### Windows下搭建Zend Studio开发环境

#### 一、下载及安装，Zend Studio 10正式版破解及汉化

下载Zend Studio 10正式版ZendStudio-10.0.0，下载破解文件com.zend.php.core\_10.0.0.201302111609 (该文件千万不要解压，也不要修改文件名)复制并替换你Zend Studio安装目录的plugins文件夹下的同名文件。

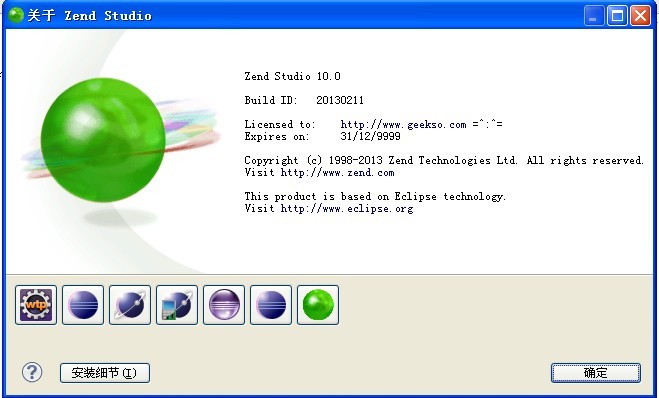
启动Zend Studio,复制如下注册码License key,粘贴(粘贴快捷键是ctrl+v)到注册框中：

34E606CF10C3E4CF202ABCEAA9B0B7A64DD2C5862A514B944AAAB38E3EB8A5F2CD735A2AB4CF9B952590EFA62BA0AB2B3E5D99C33C55309EE143165AC7F1817D626574615F3B32312F31312F323031313B392E303B3030313B313B3330

如果失效用下面这个吧：

15BD3D9635621E3AD4AE0F243B9E76CCFF6383B0E254EF646FD47B1424832332148D73E944AC95FC2A22EAF9965FE00D0BBACAD80E4B73BCFD40AADBEBD606EA4C33A74AEBB140CFB189E545E351774B7A020CDBE4FD159CDFF00F3C6A08C6AAC1A4F811EDDCF59CAFD09D46C849CB780BF6EF5983D3AFC372FC2021714E6B0B

查看是否注册成功，在菜单中“help>About Zend Studio ”。 过期时间9999年12月31日说明成功了，放一张图让没成功的人羡慕下。



#### 二、使用zend studio配置Xdebug调试PHP

接下来进行PHP代码调试，zend studio使用的是zend debugger来进行调试的。但这里要注意了：如果你的php版本是5.3以上，且是thread safe的，那么不要浪费你的时间做尝试，建议你直接改用XDebug吧。以下官方给出的解释。

The Debugger we provide supports only the non-thread-safe architecture, that is why it does not load in your configuration.

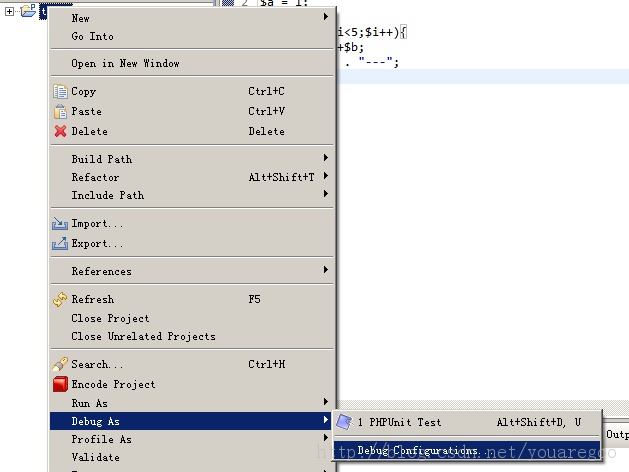
Zend stopped supporting the thread safe architecture when Microsoft also decided to implement its PHP engine with the fast-cgi architecture which means non-thread-safe.

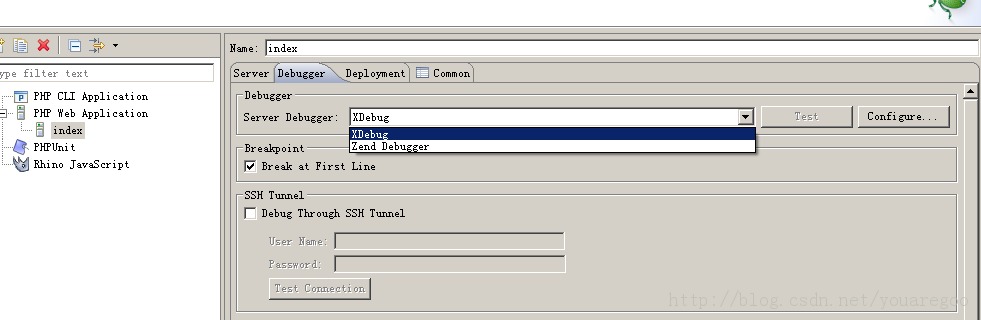
Best regards,

Massi.

XDebug在WAMP中是存在的，但需要修改Apache的php.ini文件，而不是Php目录下对应的文件。

然后，我们按文章中的设置好就可以了。但zend studio里面的设置要特别注意一下。如果你之前还用的是zend debugger进行调试的，调试单个项目又下断点不成功，你需要看一下就里设置正确没有，要设置成xdebug。





修改Apache的php配置文件php.ini，配置文件位于：wamp/bin/apache/bin/php.ini。

zend\_extension = "d:/wamp/bin/php/php5.3.13/zend\_ext/php\_xdebug-2.2.0-5.3-vc9.dll"

[xdebug]

xdebug.remote\_enable = 1

xdebug.remote\_handler="dbgp"

xdebug.remote\_host="127.0.0.1"

xdebug.remote\_port=9000

xdebug.auto\_trace=1

xdebug.collect\_params=1

xdebug.collect\_return=1

xdebug.trace\_output\_dir="d:/wamp/tmp"

xdebug.profiler\_enable = 1

xdebug.profiler\_append = 1

xdebug.profiler\_enable\_trigger = 1

xdebug.profiler\_output\_dir = "d:/wamp/tmp"

xdebug.profiler\_output\_name = cachegrind.out.%t.%p

Xdebug部分配置选项说明

xdebug.auto\_trace = 1

是否允许Xdebug跟踪函数调用，跟踪信息以文件形式存储，默认值为0

collect\_params = 1

是否允许Xdebug跟踪函数参数，默认值为0

xdebug.collect\_return = 1

是否允许Xdebug跟踪函数返回值，默认值为0

xdebug.profiler\_enable = 1

打开xdebug的性能分析器，以文件形式存储，这项配置是不能以ini\_set()函数配置的，默认值为0

xdebug.profiler\_output\_dir

性能分析文件的存放位置，默认值为/tmp

xdebug.profiler\_output\_name

性能分析文件的命名规则，默认值为cachegrind.out.%p

xdebug.trace\_output\_dir

函数调用跟踪信息输出文件目录，默认值为/tmp

xdebug.trace\_output\_name

函数调用跟踪信息输出文件命名规则，默认为trace.%c

特别说明：Xdebug的trace和profiler的输出文件名规则是可以更改的，比如将文件名命名为具体跟踪的PHP执行文件名、进程ID、随机数等，非常方便，更多的Xdebug配置选项说明，请参考官网的Xdebug配置选项说明。

至此PHP调试工具Xdebug教程之Xdebug的安装和配置就介绍完了，今后还将陆续介绍Xdebug如何在zend studio以及editplus中配置使用。