

File Control System 1.0

Product Requirements Document (PRD)

Author: Ken Rodham
Date: January 10, 2005
Revision: 2

Overview

This document specifies the requirements for the File Control System 1.0 (FCS). The purpose of FCS is to support teams of workers who need to share a common repository of files. Examples of such sharing are:

- A team of software developers who need to access a shared repository of source code files for the system they are building
- A team of writers who are collaboratively writing a book and need to access a shared repository of documents containing the different parts of the book
- A team of graphic designers who need shared access to the files containing the artwork for the project they are working on

In each of these examples, a team stores their work products in a file system consisting of files and directories. Each team member needs the ability to add new files and directories to the repository, to read and write files that are already in the repository, and to remove files and directories from the repository. All changes made to the repository must be accessible to all team members, thus allowing them to collaborate.

While network file systems such as NFS and Samba can be used to give teams shared access to files over a network, this solution is insufficient for team projects such as those listed above for the following reasons:

1. In a team context, multiple team members may try to update the file repository concurrently. If such concurrent updates conflict with each other, this may result in corruption of files and directories in the repository
2. Teams need to track the history of changes made to each file in the repository. They need access to information about what changes were made, when they were made, and who made them. They also need to be able to access prior versions of files.
3. Teams need to manage multiple versions of the file repository. At the completion of a project (or perhaps at an important milestone during a project), the team may wish to label the current repository as representing a meaningful state to which they might want to return in the future. For example, the repository might be labeled as “1.0 beta” when the beta version of the product is complete. This would allow the team to later return to the beta version of the product, even after many changes have been made to the repository.

None of these features are provided by network file systems, thus necessitating FCS.

Requirements

R01

FCS must provide multi-user access to a centralized, shared file repository. A *file repository* is a hierarchy of files and directories. Directories may be nested arbitrarily, and each directory may contain any number of files.

R02

Users must be able to create local copies of the repository on their own computers. A local copy of the repository is called a *workspace*. Users must also be able to make changes to their local workspaces, and eventually write their local changes to the shared repository. This allows users to work concurrently without interfering with each other.

R03

FCS must support concurrent access by multiple users, including concurrent updates. Concurrent access to the repository must not result in data corruption.

R04

Users must be able to add new directories to the repository.

R05

Users must be able to add new files to the repository.

R06

FCS must support both text and binary files.

R07

FCS must archive each version of every file, including the current version and all prior versions.

R08

Users must be able to remove empty directories from the repository.

R09

Users must be able to remove files from the repository.

R10

Users must be able to retrieve the current or a prior version of a file from the repository and copy it to their local workspace.

R11

Users must be able to view the differences between any two versions of a text file, similar to the Unix `diff` command.

R12

Users must be able to retrieve the contents of any repository directory and copy it to their local workspace. The contents of the specified repository directory should be copied recursively to the local workspace.

Retrieving the repository's root directory should cause the contents of the entire local workspace to be synchronized with the current contents of the repository.

Retrieving a subdirectory within the repository should cause only that subset of the workspace to be synchronized with the repository.

R13

FCS must not allow multiple users to concurrently edit (i.e., modify) the same file. Users must "lock" a file in the repository before making changes to their local copy, thus notifying other users of their intent to change the file. Only one user is allowed to lock the file at a time, thus preventing multiple users from modifying the same file at the same time.

R14

Users must be able to copy a modified version of a file from their local workspace to the shared repository, thus creating a new version of the file.

R15

If requested by the user, FCS must be able to automatically insert the following information into text files in the repository:

- (1) Relative path of the file with respect to the repository root directory
- (2) Number of this version of the file
- (3) Name of user who created this version of the file
- (4) Date/time at which this version of the file was added to the repository

This makes it easy for a user to determine exactly which version of a file they are working with.

R16

FCS must support repository versioning. FCS must allow the user to assign a version label to the current state of the repository (e.g., "BETA", "FINAL RELEASE", etc.). It must also be possible for the user to later retrieve a particular version of the repository by name (e.g., give me the "BETA" version of the repository).

R17

Users must be able to view detailed information about the repository. Repository information should include at least the following:

- (1) Name of the user who created the repository
- (2) Date/time at which the repository was created
- (3) Information about repository versions

R18

Users must be able to view detailed information about any directory in the repository. Directory information should include at least the following:

- (1) Full path of the directory within the repository
- (2) Name of the user who added the directory to the repository
- (3) Date/time at which the directory was added to the repository
- (4) List of which repository versions the directory participates in

R19

Users must be able to view detailed information about any file in the repository. File information should include at least the following:

- (1) Full path of the file within the repository
- (2) Description of the file's contents (provided by the user when the file was added to the repository)
- (3) Name of the user who added the file to the repository
- (4) Date/time at which the file was added to the repository
- (5) Current version number of the file
- (6) Name of the user who currently has the file locked (if any)
- (7) Date/time at which the user locked the file (if any)
- (8) List of which repository versions the file participates in

R20

Users must be able to view detailed historical information about the changes made to a file. At least the following information should be reported for each file version:

- (1) Name of the user who created the version
- (2) Date/time at which the version was added to the repository
- (3) Description of changes made in this version of the file