Getting Started with Eclipse PHP Development Tools (PDT)

Are you ready to take a step beyond writing code in a text editor like UltraEdit, BBEdit, or TextMate? Would you like to see those PHP and JavaScript syntax errors in the editor, without transferring files to the server or opening a browser? If so, then you're ready to jump into the world of the IDE.

Are you ready to take a step beyond writing code in a text editor like UltraEdit, BBEdit, or TextMate? Would you like to see those PHP and JavaScript syntax errors in the editor, without transferring files to the server or opening a browser? If so, then you're ready to jump into the world of the IDE — Integrated Development Environment. I'll compare the free, open source Eclipse IDE to a few of its commercial competitors Then we'll go through the steps to install Eclipse PDT All-In-One, the Zend Debugger, JSEclipse, and Subclipse.

Why an IDE? Let there be no mistake, I still love BBEdit and TextMate. I use both daily at work and at home for quick edits and I actually wrote this post in TextMate. But if you spend a good portion of your day writing PHP, an IDE will save you time in small increments by highlighting those unbalanced braces and missing semicolons, displaying PHP function arguments, and allowing you to debug your code right in the editor.

There are commercial IDEs available and I've use a few of them. I tried an early version of Eclipse with the xored studio but the combo wasn't feature-rich and didn't perform as well as Zend Studio. I later switched from Zend to Active State's Komodo. I've been happy Komodo but it's sluggish performance on my PPC Mac drove me back to giving Eclipse another try.
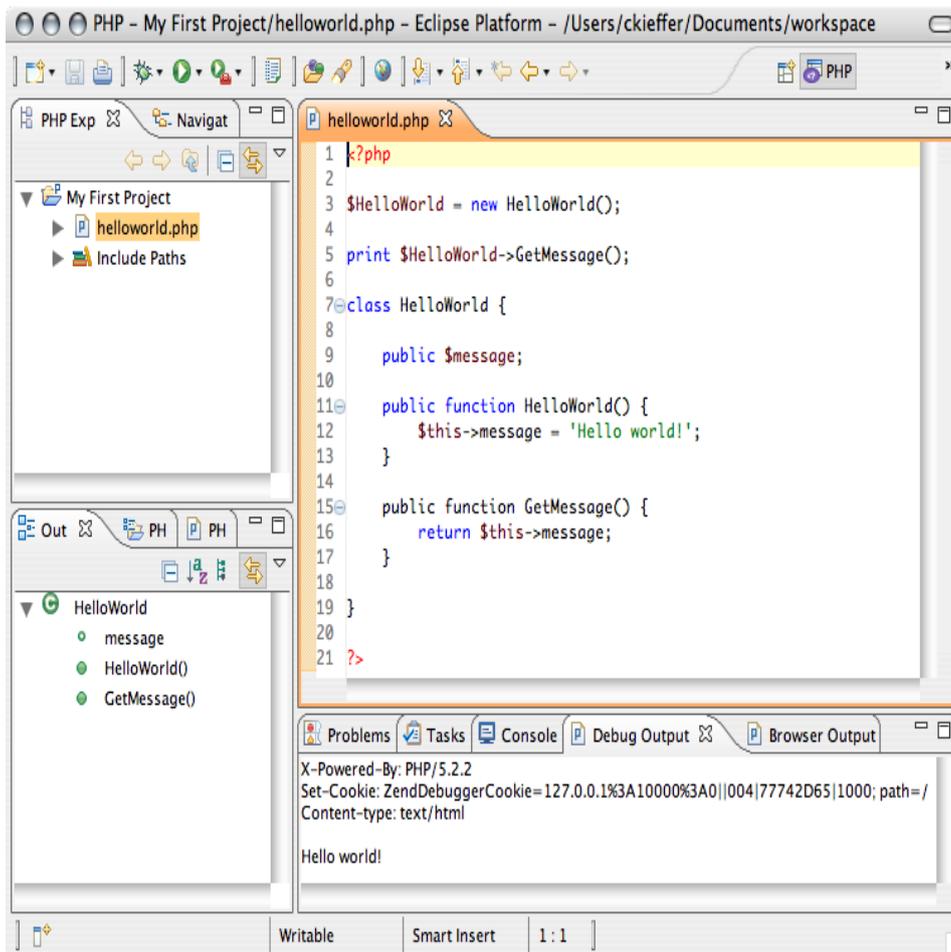
I started using Eclipse with PDT at work about a year and am amazed at how much PDT, formerly PHP IDE, has improved over the past four years. Eclipse is first a Java development environment, but extensions exist for a bunch of other languages, including Perl, Ruby, and Tcl. Okay, enough with the commentary, let's get to it, shall we?

Download and install Eclipse PDT All-In-One

Installation couldn't be easier. Visit the Eclipse PDT download page, select a stable build link, scroll down to the PDT All-in-One section, and select the download for your platform.

http://download.eclipse.org/tools/pdt/downloads/

Once the download is complete, unpack and move the 'eclipse' folder to your Applications or Program Files folder. Fire up Eclipse, create a new PHP file or project, and code away.

```php
<?php

$HelloWorld = new HelloWorld();

print $HelloWorld->GetMessage();

class HelloWorld {

    public $message;

    public function HelloWorld() {
        $this->message = 'Hello world!';
    }

    public function GetMessage() {
        return $this->message;
    }

}
?>
```

X-Powered-By: PHP/5.2.2
Set-Cookie: ZendDebuggerCookie=127.0.0.1%3A10000%3A0||004|77742D65|1000; path=/
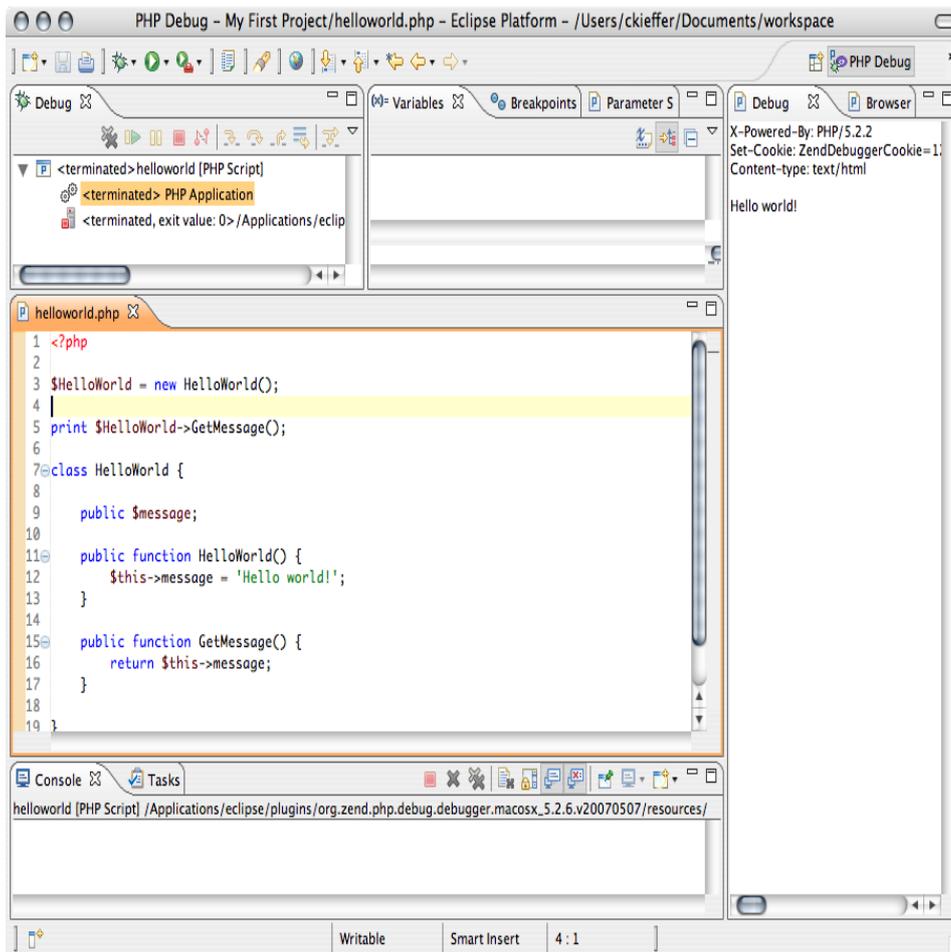Content-type: text/html

Hello world!

Install the Zend Debugger

Out of the box, Eclipse provides PHP syntax highlighting, code completion, PHP documentation, phpDoc support, and more. Local and server debugging, however, require the installation of XDebug or Zend Debugger extension. Here's how to enable local debugging with the Zend Debugger.

1. Select Help->Software Updates->Find and Install
2. Select Search for new features to install, click Next
3. Click the New Remote Site button...
   1. Name: Zend Debugger
   2. URL: http://downloads.zend.com/pdt
4. Zend Debugger now appears checked in the Sites to include in search list, click Finish
5. The update manager searches for the files to download
6. All Zend Debugger options should be checked on the Search Results screen, click Next
7. Agree to the licensing terms, click Next
8. Select the Zend Debugger in the Features to Install screen, you can change the install location, but the default is recommended, click Finish
9. Verify the Zend Debugger on the Feature Verification screen, click Install All
10. After installation you'll be asked to restart Eclipse for the changes to take affect

You should now have a PHP Debug Perspective, complete with variable tracing, breakpoints, and CLI and browser debug output views.

Install JSEclipse

To add improved JavaScript editing abilities to Eclipse, install JSEclipse from Adobe Labs. JSEclipse also provides editing support for popular JavaScript libraries, including YUI, Dojo, Prototype, and more. To install JSEclipse with the Update Manager

1. Select Help->Software Updates->Find and Install
2. Select Search for new features to install, click Next
3. Click the New Remote Site button...
   - Name: JSEclipse
   - URL: http://download.macromedia.com/pub/labs/jseclipse/autoinstall
4. JSEclipse now appears checked in the Sites to include in search list, click Finish
5. The update manager searches for the files to download
6. Check the JSEclipse version check box on the Search Results screen, click Next
7. Agree to the licensing terms, click Next
8. Select JSEclipse in the Features to Install screen, you can change the install location, but the default is recommended, click Finish
9. Verify JSEclipse on the Feature Verification screen, click Install All
10. After installation you'll be asked to restart Eclipse for the changes to take affect

To create a new JS file, select New->File->Other->Web->JavaScript. You should now have code completion and error highlighting for JavaScript files.
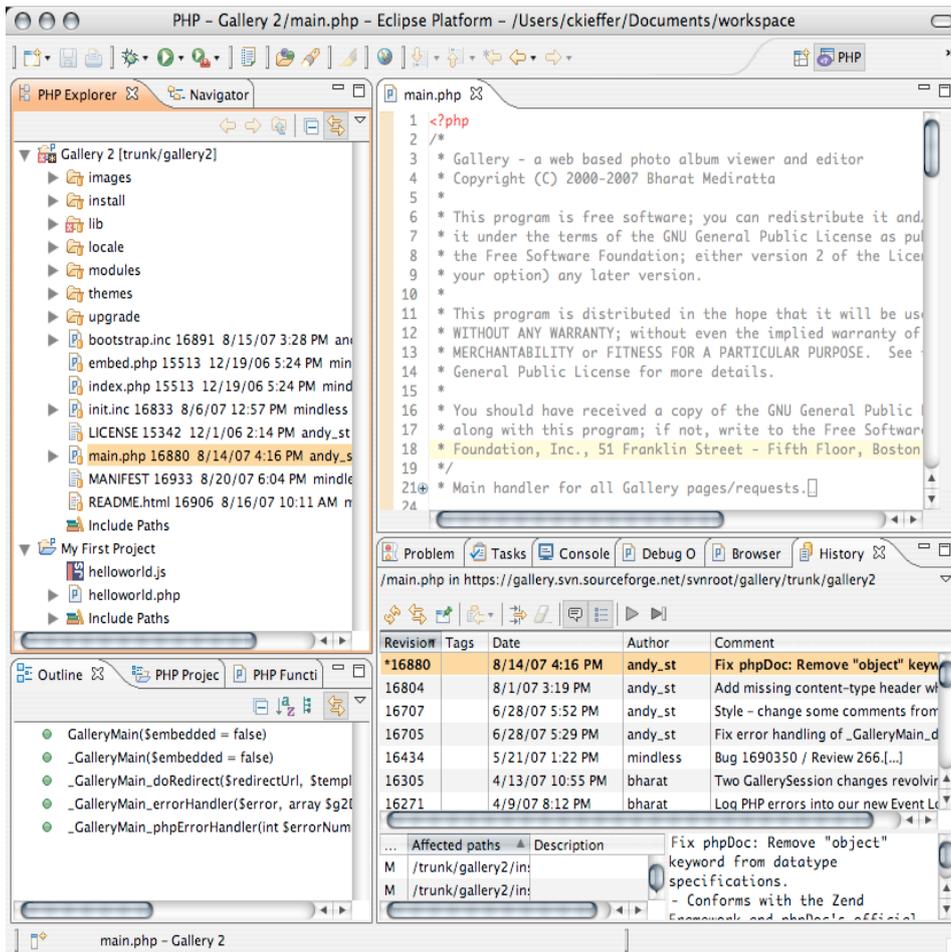
Install Subclipse

To add support for Subversion revision management you'll need to install Subclipse from tigris.org, the developers of Subversion. To install, let's use the Update Manager again.

1. Select Help->Software Updates->Find and Install
2. Select Search for new features to install, click Next
3. Click the New Remote Site button...
    - Name: Subclipse 1.2.x
    - URL: http://subclipse.tigris.org/update_1.2.x
4. Subclipse now appears checked in the Sites to include in search list, click Finish
5. The update manager searches for the files to download
6. Check the Subclipse version check box on the Search Results screen. You may need to install dependent extensions for the Subclipse integrations (I didn't select integrations), click Next
7. Agree to the licensing terms, click Next
8. Select Subclipse in the Features to Install screen, you can change the install location, but the default is recommended, click Finish
9. Verify Subclipse on the Feature Verification screen, click Install All
10. After installation you'll be asked to restart Eclipse for the changes to take affect

To checkout an existing Subversion repository

1. Select New->File->Other->SVN->Checkout Projects from SVN, click Next
2. Create a New Repository, click Next
3. Enter the URL for the SVN repository you want to checkout, click Next
4. If the repository URL is secure, accept the SSL certificate
5. Select the folder(s) to checkout
6. Create a new Project, select the Project type, PHP Project in this case, click Next
7. Enter a Project name, workspace, and any other project specific settings, click Finish
8. You'll see an overwrite warning that any SVN repository files with the same names as Eclipse settings files, normally you shouldn't need to worry about this, click OK
9. The project is checked out of the repository

I've just started to use some of the data integration and design extensions available for Eclipse and will post a followup soon. In the meantime, happy coding!

Related Links

- [The Zend Debugger Install Instructions I Used](#)
- [JSEclipse Documentation](#)
- [Subclipse documentation](#)

Use phpDoc for quick code navigation
August 23rd, 2007 at 5:48 am — Andy

- Code navigation / auto-completion:

Don't forget to add phpDoc to all your methods, else you'll miss one of the more productive features of PDT: code hyperlinks / navigation.

PDT depends on the author annotating the @param and @return data-types because of PHP's loose typing.

One thing I noticed though is that this doesn't seem to work if the return type is an array.

And one thing that is still better in PHPEclipse is that its code-autocompletion / navigation lists all candidates. PDT doesn't offer any option if it doesn't know what type a specific variable is. It would be very nice if would degrade from its accurate data-type specific autocompletion/navigation to all possible candidates.

- Code formatting:

And the one big missing feature in PDT is probably better/more code-formatting options. Java-coding has become much more productive in Eclipse 3 with the configurable code style formatter. Too bad that it isn't available for PHP or other languages yet.

It's quite painful to indent every 2nd line manually.

Source: http://2tbsp.com/node/40