

# What on Earth is... WIKI?

For many people that stumble accross this cornerstone of the Open Source ideal, the idea of letting strangers edit their work is a bit scary. **Andy Channelle** allays our fears...

## >> What on Earth is *Wiki*? It sounds like some sort of long-lost religion or art movement...

See, just one question in and already you've made a fundamental error. What we are covering here is the *WikiWikiWeb*, an alternative hypertext authoring tool created to allow collaborative development over a network. Some would argue it is actually the realisation of what the Internet promised in the first place; but then different people would probably say the exact same thing about blogging bringing the prospect of personal publishing to the masses. So now you've made a devastating *faux pas*, don't you just wish you could hit some sort of magical link and start all over again?

## >> Ah ha! I know what you're doing. Isn't some sort of 'magical link' the Big Idea behind *WikiWikiWeb* sites?

It is; and I think we'll dispense with the full name and just stick to *Wiki* for the moment, it's easier. Anyway, *Wiki* sites are indeed editable. By anyone. Beyond the normal content, which is mostly text-based, *Wiki* pages tend to have a series of set links covering things like revision history, recent changes and, of course, the Edit button. In theory this means anyone accessing the site can hit the link and make their own additions or deletions to the page. For instance, you're reading away on a *Wiki* site and you notice that the author claims the great fire and the plague hit London in 1665. Of course this is wrong, so you hit the edit link and change it to 1666. The old version gets backed up – you've specified that this is a big change – and your new version is the one the next visitor will see. The next visitor, in turn, may think that it's important to note that the fire started on Sunday September 2nd so they click the edit link and add their details. The next visitor may add geographical detail as to where the fire started, and so on.

## >> And the next visitor may like to add that Kylie Minogue started the fire by osmosis before single-handedly ruining pop music for the 21st Century. What stops people demolishing everyone else's work by defacing the page? It sounds like a recipe for chaos.

True enough, as the *Spider-Man* movies and comics say, power and responsibility go hand-in-hand. The answer is that there is nothing within the system to prevent a malicious user from going on the rampage. In the original *Wiki*, editing a page so that its only content reads 'delete' will actually delete the page.

This much control is regarded as a reason why the system works. In practice though, such damage will be picked up quite quickly – especially on a popular *Wikisite* – and can (sometimes) be rectified by rolling back to a previous version. As with any documents, it's wise to back up your *Wiki* on a regular basis to ensure you can recover from any disaster, whether it is machine- or man-created. An entry called *Why Wiki Works* claims that "to make an impact on *Wiki*, you have to generate real content... anyone can play, but only good players last".

If you're really paranoid, some of the available *Wikis* let you password protect content or even make it non-editable, but then you have to ask whether your site is a genuine *Wiki* or just simulacra.

There is a bigger issue though: when you contribute to a *Wiki* you give your permission for that contribution to be re-edited by strangers and the end result may not look anything like the original product. We don't want to get too deep, but this is the philosopher's axe in real life. For instance, two years ago, LXF began an entry in the *Wikipedia* ([http://www.wikipedia.org/wiki/List\\_of\\_British\\_monarchs](http://www.wikipedia.org/wiki/List_of_British_monarchs)) on the kings and queens of England. The initial page consisted of a simple chronological list of monarchs, but it has now acquired a lot more information. The list has been divided into the various 'houses' and most names have spawned a

page (or series of pages) of their own. The question is: to what extent can *LXF* be said to have 'authored' this page? Does anything of the original work still exist?

Open Source developers may have wrestled with this new concept of authorship and ownership, but for many *Wiki* may be their first exposure to genuine collaborative development, and if they're at all precious about their prose, it might also be their last. Having someone else edit your work can be quite disturbing.

Many users who have stumbled across *Wiki* think it will be the perfect tool for blogging and this has inspired a long-running debate within the original *Wiki* at <http://c2.com/cgi/wiki> on the concept of what has been termed *Wiki Squatting*, using a *Wiki* as your homepage. One contributor wrote: "A *Wiki* squatter puts something on *Wiki* and calls it 'his'. For example, you feel very strange editing the diary page because it seems very personal". *Wiki*, the argument goes, is for everybody.

### ➤ It's an intriguing idea. Who came up with it?

Ward Cunningham is credited with both the idea and the first *Wiki* implementation in 1995 – though Cunningham himself traces its origins back to a *Hypercard* stack he wrote at the tail end of the Eighties – at the Portland Pattern Repository which publishes *Pattern Languages* and related information. Pages were (and still are in the original *Wiki*) served up by a CGI script written in Perl, and as time passed new features were added.

The name, if you were wondering, comes from the Hawaiian phrase for *Quick*, which was chosen by Cunningham to evoke the speed at which pages or sites could be created and modified. He hated the phrase *QuickWeb*. The first *Wiki* clone was written by Patrick Mueller in REXX for OS/2.

### ➤ Wiki clone? Is there is more than one Wiki then?

Er, yes and no. There is only one *Wiki*, but there is an awful lot of *Wiki* Engines created in everything from ASP to VisualBasic. The most popular languages for *Wiki* serving are Perl, Java and PHP, but there are also implementations in Awk, Python, Lisp and even Erlang (among others) for people that prefer those languages. One of the most popular engines is *UseModWiki* (<http://c2.com/cgi/wiki?UseModWiki>) which is an easy to install and configure script written in "100 per cent pure Perl". The script itself comes in at a hefty 2,300 lines, but has a comprehensive featureset that makes it suitable for even the largest projects.

And at the other end of the scale is *TinyWiki* (<http://c2.com/cgi/wiki?TinyWiki>) which ➤





# WhatOnEarthWiki

◀ manages to do its work in a suprisingly compact 35 lines of Perl code. Both *TinyWiki* and *UseModWiki* are available under the GPL.

## ▶▶ Am I going to have to learn a whole new markup language to make any sense of Wiki pages?

In the main, writing *Wiki* is simple. You'll probably have a few hiccups at first, especially if you're trialling various Engines, but within a few minutes you get used to whichever system you're currently using. As it is primarily a text-based medium, design is limited which puts the emphasis on content rather than style. You could argue that this is one of the best reasons to use it, especially in something like a documentation project where clarity of content and presentation is more important than colour, branding or some other distracting visual noise. In a sense it is a little like the textual origins of the Internet, but with that all-important element of completely open access for all visitors.

While each *Wiki* Engine has its own style of tagging, most tend to follow Ward Cunningham's original scheme. For instance, creating a link in a document – perhaps the most fundamental formatting element – is most often accomplished by writing two capitalised words together without a space, i.e. `LinuxFormat`. This is called a *WikiName*. And here comes the clever part: if you type the *WikiName* of an existing document within your *Wiki*, the phrase will become an underlined link to the relevant page in the traditional fashion. However, if the page does not yet exist, the link becomes tentative and will have an underlined question mark at the end of it; clicking on the question mark creates the new page and opens up an edit screen to add some content.

Beyond the bounds of your particular *Wiki* there is also a scheme for building an *Interwiki* which is a system spanning a number of servers. It's a pretty complex operation, but for all its complexity though, it's a remarkable system to use, and – I might add – is surprisingly addictive.

Beyond the magical links, there are a number of formatting tags to cover styles such as bold, italic or bulleted text as well as provision for creating standard hyperlinks. These are designed to be simple and memorable. See the *Some basic formatting* table opposite for a quick introduction to these tags.

## ▶▶ Obviously I'll need some sort of web server, but what are the general requirements for building my own Wiki?

The main prerequisite is indeed a web server and most are extremely happy running on *Apache*, which is generally available as part of most common distributions. If you don't have it, you'll have to download from [www.apache.org](http://www.apache.org) and install it manually. Next up is a database to house your data; most of the *Wiki* engines available from <http://c2.com/cgi/wiki?WikiEngines> tend to favour *MySQL*, but there are also implementations for *PostgreSQL*, *MS SQL Server* and *Access*. And finally you'll need to decide which type of *Wiki* you'll be installing as this may entail additional technology. Opting for the JSP *Wiki Very Quick Wiki*, for instance, will involve installing a J2EE web container such as *Tomcat* (for *Apache*). See this last month's and this month's *Linux Pro* for more on *Tomcat*.

The most common technology for handling the management of the *Wiki* is Perl, so if you've not got that installed (open a terminal and type **which perl** to find out if you have it and where it is) you'll have to get it from your distribution disk or download it from [www.perl.com](http://www.perl.com). You'll need to know its location to adapt the *Wiki* script you intend to use. The usual location is `/usr/bin/perl`. Similarly, if you haven't got PHP (type **which php**) and you're using a PHP implementation you'll need to download it from [www.php.net/downloads/](http://www.php.net/downloads/) and install before getting to work on configuring the actual *Wiki* script. A setup like this should (hardware notwithstanding) handle all you can throw at it.

Once the infrastructure is established you can download your chosen *Wiki* script and install it in *Apache's* `cgi-bin` directory.

Of course I'm simplifying here – there's never enough space – but most of the *Wiki* scripts available will have a fairly comprehensive set of installation instructions and, once you've actually dealt with the server basics, what's involved in adding a *Wiki* is child's play.

## ▶▶ Well, at least all these things are available in a standard distribution. Isn't there an easier way?

There's always an easier way, though if you envisage creating an enormous *Wiki*, you will definitely need a database backend and the security and stability under pressure that *Apache* provides. If, however, your needs are more rudimentary you can ditch most of this and set yourself up in less than ten minutes. Firstly instead of *Apache*, try *Abyss Webserver* from Aprelium ([www.aprelium.com](http://www.aprelium.com)) this is a free (but not GPL) webserver that runs on Linux or Windows, features a very simple web interface for its various configuration options and is both PHP and Perl able. Using the Linux version is simply a matter of unpacking, cd-ing to the correct directory and typing `./abyssws`, though you'll need to enable support for your chosen CGI script (see the Aprelium website for more info).

In order to dispense with the need for a database, download the PHP *Wiki*

## Some basic formatting

### Simple and memorable

Here are a few of the main formatting elements that you'll come across using the original Wiki that's available at <http://c2.com/gci/wiki>. Be aware though, that other implementations of the Wiki scheme may have different ideas, some will even take standard HTML, which is useful when copying information across from other sources.

--	Horizontal line
<tab>*	Bulleted list
1. 2. etc	Numbered list
<space>(at the start of a paragraph)	Monospaced font
" "	Emphasis, usually italic
"" "" (triple single quotes)	Extra emphasis, usually bold
http:, ftp:, mailto: etc	Automatic link to external website, ftp server, email address
ISBN<10 digit number>	In Cunningham's Wiki and others, an ISBN number will link to the Amazon.com page for the book with that number

Processor (PWP) from <http://www.net-assistant.de/wiki/static/StartPage.html> and unzip it into the .htdocs (or the specified root directory) of the webserver and open localhost in a browser. PWP comes with a pre-made index.html, uses a flat file backend and each page you create is stored as a standard text file. Despite being simple to set up, PWP uses CSS to define formatting which means it's quite simple to integrate the visual elements into a regular website. The downside is that, as there is no database serving up pages, it is realistically limited to 300-500 pages. This is ideal if you're looking at a corporate project (say documentation development) but is not going to scale up too well.

### Ah ha, that would be fine if I was in anyway technical, but I'm new to this. Anything easier?

So far, the simplest way I've found of building your own Wiki server is *EddiesWiki* (<http://www.tinyted.net/eddie/wiki/>) which is an 85K download complete with HTTP server and no other requirements, it is genuinely an 'unzip and go' application. The downside is that currently there is no Linux version; it was written in Visual C++ and currently compiles on Visual C++ 6.0 and [Windows] GCC 2.95-2. However the download includes source code, so it's possible that someone could sort out a Linux version. Anyone want to try?

Again *EddiesWiki*, which is Postcardware, uses a flat file system for storing pages so is only suitable for small projects. It makes a perfect personal Wikiserver though.

### There's still a snag. I'm in no position to run my own accessible webserver and my ISP won't let me run Perl or PHP scripts on my personal web space unless I cough up even more money than I do already. Can I do it without having to manage a server?

The simplest (and it doesn't come any simpler than this) way of starting your own Wiki is to sign up with a *Wikifarm*. This has the benefit of off-loading all the

hassle of configuring and maintaining servers onto someone else and, in most cases, it is free for non-commercial uses. You give up some freedoms and you are at the mercy (sort of, potentially) a commercial entity, but really there aren't many cons to this route of WikiOwnership (that markup scheme is habitual you know).

Visit [www.seedwiki.com](http://www.seedwiki.com), click 'Start Your Free Wiki', choose a name, a Wiki Type (so Wikis of similar types can be corralled into communities) and language, set a code word so you can manage the backend of the site and hit 'Save'. And that's it. You can now start editing and creating pages. The good thing about *Seedwiki* is that you can format pages using standard-ish HTML tags and widgets that can, for example, display the number of visitors a Wiki page (or entire site) has had, send out a mail whenever a page is updated or include the contents of another page.

*Seedwiki* hosts some 40,000 pages ranging from a serious discussion of Wiki as a means for publishing to a forum for participants in Open University degrees, to more light-hearted and esoteric subjects. It's a nice place to hang out and edit.

### So I've set up my Wiki, clicked a few edit buttons, written a few swear words for fun. Now what?

There's plenty you can do. Wiki's text bias and incredibly easy collaborative features makes it perfect for the creation of documentation, and this is seen in the growing popularity of Wiki-based HOWTOs. Think about it: a developer creates an application and writes some elemental documentation. Users begin using the application (or other developers begin adding to it) and, in search of help, come across the HOWTO Wiki. Instead of just consuming the instruction they can add their own experiences, tips and questions which may then be answered by the developer or subsequent visitors.

As more people visit, the documentation becomes more comprehensive and the world becomes a better place... (Excuse the rampant idealism!)

It's also finding favour among commercial developers and testers who can code and document side-by-side without having to worry about the complexities of versioning in the most popular word processing packages.

Or how about really blurring the boundaries of author and reader by starting your own Wiki-novel or fanzine? If you can put it into text, it can be Wiki-fied, though of course, you not only have to submit to other people's opinions of your work, but also let them try to do better.

### If it's so good, why hasn't it been used for any thing big?

Apart from an enormous number of HOWTOs and the original *WikiWikiWebsite*, the most exciting Wiki project (and, some would argue, the ultimate use of the Internet) is *Wikipedia* ([www.wikipedia.com](http://www.wikipedia.com)), an attempt to create a free and open encyclopaedia based squarely on the concepts of peer review and freedom of information in the truest sense.

*Wikipedia* was launched in January 2001 and in just two years has managed to accrue over 100,000 English language articles and some 37,000 articles in other languages. As it's reputation as a haven of freedom of expression and knowledge has spread it has attracted the attention of academics, experts and researchers keen to expand its content. Users are no longer just consumers of information, they are creators, editors, owners.

Even better than an open encyclopaedia are the prospects of a Wiki news site (there is a nascent news community on *Wikipedia*) offering a diverse range views on current affairs or highlighting news that the mainstream media – for whatever – reason can't or won't cover and a *Wiktionary* which would be able to react to neologisms and changes in usage much faster than printed works.

And contrary to accepted wisdom, being subject to the whims and wishes of biased individuals often leads to some sort of equilibrium in articles if the common-sense guidelines are adhered to. With enough interested participants this is the very definition of seeing things from all sides.

### Hold on, Mr Boundless-Enthusiasm, it can't all be flowers and romance. What are the downsides?

To be honest, the downsides are exactly the same as the benefits. Some people won't get the concept and will plough through your ideas and effort like an ill-mannered tank commander. If subjecting your words to the potential of such an onslaught scares you, then you will find more reward in plain old static websites and should probably steer clear of Wiki.

And if you're using a Wiki for the first time, just remember to tread lightly, don't rant, and don't 'over-write' without thinking about it first. The page's progenitor may have spent hours or days honing his/her prose to the knife-sharp point you're just about to hammer flat with a throw-away phrase. 