Configuring XDebug

Configuring the XDebug tool involves:

- Downloading and installing the XDebug debugging engine
- Enabling XDebug integration with the PHP engine
- Checking the XDebug installation
- Integrating XDebug with IntelliJ IDEA

Downloading and installing the XDebug debugging engine

- Download the XDebug ⧉ extension compatible with your version of PHP and save it in the `php/` folder.

> The location of the `php/` folder is defined during the installation of the PHP engine.

> If you are using an AMP package, the XDebug extension may be already installed. Follow the instructions in the `xdebug.txt`.

Enabling XDebug integration with the PHP engine

1. Locate and open the active `php.ini` file.

   > To find out which `php.ini` file is active, create and run a test file with `phpinfo()`, then search for the Loaded Configuration File.

2. To disable the Zend optimizer, which blocks XDebug, remove the `[Zend]` section or mark it as comment in the `php.ini` file:

```
[Zend]
zend_extension_ts = "<path to ZendExtensionManager.dll>"
zend_extension_manager.optimizer_ts = "<path to Zend Optimizer>"
zend_optimizer.enable_loader = 0
zend_optimizer.optimization_level=15
zend_optimizer.license_path =
Local Variables:
tab-width: 4
End:
```

3. To enable XDebug, locate the `[XDebug]` section in the `php.ini` file and update it as follows:

   - For a *thread-safe* PHP engine, version 5.2:

```
[XDebug]
zend_extension_ts="<path to php_xdebug.dll>"
xdebug.remote_enable=true
xdebug.remote_port="<the port for XDebug to listen to>" (the default port is 9000)
xdebug.profiler_enable=1
xdebug.profiler_output_dir="<AMP home\tmp>"
```

   - For a *non-thread-safe* PHP engine, version 5.2:

```
[XDebug]
zend_extension_nts="<path to php_xdebug.dll>"
xdebug.remote_enable=true
xdebug.remote_port="<the port for XDebug to listen to>" (the default port is 9000)
xdebug.profiler_enable=1
xdebug.profiler_output_dir="<AMP home\tmp>"
```

   - For *any* PHP engine, version 5.3 and higher:

```
[XDebug]
zend_extension="<path to php_xdebug.dll>"
xdebug.remote_enable=1
xdebug.remote_port="<the port for XDebug to listen to>" (the default port is 9000)
xdebug.profiler_enable=1
xdebug.profiler_output_dir="<AMP home\tmp>"
```

4. To enable multiuser debugging via Xdebug proxies, locate the `xdebug.idekey` setting and assign it a value of your choice. This value will be used to register your IDE on Xdebug proxy servers.

5. Save and close the `php.ini` file.

Checking the XDebug installation

1. Open the XDebug checker⬈.

2. Enter the output of the `phpinfo()`.

Integrating XDebug with IntelliJ IDEA

1. Open the Project Settings and click **PHP**.

2. On the PHP page that opens, click the **Browse** button ⟨⋯⟩ next to the **Interpreter** drop-down list.

3. In the Interpreter dialog box that opens, choose **XDebug** from the **Debugger** drop-down list.

4. Define the XDebug behaviour. Click **Debug** under the **PHP** node. On the Debug page that opens, specify the following settings in the **XDebug** area:

   - In the **Debug Port** text box, appoint the port through which the tool will communicate with IntelliJ IDEA. This must be exactly the same port number as specified in the `php.ini` file:

     ```
     xdebug.remote_port = <port_number>
     ```

     By default, Xdebug listens on port 9000.

   - To have IntelliJ IDEA accept any incoming connections from XDebug engines through the port specified in the **Debug port** text box, select the **Can accept external connections** check box.

   - To have addresses of variables shown during a debugging session, select the **Show variables addresses** check box.

   - To have IntelliJ IDEA apply a workaround to overcome a known XDebug on FreeBSD crash issue 🖉, select the **Enable workaround for [XDebug on FreeBSD crash]** check box.

5. To have XDebug stop as soon as connection between it and IntelliJ IDEA is established (instead of running automatically until the first breakpoint is reached), turn on the **Run | Break at first line in PHP scripts** option on the main menu.

See Also

Procedures:

- Configuring PHP Development Environment
- Creating and Editing Run/Debug Configurations
- Multiuser Debugging via XDebug Proxies
- Configuring Zend Debugger

Reference:

- Debug
- PHP

External Links:

- http://confluence.jetbrains.com/display/PhpStorm/Xdebug+Installation+Guide 🖉
- http://www.xdebug.org/ 🖉

Web Resources:

- Developer Community 🖉