The repository is watching: automated harvesting from replicated filesystems

[This is a repost of http://jiscpub.blogs.edina.ac.uk/2011/07/15/the-repository-is-watching-automated-harvesting-from-replicated-filesystems-2/ please comment over there]

One of the final things I'm looking at on this jiscPUB project is a demonstration of a new class of tool for managing academic projects – not just documents. For a while we were calling this idea the "Desktop Repository", the idea being that there would be repository services watching your entire hard disk and exposing all the content in a local website with repository and content management services – that's possibly a very useful class of application for some academics, but in this project we are looking at a slightly different slant on that idea.

The core use case I'm illustrating here is thesis writing, but the same workflow would be useful across a lot of academic projects, including all the things we're focussing on in the jiscPUB project – academic users managing their portfolio of work, project reporting and courseware management. This tool is about a lot more than just ebook publishing, but I will look at that aspect of it, of course.

In this post I will show some screenshots of The Fascinator repository in action, talk about how you can get involved in trying it out, and finish with some technical notes about installation and setup. I was responsible for leading the team that built this software at the University of Southern Queensland. Development is now being done at the University of Central Queensland and the Queensland Cyber Infrastructure Foundation where Duncan Dickinson and Greg Pendlebury continue work on the ReDBox research data repository which is based on the same platform.

I know Theo Andrew at Edinburgh is keen to get some people trying this. So this blog post will serve to introduce it and give his team some ideas – we'll follow up on their experiences if there are useful findings.

Managing a thesis

The short version of how this thesis story might work is:

- The university supplies the candidate with a dropbox-like shared file system they can use from pretty much any device to access their stuff. But there's a twist – there is a web-based repository watching the shared folder and exposing everything there to the web.
- The university helpfully adds into the share a thesis template that's ready to go, complete with all the cover page stuff, margins all set, automated tables of contents for sections and tables and figures and the right styles – and trains the candidate in the basics of word processing.
- The candidate works away on their project, keeping all their data, presentations, notes and so on in the Dropbox and filling out the thesis template as they go.
- The supervisor can drop in on the work in progress and leave comments via an annotation system.
- At any time, the candidate can grab a group, which we call a package of things to publish to a blog or deposit to a repository at the click of a button. This includes not just documents, but data files (the ones that are small enough to keep in a replicated file system), images, presentations etc.
- The final examination process could be handled using the same infrastructure and the university could make its own packages of all the examiners reports etc for deposit into a closed repository.

The result is web-based, web-native scholarship where everything is available in HTML, not just PDF or application file formats and there are easy ways to route content to other repositories or publish it in various ways.

Where might ebook dissemination fit into this?

Well, pretty much anywhere in the above that someone wants to either take a digital object 'on the road' or deposit it in a repository of some kind as a bounded digital thing.

Demonstration

I have put a copy of Joss Winn's MA thesis into the system to show how it works. It is available in the live system (note that this might change if people play around with it). I took an old OpenOffice .sxw file Joss sent me and changed the styles a little bit to use the ICE conventions, I'm writing up a much more detailed post about templates in general, so stay tuned for a discussion of the pros and cons of various options for choosing style names and conventions and whether or not to manage the document as a single file or multiple chapters.

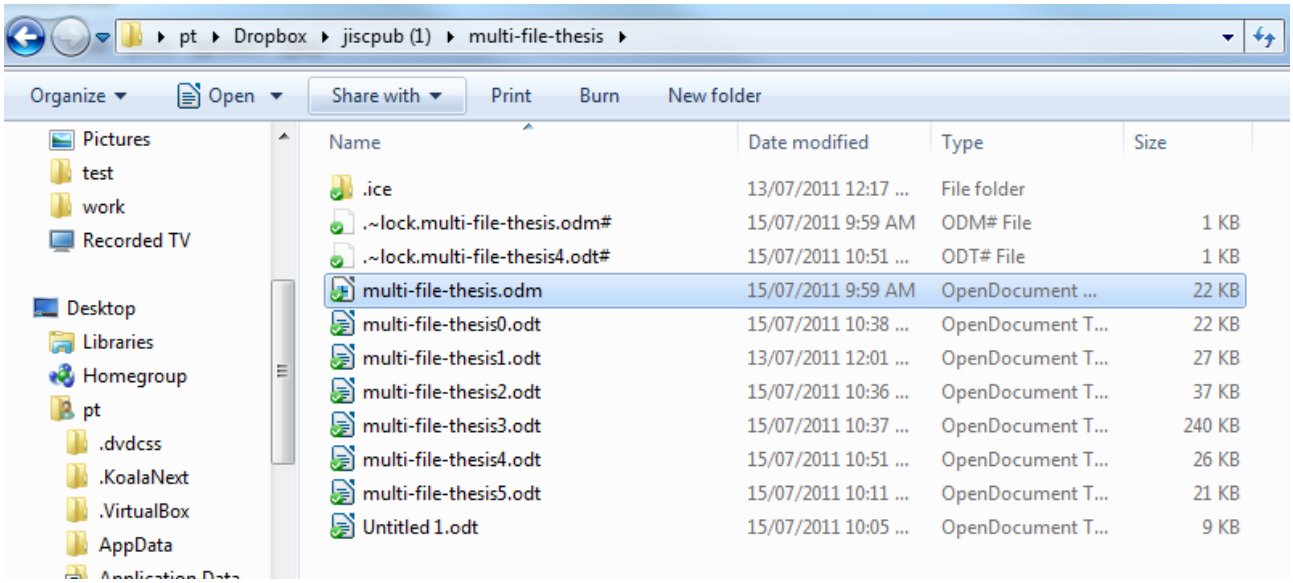

Illustration 1: The author puts their stuff in the local file system, in this case replicated by Dropbox.

Illustration 2: A web-view of Joss Winn's thesis.

The interface provides a range of actions.

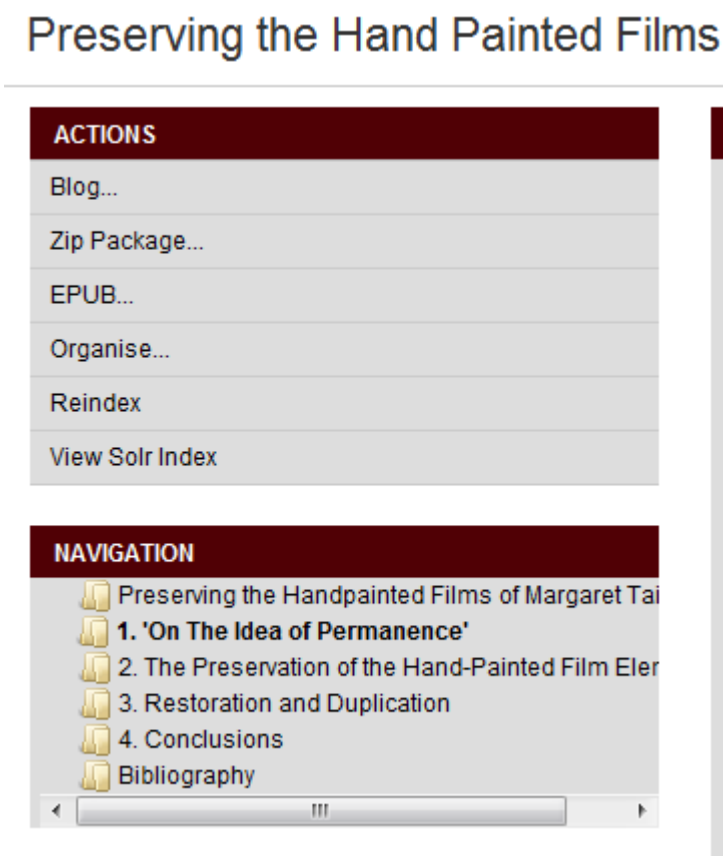

Illustration 3: You can do things with content in The Fascinator including blogging and export to zip or (experimental) EPUB

The EPUB export was put together as a demonstration for the Beyond The PDF effort by Ron Ward. A the moment it only works on packages, not individual documents, and it is using some internal Python code to stitch together documents, rather than calling out to Calibre as I did in earlier work on this project. The advantage of doing it this way is that you don't have Calibre adding extra stuff and reprocessing documents to add CSS – but the disadvantage is that a lot of what Calibre does is useful, for example working around known bugs in reader software, but it does tend to change formatting on you, not always in useful ways.

I put the EPUB into the dropbox so it is available in the demo site (you need to expand the Attachments box to get the download – that's not great usability I know). Or you can go to the package and export it yourself. Log in first, using admin as a username and a the same for a password.

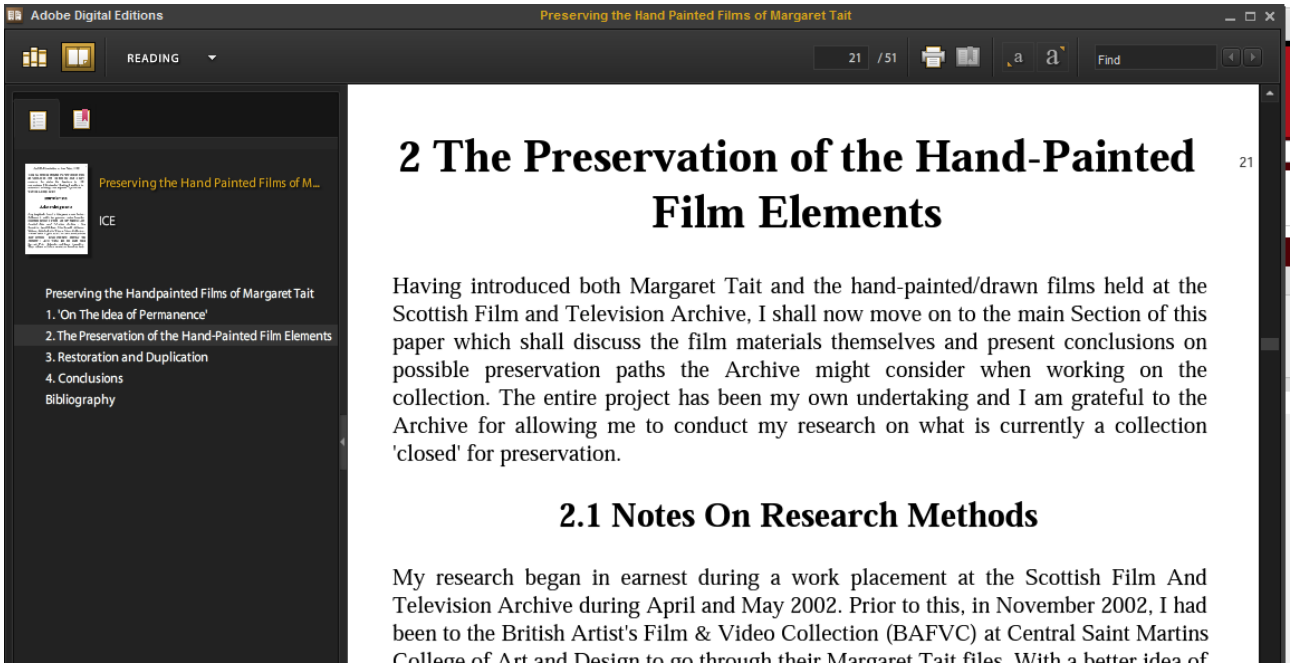

Illustration 4: Joss Winn's thesis exported as EPUB.

I looked a different way of creating an EPUB book from the same thesis a while ago which will be available for a while here at the Calibre server I set up.

One of the features of this software is that more than one person can look at the web site – and there are extensive opportunities for collaboration.

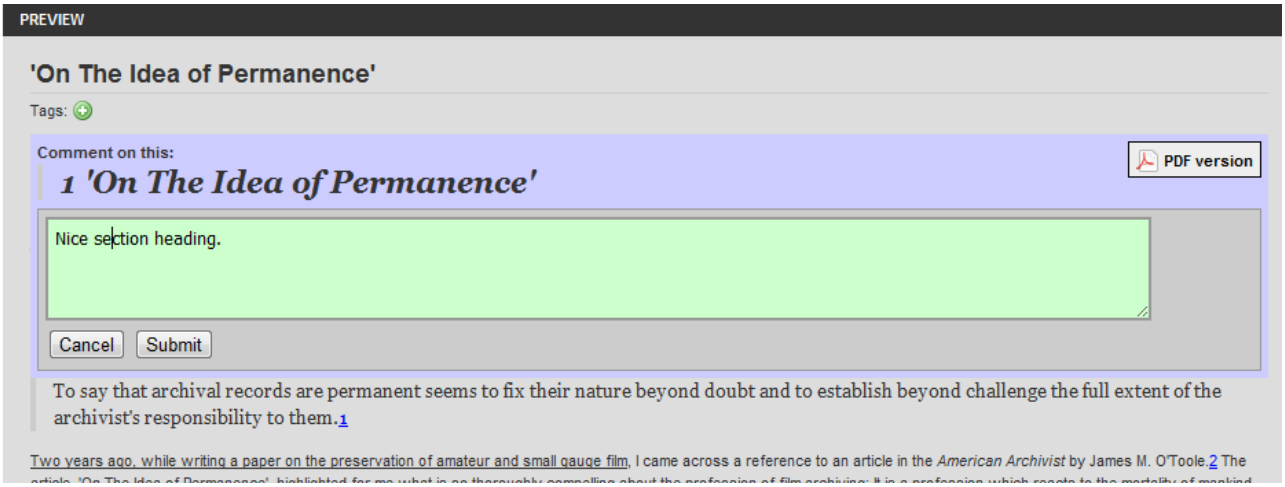

Illustration 5: Colleagues and supervisors can leave comments via inline annotation (including annotating pictures and videos)

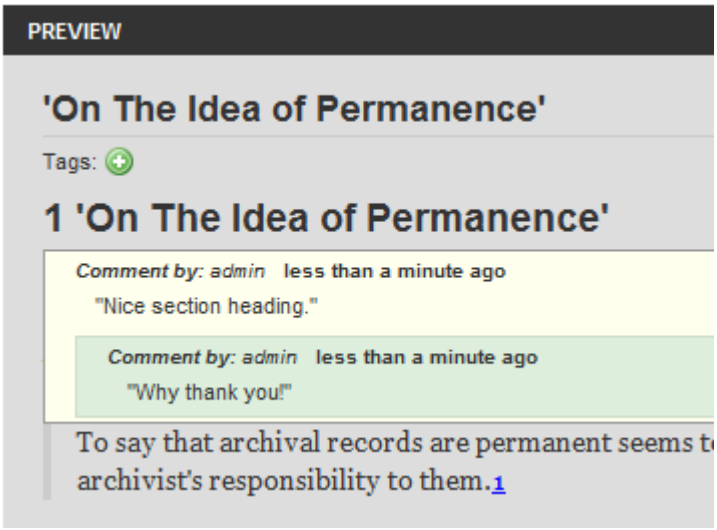

Illustration 6: Annotations are threaded discussions

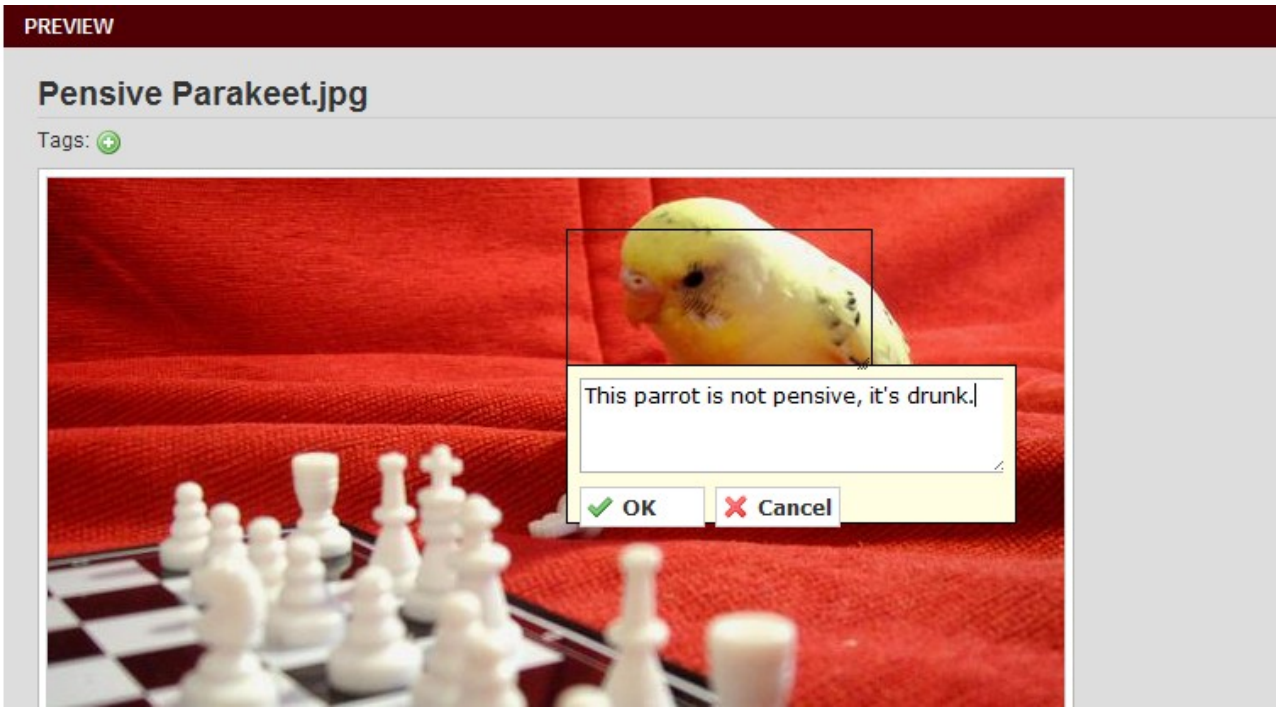

Illustration 7: Images and videos can be annotated too. At USQ we developed a Javascript toolkit called Anotar for this, the idea being you could add annotation services to any web site quickly and easily.

This thesis package only contains documents, but one of the strengths of The Fascinator platform is that it can aggregate all kinds of data, including images, spreadsheets, presentation and can be extended to deal with any kind of data file via plugins. I have added another package, modestly calling itself the research object of the future, using some files supplied by Phil Bourne for the Beyond the PDF group The Fascinator makes web views of all the content – and can package it all as a zip file or an EPUB.

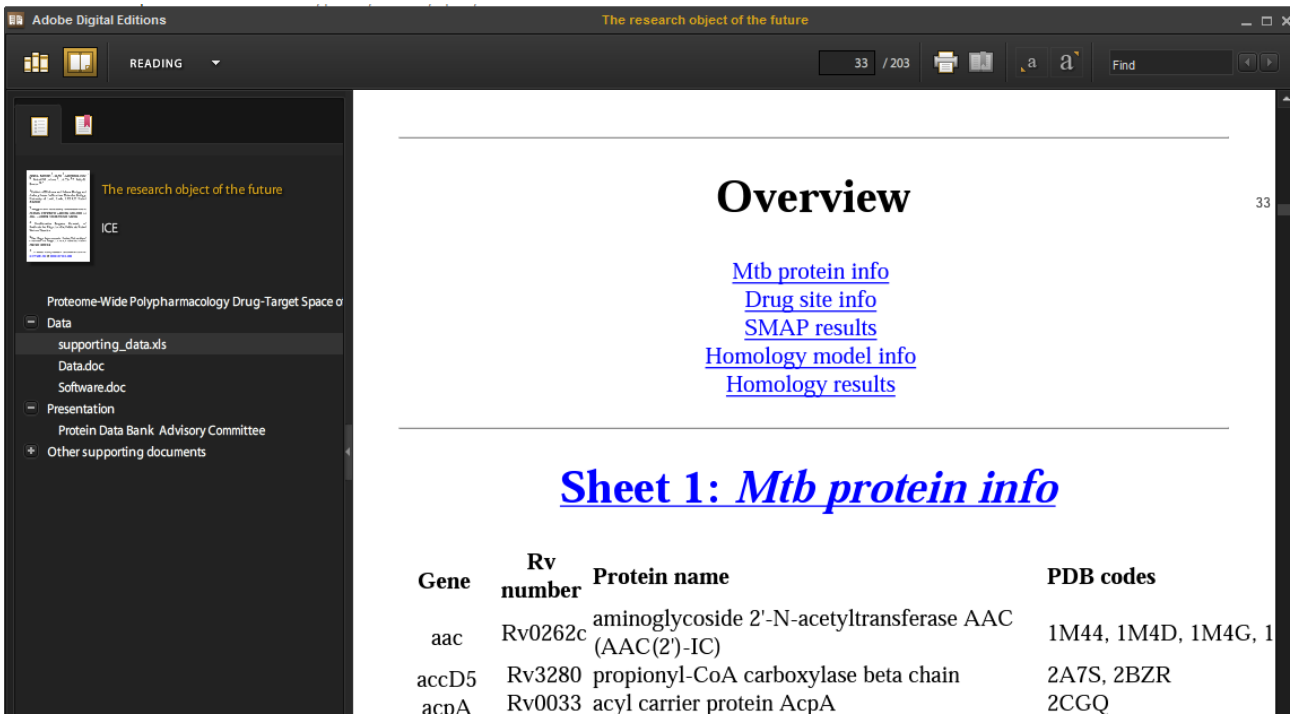

Illustration 8: A spreadsheet rendered into HTML and published into an EPUB file (demo quality only)

This includes turning PowerPoint into a flat web page.

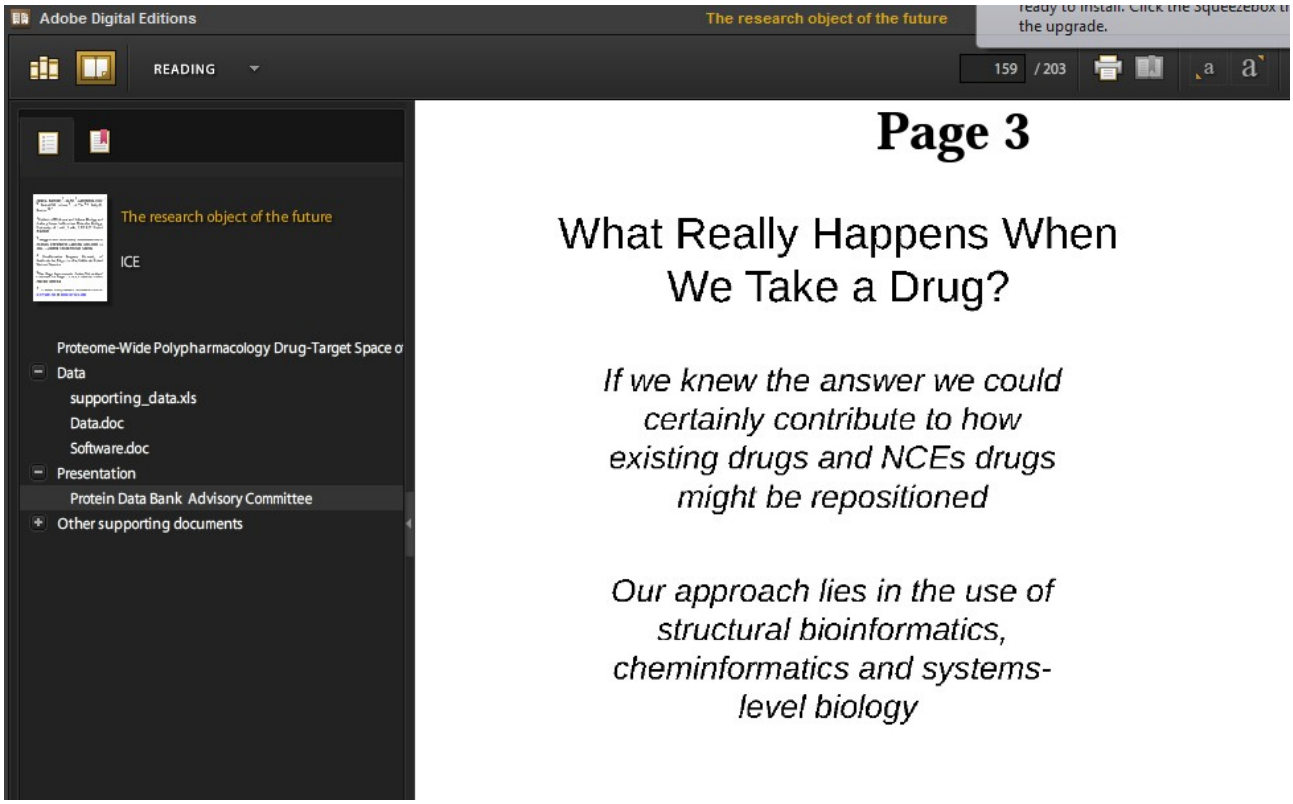

Illustration 9: A presentation exported to EPUB along with data and all the other parts of a research object

Installation notes

Installing The Fascinator (I did it on Amazon's EC2 cloud on Ubuntu 10.04.1 LTS) is straightforward. These are my notes – not intended to be a detailed how-to, but possibly enough for experienced programmers/sysadmins to work it out.

Check it out.

```
sudo svn co https://the-fascinator.googlecode.com/svn/the-fascinator/trunk
/opt/fascinator
```

Install Sun's Java

```
sudo apt-get install python-software-properties

sudo add-apt-repository ppa:sun-java-community-team/sun-java6

sudo apt-get update

sudo apt-get install sun-java6-jdk
```

http://stackoverflow.com/questions/3747789/how-to-install-the-sun-java-jdk-on-ubuntu-10-10-maverick-meerkat/3997220#3997220

Install Maven 2.

```
sudo apt-get install maven2
```

Install ICE or point your config at an ICE service. I have one running for the jiscPUB project – you can point to this by changing the `~/.fascinator/system-config.json` file.

Install Dropbox or your file replication service of choice – a little bit of work on a headless server but there are instruction linked from the Dropbox.com site.

Make some configuration changes, see below.

To run ICE and The Fascinator on their default ports on the same machine add this stuff to /etc/apache2/apache.conf (I think the proxy modules I'm using here is non-standard).

```
LoadModule  proxy_module /usr/lib/apache2/modules/mod_proxy.so

LoadModule  proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so

ProxyRequests Off

<Proxy *>

Order deny,allow

Allow from all

</Proxy>

ProxyPass        /api/ http://localhost:8000/api/

ProxyPassReverse /api/  http://localhost:8000/api/



ProxyPass       /portal/ http://localhost:9997/portal/

ProxyPassReverse /portal/ http://localhost:9997/portal/
```

Run it.

```
cd /opt/fascinator

./tf.sh restart
```

Configuration follows:

- To set up the harvester, add this to the empty jobs list in `~/.fascinator/system-config.json`

```
"jobs" : [
                    {
                        "name": "dropbox-public",
                        "type": "harvest",
                        "configFile":
"${fascinator.home}/harvest/local-files.json",
                        "timing": "0/30 * * * * ?"
                    }
```

And change /harvest/local-files.json to point at the Dropbox directory

```
"harvester": {

        "type": "file-system",

        "file-system": {

            "targets": [

                {

                    "baseDir": "${user.home}/Dropbox/",

                    "facetDir": "${user.home}/Dropbox/",

                    "ignoreFilter": ".svn|.ice|.*|~*|Thumbs.db|.DS_Store",

                    "recursive": true,

                    "force": false,

                    "link": true

                }

            ],

            "caching": "basic",

            "cacheId": "default"

        }
```

To add the EPUB support and the red branding, unzip the skin files in this zip file into the portal/default/ directory: http://ec2-50-19-86-198.compute-1.amazonaws.com/portal/default/download/551148ce6d80bfc0c9c36914f9df4f91/jiscpub.zip

```
unzip -d /opt/fascinator/portal/src/main/config/portal/default/ jispub.zip
```

[This is a repost of http://jiscpub.blogs.edina.ac.uk/2011/07/15/the-repository-is-watching-automated-harvesting-from-replicated-filesystems-2/ please comment over there]