrepancies will be treated by IntelliJ IDEA and presented in the [inspection results](#).
 2. In the **Coding standard** drop-down list, appoint the coding style to check your code against. The list contains all the coding standards installed inside the main `PHP_CodeSniffer` directory structure.

Use one of the predefined [coding standards](#) or choose **Custom** to [appoint your own standard](#).
 3. Optionally, select the **Ignore warnings** check box to have only errors reported and suppress reporting warnings. This option is equal to the `-n` command line argument.

To appoint a custom coding style to use

You can have code checked against [your own previously defined coding standard](#) with the root directory outside the default PHP Code Sniffer's Standards directory. This root directory should contain the rules themselves and the `ruleset.xml` file that points to them.

1. [Open the project settings](#), click **Inspections**. The [Inspections](#) page opens. Select the **PHP Code Sniffer validation** check box under the **PHP** node.
2. From the **Coding standard** drop-down list, choose **Custom**.
3. Click the **Browse** button .
4. In the **Custom Coding Standard** dialog box that opens, specify the path to the root directory of your own coding standard in the **Root directory**. Type the path manually or click the **Browse** button  and choose the relevant folder in the [dialog that opens](#).

The selected root directory should contain the `ruleset.xml` file that points to the rules.

To share a custom coding style with the team



1. Put the root directory of your coding standard under the [project root](#).
2. [Configure Code Sniffer as a IntelliJ IDEA inspection](#).
3. [Appoint your coding standard](#).
4. At the top of the [Inspections](#) page, select the **Share Profile** check box.
5. On the [Version Control](#) page of the **Settings** dialog box, put the root directory of your coding standard [under version control](#).

To run Code Sniffer in the batch mode

1. To [run the inspection](#), choose **Code | Inspect code** on the main menu. Specify the inspection scope and profile.
2. View the inspection results in the [Inspection](#) tool window. Errors and warnings reported by PHP Code Sniffer have the `phpcs` prefix to distinguish them from IntelliJ IDEA internal inspections.

Excluding files from inspection

When waiting for Code Sniffer response exceeds the limit specified in the **Tool process timeout** field on [Code Sniffer](#) page, IntelliJ IDEA suggests adding the file to the *ignore list*. This list is shown on the [Code Sniffer](#) page in the **Ignored files** area. For each file, IntelliJ IDEA displays its name and location.

- To delete a file from the list and have Code Sniffer process it again, select the file and click the **Remove file** button .
- To remove all the files from the list, click the **Clean the list** button .

See Also

Procedures:

- [PHP-Specific Guidelines](#)

Reference:

- [Code Sniffer](#)
- [Inspections](#)

External Links:

- http://pear.php.net/package/PHP_CodeSniffer 
- http://blog.jetbrains.com/webide/2012/08/advanced_sniffer_options/ 

Web Resources:

- [Developer Community](#) 