

Q&A Session for Zend Framework 1.0

General questions about Zend Framework

Q: Can you talk about the difference between a component library and an application framework, and where Zend "Framework" fits? I think of an application framework as a mostly-framed house, with a clear outline of where most things will go, and ready for finishing out and filling up. But frankly, I feel like the ZF is a pile of fresh, high quality lumber.

A: I think of a component library being more loosely coupled, with a use-at-will design. A framework is a reusable, abstract design for a software system, and provides both code solutions and support programs to help you employ the technology in a given project. By these definitions, ZF is more like a component library, but it's really somewhere in the middle, because ZF is designed to provide abstract classes that you can extend and reuse.

Q: Was Zend Framework built based on regular design patterns e.g. GRASP?

A: Yes, we have tried to be mindful of best practices of object-oriented design, and we use design patterns throughout the ZF code.

Q: How many developers worked and/or are currently working on the ZF?

A: A significant part of the Zend Framework project has been implemented as contributions from individuals in the community. Today we have over 230 developers who have submitted a Contributor License Agreement (CLA). Zend provides a small dedicated core of employees who work on ZF.

Q: Beyond the RC3, is the official 1.0 next?

A: Yes, we hope so. ZF 1.0.0 RC3 is scheduled to be released on Monday (June 25 2007) and we will continue making release candidates until we meet our quality goals.

Q: When will a stable release be available?

A: We hope to make the general release of Zend Framework in a few weeks, in July.

Q: To what extent do you feel that some of this functionality duplicates what was already created through PEAR modules and what is the Framework's relationship with PEAR?

A: Yes, in some cases a PEAR module exists that does something similar, so there is some duplication from a technology viewpoint. But none of the code in PEAR was contributed under the license terms that we use in the Zend Framework. Also it is hard to know which modules in PEAR are actively being developed and which have become "abandonware".

One of the goals of ZF is to provide a centralized library of technology that is maintained and supported, and all of it has some assurance that it was developed and contributed without violation of any copyrights or patents. So ZF offers some non-technical advantages even where it provides similar technical capabilities.

Q: Are there any *.zend.com websites build on ZF? Is framework.zend.com build on ZF?

A: Yes, <http://framework.zend.com> is itself a ZF application, so we do "eat our own caviar." In addition, <http://devzone.zend.com> uses ZF quite a lot too. Zend also has work in progress to employ ZF on the main website, <http://www.zend.com>.

Q: Where is Zend Framework better, where weaker?

A: We are very happy with many parts of ZF, and we think they have matured very nicely thanks to testing and feedback from our developer community. One area that I think we all agree needs to be developed is an advanced solution for processing web form input.

Q: Are there plans to offer a mini version of ZF that contains just the essentials for those that don't need things like the web-services for a particular application?

A: No, sorry, we have made the decision to offer the ZF product as a single download. However, since Zend Framework is built to support a “Use at will” philosophy you should be able to only use those parts that you need.

Q: In what ways can I expect the Framework to be supported? I mean... who ensures me that the code is still supported after 4 / 5 years?

A: One of the benefits of open source is that you don't need to worry as much about whether the code is supported, as you would for a closed-source product because you have access to the code.. The ZF team is discussing the best way that we can set up a partner program for companies wanting to offer commercial support, training and consulting services.

Q: Who are the corporate users that were mentioned and are they truly significant from the corporate standpoint?

A: The one we have featured on our website for quite a while now is IBM, who implements their QEDwiki project using Zend Framework. We are developing other case studies to feature on our website. We just opened a survey a few days ago, to ask people how they were using ZF. Dozens of companies responded, so it will take us a little while to follow up with them and create case studies.

Q: Hi, has there been a code level security audit done on the framework, and if so will the results be available?

A: We have not performed a code audit on ZF.

Questions about comparing ZF with other systems

Q: How would you compare the Zend Framework with CakePHP?

A: There are different frameworks with different goals so it is really up to the users to decide which framework is best for their situation. Each have strengths and weaknesses. One example is that Cake supports PHP4 environments, which of course Zend Framework doesn't do.

Q: I am currently evaluating frameworks and would like to get a grip on Zend's framework as well, but don't know how it is from a maturity perspective.

A: We are ready to make our first general release, and we know that it is already used in dozens of web applications in production. So we are comfortable calling it “stable” at this point. As for how good it is, the real question is whether it satisfies the needs of your project and results in productivity improvements for you. I cannot answer that, because it depends on the nature of your project.

Q: How would you advise a developer about usage of Zend Framework vs. something like PEAR in conjunction with Cake or Symfony, etc? I'd also like to be confident that I'm not missing anything using Framework vs. any of these. I'm very concerned about quality and support.

A: In each project, one needs to evaluate the available solutions both for its technical advantages, and its business advantages. We've already talked a lot about the technical strengths of Zend Framework, so here I'll just mention that ZF is the only major framework for PHP 5 that uses the

very business-friendly New BSD license, and has a Contributor License Agreement policy. Many of the current users of ZF tell us this was a significant factors in choosing Zend Framework.

Q: Can you comment on how the ZF compares to CodeIgniter? At first glance they look quite similar. What advantages does the ZF have over it?

A: I'm not a user of CodeIgniter, so I'm not the best person to answer that.

Q: Is Zend Framework more loosely-coupled than ezComponents? When talking to a PHP developer I was told that the opposite was true (this was about 2 months ago)

A: I'm not a user of ezComponents so I'm not the best person to answer that.

Q: Are there plans to build a true framework on top of the ZF components, something at the level of Ruby on Rails?

A: Yes, the ZF community is starting to think in terms of designing helper scripts to automate some application development tasks to give similar benefits as RoR. So some tools to solve similar needs to those in Ruby on Rails are possible, though we don't have a specific proposal for such tools at this time.

Questions about supported environment

Q: Is Framework dependent on any particular PHP5 version?

A: Yes, ZF requires PHP 5.1.4 minimally. This is because ZF uses some object-oriented interfaces that weren't finished in earlier PHP releases. ZF uses the PHPUnit testing tool extensively, which requires PHP 5.1.4. It would probably be to your great benefit to upgrade to at least PHP 5.2.1 or later, for the additional performance and security improvements available in the recent versions.

Q: Will ZF Be offered for older versions of PHP ?

A: ZF depends on some features of PHP that were not finished prior to PHP 5.1.4. I'm sorry, but we cannot support older versions of PHP and at the same time gain the benefit of technology that is provided only in newer versions.

Q: Will the Zend Framework also work with the upcoming PHP6?

A: Zend Framework seeks to support the version of PHP that is currently best to use for production applications. PHP 6 is still under development and we assume only the most intrepid users would attempt to deploy PHP 6 in production. So right now we do not test ZF with PHP 6. By the time PHP 6 is adopted by the mainstream PHP community, Zend Framework may be on its version 2.0, so we expect to have ample opportunity to support PHP 6,

Q: Do you have any knowledge of the Framework being used with an application running in PASE on an i5?

A: I'm not aware of any such applications currently in production, but I'm not an expert on the i5/OS platform, so I'm not the best person to ask.

Questions about the presentation

Q: What is the URL that we can access the recording of this presentation and the source code to the sample application?

A: Yes, they should be posted to <http://www.zend.com/webinar>

Q: I'm just starting to learn PHP and MySQL. Are you guys going to have a presentation that won't be as advanced as this one was?

A: We've been assuming that our audience has a bit of experience using PHP 5, object-oriented programming skills, and knows a database like MySQL. As we develop tutorials, we will probably continue to do so. It's probably better to find some existing introductions to PHP and MySQL. There are a lot of very good resources available, and it wouldn't make much sense for us to duplicate them.

Questions about examples, tutorials, documentation

Q: Can you point me to a good Hello World code using MVC in ZF or to this sample application you were talking about?

A: Yes, I will make the example application available at the same location as the webinar archive: <http://www.zend.com/webinar>

Q: Will there be a "real" example application which shows the right usage of all components and which provides a basis for own development?

A: We have begun developing an application that serves as an example, and is accompanied by a tutorial document. This work is in a draft state right now and we will continue to enhance it. See <http://framework.zend.com/wiki/display/ZFDEV/tutorial>.

Q: Will there be an official bootstrap script/approach?

A: Yes, this is an interesting idea that we have been discussing. Zend Framework is very flexible and does not force you into a specific design for your directory layout and bootstrap script. But it can be very helpful to have a "skeleton" project from which to start a new application. So we will develop a good reference script, but we will also maintain ZF as a flexible solution.

Q: Sort of a suggestion/question: any plans on doing screencasts. Visual learners really enjoy that and I think it has helped with adoption of other frameworks. Any plans?

A: It's a good idea and it has been suggested. But we don't have such a task in progress at this time. I'm sure we'll get to it.

Q: Is the Zend Framework limited to the class library or does Zend have any plan to start a "best practices" list or other methodology?

A: We currently have a document of coding standards and practices in our wiki site (<http://framework.zend.com/wiki/x/PQ>), but these are general PHP issues and not specific to ZF. We would like to do more, to provide recommendations and standards for application structure.

Questions about extending/customizing/integrating

Q: Are there no configuration files at all? How do you customize and adapt?

A: That's right; no part of ZF requires configuration files. Typically, resources are established in the bootstrap script. If you prefer to store your application-specific data in a configuration file, you can do so, and then use `Zend_Config` to read it.

Q: How is the `Zend_View` module different from Smarty, a popular template engine? Can Smarty be used? What advantage is there with Zend solution?

A: `Zend_View` scripts are simply PHP. If you know PHP, you know `Zend_View`. Certainly, Smarty can be used as an alternative to the `Zend_View` class. The ZF manual has an example of how to do this (see <http://framework.zend.com/manual/en/zend.view.scripts.html>).

Also keep in mind that Smarty uses the LGPL license, which is not as business-friendly as the New BSD license used by ZF.

Q: Can Zend Framework be integrated easily with other solutions like Smarty and ADODB?

A: Yes, ZF is designed to have very little interdependence between its components, especially to support alternative solutions like those you mention. The manual shows an example of integrating Smarty as an alternative scripting engine (see <http://framework.zend.com/manual/en/zend.view.scripts.html>).

Similarly, you can use ADODB as an alternative database access solution that you can use in your Model classes; you are not constrained to use Zend_Db. You can also use Propel or MDB or phpDoctrine, which are all fine solutions that solve the needs for which they were designed.

Q: How easily can I just use parts of the ZF without migrating my whole app over?

A: The loose coupling between components and simple but powerful API's make it very easy to integrate individual components, at your own pace and how it makes sense for your project. During my Zend_Db presentation at the MySQL Conference in April, a developer told me he integrated ZF's Zend_Log component into his own website, creating a solution to log all his traffic to a database. He said he did this work while sitting in the room listening to my presentation, using the hotel's wi-fi network to access his site. It took him about 20 minutes.

Q: Sounds like if I use Symfony, I could start adding Zend Framework components and slowly migrate over to Zend Framework, is that right?

A: Yes, that's right. It's easy to employ individual components from Zend Framework. There is no requirement to do it all at once. In the case of MVC, you should use one MVC solution throughout your application. But many people utilize some other individual components of ZF like the database access, search engine, or feed client, even if they haven't integrated their application with Zend_Controller yet.

Q: A very practical question: how to structure the directories so that HTML template files (for example Smarty) are correct when viewed from Web Design Tools and also when showed in browser. I'm thinking about correct path to images and CSS files, JavaScript files, etc.

A: The directory layout I showed in the example application is what our MVC solution uses as a default, but also it allows you to specify the location of the controllers, views, and other directories. So you literally can put the files where it works for your web design tool, and in a couple of lines of code, tell the MVC controller where to find the files.

Q: How about future cooperation with Java integration projects like PHP/java bridge? Could ZF have components to encapsulate PHP/java bridge connectivity?

A: Zend Framework is by design 100% PHP code and it does not implement PHP extensions or alter the PHP environment in which it runs. As such, ZF currently has no plans to implement Java bridging technology. My understanding is that Zend Platform is the preferred solution with which one can bridge PHP and Java code.

Q: How easy is it to upgrade to a new version of Zend Framework?

A: One of the chief goals of ZF was to provide a class library that could commit to backward-compatibility and allow users to have assurance that their applications continue to work with subsequent versions of ZF. Therefore, once we release ZF 1.0, we have an intention to maintain the class interface, so that upgrading to new versions will be virtually seamless.

Questions about IDE tool support

Q: Which visual IDE's are slated for integration with Zend Framework? Do you plan to integrate development features to Eclipse IDE? Regarding Zend IDE, will it also be open source or commercial?

A: Zend Studio already contains support for Zend Framework. Zend Studio is a commercial product offering of Zend. See http://www.zend.com/products/zend_studio. We don't have any plans right now to contribute support for Zend Framework to the PDT, but we encourage developers of commercial Eclipse plugins to support Zend Framework.

Questions about performance

Q: Isn't there a big overhead with frameworks that reduces performance?

A: These questions are very subjective. We've tried to design solutions that are very simple. This often has the benefit the code is light-weight and lower overhead. ZF has a use-at-will architecture, so only classes that you need are loaded. You don't have to load the entire collection of classes; they just sit there on your file system.

Q: Some other frameworks (Solar, CodeIgniter) claim to be much faster than ZF. Has Zend done any performance testing and gathered stats around using Zend's MVC?

A: I'm reminded of this famous quote:

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. Yet we should not pass up our opportunities in that critical 3%." – Donald Knuth, paraphrasing Tony Hoare

Anyway, to answer your question: no, we have not tried to generate benchmarks or statistics of ZF performance relative to other solutions. Software benchmarks generally do not say anything that is accurate, conclusive, or interesting.

Q: I noticed ZF increases memory footprint. How will applications handle large number of visitors?

A: ZF includes a couple of components that help web applications scale at the application layer, for example, Zend_Cache and Zend_Memory.

But the best solution for making a web application scalable – regardless of the specific technology you use – is an effective architecture that includes strategies for load-balancing and caching. There are multiple layers for doing this, including the application layer, the PHP engine, the database server, the web server, the sever hardware, the network and router, etc. Don't depend on any single one of these layers to solve your scalability architecture, and don't assume that any given layer is always the bottleneck.

Q: I am developing a high traffic web site, and our system administrators disabled .htaccess in order to get a little more performance. Isn't the ZF routing much slower then mod_rewrite?

A: You can use mod_rewrite without using .htaccess files. I do not think that ZF routing is not significantly slower than using mod_rewrite.

Questions about MVC

Q: What is the second argument in the _getParam() method?

A: The second argument to the _getParam() method shown in the Controller classes is a default value. The default value is returned if the specified parameter is not present in the web request.

Q: Is RewriteEngine On required? Will we be able to deploy ZF applications on shared hosting services which do not allow rewriting?

A: If you cannot use Apache's RewriteEngine, you can still use Zend Framework, but the URL's for your application won't look as nice.

Q: How would ZF - the way it uses the URL to pass params - support user directories like <http://www.mydomain.com/USERNAME> ?

A: You should look into Apache's "RewriteBase" directive. See http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html#rewritebase. With this solution, you can serve a ZF application from a subdirectory; you don't have to put it at the top of the document_root. I use this in the example application.

Q: How will the way ZF uses the URL to pass parameters affect SEO?

A: I assume you mean that if the URL parameters look like a series of subdirectories, it will make search engines think that they are different pages. It would be a good policy to use this format of request parameter only when it is appropriate to treat the pages as different targets for a search engine. Zend_Controller also supports GET and POST request parameters.

Q: Is it possible to create a URL, such as `"/catalog/(catalogID)"` using the ZF routers? I mean, like a URL without an action, and with a parameter.

A: Yes, the default routing like `"/controller/action/param1/value1"` can be customized easily. You can define your own routing rules and add them to the Front Controller (see <http://framework.zend.com/manual/en/zend.controller.router.html>).

We use some custom routes on our web site. For example, in an URL like <http://framework.zend.com/manual/en>, the controller is "manual" and the locale code "en" is a request parameter. The action name is not represented in the URL, but the routing rule defines it.

Q: Should we use relative or absolute URLs to images and CSS files, etc?

A: This is a question that is applicable to web development in general, regardless of the framework used. Certainly ZF allows you to use either relative or absolute HREF's. The Controller class also has convenience methods so the View script can get the base URL of the application. This allows you to make absolute HREF's without making your application fixed to a certain location.

Q: What is the best practice for creating complex views (e.g. widgets, common headers/footers)?

A: This is an ongoing discussion right now, and we haven't settled on a single best practice. I would guess that this indicates that the problem is complex and there isn't a single best solution. But it's clear to us that the ZF needs some documentation to provide guidelines for web application design best practices.

Q: Why is the `_getParam()` underscored? Isn't that against the coding standards, because we use it like a "public" method?

A: The convention in PHP is that protected or private methods are spelled with a leading underscore, and public methods are not. In a controller action method, you can use the protected method like `"$this->_getParam()"` or you can access the controller's request object and use its public method like `"$this->getRequest()->getParam()"`. These are equivalent; the former simply calls the latter.

Q: Regarding templates, can ZF be adapted to use other template system like Dreamweaver DTW templates? The problem we have here is that our designers like to work with that and we developed our own template (in fact a complete framework) using DWT.

A: Zend_View is an interface, so you can write your own view class implementing that interface, which reads DTW templates. See <http://framework.zend.com/manual/en/zend.view.scripts.html>

Q: Do you have a recommended URL rewriter for IIS? So far I have had the most experience with Ionics Isapi Rewrite Filter (IIRF)

<http://www.codeplex.com/IIRF/Release/ProjectReleases.aspx?ReleaseId=5018>

A: I'm only an occasional user of IIS, so I'm not the best person to answer this question.

Q: Does the ZF is suitable for application based on workflow?

A: The MVC pattern can be used to support workflow-based application architecture, even though that is not the primary purpose for MVC, and merely using MVC does not give an application a workflow-based architecture. Making a complete workflow solution is outside of the scope of what a humble framework can achieve.

Questions about Db

Q: Why is there a limitation of only using PDO? Is there a work around?

A: Zend_Db already provides Adapters for several non-PDO database extensions. Currently you can use MySQL, Oracle, and IBM DB2 with no requirement to have PDO.

See a list of supported adapters here:

<http://framework.zend.com/manual/en/zend.db.html#zend.db.adapter>

Q: I would also be interested in knowing more about plans for a higher-level DB abstraction, such as ActiveRecord.

A: ZF has classes Zend_Db_Table and Zend_Db_Table_Row, implementing the design patterns known as "Table Data Gateway" and "Row Data Gateway". These patterns are very similar to ActiveRecord.

Q: Does the Zend_Db Adapter allow you to access databases without having the database DB extension loaded?

A: No, Zend_Db_Adapter merely provides a consistent class interface so that you can use the PHP drivers for different databases without being required to change your application code. You do need to have the PHP extension for the respective database present and enabled.

Q: Is the DB2 adapter based on ODBC driver or is it independent?

A: The Zend_Db Adapter for IBM DB2 is based on the ibm_db2 extension in PHP. In addition, there is a new Adapter under development, which uses the pdo_ibm extension.

We do intend to develop a Zend_Db Adapter for the ODBC extension.

Q: What is the difference between Table Gateway and Zend_Db_Table?

A: Table Data Gateway is an abstracted design pattern described in Martin Fowler's book "Patterns of Enterprise Architecture." Zend_Db_Table is an implementation of that design pattern in PHP 5.

Q: Is there a preferred adapter to use for MySQL databases? How mature are Mysqli and Pdo_Mysqli adapters?

A: Zend_Db offers two Adapters for accessing MySQL. One utilizes the pdo_mysql extension, and the other utilizes the mysqli extension. They are both functioning fine and we test them both with an identical suite of tests. Both adapters are passing with flying colors. You can run the test suites yourself – they’re part of the ZF download.

Questions about other specific components

Q: Is Zend_Pdf capable of generating PDF archives based on CSS styles?

A: No, I don’t believe so.

Q: Has Zend_Log class any log rotating functionality which avoids big log files?

A: Log rotation is best handled outside of PHP. For example, Linux has a tool “logrotate” that is used generically for log management of many types of services. You can use logrotate for your own applications too.

Q: Does Zend_Search_Lucene work with IIS web server?

A: Yes, I believe so. But I’m not an expert on using IIS; I’m an Apache user.

Q: I want to know if filter supports UTF-8 and foreign language that's not ASCII?

A: Yes, we recently updated Zend_Filter and Zend_Validate classes to support UTF-8.

Q: Does the Zend PDF generator require the PDFLib extension?

A: No, I don’t think Zend_Pdf requires that extension. It includes its own complete PDF file parser and generator.

Q: What benefits does Zend Framework offer with AJAX style apps?

A: ZF includes a component called Zend_Json, which facilitates converting PHP data to and from JSON format. So this makes it easy to share information with an AJAX front-end. Also, you can design MVC controllers that interact with AJAX clients, and set their response mime-type to data that will be readily consumed by AJAX.

Q: Are there any known problems with UTF-8 and ZF?

A: Yes, we currently have a few open issues regarding improving support for UTF-8, but they are minor issues. We will address these issues and support UTF-8.

Questions about future development

Q: Is there a means of generating a web services WSDL on the fly?

A: We currently have SOAP and WSDL support under development, but they will not be finished for ZF 1.0.

Q: Is there a rough estimate of when the web form functionality will be available?

A: I don’t want to quote a date because I would rather let that be determined by our quality goals. It’s likely to be several months. Designing a solution for processing forms is a complex task. We want to design something that satisfies the goals that it solves 80% of users’ needs, allows for extensibility to satisfy most of the remaining 20%, and still provides an “extremely simple” interface.

Q: Has any work started on the advanced web forms component of the Zend Framework?

A: First we needed some of the lower-level classes to support a form-processing solution. The MVC and also Zend_Filter/Zend_Validate were steps in this direction, establishing a foundation

on which to build. Following the release of ZF 1.0, we will begin designing and implementing a forms component in earnest.

Q: I heard the logrotate question and that makes me wonder, what is the current methodology for determining what should be and what should not be in Zend Framework. I would love to know it will never become bloated.

A: The method is that the team of developers at Zend reviews proposals for components to include in Zend Framework. In general, we try to include components that meet the needs of very common web application development tasks.

We have turned down proposals that are designed to solve exotic problems, outside of the needs of a great majority of web developers. We have also turned down proposals that offer no benefit over similar extensions available in standard PHP. Also proposals that are overly “generic” and do not address a specific use case are rejected.

Q: Are there any plans for making Zend_Debug fancier? In a way of looking like Xdebug.

A: No, currently there is no specific proposal for enhancing the Zend_Debug class.

Q: Do you support SOAP web services in addition to REST web service clients and servers? Or better stated, does ZF provide a SOAP webservice client and server?

A: Yes, Zend_Service_StrikeIron is an example of a web service supported in ZF that is SOAP-based. It uses the standard SOAP extension available in PHP. We also have a SOAP server component with WSDL support but it is not going to be finished in time for ZF 1.0.

Q: Does the Zend Framework have MS-Word or MS-Excel document generation capabilities?

A: That’s a great idea, and we thought of this too. In Microsoft Office 2007, the file formats are based on XML, so they may be easier for tools to parse or generate than the proprietary binary formats like BIFF used in earlier releases. A PHP class to work with the MS Office formats would be great, but so far we do not have a proposal to implement this solution.

Q: Are planned components to work with video/VOIP?

A: We have no proposal to develop components for video or VOIP. I think this would qualify as an exotic application. In general ZF tries to provide solutions that are needed by a majority of typical web applications, so we have not sought to work with advanced media technology.

Q: Do you plan on developing an AJAX object that encapsulates the functionality of popular API’s like Prototype and jQuery?

A: Yes, we do have intentions of developing PHP classes to support rich-client interaction patterns.