Improving VALET - part 1

This week the ARROW community is having get together for developers to work on the VALET repository ingest tool. This is probably of little interest if you're not a repository person (or rat) but if you are then this may be of interest whether you are associated with the VITAL / Fedora world or not.

VALET is a deposit tool designed to allow self-deposit of electronic stuff into a Fedora repository, specifically one running VTLS VITAL. The bit about VITAL is crucially important – Fedora is an underlying storage layer, a kind of database, and different software will use it in different ways. VITAL has some tricks for storing datastreams derived form other assets, such as full-text extracted from PDF that other software like Fez would not understand.

VALET comes in two versions.

1. There's an open source one Valet for ETDs which is set up initially just to deal with Electronic Theses and Dissertations (ETDs). It's available from the VTLS website or from Google Code (last week the one at the VTLS site was out of date, and the package for download from Google Code was slightly less out of date but I think they might be up-to-date now).
2. The other version is mostly the same but is not free. It is important to make the distinction because if you customize the non-free version then you would have to ask VTLS for permission to redistribute it, possibly even within your own institution. I am not a lawyer (although I have a 10 year old who is threatening to become one) but I would be very cautious about changing a file that says (c) <Some Corporation> All rights reserved (Her other potential career is being a computer programmer might be a good idea to do both so she can be rich **and** happy).

So the outcome of the workshop will be to get a version of the open-source VALET with the best of the modifications that people have made at their sites, with maybe some new features.

One much requested feature for VALET (and for VITAL too) is to be able to edit submissions that have already been approved and pushed through VALET workflow into the repository. It's kind-of surprising that VALET doesn't do this already but it doesn't.

I had an idea about how this might work last week, and Tim McCallum has implemented the first part of it already. To explain it we have to go into a little bit of detail about how VALET works. VALET takes a very simple approach to workflow, of which I for one approve. In simple terms:

- An administrator defines a workflow with a set number of steps and says who can approve a submission at each step.
- An administrator defines a web form, based on the example(s) shipped by VTLS to collect the metadata required for a submission.
- At each stage the software simply serializes the information in the form into XML and saves it on disk.
- For each new stage the program picks up the information from disk and puts the values back into the form.
- At the final stage the program runs XSLT stylesheets (supplied by the administrator) to transform the serialized form data into the 'proper' metadata for the repository.

What Tim has done is simply to create an additional data stream containing the form data along with the other data streams when an item is approved. This means that it will be there alongside the repository item and all the other metadata streams. I think this will be really useful in solving some of the ongoing issues people are having with their repositories. For example, you might want to capture author email addresses but there is no sensible place to put them in a MODS datastream.

I know, some of you are thinking about standards – how can I save my important data in a non-standard format? To which I say, better to save your data in a form which is not standard and not pretending to be standard, than to rush into inventing a new standard which only you support. Is there a standard out there that captures all the data you want to save? Then use it. If not, capture the data now and work with the community to define the standard you need.

I'm not the only one who had this idea. I found out that Vicki Picasso from Newcastle also thought it would be good to capture the VALET form.

This approach is actually very similar to what you do in ePrints – you can define any old metadata you want (as long as it's flat name-value pairs) and map it to Dublin Core as you see fit for dissemination purposes.

In VITAL, and in our Sun Of Fedora repository portal project you can index any XML datastream you like. So if you want to collect HERDC categories (that's to do with reporting research publications to the Australian Government – very important stuff) then you can, without having to jam them into a metadata schema that was not designed to take them.

Next steps in the work Tim started:

1. Work out how to search for and retrieve an item to be re-edited, putting it back in the workflow.
2. Work out how to create the formdata from existing items that did not get put in the repository. We already have some experience with generating VALET form data based on a very cool idea by Simon McMillan of UNE who can't make it to the workshop. Get well Simon!

(I put it to my daughter that she could be a programmer and a lawyer and that would make her rich and happy. She said of course being a lawyer would make her rich and happy. I asked what would being a programmer make her? A nerd, apparently.)