FedoraShare

What is it?

An application front-end deployed on top of a Fedora repository which will allow users to easily add (either as a file upload or via linking to an external site), discover, and view content as well as add tags, comments, and relationships to that content.

What will it do?

- Provide content viewers through both external hosting sites (such as YouTube for videos) and via local service disseminations.
- Provide users with the opportunity to add metadata about the content.
- Allow all content added to be viewable by all users.
- Allow users to comment on and tag content.
- Allow users to create named collections of content.
- Allow users to create relationships between pieces of content. Relationships may be pre-defined or custom.
- Provide the ability to search for content using at least the content's metadata and tags.
- Provide the ability to browse through the repository by clicking on tags, collections, or relationships surrounding content.
- Allow an administrator to indicate "Featured" content items, which will be displayed to all users.
- Track usage and provide suggestions such as a "Most Popular" and "You Might Like". *
- Hide the underlying Fedora object structure from the user.

Why is this interesting?

- FedoraShare goes beyond existing sites (YouTube, Flickr, etc) by allowing users to combine content of different types and from different locations in a single interface.
- FedoraShare would allow users to comment on and tag collections as well as individual files.
- FedoraShare would allow users to build relationship graphs over the content, linking to specific objects and collections, and asserting any kind of relationship.
- FedoraShare would allow users to store their content in a repository for preservation but still have it be accessable through high-traffic web sources.
- FedoraShare would provide a location where information about content (metadata) can included with or linked to that content.
- FedoraShare would provide a way for Fedora to integrate with the content already available on the web.

How will it work?

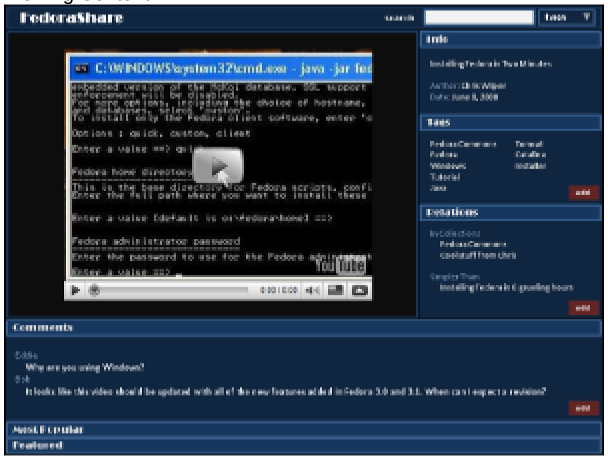
- Content will be stored as datastreams. There will be one content item per object. This allows linking to a content file without having to include a RELS-INT
    - All managed content will need to indicate the type of file so that the appropriate content model (and thus disseminations) can be attached.
        - All file types will be available for download.
        - Certain types of files (primariliy video and slides) will be pushed to external sites (such as YouTube and SlideShare), then disseminated using the APIs provided by those sites.
    - Linked content can be just links to a site where the content is hosted. For popular destinations further support will be provided, either by accessing APIs via disseminations or creating simple dissemination schemes (an example of this would be using the YouTube and SlideShare provided embed tags to construct an html page which displays the content.)
    - Content which is not multimedia (text, Word docs) will be handled by the browser, which will either open or provide a download option.
    - Content models will be created for a variety of media types. Some of these will have sdef/sdep pairs to provide disseminations. All media type CModels (including a catch-all CModel for unknown types) will have an icon indicating the format, which can be displayed in search results.
- Metadata will be dublin core, to keep things simple and provide easy searching capability. Further metadata can always be added as another content item.
- Comments on content will be stored as an inline xml datastream within the object where the content is stored.
- Tags will be stored as RDF in RELS-EXT. When displaying a content object all of its tags will also be displayed as links which perform an RDF query to retrieve all other objects with the same tag.
- Collections will be created as an object with no content. Items making up the collection will be included in the RELS-EXT of the collection so that updating the members of the collection will only involve changes in one object.
- User asserted relationships between objects will be stored in the content object's RELS-EXT.
- Users will be stored as objects. Relationships will be added to the user for each object that they create, this will allow users to update only their own content and collections (without the implementation needing to get into AuthZ policies).
- Search will provide a few options
    - FieldSearch returning a default field selection (such as label and description)
    - Search for tags, accomplished via RDF search
    - Search within full text and comments will be provided using GSearch *
- Browsing will be possible by collection, tag, or relationship based on underlying RDF queries.
- Tag and relationship clouds will be created using RDF queries to determine all available tags and relationships as well as their prevalence. *
- The Featured items list will be a collection that is created by an administrator and available to all users.
- In order to provide a listing of most popular items, Fedora will be updated to produce JMS messages on API-A calls as well as API-M. A new project "FedoraStats" will listen for getDissemination request messages and update a DB table which stores the number of disseminations per PID. *

* - This feature will likely not be included in the first iteration

What will it look like?

The following are mockup screenshots to provide an idea of what the application may look like.

Viewing Content



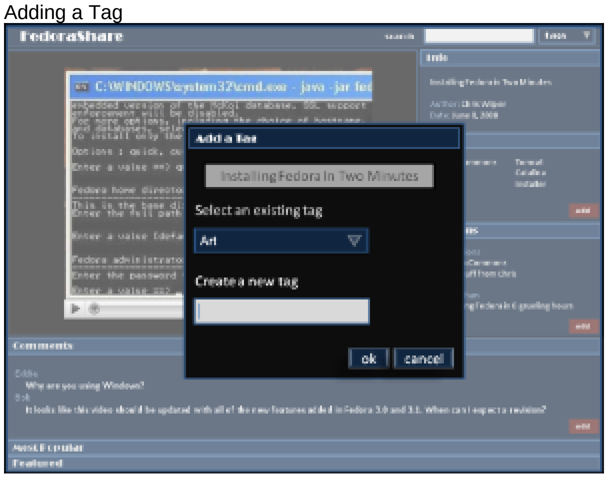
Installing Fedora in Two Minutes
Author: Chris Wilper
Date: June 8, 2008

Fedora Commons, Fedora, Windows, Tutorial, Java, Tomcat, Catalina, Installer

In Collections: Fedora Commons, Cool stuff from Chris
Simpler Than: Installing Fedora in 6 grueling hours

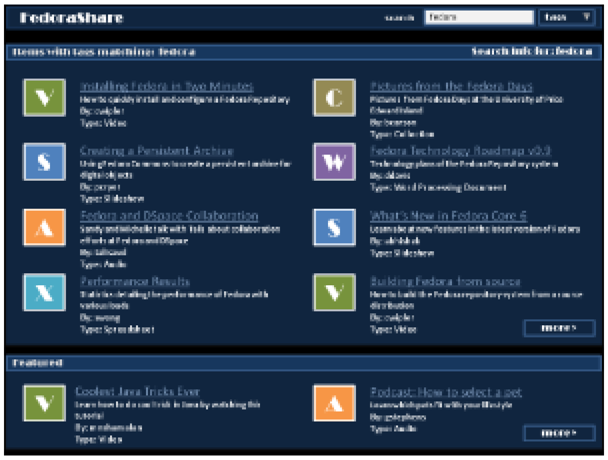
Eddie: Why are you using Windows?
Bob: It looks like this video should be updated with all of the new features added in Fedora 3.0 and 3.1. When can I expect a revision?

Adding a Tag



Installing Fedora In Two Minutes

Searching



Installing Fedora in Two Minutes
How to quickly install and configure a Fedora Repository
By: cwilper
Type: Video

Pictures from the Fedora Days
Pictures from Fedora Days at the University of Prince Edward Island
By: bbranan
Type: Collection

Creating a Persistent Archive
Using Fedora Commons to create a persistent archive for digital objects
By: payette
Type: Slideshow

Fedora Technology Roadmap v0.9
Technology plans of the Fedora Repository system
By: ddavis
Type: Word Processing Document

Fedora and DSpace Collaboration
Sandy and Michele talk with Tufts about collaboration efforts of Fedora and DSpace
By: tufts
Type: Audio

What's New in Fedora Core 6
Learn about new features in the latest version of Fedora
By: [illegible]
Type: Slideshow

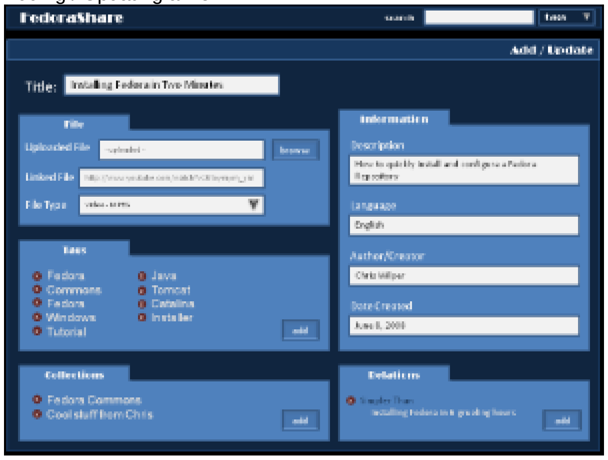
Performance Results
Statistics detailing the performance of Fedora with various loads
By: [illegible]
Type: Spreadsheet

Building Fedora from source
How to build the Fedora repository system from a source distribution
By: cwilper
Type: Video

Coolest Java Tricks Ever
Learn how to do cool trick in Java by watching this tutorial
By: [illegible]
Type: Video

Podcast: How to select a pet
Learn which pets fit with your lifestyle
By: [illegible]
Type: Audio

Adding / Updating a file



Title: Installing Fedora in Two Minutes
Uploaded File: -uploaded-
Linked File: http://www.youtube.com/watch?v=[illegible]
File Type: video-[illegible]
Tags: Fedora, Commons, Fedora, Windows, Tutorial, Java, Tomcat, Catalina, Installer
Description: How to quickly install and configure a Fedora Repository
Language: English
Author/Creator: Chris Wilper
Date Created: June 8, 2008
Collections: Fedora Commons, Cool stuff from Chris
Relations: Simpler Than: Installing Fedora in 6 grueling hours