

Enabling a Command Line Tool

The PHP command line tools that are available from IntelliJ IDEA are listed in the [Command Line Tool Support](#) page of the [Settings](#) dialog box. Integration with command line tools is provided at the IntelliJ IDEA level, so once configured, a tool can be used in all your IntelliJ IDEA projects. Just activate or de-activate it when necessary depending on the needs of each specific project.

You can download and [activate an existing third-party tool](#) or [create your own tool](#). You can [update the command definitions](#) of third-party or a custom tool right in the IntelliJ IDEA editor.

In this section:

- [Enabling integration with a third party PHP command line tool](#)
- [Creating a custom PHP command line tool](#)

To enable integration with a third party PHP command line tool

Integration with command line tools is supported via the *Command Line Tool Support* plugin.

The plugin is not bundled with IntelliJ IDEA, but it is available from the [JetBrains plugin repository](#). Once enabled, the plugin is available at the IDE level, that is, you can use it in all your IntelliJ IDEA projects.

1. Download and install the desired tool. Currently IntelliJ IDEA supports integration with the following tools:
 1. [Symfony](#) version 1.1 or higher.
 2. *Zend Framework 1*, see <http://framework.zend.com/download/overview>.
 3. *Zend Framework 2 (ZFTool)*, see <http://framework.zend.com/downloads/latest>.
 4. [Composer Dependency Manager](#).
 5. [Drush](#), version 5.8 and higher.
 6. Symfony console-based tools [Laravel](#) and [Doctrine](#). These tools are installed through *Composer*.
2. [Open the IntelliJ IDEA settings](#) and click **Command Line Tool Support**.
3. In the [Command Line Tool Support](#) page, click the **Add** button. In the **Choose Tool to Add** dialog box that opens, choose **Zend Framework** or **Symfony**.
4. In the dialog box that opens, specify the path to the definition file of the chosen tool in the **Path to <tool>** text box. Depending on the chosen tool, specify the location of the following file:
 - <Zend-Framework-home>/zf.bat for *Zend Framework 1*.
 - <Zend-Framework-home>/zf.php OR <Zend-Framework-home>/zf.php for *Zend Framework 2*.
 - <Symfony-home>/data/bin/symfony for *Symfony 1.**.
 - <Symfony-home>/app/console for *Symfony 2*.
 - <Composer-home>/composer.phar for *Composer*
 - C:/ProgramData/Drush/drush.bat on Windows and /usr/bin/drush on Mac OS or Linux for *Drush*.
 - For *Laravel*, the path to the *artisan* bundled with the tool, for example, C:\MY_PROJECTS\PHP_PROJECTS\MyLaravelProject\artisan.
 - For *Doctrine*, the path to the tool home.IntelliJ IDEA searches the contents of the specified file for definitions of commands. When the file analyzes is completed, IntelliJ IDEA displays the specified file in the list.
5. In the **Alias** text box, specify the alias to use in calls of tool commands. Accept the default alias or edit it, if necessary.
6. To activate the detected command set, select the **Enable** check box.
7. [Customize the command set](#), if necessary.
8. In the **Show console in** area, specify where you want the **Input** pane for [typing commands](#) opened:
 - To have the **Input** pane opened in a pop-up window, choose the **Pop-up** option.
 - To have the **Input** pane opened as a text box at the bottom of the [Command Line Tools Console](#) tool window, choose the **Tool window** option.

To create a custom command line tool

1. Open the [IntelliJ IDEA settings](#) and click the **Command Line Tool Support**.
2. In the [Command Line Tool Support](#) page, click the **Add** button. In the **Choose Tool to Add** dialog box that opens, choose **Custom Tool**
3. In the [Tool Settings](#) dialog box that opens, specify the following:
 1. Tool name and location.
 2. The alias to use in command calls instead of the full path to the tool.
 3. Brief explanation of the tool functionality.
4. When you click **OK**, IntelliJ IDEA brings you to the [Command Line Tool Support](#) page, where the new tool is added to the list of available tool.
5. Select the newly created tool and click the **Open definition in editor** button. In the tool definition `.xml` file that opens in the editor, define the tool commands.
6. In the **Show console in** area, specify where you want the **Input** pane for [typing commands](#) opened:
 - To have the **Input** pane opened in a pop-up window, choose the **Pop-up** option.
 - To have the **Input** pane opened as a text box at the bottom of the [Command Line Tools Console](#) tool window, choose the **Tool window** option.

See Also

Procedures:

- [Using Command Line Tools](#)
- [Running Command Line Tool Commands](#)
- [PHP-Specific Guidelines](#)

Reference:

- [Command Line Tool Support](#)
- [Command Line Tools Console Tool Window](#)
- [PHP](#)

Web Resources:

- [Developer Community](#) 