Dandi DMS

1 What is it?

Dandi DMS is a document management system that uses the Drag 'n' Drop metaphor for ease of use.

Most DMS systems use one of two approaches, either they are web-based or they extend the client application in some way, both of these methods have their problems. Web-based approach requires a level of computer competence that largely excludes the novice, where as extending the client application is a tread-mill that requires all editors for all document types to be extended to include some form of DMS interface.

The Cleanest method of including DMS on a system would be to extend the filing system itself to incorporate some form of relational mapping, as this is not going to happen soon I felt an intermediate method was required.

Dandi-DMS is a compromise between the flexibility of the Web-based DMS and the usability of extending applications.

1.1 How does it work?

1.2 Local Persistence Model

Each interaction with the server either committing a new file or downloading a file adds the file to you local history list and creates a file of the same name with ".dnd" appended. When you want to interact with the server again just find the file in you local history and perform one of the predefined actions. The local history keeps the following information.

Description	*Value*
Filename	The path to the file on you local system.
index	Unique identifier of the file on the server
sha1	A digital signature of the file that was last downloaded or uploaded to the server.
version	Version number of the file on the server.
Locked	The name of the person who has locked the file or blank if the file is not locked.

Having this data for each file in your local history allows the following rules to be enforced.

The terms used in this are as follows.

Authorative Information about the file from the server.

History Information from the local history

Working Information obtained from the local file.

When discussing each of these objects, we will use the form Authorative.sha1 to reference the digital signature of the file on the server.

1.2.1 Lock

If the History.Version does not equal the Authoratitive.Version a lock will not be acquired and error will be displayed and the action cancelled.

If the file is already locked by anyone other than yourself and error will be displayed and the action cancelled.

If successful the server will mark the file locked by you and your local history will be updated.

1.2.2 Release

The server releases a lock on the file (Need some rules).

1.2.3 Commit

History.Locked must reference the client or an error is displayed and the action is cancelled.

History.Version must equal Authoratitive.Version an error is displayed and the action cancelled.

Working.Sha1 must be different to History.Sha1 or an error is displayed and the action cancelled.

The file is committed to server, the server increments the version number and the local history is updated.

1.2.4 Update

History.sha1 must be the same as Working.sha1 or and error message is displayed and the action is cancelled.

The current version of the file is downloaded and the history updated.

1.3 Installation

To be filled out at V1.

1.4 Bugs

1.4.1.1 Version 0.4.8b

loses files from the local list all to easily.
Need to fix problems with mail system.

File permissions on the server and other nasty errors
Fixed mail permissions.

1.4.1.2 Version 0.4.9a

It is possible to add the same folder twice when committing a file if the config is edited between additions.

1.5 Problems

1.6 Ideas