Zend Framework in Action White Paper

Zend Framework in Action
Green Paper
EARLY ACCESS EDITION

MEAP Release: September 2007
Softbound print: October 2008 (est.) | 425 pages
ISBN: 1933988320

This paper provides an introduction to the Zend Framework and the book Zend Framework in Action. It includes an overview of the Zend Framework and its benefits and then explores Zend Framework in Action and its target audience.

What is the Zend Framework
In late 2005, Zend Technologies, a company that specializes in PHP, started the Zend Framework. The Zend Framework is an open source project that provides a web framework for PHP and is intended to become one of the standard frameworks that PHP applications of the future will be based on. It provides a set of components to enable you to build PHP applications more easily, which will be easier to maintain and extend over the lifetime of the application. Version 1.0 was released on 30 June 2007 with maintenance updates approximately every two months.

The Zend Framework is composed of many distinct components grouped into a set of top level modules as shown here:
As a complete framework, there is everything required to build enterprise ready web applications. However, the system is very flexible and has been designed so that only the relevant parts of the framework are used for any given application. Following on from the high level overview, this table shows all the components:

<table>
<thead>
<tr>
<th>Core:</th>
<th>Inter-application communication:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zend_Controller</td>
<td>Zend_Json</td>
</tr>
<tr>
<td>Zend_View</td>
<td>Zend_XmlRpc</td>
</tr>
<tr>
<td>Zend_Db</td>
<td>Zend_Soap</td>
</tr>
<tr>
<td>Zend_Config</td>
<td>Zend_Rest</td>
</tr>
<tr>
<td>Zend_Filter &amp; Zend_Validate</td>
<td>Web Services:</td>
</tr>
<tr>
<td>Zend_Registry</td>
<td>Zend_Feed</td>
</tr>
<tr>
<td>Authentication and Access:</td>
<td>Zend_Service_Amazon</td>
</tr>
<tr>
<td>Zend_Acl</td>
<td>Zend_Flickr</td>
</tr>
<tr>
<td>Zend_Auth</td>
<td>Zend_Service_Yahoo</td>
</tr>
<tr>
<td>Zend_Session</td>
<td>Utility:</td>
</tr>
<tr>
<td>Internationalization:</td>
<td>Zend_Cache</td>
</tr>
<tr>
<td>Zend_Date</td>
<td>Zend_Search</td>
</tr>
<tr>
<td>Zend_Locale</td>
<td>Zend_Pdf</td>
</tr>
<tr>
<td>Zend_Measure</td>
<td>Zend_Mail / Zend_Mime</td>
</tr>
<tr>
<td>Http:</td>
<td>Misc!</td>
</tr>
<tr>
<td>Zend_Http_Client</td>
<td>Zend_Measure</td>
</tr>
<tr>
<td>Zend_Http_Server</td>
<td></td>
</tr>
<tr>
<td>Zend_Uri</td>
<td></td>
</tr>
</tbody>
</table>

Each section of the framework consists of a number of components, which is usually the name of the main class too. For example, Zend_View is the concrete view class used by applications. Each component also contains a number of other classes too that are not listed in the above table.

The Core Components
The core components provide a full-features Model-View-Controller (MVC) system for building applications that separate out the view templates from the business logic and controller files. The three main classes are Zend_Controller_Front and Zend_Controller_Action acting as a controller system, Zend_View, view template system and Zend_Db_Table which is a gateway class to a database table.

Security is very much on the minds of every professional PHP developer. Input data validation and filtering is the key to a secure application. Zend_Filter and Zend_Validate are provided to help the developer ensure that input data is safe for use in the application.

The Zend_Filter class provides a set of filters that typically remove or transform unwanted data from the input as it passes through the filter. For example, an numeric filter would remove any characters that were not numbers from the input and an HTML entities filter would convert the “<” character to the sequence “&lt;”.

©Manning Publications Co. For Source Code, Sample Chapters, the Author Forum and other resources go to http://www.manning.com/allen
Appropriate filters can be set up to ensure that the data is valid for the context it will be used in.

ZendValidate provides a very similar function to Zend_FILTER, except that it provides a yes/no answer to the question “is this data what I expect?”. Validation is generally used to ensure that the data is correctly formed. For example, validating that the string provided as an email address is actually an email address. In the case of failure, ZendValidate also provides a message indicating why the input failed validation so that appropriate error messages can be provided back the end user.

Authentication and Access Control Components
Not every application needs to identify their users, but it is a surprisingly common requirement. Access control is the process of providing access to a given resource, such as a web page, to an authenticated user. That is, authentication is the process of identifying and entity, usually via a username/password pair, but could equally be via a fingerprint. Access control, sometimes called Authorization, is the process of deciding if the authenticated entity is allowed to have access to, or perform operations on, a given resource, such as a record from a database.

As there are two separate concept, the Zend Framework provides two separate components: Zend_Acl and Zend_Auth. Zend_Auth is used to identify the user and is typically used in conjunction with Zend_Session to hold that information across multiple page requests (known as token persistence). Zend_Acl is then uses the authentication token to provide access to private information using the Role Based Access Control List system.

Internationalization Components
As we live in a multi-cultural world with multiple languages, the framework provides a rich set of functionality to allow for localizing your application to match your target users. This covers minor issues like ensuring that the correct currency symbol is used through to full support for changing all the text on the page to the correct language. Date and time routines are also provided with a simple object oriented interface to the multitude of ways that date and time is displayed.

Http Components
The Zend Framework provides a component to read data from other websites. Zend_Http_Client makes it easy to collect data from other web sites and services and then present it on your site. A server component is also provided to allow for PHP based serving for web pages. Obviously this component is intended for development and other specialized requirements rather than general web page serving, as proper web-servers like Apache are orders of magnitude faster!

Inter-application Communication Components
When you need to communicate with another application over HTTP, the most common transfer format is one of two flavors of XML: XML-RPC and SOAP. As you would expect from an enterprise-class framework, the Zend Framework provides components to allow for easy processing of both XML-RPC and SOAP protocols. More recently, the lightweight JSON protocol is gaining favor, mainly due to how easy it is to process within the JavaScript of an Ajax application. Zend_Json provides a nice solution to both creating and reading JSON data.
Web Services Components
The Zend Framework provides a rich set of functionality to allow access to services provided by other suppliers. These components cover generic RSS feeds along with specific components for working with the public APIs from Google, Yahoo! and Amazon. RSS has come a long way from its niche amongst the more technologically minded bloggers and is now used by the majority of news sites. Zend_Feed provides a consistent interface to reading feeds in the various RSS and atom versions that are available without having to worry about the details.

Utility Components
There are a set of other components provided with the Zend Framework that do not fit easily into any category, so they are grouped together here into the utility category. This potpourri of components includes caching, searching, pdf creation, email and the rather esoteric measurement class.

Everyone wants a faster website and caching is one tool that can be used to help speed up your website. The Zend_Cache component provides a generic and consistent interface to cache any data in a variety of back end systems such as disk, database or even with APC’s shared memory. This flexibility ensures that Zend_Cache solutions can be scaled as the load on the website increases.

Zend_Search is based on the Apache Lucene search engine for Java and provides an industrial strength text search system that will allow your users to find what they are looking for. As required by a good search system, it supports ranked searching so that the best results are at the top, along with a powerful query system.

Another useful component is Zend_Pdf which covers the reading and creation of PDF files. PDF is a portable format for creating documents intended for printing. This is because you can control the position of everything on the page with pixel-perfect precision without having to worry about differences in the way web browsers render the page. Zend_Pdf is written entirely in PHP and can create new PDF documents or load existing ones.

Email Components
The Zend Framework provides a strong email component to allow for sending emails in plain text or HTML. As with all Zend Framework components, emphasis has been placed on flexibility combined with sensible defaults. Within the world of email, this means that the component allows for sending email using SMTP or via the standard PHP mail() command. Additional transports can be easily slotted into the system by writing a new class that implements Zend_Email_Transport_Interface.

Why use the Zend Framework?
The Zend Framework provides a standardized set of components that allow for easy development of web applications. These applications can be easily developed, maintained and enhanced.

The key features of the Zend Framework are:

- Everything in the box
- Modern design
- Easy to learn
• Full documentation
• Rapid development

Everything in the box
The Zend Framework is a comprehensive full stack framework that contains everything you need to develop your application. This includes a robust MVC component to ensure that your website is structured according to best practices. Accompanying the MVC component, there are components for authentication, searching, localization, PDF creation, email and connecting to web services, along with a few other more esoteric items.

A core feature of the design of the framework is that it is easy to use just those bits you want to use with the rest of your application or with other libraries such as PEAR, the Doctrine ORM or the Smarty template library.

Modern design
The Zend Framework is written in object-oriented PHP5 using the modern design techniques, known as design patterns. Software design patterns are recognized high level solutions to design problems and, as such, are not a specific implementation of the solution. The actual implementation depends on the nature of the rest of the design. The Zend Framework makes use of many software design patterns and its implementation has been carefully designed to allow the maximum flexibility for application developers without making them do too much work!

The framework recognizes the PHP way and doesn’t force you into using all the components, so you are free to pick and choose between them. This is especially important as it allows specific components to be integrated into an existing site.

Easy to learn
The Zend Framework is modular and has a design goal of simplicity which makes it easy to learn, one step at a time. Each component doesn’t depend on lots of other components and so is easy to study. The design of each component is such that it can be easily used with defaults and then more advanced features can be used later. This helps to reduce the barrier to entry for most users.

Full documentation
The Zend Framework is aimed at developers who do not want to have to dig through all the source code to get their job done and so the Zend Framework project puts documentation on an equal footing with the code. There are two types of documentation supplied with the framework: API and end-user. The API documentation is created using PHPDocumenter and is automatically generated using special “docblock” comments in the source code. These comments are typically found just above every class, function and member variable declaration. The key advantage of using docblocks is that IDEs such as PHPIDE in Eclipse or Zend’s Studio are able to supply auto-completion tool tips whilst coding and so improve developer productivity.

The Zend Framework also supplies a full manual as part of the download and also available online at http://framework.zend.com/manual. The manual provides details on all components of the framework and shows what functionality is available.

©Manning Publications Co. For Source Code, Sample Chapters, the Author Forum and other resources go to http://www.manning.com/Allen
Examples are provided to help you get started in using the component in an application. More importantly, in the case of the more complicated components (such as Zend_Controller), the theory of operation is also covered, so that you can understand why the component works the way it does.

The documentation provided with the framework does not explain how to fit all the components together to make a complete application. As a result, a number of tutorials have sprung up on the web by community members to help developers get started on the framework. These have been collated on a web page on the framework’s wiki at http://framework.zend.com/wiki/x/q.

Rapid Development
As the Zend Framework provides many of the underlying components of an application, the developers are free to concentrate on the core parts of the application that add value, rather than on the underlying foundation. Hence, it is easy to get started quickly on a given piece of functionality and immediately see the results.

Clean IP
All contributors to the Zend Framework have signed a Contributor License Agreement. This is an agreement with Zend which defines intellectual property status of the contribution. That is, the contributor warrants that (to the best of her knowledge), she is entitled to make the contribution and that no one else’s intellectual property rights are being infringed. This is intended to help protect all users of the framework from potential legal issues related to IP and copyright.

The Zend Framework’s MVC Components
Zend_Controller is the centrepiece of the Zend Framework’s MVC system. MVC stands for Model-View_Controller and is a design pattern that is used to separate an application’s code to make it more maintainable as shown in this diagram:

The Model
The model part of the MVC pattern is all the code that works behind the scenes related to how this particular application works. This is known as business logic. This is the code that decides how to apply the shipping cost to an e-commerce order or the code that knows that a user has a first name and a surname. It follows therefore that retrieving and storing data to a database is within the model layer. In terms of the
code, the Zend Framework provides the Zend_Db_Table class which provides table level access to the database and allows for easily manipulating the data used by the application.

The View
The view is the display logic of the application. For a web application, this is usually the HTML code that makes up the web pages, but can include, say, XML that is used for an RSS feed. Also, if the website allows for export in CSV format, the generation of the CSV would be part of the view. The view files themselves are known as templates as they usually have some code that allows for the displaying of data created by the model. It is also usual to move the more complex template related code into functions known as View Helpers, View Helpers improve the re-usability of the view code. By default the Zend Framework’s view class (Zend_View) uses PHP within the template files, but another template engine such as Smarty or PHPTAL may be substituted.

The Controller
The controller is the rest of the code that makes up the application. For web applications, the controller code is the code that works out what to actually run in response to the web request. For Zend Framework applications, the controller system is based on the design pattern known as Front Controller which uses a handler (Zend_Controller_Front) and action commands (Zend_Controller_Action) which work together in tandem. The front controller handler accepts all server requests and runs the correct action function within the action command. This process is known as routing and dispatching. The action class is responsible for a group of related action functions which perform the “real” work required from the request. Within the Controller of the Zend Framework, it is possible to have a single request result in the dispatch of multiple actions.

The various parts that make up the MVC system are shown in the following diagram:
The front controller, Zend_Controller_Front controls the routing and dispatching of actions. The action command then uses Models, which are usually linked to a database using Zend_Db_Table, and sets up the view (an instance of Zend_View) which is used for creating the output that is rendered to the response object.
Conclusion

The Zend Framework is a compelling framework for developing PHP applications for a number of reasons. For long term stability, the Framework is created and managed by Zend Technologies who are a leading PHP company. Also supporting the framework are companies like IBM. The license is the open source New BSD license which is very friendly for open source projects and corporations alike.

With a robust MVC system at its centre, the Zend Framework provides everything that is required to build enterprise level, modern web applications that can interface to other web services as required. The framework follows the PHP-way and doesn’t mandate an application must be set up or configured and so provides maximum flexibility for developers. Each component is unit-tested and fully documented so as to provide high quality code with simple usage.