Xdebug: Documentation

XDEBUG EXTENSION FOR PHP | DOCUMENTATION
» Documentation for: **Xdebug 2**
» Feature: **Remote Debugging**

Xdebug provides an interface for debugger clients that interact with running PHP scripts. This section explains how to set-up PHP and Xdebug to allow this, and introduces a few clients.

Introduction

Xdebug's (remote) debugger allows you to examine data structure, interactively walk through your and debug your code. There are two different protocols to communicate with the debugger: the old GDB-like command protocol (GDB) which is implemented in Xdebug 1.3 and 2; and the DBGp [1] protocol which is implemented in Xdebug 2.

Clients

Xdebug 2 is bundled with a **simple command line client** for the DBGp [2] protocol. There are a few other client implementations (both free and commercial) as well. I am not the author of any of those, so please refer to the original authors for **support**:

- **Dev-PHP [3]** (IDE: Windows)
- **Eclipse plugin [4]**, which has been submitted as an enhancement for the PDT [5] (IDE).
- **Emacs plugin [6]** (Editor Plugin).
- ActiveState's **Komodo [7]** (IDE: Windows, Linux, Mac; Commercial).
- **MacGDBP [8]** - Standalone Mac client.
- **NetBeans [9]** (IDE: Windows, Linux, Mac OS X and Solaris.

- **Notepad++ [10]** plugin [11] (Editor: Windows).
- WaterProof's **PHPEdit [12]** (IDE, from version 2.10: Windows; Commercial).
- Anchor System's **Peggy [13]** (IDE: Windows, Japanese; Commercial).
- MP Software's **phpDesigner [14]** (IDE: Windows, Commercial).
- **PHPEclipse [15]** (Editor Plugin).
- JetBrain's**PhpStorm [16]** (IDE; Commercial).
- **Protoeditor [17]** (Editor: Linux).
- **tsWebeditor [18]** (Editor: Windows).
- Xored's **TrueStudio IDE [19]** (IDE; Commercial).
- **VIM plugin [20]** (Tutorial [21]) (Editor Plugin).
- jcx software's **VS.Php [22]** (MS Visual Studio Plugin; Commercial).
- **XDebugClient [23]** - Standalone Windows client.

A simple command line client for the GDB protocol is bundled with Xdebug 1.3. A client implementation of the deprecated GDB protocol can also be found in the free editor WeaverSlave [24].

Starting The Debugger

In order to enable Xdebug's debugger you need to make some configuration settings in php.ini. These settings are xdebug.remote_enable to enable the debugger, xdebug.remote_host and xdebug.remote_port to configure the IP address and port where the debugger should connect to. If you want the debugger to initiate a session when an error situation occurs (php error or exception) then you also need to change the xdebug.remote_mode setting. Allowed values for this setting are "req" (the default) which makes the debugger initiate a session as soon as a script is started, or "jit" when a session should only be initiated on an error.

After made all those settings Xdebug will still not start a debugging session automatically when a script is run. You need to activate Xdebug's debugger and you can do that in three ways:

1. When running the script from the command line you need to set an environment variable, like:
   ```
   export XDEBUG_CONFIG="idekey=session_name"
   php myscript.php
   ```
   You can also configure the xdebug.remote_host, xdebug.remote_port,

xdebug.remote_mode and xdebug.remote_handler in this same environment variable as long as you separate the values by a space:

```
export XDEBUG_CONFIG="idekey=session_name remote_host=localhost profiler_enable=1"
```

All settings that you can set through the XDEBUG_CONFIG setting can also be set with normal php.ini settings.

2. If you want to debug a script started through a web browser, simply add XDEBUG_SESSION_START=session_name as parameter to the URL. Refer to the next section to read on how debug sessions work from within a browser window.
3. Another way to activate the debugger while running PHP through a web server is by installing one of the following three browser extensions. Each of them allows you to simply enable the debugger by clicking on one button. The extensions are:
   easy Xdebug for FireFox

   This extension [25] for Firefox was built to make debugging with an IDE easier. You can find the extension at https://addons.mozilla.org/en-US/firefox/addon/58688/ [26].

   Xdebug Helper for Chrome

   This extension [27] for Chrome will help you to enable/disable debugging and profiling with a single click. You can find the extension at https://chrome.google.com/extensions/detail/eadndfjplgieldjbigjakmdgkmoaaaoc [28].

   Xdebug Toggler for Safari

   This extension [29] for Safari allows you to auto start Xdebug debugging from within Safari. You can get it from Github at https://github.com/benmatselby/xdebug-toggler [30].

   Xdebug launcher for Opera

   This extension [31] for Opera allows you to start an Xdebug session from Opera.

Before you start your script you will need to tell your client that it can receive debug connections, please refer to the documentation of the specific client on how to do this. To use the bundled client simply start it after compiling and installing it. You can start it by running "debugclient". If you want to use the GDB commandset to debug your scripts, make sure you use a debugclient as bundled with Xdebug 1.3 as the one bundled with Xdebug 2 only works with the DBGp commandset.

When the debugclient starts it will show the following information and then waits until a connection is initiated by the debug server:

```
Xdebug Simple DBGp client (0.10.0)
Copyright 2002-2007 by Derick Rethans.
```

```
- libedit support: enabled

Waiting for debug server to connect.
```

After a connection is made the output of the debug server is shown:

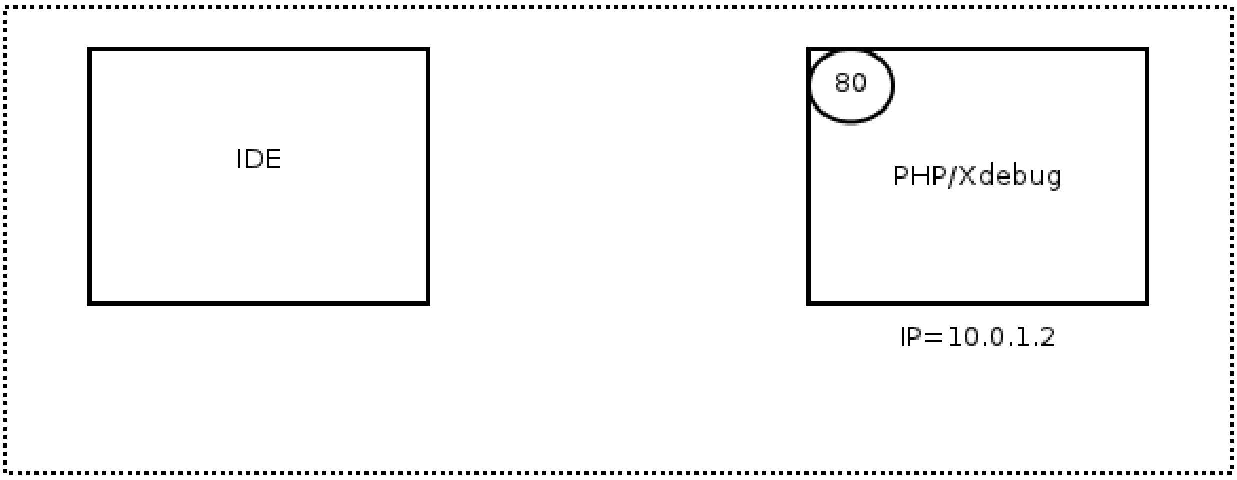
```
Connect
<?xml version="1.0" encoding="iso-8859-1"?>
<init xmlns="urn:debugger_protocol_v1"
      xmlns:xdebug="http://xdebug.org/dbgp/xdebug"
      fileuri="file:///home/httpd/www.xdebug.org/html/docs/index.php"
      language="PHP"
      protocol_version="1.0"
      appid="13202"
      idekey="derick">
  <engine version="2.0.0RC4-dev"><![CDATA[Xdebug]]></engine>
  <author><![CDATA[Derick Rethans]]></author>
  <url><![CDATA[http://xdebug.org]]></url>
  <copyright><![CDATA[Copyright (c) 2002-2007 by Derick Rethans]]></copyright>
</init>
(cmd)
```

Now you can use the commandset as explained on the DBGp [32] documentation page. When the script ends the debug server disconnects from the client and the debugclient resumes with waiting for a new connection.

Communication Set-up

With a static IP/single developer

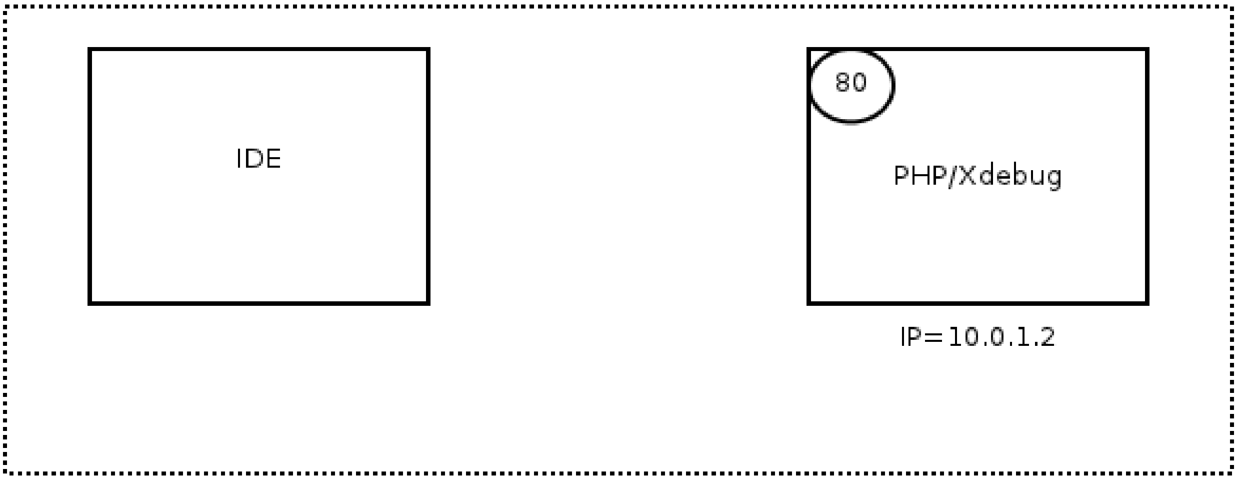
With remote debugging, Xdebug embedded in PHP acts like the client, and the IDE as the server. The following animation shows how the communication channel is set-up:



- The IP of the server is 10.0.1.2 with HTTP on port 80
- The IDE is on IP 10.0.1.42, so xdebug.remote_host is set to 10.0.1.42
- The IDE listens on port 9000, so xdebug.remote_port is set to 9000
- The HTTP request is started on the machine running the IDE
- Xdebug connects to 10.0.1.42:9000
- Debugging runs, HTTP Response provided

With an unknown IP/multiple developers

If xdebug.remote_connect_back is used, the set-up is slightly different:



- The IP of the server is 10.0.1.2 with HTTP on port 80
- The IDE is on an unknown IP, so xdebug.remote_connect_back is set to 1
- The IDE listens on port 9000, so xdebug.remote_port is set to 9000
- The HTTP request is made, Xdebug detects the IP addres from the HTTP headers
- Xdebug connects to the detected IP (10.0.1.42) on port 9000
- Debugging runs, HTTP Response provided

HTTP Debug Sessions

Xdebug contains functionality to keep track of a debug session when started through a browser: cookies. This works like this:

- When the URL variable XDEBUG_SESSION_START=name is appended to an URL Xdebug emits a cookie with the name "XDEBUG_SESSION" and as value the value of the XDEBUG_SESSION_START URL parameter. The expiry of the cookie is one hour. The DBGp protocol also passes this same value to the init packet when connecting to the debugclient in the "idekey" attribute.
- When there is a GET (or POST) variable XDEBUG_SESSION_START or the XDEBUG_SESSION cookie is set, Xdebug will try to connect to a debugclient.
- To stop a debug session (and to destroy the cookie) simply add the URL parameter XDEBUG_SESSION_STOP. Xdebug will then no longer try to make a connection to the debugclient.

Multiple Users Debugging

Xdebug only allows you to specify one IP address to connect to with xdebug.xdebug.remote_host) while doing remote debugging. It does not automatically connect back to the IP address of the machine the browser runs on, unless you use xdebug.xdebug.remote_connect_back.

If all of your developers work on different projects on the same (development) server, you can make the xdebug.xdebug.remote_host setting for each directory through Apache's .htaccess functionality by using php_value xdebug.remote_host=10.0.0.5. However, for the case where multiple developers work on the same code, the .htaccess trick does not work as the directory in which the code lives is the same.

There are two solutions to this. First of all, you can use a **DBGp proxy**. For an overview on how to use this proxy, please refer to the article at Debugging with multiple users [33]. You can download the proxy on ActiveState's web site [34] as part of the python remote debugging package. There is some more documentation in the Komodo FAQ [35].

Secondly you can use the xdebug.xdebug.remote_connect_back **setting** that was introduced in Xdebug 2.1.

Related Settings

xdebug.extended_info

Type: integer, Default value: 1
Controls whether Xdebug should enforce 'extended_info' mode for the PHP parser; this allows Xdebug to do file/line breakpoints with the remote debugger. When tracing or profiling scripts you generally want to turn off this option as PHP's generated oparrays will increase with about a third of the size slowing down your scripts. This setting can not be set in your scripts with ini_set(), but only in php.ini.

---

xdebug.idekey

Type: string, Default value: *complex*
Controls which IDE Key Xdebug should pass on to the DBGp debugger handler. The default is based on environment settings. First the environment setting DBGP_IDEKEY is consulted, then USER and as last USERNAME. The default is set to the first environment variable that is found. If none could be found the setting has as default ''. If this setting is set, it always overrides the environment variables.

---

xdebug.remote_autostart

Type: boolean, Default value: 0
Normally you need to use a specific HTTP GET/POST variable to start remote debugging (see Remote Debugging [36]). When this setting is set to 1, Xdebug will always attempt to start a remote debugging session and try to connect to a client, even if the GET/POST/COOKIE variable was not present.

---

xdebug.remote_connect_back

Type: boolean, Default value: 0, Introduced in Xdebug > 2.1
If enabled, the xdebug.remote_host setting is ignored and Xdebug will try to connect to the client that made the HTTP request. It checks the $_SERVER['REMOTE_ADDR'] variable to find out which IP address to use. Please note that there is **no** filter available, and anybody who can connect to the webserver will then be able to start a debugging session, even if their address does not match xdebug.remote_host.

---

xdebug.remote_cookie_expire_time

Type: integer, Default value: 3600, Introduced in Xdebug > 2.1
This setting can be used to increase (or decrease) the time that the remote debugging session stays alive via the session cookie.

---

xdebug.remote_enable

Type: boolean, Default value: 0
This switch controls whether Xdebug should try to contact a debug client which is listening on the host and port as set with the settings xdebug.remote_host and xdebug.remote_port. If a connection can not be established the script will just continue as if this setting was 0.

---

xdebug.remote_handler

Type: string, Default value: dbgp
Can be either 'php3' which selects the old PHP 3 style debugger [37] output, 'gdb' which enables the GDB like debugger interface or 'dbgp' - the debugger protocol [38]. The DBGp protocol is the only supported protocol.

Note: Xdebug 2.1 and later only support 'dbgp' as protocol.

---

xdebug.remote_host

Type: string, Default value: localhost
Selects the host where the debug client is running, you can either use a host name or an IP address. This setting is ignored if xdebug.remote_connect_back is enabled.

---

xdebug.remote_log

Type: string, Default value:
If set to a value, it is used as filename to a file to which all remote debugger communications are logged. The file is always opened in append-mode, and will therefore not be overwritten by default. There is no concurrency protection available. The format of the file looks something like:

```
Log opened at 2007-05-27 14:28:15
-> <init xmlns="urn:debugger_protocol_v1" xmlns:xdebug="http://xdebug.org/dbgp/x ... ight></init>

<- step_into -i 1
```

```
-> <response xmlns="urn:debugger_protocol_v1" xmlns:xdebug="http://xdebug.org/db ... ></response>
```

xdebug.remote_mode

Type: string, Default value: req
Selects when a debug connection is initiated. This setting can have two different values:

req
Xdebug will try to connect to the debug client as soon as the script starts.
jit
Xdebug will only try to connect to the debug client as soon as an error condition occurs.

xdebug.remote_port

Type: integer, Default value: 9000
The port to which Xdebug tries to connect on the remote host. Port 9000 is the default for both the client and the bundled debugclient. As many clients use this port number, it is best to leave this setting unchanged.

Related Functions

bool xdebug_break()

Emits a breakpoint to the debug client.

This function makes the debugger break on the specific line as if a normal file/line breakpoint was set on this line.

1. http://xdebug.org/docs-dbgp.php
2. http://xdebug.org/docs-dbgp.php
3. http://devphp.sf.net/
4. https://bugs.eclipse.org/bugs/show_bug.cgi?id=169408

5. http://www.eclipse.org/php/
6. http://code.google.com/p/geben-on-emacs/
7. http://activestate.com/products/komodo_ide/?src=AScom&type=bn&X=HP&campaign=KMD
8. http://www.bluestatic.org/software/macgdbp/index.php
9. http://php.netbeans.org/
10. http://notepad-plus.sourceforge.net/uk/site.htm
11. http://sourceforge.net/project/showfiles.php?group_id=189927&package_id=236520
12. http://www.waterproof.fr/products/PHPEdit/
13. http://www.anchorsystems.co.jp/anchor/ashp/peggy/pegindex.html
14. http://www.mpsoftware.dk/products.php
15. http://www.phpeclipse.com/
16. http://www.jetbrains.com/phpstorm/
17. http://protoeditor.sourceforge.net/
18. http://tswebeditor.net.tc/
19. http://www.xored.com/trustudio
20. http://www.vim.org/scripts/script.php?script_id=1929
21. http://tech.blog.box.net/2007/06/20/how-to-debug-php-with-vim-and-xdebug-on-linux/
22. http://www.jcxsoftware.com/vs.php
23. http://code.google.com/p/xdebugclient/
24. http://weaverslave.ws/
25. https://addons.mozilla.org/en-US/firefox/addon/58688
26. https://addons.mozilla.org/en-US/firefox/addon/58688/
27. https://chrome.google.com/extensions/detail/eadndfjplgieldjbigjakmdgkmoaaaoc
28. https://chrome.google.com/extensions/detail/eadndfjplgieldjbigjakmdgkmoaaaoc
29. http://benmatselby.posterous.com/xdebug-toggler-for-safari
30. https://github.com/benmatselby/xdebug-toggler
31. https://addons.opera.com/addons/extensions/details/xdebug-launcher/?display=en
32. http://xdebug.org/docs-dbgp.php
33. http://derickrethans.nl/debugging-with-multiple-users.html
34. http://code.activestate.com/komodo/remotedebugging/
35. http://community.activestate.com/faq/komodo-ide-debugger-proxy-pydbgpproxy

36. http://xdebug.org/docs/remote

37. http://www.php.net/manual/en/debugger.php

38. http://xdebug.org/docs-dbgp.php