Chapter 1
Yet Another Boring
Introduction

Wherein I describe why a software developer needs a wiki, why I chose DokuWiki over all of the other guys, what I'm going to cover in this book, and how to pronounce 'DokuWiki' the way all the cool kids do.

I had a problem. I'm a freelance software developer based in Milwaukee, WI, which is, to borrow a line from Mal in Josh Whedon's Firefly, at the corner of 'No' and 'Where'. In other words, my customers are everywhere but Milwaukee - East Coast, West Coast, Europe, Far East, why, even a couple in that strange, foreign land known as "Texas."

The Problem

For years, I have developed specifications via an OpenOffice.org document - I'd write something and send it to my contact at the customer. They'd review it, mark it up, perhaps pass it by their folks, and then send it back, hopefully having turned revision marks on so I could see their changes. Then I'd update the document again, send it to them, and so on and so forth. We'd use email and the telephone to flesh out concepts and ideas, and occasionally I'd visit them, but the document we passed back and forth served as the ultimate arbiter of what we'd discussed through other channels. Most of the time I'd also create a prototype of what was being described in the specification, and send them a program.

Worked great.

Eventually, though, I ran into situations that this process didn't address well. There were scenarios where more than one person wanted to work on the spec at the same time. Often, once coding started, and always, once pieces of the application were being delivered, changes to the application were made - meaning the spec was now out of touch with reality. I did my best to keep the document updated, but this was time-consuming and, frankly, no one but me ever looked at the spec after it was accepted.

The Solution

A better vehicle for sharing information was needed. Enter the wiki.

A wiki is a program that resides on a Web site which allows visitors to edit the pages. Changes made to the pages are tracked through a version control system where users can see what older versions of the pages looked like and who made the changes. It's a wonderful tool for collaborating on the creation of documents as well as an excellent repository for knowledge acquired by a group of people. The Wikipedia, www.wikipedia.org, is probably the most well-known wiki.

With the increasing popularity of wikis, people use the term 'wiki' to both define the software as well as the Web site it runs. In this book, the distinction - when I'm talking about the DokuWiki software and when I'm talking about the wiki - should be clear.

Choosing

A wiki is a type of program. Just as there are many brands of word processors - such as Microsoft Word and OpenOffice.org - there are many brands of wikis, and choosing the right one can be difficult. I'd taken a look at the Wiki Matrix (www.wikimatrix.org) and gotten overwhelmed quickly. It currently has 120 wikis to compare and contrast. I had several non-negotiable requirements:

1. Installable on a shared Web host
   - I didn't want to have to deal with hosting on a local box or require esoteric functionality not always provided with inexpensive hosting accounts.

2. Restricted access
   - While the general concept of a 'wiki' is public access, for this use, I had to be able to restrict access to a limited number of users, in a way that my customers would feel comfortable with - since it was their information that was going on the wiki.

3. Either PHP or Python based
   - I didn't want to learn Java or C++ or Lisp.

Additionally, I didn't want to deal with a 'toy' wiki that would be good for a couple of users on a local machine, but wouldn't scale well for thousands of pages and perhaps a dozen or more users. I didn't see myself using a wiki for scope larger than that, but didn't want to unnecessarily limit myself either.

Using the customized search tool on the WikiMatrix site, I'd eventually narrowed my list down to MoinMoin and MediaWiki. MoinMoin is Python based while MediaWiki uses PHP. They both can use MySQL as the backend database to store content.

Once I started investigating the tools, troubles began. First, MoinMoin's installation involved more work than I wanted to invest. To be fair, part of this complexity is because MoinMoin is really flexible. But the questions come fast and furious - and without enough background info to make informed decisions. CGI or Standalone? Apache on Linux? Mac?

Windows? IIS? WebLogic? Apache with ModWSGI? Or ModPython? Twisted Web? FastCGI on Lighttpd? And worse, once made those decisions, then the real work began.

For instance, suppose you choose Apache on Linux. Your first step was to check that Python was installed. The correct version, of course. Then download and install M-M via the Command Line, choosing either the default - or another specific location. Debug. Test installation. THEN.... 1. Create a wiki instance. 2. Install moin.cgi 3. Configure moin.cgi 4. Configure Apache 5. Configure MoinMoin 6. Test the wiki. This was a lot of work... too much work without sufficient motivation - more motivation than I had.

So I jumped over to MediaWiki. Compare the Moin-Moin installation To Do list to the following steps required to install MediaWiki on a Linux/Apache box:

```
root# yum install mediawiki
```

Yes, a bit easier than MoinMoin. This didn't solve my requirement of installing on a shared Web host - I was pretty sure, given I didn't have root access, that it wouldn't be quite that easy. Still, I was up and running with MediaWiki in about five minutes. I also really liked the basic look of MediaWiki, and the documentation was very helpful.

Where MediaWiki fell down was restricting access to a selected group of users. They're very upfront about it - MediaWiki is for public collaboration. However, that means their tools for using it as a private wiki are somewhat limited and convoluted to use. For example, in the "Preventing Access" topic, they say "Schools and other institutions may want to block all edits not from a few specified IP address ranges. To do so, all other ranges must be blocked. The only way to do this at present without modifying the code is to go to Special:Blockip and systematically block every one of the 65,536 CIDR Class B ranges that you don't want to be able to edit."

Ugh.

And the Winner Is...

While I was wrestling with this, I stumbled upon (not literally!) DokuWiki, billed as "aimed at creating documentation of any kind. It is targeted at developer teams, workgroups and small companies." Bingo!

It's easy to install locally or on a shared Web host: it took five minutes to get a five page wiki up and running, and another five to restrict access both via IP and through a login mechanism. It's also available for installation directly from popular distributions like Ubuntu.

It's written in PHP.

There are dozens of templates and hundreds of plugins that customize and extend the base engine.

And although its data store is simple text files, the engine is built to be able to handle tens of thousands of pages and access by dozens of users. While this may not be suitable for the specifications for an airline ticketing system or a replacement for the U.S. Internal Revenue Service tax collection software, it's fine for the 500 to 2,000 hour-long systems I write.

In fact, Andreas Gohr, the author, specifically used flat files instead of a database for a reason. He says, "DokuWiki was designed to hold documentation of all kinds. So what do you do when your database or your web server is down and the documentation to bring it up again is only accessible through this web server or database? Yikes! So DokuWiki stores all this vital information as plain text which is always readable, even if you only have some tape backups left of your beloved server."

Why a Book?

As I worked with DokuWiki, I figured it might be helpful if I wrote down what I did and share it with others. At this point, some may point out, "Ummm, there is already a wiki for DokuWiki - you should add your material there!" Others may offer, "Well, you're still talking about a wiki? If you feel the existing DokuWiki wiki is lacking, why not create your own wiki Web site?"

Fair enough.

First of all, let me make clear that I think the existing DokuWiki wiki is excellent. It really is - and time reading through it is time well spent. Considering all of the software I've used over the years, the DokuWiki wiki is way better than most - clearly laid out, easy to use, easy to find things, chock full of information.

But it's also good to have access to a different point of view. One of the first software manuals I used was the dBASE II documentation. In 1981. While it was head and shoulders above anything else at the time, it still suffered from tunnel vision. They would use the same example in multiple help topics - an approach that failed miserably when (1) you didn't understand that example, or (2) there was an error in the example. Furthermore, not everyone learns the same way. Thus, I would argue that it's much better to have a couple different examples to provide varying viewpoints, and so, when set next to the DokuWiki wiki, this book acts as an additional viewpoint.

Second, I wanted to support the DokuWiki community. For some odd reason, having a book on the shelf (actually, that should be 'a book on Amazon', since bookstores these days are too full of music and stationary and stuffed animals to be bothered with carrying, of all things, books) adds legitimacy to a piece of software. It means that another company felt strongly enough about an application that they devoted resources to producing a book for it.

And third, there are those folks who prefer to read material while off-line. You know - in an armchair with a glass of cold milk and a couple of chocolate chip cookies. This is true more for 'concept' books than books that are simply syntax reference books - and this

DokuWiki book has buckets of concept-oriented material that is perfect for reading in a hammock on a lazy Sunday afternoon with nary a computer in sight.

The downside to a book is that once it hits the printer... well, it's more difficult to change it than, oh, say, a wiki. I'm not going to sweat this point. Since this book was sent out into the wild shortly after the new February 2009 release was released, the book will be fresh for a while. And many of the concepts will be true even as subsequent releases are produced.

So this is my guide to you, a fellow software developer, on how to use DokuWiki to collaborate on software specifications and other related documents.

Let's do it!

The Road Ahead

Here's the setting for this book: picture a Saturday afternoon in winter. You're in the upper Midwest - it's been snowing since October. So the sky is bright blue, there's 3 1/2 foot of snow on the ground outside, cars are doing more sliding than driving down the minimally maintained village streets and the occasional power walkers are motoring past in full cold-weather gear, mittens and hats pulled on tight.

You've come over to learn about this DokuWiki software that I've been raving about on the local user group's mailing list. We're settled in my office for the afternoon, sitting in front of a pair of widescreen monitors, keyboards and mice at the ready, hot chocolate steaming away in mugs, and the Cream reunion concert at Royal Albert Hall playing softly in the background. I'm going to give you the nickel tour of how to install and use DW, and once you're comfortable with the basics, we'll discuss advanced DW administration and usage, tips and tricks. We might even try our hand at solving all of the world's problems before the sun disappears from office's western window.

This book is the written version of what will transpire this afternoon.

Part I - Getting Started and into Production

This book is roughly broken into three sections. The first is the technical instruction - how to use DokuWiki - getting a basic wiki up and running on a live Web server. Not a lot of frills - the basics plus some essential add-ons.

First, we'll install DW on a local machine (as opposed to a production server or a third-party host), and discuss the various installation options in detail. Then we'll get into building some pages in the wiki, showing you how to edit and format pages, and how to organize a wiki, with a nod given to our specific purpose of building software specifications. And with that, we'll look at some of the major features of DokuWiki - handling changes, revisions, and drafts.

Now that we're comfortable with the wiki in general and DokuWiki in particularly, it's time to configure and customize. First comes basic configuration, discussing how

configuration works and what things you might want to change immediately. Then, we'll customize DW, installing templates and plugins to make DokuWiki do what you need it to do.

And now it's time to deploy to a production server - either a box under your control or a shared host. We now have a live, working wiki.

Part II - Advanced Usage and Administration

The second section of this book are made up of standalone chapters; each addressing a specific topic that can be read by itself.

First we'll look at what's going on under the hood - investigating all of those folders and files in a wiki, and open up the top few PHP scripts to see how DokuWiki gets started and where it heads.

Then we'll take a long, hard look at security, starting with Access Control Lists (ACLs), then investigating user management, login restrictions and various scenarios you might want to emulate, and then finally consider the use of .htaccess to restrict access to the Web server itself.

Seeing as this is 2009, it's very possible that your wiki doesn't live by itself, but rather, in an environment where it has to play nicely with others. So we'll cover how to let users authenticate against a backend like MySQL instead of requiring them to maintain yet another set of credentials just for DokuWiki. Finally, I'll show you how to set up DokuWiki to send administrative email and notifications about content changes.

The last chapter in this section is a synopsis - a cookbook recipe that summarizes these steps so you can set up your next DokuWiki quickly, and without missing anything.

Part III - About Collaborative Software Development Documentation

So far we've just talked about the wiki itself. Isn't this about software development documentation? Well, yes. While I've endeavored to make the examples up to this point relevant specifically to the task at hand, they've still been more of an afterthought than the main focus. Now let's talk about writing, er, collaborating on software document, using this brand new widget in our utility belt as our primary tool.

What we've set up is the infrastructure that's well suited for developer documentation. The trouble is that, unlike, say, accounting, where there's a generally accepted method to doing a balance sheet, there isn't a single 'generally accepted practices' for software documentation. Everyone does it their own way. What I'll discuss here are some pieces that you might find useful.

How to Pronounce (and Abbreviate) "DokuWiki"

DokuWiki mailing list member Jon Schneider posted a question about pronouncing DokuWiki a while back. His reasoning was that he was about to sell it to other members of his team, and he wanted to get off on the right foot by pronouncing it correctly.

Com'on, 'fess up. I bet a few of you have been wondering the same thing. "DOCK-you"? Or "do-COO"? Or "DOE-kuh"? Or maybe some combination in between?

The correct answer is "DOE-coo". "DOE" as in "donut" and 'coo' as in 'cool'. Here's why.

Andi says "Doku is short for "Dokumentation" – the German word for documentation (and pronounced DOE-coo...) I'm from Germany so this was the very first working title..." He adds, "Most Americans tend to speak it like DocuWiki (like in document)." So now you know.

Before we get started, a couple of notes about what's inside this book, and what additional resources are available.

1. First of all, please see "How to download the files" for instructions on how to grab files from our website. The source code consists of one big ZIP file that consists of separate ZIP files for each chapter. Check out the "README.TXT" files inside each chapter's ZIP file for details.

2. The screen shots and URLs throughout this book were current as of the writing and/or printing. Depending on when you are reading this, they might have changed. ("Might have"? Of course they have!) See the errata on our Web site, http://www.hentzenwerke.com, for updates as we find them.

3. I've written a number of articles that you might find useful; they can be found under the "Resources" link on our website.

That's all for now. Enjoy the ride.

Conclusion/Summary

It's 2009. (Or maybe 2010, or 2011...) There's no excuse for even the most casual of software developers not to have a full suite of support tools available, and a mechanism for collaborating over documents is one of them. DokuWiki fits the bill perfectly - full featured, Open Source, easy to install, use and administer, and... inexpensive. Let's install and start using it.

> Updates and corrections to this chapter can be found on Hentzenwerke's Web site, www.hentzenwerke.com. Click "Catalog" and navigate to the page for this book.