NAME

trm - A Safer Delete Utility

SYNOPSIS

`trm.sh [-Vcdhistvx] [-g graveyard] file|dir [...]`

DESCRIPTION

`trm` is a file- and directory removal tool with several key features:

- Files are not actually removed, but put in a "graveyard" where they can later be retrieved. By default, this is in `~/.graveyard`, but this can be overriden on the command line with the `-g` option.
- There is a "test mode" which doesn't actually do anything, but only shows you what the program *would do* if it actually executed,
- Deleted items are stored in the graveyard in the same directory hierarchy they were originally in.
- By default, deleted items have a datetime stamp or "serial number" appended to them. This allows the graveyard to accumulate different versions of a file- or directory tree as they are deleted over time.
- `trm` can also be use to simply copy a set of files- and/or directories into the graveyard, without actually removing them from their original location. This is handy if you want to take advantage of the serial number capability and keep versioned "deletions" around, but not actually remove the original files- or directories.
- Delete or copies can be invoked interactively so you can select which of the targets you actually want affected.
- You can either run the program as a standalone utility - via the `trm.sh` program - Or, you can source it - `source trm.sh`. This loads the `trm` shell function into the current shell context for subsequent use from the command line or another program.
- You can put commonly used command line options in the `$TRM` environment variable. For instance, you may want to default to test mode operation by doing something like `export TRM="-t"`.

OPTIONS

You can override the program's default behavior with a number of command line options:

-V	display version control commit ID
-c	copy targets to graveyard, don't remove them
-d	empty the current graveyard (permanent removal)
-h	display this help screen
-g graveyard	use named graveyard instead of default
-i	interactive removal/copy

-s	don't generate serial number suffixes
-t	test mode, just show what would be done
-v	verbose mode - be noisy
-x	execute, overrides previous -t

OPERATING NOTES

- Note that, in these examples, `trm.sh` means we're invoking the program as a standalone utility that is presumably somewhere on your `$PATH`. `trm` means you've sourced it previously and it's being called as shell function from your current command line context.
- You can use `trm` just like you do conventional `rm`:

  ```
  trm /tmp/foo /home/me/bar
  ```

  So the files/directories called `foo` and `bar` would be "removed‘ from their current location and placed in `~/.graveyard/tmp/foo` and `~/.graveyard/home/me/bar` respectively. If the required directory structure does not exist in the graveyard, `trm` will create it automatically.
- Directory removal or copy is always recursive.
- Options are processed left-to-right and take effect in that same order. For example, `trm.sh -ve` will clear out the graveyard and report on everything it is removing. However, `trm.sh -ev` will not because the deletions will occur before the verbose option is recognized.
- If you are using multiple graveyards, you can use this property to your advantage to clear them all with one command:

  ```
  trm -v -g gy1 -d -g gy2 -d -g gy3 -d ....
  ```
- Serial numbers are appened to the target *as named on the command line*. If you delete a file, then it wil go into the graveyard as `file.serialnumber`. Similarly, a directory goes into the graveyard as `dir.serialnumber`. However, the names of anthing *within* that directory, are left untouched. This allows you to decide at what level of detail you want delete versioning to be maintained.
- If one of your delete/copy targets is a symlink, it will simply be moved or copied to the graveyard as-is. However, if you refer to a symlink somewhere in the path to a target, it *will* be expanded. That target will be moved or copied to a directory tree within the graveyard that mirrors the canonical location of the original target.

  For example, say you have a symlink such as `/local -> /usr/local`. Then:

  ```
  trm /local  # Moves the symlink to ~/.graveyard/local
  ```

  But:

  ```
  trm /local/bin/x # Moves x to ~/,graveyard/usr/local/bin/x
  ```

BUGS AND MISFEATURES

None known as of this release.

COPYRIGHT AND LICENSING

AUTHOR

Tim Daneliuk
trm@tundraware.com

DOCUMENT REVISION INFORMATION

$Id: '1ca95bc tundra Fri Oct 28 09:27:57 2016 -0500'

This document was produced with emacs, RestructuredText, and TeX Live.

You can find the latest version of this program at:

http://www.tundraware.com/Software/trm