1
Theming Plone 3: An Overview

For those of you who are unfamiliar with Plone or this is your first introduction to it, Plone is a content management system (CMS). A CMS is a tool that enables webmasters to manage their web sites' content easily **through the web**. Plone provides additional support by auto-generating navigation elements, making content searchable, allowing for multilingual content, handling permissions and security, and much, much more. In comparison to other CMS platforms, Plone is extremely stable, secure, and is actively working toward increased adoptability.

One of the most common needs with a CMS web site is the ability to change the look and feel. As with any framework, building a design around a CMS introduces special challenges. With Plone, the process of separating content from presentation is not entirely straightforward. By the end of this book, you will have the knowledge to create your own robust Plone themes and understand where the line is drawn between content and presentation. Specifically, this book is geared toward web designers with little or no knowledge of Plone who need to theme a Plone web site.

This chapter will explore the origins of Plone as well as analyze the current state of theming in Plone.

Background

The origin of Plone is one of community lore. In 1999, founders Alexander Limi and Alan Runyan had a fortuitous meeting on **#zope**, the IRC channel dedicated to **Zope** development. Zope is the framework upon which Plone is built. The two of them forged an online friendship based on a mutual love of Zope, Python, and music.

These two men, with too much time on their hands and the encouragement of Paul Everitt (one of the founders of Zope Corporation), built a CMS named after an electronica band with questionable musical talent. In the process, they also gathered a thriving community of people around them. They continue to work in the service of Plone today and to grow the ranks around them.

The last year or so has brought tremendous change to Plone, as we have moved from the Zope 2 architecture to the mixed implementation of Zope 2 and Zope 3. Don't worry too much yet about what this means; suffice it to say that when Plone 3 was released, the ground shifted underneath the feet of Plone users worldwide. This shift not only brought a lot of power to the table, but also introduced a lot of fear, particularly in the hearts of themers. Most of this fear is exaggerated, and this book aims to quell much of this fear. The theming process for Plone may be complex at the moment, but it's still possible to generate beautiful, high-impact themes.

Plone is useful for all kinds of web sites, from large enterprise sites to educational and government sites, and even small environmental sites. I work at an environmental non-profit named ONE/Northwest. (We hosted the 2006 Plone Conference in Seattle.) ONE/Northwest has produced almost 200 attractive, high quality, but generally small Plone sites that empower other environmental groups. These web sites give our clients the tools they need to reach their audiences and to hopefully preserve our natural environment. Every day brings new excitement, revelations, more complex sites, and plenty of ideas on how to improve Plone. As an open source CMS, it's especially thrilling to be able to contribute ideas and put efforts that incrementally improve Plone.

For many members of the Plone community, we stay with it because it's all about contributing and the community. We are a vast and far-flung crowd, yet surprisingly close-knit. At any given point in time on **#plone**, the IRC channel for Plone users on freenode.net, you can be talking simultaneously to the people in Belgium, South Africa, Australia, Israel, or the United States. Sometimes the community gets together in person to work, sometimes at conferences, but more often at sprints (small gatherings geared toward solving a particular problem). People have sprinted on projects in castles, on archipelagos, on beaches, in the swank offices of Google Corporation and online, all over the world. The community has proven to be extremely welcoming, bright, ambitious, and inclusive to those who are willing to contribute back to the community.

As a themer, I continue to use and improve Plone because it's a rewarding platform with which to work, and because I've witnessed the change in the world it can effect. My personal pet project with Plone is a small non-profit called **Safe Passage** that helps to educate, feed, and protect children who work in the garbage dumps of Guatemala City. With a very simple (and sadly outdated) theme, and a form-building product called **PloneFormGen**, Safe Passage has managed to get more than 900 children sponsored, fed, and in school. It's a small, unassuming site, but without Plone, we could not have achieved such a tremendous accomplishment. Plone is a wondrous tool, frustrating at times, but absolutely worth the time spent and valued by the organizations that use it to manage their content and spread their message. From the simple theme I built for Safe Passage to the intricate themes I build now, I've learned that Plone is a deep and sometimes swift river, but not impossible to cross. The best part is that even as you read this book, work is being done in the Plone community to ease the pain of theming. This book is one small step in that direction.

Before we move into the theming portion of this book, let's learn a little more about Plone and why it might be a good choice for your organization.

What is Plone, really?

The following is an overview of Plone:

- **Plone is an open source CMS**: Plone is a downloadable content management system that is built on the powerful and free Zope application server. Plone is easy to set up, extremely flexible, and provides tools for managing the content of large web sites, extranets, intranets, government and educational sites, and even social networking sites. Plone is licensed under the **GNU General Public License**, the same license used by Linux. This gives you the right to use Plone without a license fee, and to improve upon the product.
- **Plone has the tools you need**: Plone provides numerous out-of-the-box and add-on tools that make working with content easier. These include **kupu**, the powerful visual editor built into Plone, PloneFormGen for creating quick and easy forms. Plone also includes the ability to integrate with other open source tools, page compositors, e-commerce solutions, and more. Plone may not have as many add-ons as frameworks like Drupal, but you can feel secure in knowing that the popular add-ons for Plone are generally quite stable and thoroughly vetted by the community.

- **Plone is easy to install**: You can install Plone by going to `http://plone.org/products/plone` and downloading the Plone installer that is the right option for you. Simply run the installer, follow the `README.txt`, and you will have a working content management system in minutes. As of Plone **3.1.2**, Plone is by default installed using a system called **buildout**, which we will cover later. It's more complex than WordPress's "famous 5-minute install", but installing Plone and add-on products is simple once you get used to the process.
- **Plone is easy to use**: Plone's development team has usability experts who have made it simple for content managers to add, update, and maintain content. Plone's UI team is constantly looking for ways to simplify and improve the end user experience, and founder Alexander Limi is on the forefront of this charge. I've had the joy of having content managers from completely different continents grasp the concepts of how to manage content in Plone without problem, and the average web-savvy user will have the same experience.
- **Plone is secure**: While the Drupal platform may have much to offer to end users, as a PHP-based CMS, it has many security problems that Plone does not have. Plone has had only one security-related patch in the last two years, whereas Drupal has had several in the last few months. The difference is that Plone runs on Python, which is markedly safer. Having a secure CMS is critical, and Plone is clearly the frontrunner here.
- **Plone provides international support**: The Plone interface has more than 35 translations, right-to-left support, and tools such as **LinguaPlon**e for managing multilingual content.
- **Plone is compliant**: Plone rigorously follows standards for usability and accessibility, including US Section 508, and the W3C's AAA rating for accessibility.
- **Plone is protected**: The nonprofit Plone Foundation (`http://plone.org/foundation`) was formed in 2004 to promote the use of Plone around the world and protect the Plone Intellectual Property and trademarks.
- **Plone has planned development and supports contributors**: The Plone development keeps a close eye on the future of Plone, gives considered thought to new features, and presents a unified front. This is different from a CMS such as Drupal, which has more sprawl in its development processes, especially in terms of add-on products. As of this writing, the biggest developments in Plone include dramatically lowering the bar on theming, making the page compositing experience much simpler, and enabling through-the-web creation of content types. Ultimately, the focus for Plone 4 is on integrators and themers, which is right where the focus needs to be in order to have an adoptable CMS.

- **Upgrades are easy**: Upgrades are less frequent, and releases are carefully coordinated to make the transition easier. For anyone who has suffered a painful upgrade, you'll appreciate this fact more than anything.

Technical overview

Plone sits on top of the Zope technology stack. Zope is an open source application server for building content management systems, intranets, portals, and custom applications. The Zope community consists of hundreds of companies and thousands of developers all over the world who work on building the platform and other Zope applications.

Zope and Plone are both written in Python, an easy-to-learn, widely used and supported open source programming language. The security benefits available with Python, as well as the cleanliness of code, are great advantages for Plone.

By default, Plone stores its contents in Zope's built-in transactional object database, the **ZODB**. There are products and techniques, however, that allow sharing of information with other sources, such as relational databases, LDAP, and more. WSGI support is also now available with Plone, which means even greater integration with other web applications. WSGI is the basis of what makes **Deliverance** (the future of theming for Plone) possible. Read the last chapter for more exciting information on Deliverance.

Plone runs on Windows, Linux, BSD, Mac OS X, UNIX, Solaris, and other platforms. Double-click installers are available for all platforms. For full information and to download Plone, see `http://plone.org/products/plone`.

Books about Plone

Plone currently has approximately 7 books available, several of which are out of date, but a few of which are extremely helpful. None of the books are specifically geared towards themers, although sections of each are relevant.

- Released in February 2009 is a community-written book, *Practical Plone 3*, which is intended for integrators and individuals new to Plone. It contains several chapters geared towards themers, plus a wealth of other information for integrators.
- Martin Aspeli's developer's guide, *Professional Plone Development* (2007). While it is geared towards developers, some of the technical information in this book is pertinent to themers.

- The defacto Plone book, *The Definitive Guide to Plone* by Andy McKay. It is woefully out of date, but is relevant for giving a broad understanding of Plone and the templating language used by Plone. A rewrite to this book is expected in 2009.
- Philipp von Weitershausen published *Web Component Development with Zope 3* in 2007. It is a helpful book conceptually, but geared specifically towards pure Zope 3 development, not Plone.
- *Plone Live*, which had regular updates for years, but is now languishing since the Plone **3.0** release. The information it contains is still valid, however.
- *Content Management with Plone: Handbook for Authors and Editors* (available in English and German, updated for Plone 3). This book is intended for end users and not generally helpful to themers.
- James Cameron Cooper's *Building Websites with Plone* (2002). Mostly out of date, and not especially relevant to themers.

Theming and other CMS frameworks

To understand the current state of theming with Plone, it's helpful to examine a rival content management system's theming story. In this case, we will look at Drupal, a popular open source CMS that is written in PHP, and touch briefly on WordPress.

Both Plone and Drupal provide online theming manuals. Plone also has additional quick start documents that explain how to build a theme in Plone 3. These documents are steadily being integrated into the Plone theming manual. As of this writing, both projects have solid theming documentation, although Drupal's might be organized slightly better.

Drupal also has more theming books than the Plone community, but it's worth mentioning that the quality of the books about Plone with relation to theming is quite helpful. Unlike Drupal, the book you are reading now is the first book specifically geared toward theming with Plone, but it is the beginning of a trend. Basically, what this means is that either CMS is a good choice, and Plone has the documentation ready for new themers.

Another telling point is the availability of open source themes. Interestingly, at one point, Drupal had the `themes.drupal.org` site that allowed users to test-drive the available themes for Drupal. As of Drupal **5.0**, the site was abandoned due to the shortage of volunteers. WordPress also offers a number of add-on themes, though most of them are geared specifically toward bloggers. In contrast, Plone volunteers are currently working at putting in place a better user interface for finding themes (blog-like and otherwise), writing and improving documentation, and looking at a themes.plone.org-type web site for test driving open source themes.

This is the most active time for the Plone theming community to date, and the entire community has taken notice.

More than books, tutorials, and the availability of open source themes, the actual skinning process is where comparing Plone and Drupal is most critical. It's generally acknowledged that getting started with skinning in Drupal and WordPress are easier because users only have to worry about CSS, but once you get past the initial theme, you have to worry about PHP, which can be spidery and hard to understand. Conversely, Plone's theming framework is tough at the outset, but easier, more logical, and sophisticated over the long haul. Both situations present special challenges.

According to Larry Garfield in **#drupal**:

> *I'd say the hardest thing to get used to with Drupal is letting Drupal do things the Drupal Way rather than trying to force it into your mold. Drupal can bend in all sorts of weird and exciting ways, if you bend it where it's designed to bend. But if you try to shoe-horn it into the way you wish it worked rather than the way it does, you'll waste a lot time needlessly,* especially *at the theming layer.*

The same could be said about Plone, but Plone's problem is less about shoe-horning and more about having a lot of concepts that need to be understood before the real progress can be made. Additionally, Plone currently provides more than one method of accomplishing theming tasks, which can cause confusion. Those various methods are steadily being unified and simplified, and the goal of this book is to demystify the complexity of Plone with regards to theming.

Larry Garfield from #drupal followed his previous comment by saying that:

> *Drupal 6 includes a heavily rewritten theme layer that is a lot nicer to work with. Plone doesn't currently have an abstracted theming layer, though there are certain tools that are available to themers that will be discussed in this book. Moreover, the future of Plone is Deliverance, a non-Plone-specific tool that will turn the theming process into a nearly CSS-only experience. Deliverance will position Plone over with the Drupals and WordPresses of this world, and with much greater power.*

It's worth spending time with Plone, especially once you get a glimpse at the power under the hood. The real potential is in the value to the content manager. Heavyweight blogs such as WordPress, or lightweight CMS's such as Drupal, simply don't offer the necessary flexibility or ease of use, nor do they have the same level of extensibility. If you can build a Plone theme, you'll appreciate the real value inherent in Plone, and building a theme doesn't need to be intimidating once you know where to start.

While there's always going to be the question of which CMS is the most appropriate for the job, it's clear that Plone has a bright future, and is keeping pace with other CMS frameworks.

The evolution of skinning for Plone

Plone has always offered a robust base skin from which to start with solid CSS hooks, and it also has cleanly separated the CSS files that make skinning a fairly straightforward process. The actual process of working with the skin and CSS is what has changed dramatically over time.

As of Plone **2.0**, skinning was most commonly done through-the-web via the **ZMI** or **Zope Management Interface**. Working TTW (through the web) resulted in a mixture of content and code that could not be easily pulled out of the ZMI. While tools exist that helped extract skin-related elements from the ZMI, it was still difficult to back up sites or to preserve the configuration in the event that a site needed to be moved to another Zope instance.

When Plone **2.1** was released, a tool known as **DIY Plone Style** was introduced that allowed users to quickly create a skeleton product so that users could work more easily on the filesystem. Additionally, a system known as **GenericSetup** was gaining momentum as a means of exporting settings to a skin product. TTW management was still possible, though pressure was mounting to move away from working through the ZMI.

Very little changed for themers in the transition from Plone **2.1** to Plone **2.5**. Generally speaking, themers only had to know where to find certain knobs and switches in the ZMI, modify Plone's default templates, understand the TAL templating language, and work with CSS to make changes to their sites. While modification of basic templates is still a legitimate approach for writing templates and business logic, there is an extra layer of templating involved now, known loosely as "viewlets" and "portlets", that themers need to know about. Filesystem development is now the present state of theming. While it may sound daunting to work on the filesystem, once you get started, it's actually quite liberating.

As of Plone **3**, a system known as **buildout** was introduced, and new visual design and deployment schemes were revealed that enabled developers to create a repeatable, testable development environment that could be shared across teams. For themers, this meant learning how to manage a development environment without necessarily having the programmatic knowledge to do so. Thankfully, Plone's installers mostly take care of this now.

In addition, themers not only had to understand the previously mentioned technologies, but they also had to learn about Zope, including such programmatic jargon as "multi-adapters", "browser layers", "boilerplate", "ZCML", and "GenericSetup profiles". Worse, they needed to learn how to work with Python classes and understand the difference between Zope 2 and Zope 3 templates and how to use them in their skin products. The complexity involved in skinning a site doubled, if not tripled, and the theme development time increased.

The positive side of this was a dramatic increase in the robustness, reusability, and flexibility of theme components. Once learned, the Plone theming framework provides tremendous leverage for web-design professionals. And, much of the Plone theme framework is unit testable, which is amazing! While there is a lot to know, it's worth stating that theming for Plone is a challenging but exhilarating experience. Even better, it has encouraged the Plone community to spend some real time focusing on the theming needs of the Plone users, meaning that the road forward will be easier to tread.

It's also important to mention that community resources are available if you have questions or problems. In particular, themers should be aware of:

- The Plone-users email list that can be found at `http://plone.org/support/lists`
- The Plone forums found at `http://plone.org/support/forums`
- The IRC chat channel that can be accessed at `http://plone.org/support/chat`.

You should always feel that no question is too stupid to ask, and someday you might even be able to answer someone else's questions. It's how most of us have arrived in the community, and we're always happy to have more contributors.

In the meantime, this book attempts to break down the barriers to Plone 3 theming in a way that makes it easier for non-programmers to successfully change the look and feel of their sites. While themers who are accustomed to working with systems such as WordPress and Drupal may find the learning curve challenging, with some effort, it is still possible to generate attractive and robust themes. I hope you enjoy the ride.

Summary

In this chapter, we have learned that:

- The current theming story for Plone is challenging for new themers, and has become even more complex with the newest release of Plone
- Plone is on par with other CMS frameworks in terms of the theming story
- It is possible to create attractive themes in Plone 3.0, but it requires extra effort and some knowledge of programmatic concepts

Next, we will look at the tools necessary for any themer to be productive and then dive deeper into the actual theming process. Onward!