



Monday, July 2. 2007

Released into the wild: Zend Framework 1.0.0

Today saw the much anticipated release of the [first production release of the Zend Framework](#). I've been following the project keenly since last Summer, so congratulations to the developers, contributors and the ever omniscient Zenders for their hard work to date! 


The Zend Framework has evolved into a loosely coupled, extremely flexible and easy to modify framework. Built across months of debate, hard work and the occasional emotional tirade (as I've been guilty of recently )

the 1.0.0 release now carries a stable API and an impressive array of features. Personally, I'm impressed on the insistence of obtaining a stable API by a fixed date at the cost of some completion concerns. It really is essential these days that adopters of a new technology are given the stability they need to get things done effectively now - not months down the line. I know from migrating several applications across 0.6, 0.7 to 0.9 that the lack of stability is a costly issue, much more costly than a few missing or incomplete features one can workaround for the moment.

I know most people will shout MVC a dozen times before collapsing into a state of ecstatic delirium, but there is a lot more to the Zend Framework than MVC. The MVC components are absolutely essential, but are not the sum total of the Framework.

Over the months I've come to appreciate the fixation on web services. Easy integration with web services is hugely important these days - from the simplicity of agregating RSS/Atom feeds to the complexity of building your own RESTful service. The Zend Framework is all geared up on web service steroids with support for RSS/Atom feeds, Google, Strikelron (which is quite cool), Yahoo, Flickr, and the usual suspects. Not to mention the more general support offered by Zend_Rest and Zend_XmlRpc.

My other favourites are the i18n support offered by Zend_Locale, Zend_Date and Zend_Translate. For those who will use Zend_Validate classes, many of the character string validators attempt to use a Unicode mode so you're not tied to the usual ASCII range. These components make it very easy to internationalise your application, support multibyte languages, and minimise the mountain of work a typical PHP application needs to go international correctly. Zend_Uri even has support for validating a number of Unicode IRIs (internationalised resource identifiers). Hopefully support for this improves over time given how many domain names are availing of IRI support - especially now that many of the popular browsers (incl. IE7) have built in the IDNA standard.

Finally, the Zend Framework website has seen a few changes. You can read the current future roadmap over at <http://framework.zend.com/whyzf/future/>. It's interesting to see both OpenID (and to a lesser extent Vista's CardSpace) mentioned along with support for the YAML format. I've worked with both extensively and have related proposals (Dmitry Stogov is leading the [OpenID proposal](#), me the [related Yadis Protocol one](#), as well as a [Zend Yaml proposal](#)) on the ZF Developer's Wiki. OpenID is a very interesting authentication service so spare the proposals a read if you have a chance 

On a last unrelated note:

if you've read or intend [reading my Acceptance Testing with PHPUnit/Selenium article on Devzone](#) you should be aware that Sebastian Bergmann just announced the release of PHPUnit 3.1.0 which replaces the requirement for PEAR's Testing_Selenium with an internal implementation. I'll be testing later on to see if an

article addendum is warranted.

Posted by Pádraic Brady in Openid and Yadis, PHP General, PHP Security, Zend Framework at 17:27

Does it have OpenID/Yadis? 

Anonymous on Jul 3 2007, 11:28

Not yet, Dmitry 

. Dmitry #2 (Stogov) at Zend has published a proposal for OpenID, and I have functional OpenID 2.0 code (as well as a Yadis 1.0 proposal) which I was putting together long for the ZF long before Zend jumped on the bandwagon two weeks ago so we'll both likely be doing a lot of talking, debating and work to get OpenID online in the Zend Framework over the next few months.


I'm also porting my library to PEAR independent of the ZF effort (the Yadis proposal should be up next day or two there). Anonymous on Jul 3 2007, 22:17

I've just made initial implementation of writing yaml-files using syck! <http://blog.milkfarmsoft.com/?p=57>

My next aim is object-serialization " I guess we should come to some common solution with you, here, to make our libraries interoperable


Take a look at my TODO:

<http://code.whytheluckystiff.net/svn/syck/trunk/ext/php/TODO>

I copied object-notation from the one in ruby " if you have better idea " let me know 

Anonymous on Jul 6 2007, 07:18


Sorry about the OT, but I just got a comment notification in Russian... Anonymous on Jul 6 2007, 10:34

Looks good 


. Not sure if the YAML proposal mentions it but if my Zend_Yaml proposal is adopted it will delegate to your syck binding when available.


Are there any plans to port this to PECL? I think it would be such a cool addition and raise its visibility a bit more.

I'm not too worried about object serialisation forms. If you implement a solution I'm happy enough to ensure Zend_Yaml sticks to it. Anonymous on Jul 6 2007, 11:24

I am actually waiting for PECL-account approving now 

Anonymous on Jul 6 2007, 11:28

Definitely blogging that arrival 

. Having YAML support inside PECL will make it easier to point to and I can tell people to go to PECL if they want a really fast Zend_Yaml 

. Anonymous on Jul 9 2007, 14:44