Local workstation xampp/Eclipse PDT/xdebug on WinXP from start to end

Last week, I got an xampp/eclipse-PDT/xdebug installation working on my Windows XP laptop. During the configuration, I found that I had to do a lot of jumping around to different sites to get the information that I needed to get it working. In the article below I will outline the steps that I took, with hopes that it will help someone else.

Note: The steps to install this on Linux are significantly different. As such, I have written the Ubuntu Linux 7.04 instructions here:

http://malibugarage.blogspot.com/2007/07/local-workstation-xamppeclipse.html

So.. Here we go!

Step 1: Download xampp

You can download xampp for Windows here:
http://www.apachefriends.org/en/xampp-windows.html
(I recommend the 7-zip self-extracting archive)

Step 2: Install xampp

Uncompress the file you downloaded. The configuration files within are all set up to run from C:\xampp. So if you can, copy the xampp folder that comes out of the compressed file to that location. If you can't, you should probably search each file for /xampp/ and \xampp\ and replace each instance with your preferred path. Do this before you get to the Eclipse step and it will save you a lot of hassle. In case you choose this path, I will refer to the xampp home directory as <xamppHome> from here on on.

Step 3: Start xampp and test

Double-click on 'start_xampp' in the <xamppHome> folder.

Swing a browser on your xampp machine to 'http://localhost' to test the installation. You should see a flashy splash screen.

You might also want to put a file called phpinfo.php in your htdocs directory. You'll want it to have the following contents:

<?php phpinfo() ?>

Hit the page 'http://localhost/phpinfo.php' and you've just executed a php script that tells you all about your php installation.

Step 4: Download Eclipse PDT

Eclipse PDT is a version of Eclipse that is bundled specifically for PHP developers. It's a good place to start when you're a PHP developer.

I recommend the latest integrated test version. You can download it at:
http://download.eclipse.org/tools/pdt/downloads/?release=I20070712

You want the 'all-in-one' package for your platform. It's Java, so at it's heart it is platform independent, but it uses widgets that are platform specific.

Step 5: Install Eclipse PDT

Uncompress the file you just downloaded. Somewhere inside there you will find a directory called 'eclipse'. Move that one into your xampp install directory.

Step 6: Download xdebug server install and configure in php.

Download at: http://www.xdebug.org

Windows users can download the dll.

Copy the .dll file into the directories <xamppHome>/php/ext and <xamppHome>/php/extensions.

In both <xamppHome>/apache/bin/php.ini **and** <xamppHome>/php/php.ini, make the following configuration changes:

- set 'implicit_flush = On'
- Comment out (type ; at beginning of the line) ALL the lines under [zend]
- Add the following:

```
[xdebug]
xdebug.remote_enable=1
xdebug.remote_host="localhost"
xdebug.remote_port=9000
xdebug.remote_handler="dbgp"
zend_extension_ts="<xamppHome>/php/ext/php_xdebug-2.0.0rc4-5.2.1.dll"
```

Note: The _ts above corresponds to THREAD SAFE. If you have trouble seeing xdebug in your phpinfo output, check the Thread Safe field in the top section. If it is Thread Safe: No, then take the _ts off and just use zend_extension=.
Also, check the 'Debug Build' field. If it is 'yes' you will need 'debug' in this field name. Here are the possible combinations you could end up with:

zend_extension (I ended up with this on Ubuntu+xampp)
zend_extension_ts (I ended up with this on Windows XP+xampp)
zend_extension_debug
zend_extension_debug_ts

Step 7: Restart xampp and test xdebug

Double click C:/xampp/restart_xampp

Now browse to http://localhost/phpinfo.php. You should see a section for xdebug. This means that the xdebug extension is running properly.

Step 8: Download Eclipse xdebug plugin

You want the 'Prebuilt Binary V0.2.3'. Here's the download link:
https://bugs.eclipse.org/bugs/attachment.cgi?id=74296

Extract the file and move the two jar files within into the <xamppHome>/eclipse/plugins directory.

Step 9: Point Apache into the Eclipse workspace

If this is to be purely an Eclipse/xdebug workstation, it's best to change the xampp Apache DocumentRoot so that it will look to the Eclipse workspace for any web page that you open. Edit <xamppHome>/apache/conf/httpd.conf and change the following:

From:
DocumentRoot /xampp/htdocs
To:
DocumentRoot <xamppHome>/eclipse/workspace

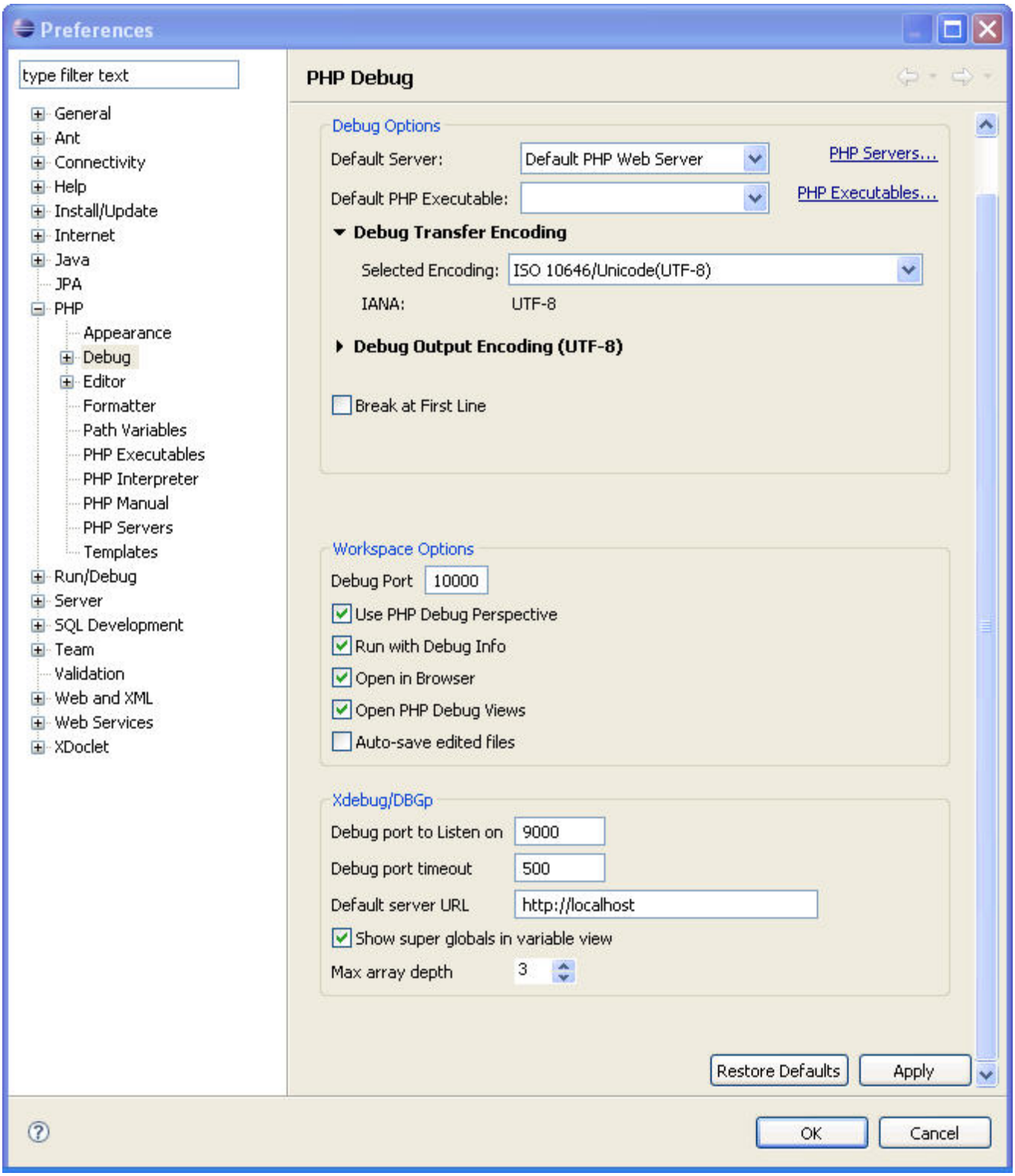
Also,
From:
<Directory "/xampp/htdocs">
To:
<Directory "<xamppHome>/eclipse/workspace">

Because you have changed httpd.conf, you should restart xampp now.

Step 10: xdebug configuration within Eclipse

Fire up Eclipse by executing the appropriate binary in <xamppHome>/eclipse. When it asks you about the workspace, change it to <xamppHome>/eclipse/workspace. You can check the "Don't ask me later" mark; it's really easy to change again later.

When Eclipse starts, go to Window->Preferences.. PHP->debug. The pane should look like this. (Notice the xdebug section). I didn't actually have to change anything here, as it's all in the php.ini.



Step 11: Create a PHP Project

You're finally ready to debug a web-based PHP script. Without further ado, within Eclipse select File-

>New->Project.. Now PHP->PHP Project.

For 'Project Name', enter 'debugtest' and then select 'Finish'. You will see the project pop up in the PHP Navigator. Highlight the project. Select File->New->PHP File and name it phpinfo.php (ok I'm not feeling very original here).

In the editing pane, make sure the file contents are:

<?php phpinfo() ?>

Note: Newlines get ignored.

Save the file.

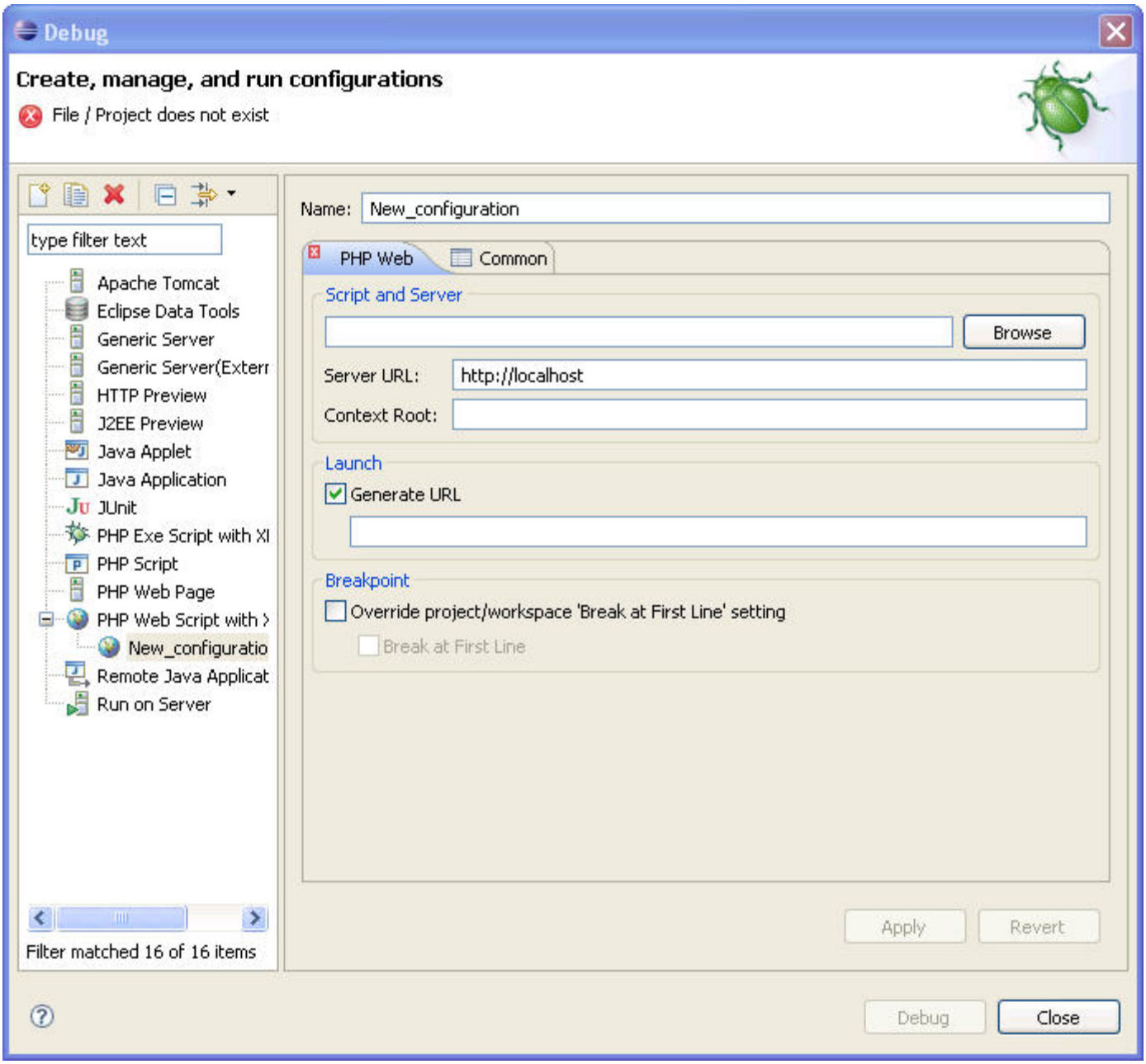
Navigate to http://localhost/debugtest/phpinfophp. You should see the same info you saw before.

Step 12: There ain't no more.

Now for one last, momentous step. You must now go into Eclipse and tell the xdebug plugin how to start a debug session so that you can catch a breakpoint in your new script. Select Run->Open debug dialog.. and double-click on 'PHP Web Page with Xdebug'.

NOTE: There must be menu items 'PHP Exe script with XDebug' and 'PHP Web Script with XDebug'. If these aren't there then something has gone wrong with the plugin installation.

You will come to a dialog screen that looks like this:



Under Script and Server, browse to the file phpinfo.php in project debugtest.

Make sure the url field has http://localhost/debugtest/phpinfo.php. If it doesn't, uncheck the box and edit it manually. This is important because it is the site that Eclipse will go to when you start your debug session. I had to do this on my laptop just now when I tested it.

Select 'Apply' and 'Debug'.

You should have seen your phpinfo output come up in the internal Eclipse browser. If it came up in an external browser don't sweat it, because that will work to. With some playing around with the configuration you can get it the way you want.

Now select the tab for phpinfo.php and right-click on the vertical bar between your phpinfo.php line numbers and the edge of the edit pane and select 'Toggle Breakpoints'. You will see a blue circle appear, indicating that there is a debug breakpoint at that location.

Click on the down-arrow next to the bug in the top toolbar and select the debug test session that you just created.

You should now have caught your first breakpoint.