VERSIONDEFS

By Ron Kaplan

This document was last edited in December 2024.

VERSIONDEFS provides basic tools for working with (but keeping separate) the different definitions of an item on different Medley file-versions. A version is specified as a positive or negative integer. The positive integers count from the oldest file (version 1 is the oldest, version 2 is the second oldest) while negative integers count down from the most recent version (-1 is the newest, -2 is the next newest, etc.). The atoms OLDEST, FIRST, F are synonyms for 1, NEWEST, N and the number 0 are synonyms for -1. (With a little more work, a version could be specified as before or after a particular date.)

Currently this operates on files that already exist in the local environment. The shell utility scripts/restore-versions.sh can be used to recreate locally the medley-versions of files in a remote git repository, in the order of their commits. Those files could then be specified by their ordinal numbers. A further extension would be to use git branch, commit, or PR identifiers to refer to specific file versions.

(VERSIONP X) [Function]

Returns the integer form of the version if X denotes a version, otherwise NIL.

(FINDFILEVERSION FILE VERSION DIRLIST NOERROR) [Function]

Returns the full directory name of the specified VERSION of FILE found somewhere in DIRLIST (or DIRECTORIES). If NOERROR, returns NIL if such a file does not exist, otherwise causes an error.

(GETVINFO NAME TYPE FILE VERSION DIRLIST) [Function]

Retrieves the definition of NAME as TYPE on the specified VERSION of FILE, returning a 3-element list (VNAME TYPE DEF). VNAME is an annoated version of NAME that has the actual version of the file from which the definition was obtained. For example, if VERSIONDEFS;3 is the oldest version of that file, the VNAME returned for (GETVINFO 'VERSIONP NIL NIL 1) will be GETVINFO;3.

VERSIONDEFS includes three simple editing applications of these version primitives. These open SEDIT windows on versioned definitions without overwriting or otherwise confusing or interfering with the current in-memory definition. That is because the version-definitions are associated with the annotated VNAME's and are not actually install in the environment.

(EDV NAME TYPE FILE VERSION DIRLIST) [Nlambda no-spread Function]

A generic version-editor akin to ED: brings up the definition from the version-file of NAME as TYPE. Expressions can be examined and copied, and even modified--but the modifications will fall on the floor.

(DFV NAME FILE VERSION DIRLIST) [Nlambda-nospread Function]

Calls EDV with NIL as the type, akin to DF.