organising sysadmin documentation

Jon Dowland
Information Systems and Services
Newcastle University

Overview

- History and first attempt
- Findings
- Next generation
- Conclusion

History and first attempt

Introduction

- Newcastle University
    - ~4,800 staff; ~20,000 students
- Information Systems and Services
    - ~160 people

- Diverse ecosystem
- 50+ years of computing

ISS UNIX Team

Joined 2 years ago
5-strong team
widely varying age and experience

system administration and infrastructure
higher-level (user-facing) services
legacy systems

sister teams: networks; windows; SAP; etc.

Inherited situation

directory full of text files
categorization by directory
lots of out-of-date information
lots of undocumented stuff

What do we keep in there?

- Procedure documents
- HOWTOs

Existing experience

- wikipedia editing
- webteam-based mediawiki hosting service
- version control (CVS, SVN)
- On campus Web/SSO system

First-cut requirements

- import existing data
- support reviewing material
- version control

- web-based
- multimedia
- integrated with SSO

Mediawiki

- **<http://www.mediawiki.org/>**
- powers wikipedia
- LAMP
- GPL (v2)
- CMS
- On-campus hosting service

Mediawiki

SSO integration

Shibboleth SSO software

<http://shibboleth.internet2.edu/>

mediawiki Shibboleth plugin

<**http://www.mediawiki.org/wiki/Extension:Shibboleth_Authentication**>

Apache configured with mod_shibd

```
AuthType shibboleth
ShibRequireSession On
require eppn someuser@ncl.ac.uk
…
```

Mediawiki markup

- simple
- proprietary

```
the '''quick''' brown ''fox''

* jumps over

 the [[lazy]]

[http://example.com/ dog]
```

the **quick** brown *fox*

- jumps over

the lazy

dog

Categories

a tagging mechanism

adding the following to the 'Sandbox' page

[[Category: Example]]

yields the page 'Category:Example'

Articles in category "Example"

There is one article in this category.

S

- Sandbox

Importing data

- one page per old file
- each page is categorized with
  - "imported from fireman FAQ"
  - "articles to review"

- Imported files on a per-parent-directory basis
  - vary categories for each directory

WWW::MediaWiki

perl module to manipulate mediawikis
VCS-like command-line tool front-end

```
$ mvs login -d (host) -u (user) -p (pass) -w (path)
```

Create a new page, or checkout an existing one:

```
$ mvs update Somepage.wiki
```

make and commit a change:

```
$ echo hello world >> Somepage.wiki
$ mvs update Somepage.wiki
```

Importing: categorisation

```
$ echo '[[Category: articles to review]]' \
    >> Somepage.wiki

$ echo '[[Category: Imported from fireman FAQ]]' \
    >> Somepage.wiki
```

Side-stepping the SSO

A separate apache vhost for the script
Separate mediawiki install

    different URI path

    the same DB tables

```
<Location /unix/localwiki>
        Order deny,allow
        Deny from all
        Allow from localhost
</Location>
```

Findings

Findings

- in use for 2 years
- Number of articles doubled
- removed **some** older material

- scope has grown

What do we now keep in there?

- *Procedures*
- *HOWTOs*
- **meeting notes**
- **project outlines**
- **TODO lists**
- **documents for other teams**

Audience

- originally, just us
- now other teams too
  - meeting minutes
  - cross-team procedural documents

Access control

- SSO layer gives us "full on, full off" access
- mediawiki offers tiered write-access control
- everyone has full read access

Types of contributor

willing / reluctant

structural / brain dump

novice / experienced

"A major killer of wiki projects is the imposition of too much structure during the early stages"

The Practise of System and Network Administration, p254

Editor role

Subset of team take on responsibility for maintaining the wiki

The remainder read/brain dump/suggest

Accessing/organising data

- "search, don't sort"
- Three Letter Acronyms

High Availability

- failure points for old system:
  - filesystem
  - network

- failure points for new system
  - web server
  - database server
  - authentication paths
  - network

Mediawiki UI revisited

Revised requirements

- *Import existing data*
- *support reviewing material*
- *version control*
- *online*
- *multimedia*
- *integrated with SSO*

- **High availability**
- **Search**
- **get data out again**

Next generation

Candidates

- freshmeat.net: 209 matches for 'wiki'
- sourceforge.net: 893 matches

...other "document management" / CMS software, besides wikis
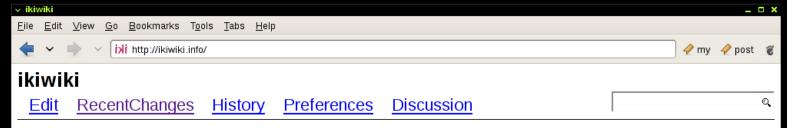
- sharepoint
- ikiwiki

Sharepoint

- DB backend
- proprietary
- UI usability concerns
- ...

IkiWiki

- **<http://ikiwiki.info/>**
- perl-based
- open source
- static rendering
- real VCS backend
- plugin support

IkiWiki

http://ikiwiki.info/   🖉 my   🖉 post

ikiwiki

Edit   RecentChanges   History   Preferences   Discussion

Ikiwiki is a **wiki compiler**. It converts wiki pages into HTML pages suitable for publishing on a website. Ikiwiki stores pages and history in a revision control system such as Subversion or Git. There are many other features, including support for blogging, as well as a large array of plugins.

using ikiwiki

Setup has a tutorial for setting up ikiwiki, or you can read the man page. There are some examples of things you can do with ikiwiki, and some tips.

All wikis are supposed to have a SandBox, so this one does too.

This site generally runs the latest release of ikiwiki; currently, it runs ikiwiki 2.68.

developer resources

The RoadMap describes where the project is going. Bugs, TODO items, wishlist items, and patches can be submitted and tracked using this wiki.

ikiwiki is developed by Joey and many contributors, and is FreeSoftware.

___

Links: TourBusStop todo/toplevel index

Last edited 1 month and 24 days ago

ikiwiki

- News
- Download
- Setup
- Security
- Users
- SiteMap
- Contact

The basics

a command-line tool
a compiler
most of the code in a library

The basics

```
$ find .
.
./src
./src/somefile.extension
./src/foo.mdwn
$ ikiwiki src dest
$ find dest
...
dest/somefile.extension
...
dest/foo/index.html
...
```

The basics

modify behaviour with command-line arguments

```
$ ikiwiki --no-usedirs ./src ./dest
$ find ./dest
...
./dest/somefile.extension
...
./dest/foo.html
...
```

or a configuration file

```
$ ikiwiki --setup config.setup ./src ./dest
```

Plugins

lots of hooks for plugins (24+)
perl native modules
external plugins via XML RPC

Version control integration

via plugins
git; bzr; subversion; monotone; mercurial; tla

helper tool to migrate setup a repository

```
$ ikiwiki-makerepo <vcstype> \
    <srcdir> <repository>
```

then adjust your .setup file accordingly

Web integration

A CGI script for web-edits
VCS hooks for refreshing as a result

setuid (to owner of srcdir/repository)

urgh
small, easily-audited code

Syntax

- via plugins
- "default" is markdown

  o simple
  o not proprietary

```
Here's a paragraph.

Here's another one with *emphasised* text.

# Header

## Subheader

> This is a blockquote.
>
> This is the first level of quoting.
>
> > This is nested blockquote.
>
> Back to the first level.
```

Here's a paragraph.

Here's another one with *emphasised* text.

Header

Subheader

> This is a blockquote.
>
> This is the first level of quoting.
>
> > This is nested blockquote.
>
> Back to the first level.

IkiWiki pagespec

page specification

booleans
```
foo or bar or baz
```

negations
```
foo* and !*baz
```

functions
```
link(somepage)
```
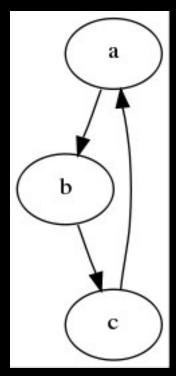
Useful plugins

Mediawiki syntax plugin

<http://u32.net/Mediawiki_Plugin/>

some limitations

useful plugins: graphviz

inline graphs

```
[[!graph src="a->b ; b->c; c->a"]]
```

useful plugins: img

- scaled images
- link to fullsize
- uses ikiwiki pagespecs

```
[[!img foo.png alt="alt text"
size="100x400"]]
```

useful plugins: inline and map

```
[[!inline pages="blog/*
    and !*/Discussion"]]
```

copies the matching pages into the page

```
[[!map pages="* and
    !blog/* and !*/Discussion"]]
```

creates a bullet-list of links to matching pages

useful plugins: toc

table of contents
derived from headers in the page

```
[[!toc levels=2]]

# hello

## more

## headers

### yes

# indeed
```

I. **hello**
    1. **more**
    2. **headers**
II. **indeed**

hello

more

headers

yes

indeed

useful plugins: template

Uses HTML::Template perl module
define a home for templates in the wiki

templates/example.mdwn:

```
	hello <TMPL_VAR name>
```
index.mdwn:
```
	: [[!template id=example name="Jon"]] ;
```

result:
```
	: hello Jon ;
```

useful plugins: tag

essentially a hidden wikilink
add a tag in a page

```
[[!tag foo bar baz]]
```

list pages which are tagged foo:

```
[[!map pages="link(foo)"]]
```

useful plugins: pagecount

how many pages match a pagespec?

```
[[!pagecount pages="* and
    !*/Discussion"]]
```

Possible migration strategy

- old/new running in tandem
- automatic import from one to the other

- don't rely on mediawiki syntax forever

Migrating data

- One approach:
- <http://u32.net/Mediawiki_Conversion/>
  - o Scott Bronson's excellent guide

Procedure involves

- Hand-extracting a list of pages (urgh)
- exporting data in an XML container
- various scripts removing mediawiki-isms
- Mediawiki syntax plugin

Another approach

- Parse HTML to extract page names (urgh)
  - get_pagenames.py script

- use WWW::MediaWiki to fetch page contents
- various scripts removing mediawiki-isms

- another: mine the DB backend

get_pagenames.py

```python
from xml.dom.minidom import parse, \
    parseString

dom = parse('Special:Allpages')
tables = dom.getElementsByTagName("table")
pagetable = tables[-1]
anchors = \
    pagetable.getElementsByTagName("a")
for a in anchors:
    print a.firstChild.toxml()
```

Tackling high-availability

static copies on separate machines

Tackling access control

Our needs are for tiered read control selectively publish static copies of sub-sections?

IkiWiki drawbacks

- search
  - xapian plugin
  - google site search plugin
- Speed

  mod_perl?

Comparison

Comparison

Edit **RecentChanges** **Preferences** **?**Discussion

?right

Welcome to the ISS Unix Team Wiki! Find out more about current work going on the wiki at **Todo**.

- **TeamContacts**

Subsections

All the content on the wiki is (or should be) organised into categories. The top-level categories are:

- **Category:Services** - services offered by the UNIX team
- **Category:Projects** - projects we are undertaking
- **Category:Howto** - how to achieve various things

older stuff

- **Projects** - stuff that we're doing/thinking about doing
- **Maintenance** - Sun maintenance agreements etc
- **Random Thoughts** - ramblings on possible developments/changes
- **To Do List** - short term to do list
- **?**Category:Root - a category for "top-level" topics

Useful external links

- **UNIX team shared area**
- monitoring:
  - *Nagios*

Conclusions

Sysadmin specifics

- high-availability
- data fidelity
- "hackability"

Software specifics

- "text files in a dir" inadequate
- mediawiki/LAMP hard to HA
- ikiwiki promising

- not just a software problem

People

uptake crucial

References

The Practise of Network and Systems Administration
Thomas A. Limoncelli, Christina J. Hogan
Addison-Wesley Professional
ISBN 0-201-70271-1

Time Management for System Administrators
Thomas A. Limoncelli
O'Reilly Press
ISBN 0-596-00783-3

Slides, etc.

http://www.staff.ncl.ac.uk/jon.dowland/unix/docs/

jon.dowland@ncl.ac.uk