

Using PHP Mess Detector

In addition to built-in coding assistance, IntelliJ IDEA provides checking the source code through integration with the [PHP Mess Detector](#) tool, which detects [potential problems](#) related to code size, inconsistency, unused code, violation of naming conventions, poor design, etc.

You can have [predefined rules](#) applied or define your own [custom set of rules](#).


On this page:

- [Preparing to use PHP Mess Detector](#)
- [Registering PHP Mess Detector in IntelliJ IDEA](#)
- [Specifying advanced PHP Mess Detector options](#)
- [Configuring PHP Mess Detector as a IntelliJ IDEA inspection](#)
- [Specifying the rules to apply](#)
- [Running Mess Detector in the batch mode](#)
- [Excluding files from inspection](#)

Preparing to use PHP Mess Detector

1. Download *PHP Mess Detector*. The easiest ways to do that is use the *Composer Dependency Manager*. See [Using Composer Dependency Manager](#).
2. Make sure the *PHP repository plugin* is [downloaded, installed, and enabled](#).

Registering PHP Mess Detector in IntelliJ IDEA

1. [Open the project settings](#), and click **Mess Detector** under the **PHP** node.
2. In the **PHP Mess Detector path** text box, specify the location of the Mess Detector start-up script `phpmd.bat` for WIndows and `phpmd` for Linux. Type the path manually or click the **Browse** button  and select the path in the dialog box, that opens. If you installed the tool through *Composer*, the default location is `<project root folder>\vendor\bin\phpmd.bat`.

To check that the specified path to `phpmd.bat` or `phpmd` ensures interaction between IntelliJ IDEA and Mess Detector, that is, the tool can be launched from IntelliJ IDEA and IntelliJ IDEA will receive problem reports from it, click the **Validate** button. If validation passes successfully, IntelliJ IDEA displays the information on the detected Mess Detector version.

Specifying advanced PHP Mess Detector options

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

1. [Open the project settings](#), and click **Mess Detector** 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 Mess Detector messages

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

Configuring PHP Mess Detector as a IntelliJ IDEA inspection

1. Open the [project settings](#), and click **Inspections**.
2. On the [Inspections](#) page that opens, select the **PHP Mess Detector validation** check box under the **PHP** node.
3. On the right-hand pane of the page, configure the PHP Mess Detector tool using the controls in the **Options** area:
 1. From the **Severity** drop-down list, choose the [severity degree](#) for the Mess Detector inspection. The selected value determines how serious the detected discrepancies will be treated by IntelliJ IDEA and presented in the [inspection results](#).
 2. [Appoint the rules to apply](#).

Specifying the rules to apply

You can have *predefined rules* applied or define your own *custom set of rules*.

1. Open the [project settings](#), and click **Inspections**.
2. On the [Inspections](#) page that opens, select the **PHP Mess Detector validation** check box under the **PHP** node.
3. Do one of the following:
 - To use a predefined rules, in the **Options** area, select the check boxes next to the validations to be performed. For more details on *predefined rules*, see <http://phpmd.org/rules/index.html>.
 - To use a custom ruleset:
 1. Create and save one or several *ruleset files*. A valid *ruleset file* is an `.xml` file that contains the root element `<ruleset>` with the attribute `name`. For more details on *custom rulesets*, see <http://phpmd.org/documentation/creating-a-ruleset.html>.
 2. In the **Custom rulesets** area, click the **Add Rule** button **+** and select the relevant rule definition file in the dialog box that opens. When you click **OK**, a new item is added to the **Custom rulesets** list, where the **File** field shows the location of the selected ruleset file and the **Name** field shows the ruleset name retrieved from the attribute `name` within the `<ruleset>` tag.

Running Mess Detector 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 Mess Detector have the `phpmd` prefix to distinguish them from IntelliJ IDEA internal inspections.

Excluding files from inspection

When waiting for Mess Detector response exceeds the limit specified in the **Tool process timeout** field on [Mess Detector](#) page, IntelliJ IDEA suggests adding the file to the *ignore list*. This list is shown on the [Mess Detector](#) 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 Mess Detector 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 **X**.

See Also

Procedures:

- [Using PHP Code Sniffer Tool](#)

Reference:

- [Mess Detector](#)

External Links:

- <http://blog.jetbrains.com/webide/2012/12/php-mess-detector-support/> 
- <http://phpmd.org/> 

Web Resources:

- [Developer Community](#) 