

TLA Reference tla tla-1.2-2 from regexps.com

help

help : provide help with arch

User Commands

my-id : print or change your id
my-default-archive : print or change your default archive
register-archive : change an archive location registration
whereis-archive : print an archive location registration
archives : Report registered archives and their locations.

Project Tree Commands

init-tree : initialize a new project tree
tree-root : find and print the root of a project tree
tree-version : print the default version for a project tree
set-tree-version : set the default version for a project tree
undo : undo and save changes in a project tree
redo : redo changes in project tree
changes : report about local changes in a project tree
file-diffs : show local changes to a file

Project Tree Inventory Commands

inventory : inventory a source tree
tree-lint : audit a source tree
id : report the inventory id for a file
id-tagging-method : print or change a project tree id tagging method
add-id : add an explicit inventory id
add : (alias for add-id)
delete-id : remove an explicit inventory id
delete : (alias for delete-id)
rm : remove a file (or dir, or symlink) and its explicit inventory tag (if any)
move-id : move an explicit inventory id
move : (alias for move-id)
mv : move a file (or dir, or symlink) and it's explicit inventory tag (if any)
explicit-default : print or modify default ids
default-id : (alias for explicit-default)
id-tagging-defaults : print the default =tagging-method contents

Patch Set Commands

changeset : compute a whole-tree changeset
mkpatch : (alias for changeset)
apply-changeset : apply a whole-tree changeset
dopatch : (alias for apply-changeset)
show-changeset : generate a report from a changeset

Archive Transaction Commands

make-archive : create a new archive directory
archive-setup : create new categories, branches and versions
make-category : create a new archive category
make-branch : create a new archive branch
make-version : create a new archive version
import : archive a full-source base-0 revision
commit : archive a changeset-based revision
get : construct a project tree for a revision
get-changeset : retrieve a changeset from an archive
lock-revision : lock (or unlock) an archive revision
archive-mirror : update an archive mirror

Archive Commands

abrowse : print an outline describing archive contents
rbrowse : print an outline describing an archive's contents
categories : list the categories in an archive
branches : list the branches in an archive category
versions : list the versions in an archive branch
revisions : list the revisions in an archive version
ancestry : display the ancestry of a revision
ancestry-graph : display the ancestry of a revision
cat-archive-log : print the contents of an archived log entry
cacherev : cache a full source tree in an archive
cachedrevs : list cached revisions in an archive
uncacherev : remove a cached full source tree from an archive
archive-meta-info : report meta-info from an archive
archive-snapshot : update an archive snapshot
archive-version : list the archive-version in an archive
archive-fixup : fix ancillary files (e.g. .listing files) in an archive

Patch Log Commands

make-log : initialize a new log file entry
log-versions : list patch log versions in a project tree
add-log-version : add a patch log version to a project tree
remove-log-version : remove a version's patch log from a project tree
logs : list patch logs for a version in a project tree
cat-log : print the contents of a project tree log entry
changelog : generate a ChangeLog from a patch log
log-for-merge : generate a log entry body for a merge
merges : report where two branches have been merged
new-merges : list tree patches new to a version

Multi-project Configuration Commands

build-config : instantiate a multi-project config
cat-config : output information about a multi-project config

Commands for Branching and Merging

tag : create a continuation revision (aka tag or branch)
update : update a project tree to reflect recent archived changes
replay : apply revision changesets to a project tree
star-merge : merge mutually merged branches

apply-delta : Compute a changeset between any two trees or revisions and apply it to a project tree
missing : print patches missing from a project tree
join-branch : construct a project tree for a version
sync-tree : unify a project tree's patch-log with a given revision
delta : Compute a changeset (or diff) between any two trees or revisions

Local Cache Commands

changes : report about local changes in a project tree
file-diffs : show local changes to a file
file-find : find given version of file
pristines : list pristine trees in a project tree
lock-pristine : lock (or unlock) a pristine tree
add-pristine : ensure that a project tree has a particular pristine revision
find-pristine : find and print the path to a pristine revision

Revision Library Commands

my-revision-library : print or change your revision library path
library-config : configure parameters of a revision library
library-find : find and print the location of a revision in the revision library
library-add : add a revision to the revision library
library-remove : remove a revision from the revision library
library-archives : list the archives in your revision library
library-categories : list the categories in your revision library
library-branches : list the branches in a library category
library-versions : list the versions in a library branch
library-revisions : list the revisions in a library version
library-log : output a log message from the revision library
library-file : find a file in a revision library

Published Revisions Commands

grab : grab a published revision

Miscellaneous Scripting Support

parse-package-name : parse a package name
valid-package-name : test a package name for validity

help

help : provide help with arch

-h, --help Display a help message and exit.
-H Display a verbose help message and exit.
-V, --version Display a release identifier string
 and exit.

This command prints a list of the available commands.

To see just a list of the options to a particular command, use:

```
tla $cmd -h
```

(where \$cmd is the name of the command). For additional explanation about a given command, use:

```
tla $cmd -H
```

User Commands

my-id : print or change your id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-e, --errname  specify program name for errors
-u, --uid      print only the UID portion of the ID
```

With no argument print your arch id.

With an argument, record ID-STRING as your id in ~/.arch-params/=id

Your id is recorded in various archives and log messages as you use arch. It must consist entirely of printable characters and fit on one line. By convention, it should have the form of an email address, as in this example:

```
Jane Hacker
```

The portion of an id string between is called your uid. arch sometimes uses your uid as a fragment when generating unique file names. The option -u (--uid) causes only the uid part of your id string to be printed.

my-default-archive : print or change your default archive

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-e, --errname  specify program name for errors
-d, --delete   unspecify your default archive
-s, --silent   suppress reassuring messages
```

With no argument, and without -d, print the name of your default archive.

With an argument, record ARCHIVE as your default archive in ~/.arch-params/=default-archive

With the option -d (--delete) and no argument, ensure that you do not have a default archive set in ~/.arch-params.

Your default archive is determined this way:

If the option -A (--archive) is given and not empty, that archive is the default (which makes this script useful for processing a -A argument that was passed to another script).

If -A is not given, but ~/.arch-params/=default-archive exists and is not empty, that is your default archive.

Otherwise, your default archive is the name of the local archive rooted at the argument to -R (--root) or specified in the environment variable ARCHROOT.

If no default archive can be found by any of these means, the program exits with status 1, printing an error message unless the -s (--silent) option is given.

register-archive : change an archive location registration

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-f, --force    overwrite existing location
-d, --delete   delete archive registration
```

Record the location of ARCHIVE.

With -d, remove the registration of a previously registered archive. When accompanied by -f, override permissions on the registration file and don't complain if the archive is not registered.

A LOCATION should be either a directory name or a distant URL.

When registering a new archive, if no ARCHIVE's name is passed on the command line, then the archive's name will be read automatically from the archive's meta data.

Archive locations are stored in ~/.arch-params/=locations.

You must register the location of a remote archive before you access it. It is not strictly necessary to register the locations of local archives (you can always specify their location using command line arguments and/or environment variables), but registering local archive locations is recommend (for simplicity).

whereis-archive : print an archive location registration

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
```

Print the registered location of an archive.

Usually the archive must have been previously registered with "tla register-archive".

As a special exception, the the archive is not registered, but is the name of the archive rooted at the location given with the option -R (--root) or in the environment variable ARCHROOT then print that root directory.

archives : Report registered archives and their locations.

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-n, --names     print archive names only
-R, --exclude-remote Exclude MIRROR and SOURCE archives.
```

Print a list of registered archives and their locations

If [search regex] is given then only archives with names that match [search regex] will be shown

Project Tree Commands

init-tree : initialize a new project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-A, --archive   Override `my-default-archive'
-d, --dir DIR   cd to DIR first
--nested       initialize a nested project tree.
```

Initialize DIR as a new project tree.

An empty patch-log for VERSION will be created, and VERSION will be made the default location for check-ins.

tree-root : find and print the root of a project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
--accurate     error for mid-txn trees
-s, --silent    exit status only
```

Find and print the root of the project tree that contains DIR (or the current directory). The option --accurate checks for three possibilities:

- 1) The project tree was the subject of a commit that was killed before the log entry was added to the tree's patch log. It is unknown whether or not the commit took place in the archive. The {arch} directory contains the log file as "++mid-commit".

- 2) The project tree was the subject of a killed commit, but it is certain the commit took place. The log file is stored as "++commit-definite".
- 3) The project tree patch log is fully up-to-date (as far as arch knows).

In case 1, exit with an error and error message.

In case 2, install the log file before printing the tree root and exiting normally.

In case 3, print the tree root and exit normally.

tree-version : print the default version for a project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

Print the default version of project tree DIR (or the current directory).

set-tree-version : set the default version for a project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-d, --dir DIR  cd to DIR first
```

This operation sets the default version on which to check-in the source tree DIR (default: the current directory) during a commit.

You can specify the archive for this version as an ordinary command line argument (ARCHIVE/VERSION) or via -R, -A and the environment as usual (try "tla my-default-archive --help").

undo : undo and save changes in a project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-d, --dir DIR  cd to DIR first
-o, --output PATCH-DIR save changeset in PATCH-DIR
-n, --no-output do not save the changeset
-N, --forward  pass the --forward option to `patch'
-q, --quiet    no progress reports while computing changeset
```

Compute a patch set describing the changes from REVISION to the project tree containing DIR. Save the patch in PATCH-DIR (which must not already exist) and apply the patch in reverse to the project tree containing DIR.

The effect is to remove local changes in the DIR project tree but save them in a convenient form.

If REVISION is not specified, the latest ancestor of the default version of project tree is used.

If REVISION is specified as a VERSION, the latest ancestor of the project tree in that VERSION is used.

If DIR is not specified, "." is assumed.

If PATCH-DIR is not specified, a temporary file-name of the matching „undo-*" is used.

If --no-output is specified, the patch set is not saved.

See also "tla redo -H" and "tla changes -H".

redo : redo changes in project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
```

-V, --version Display a release identifier string and exit.

-N, --forward pass the --forward option to `patch`

-k, --keep do not delete the patch

-q, --quiet no progress reports while computing changeset

-d, --dir DIR Operate on project tree in DIR (default `.`)

Apply CHANGESET to the project tree and then delete CHANGESET.

If CHANGESET is not specified, the highest numbered „undo-N directory in the project tree root is used.

If --keep is given, the changeset directory is not deleted.

See also "tla undo --help" and "tla apply-changeset --help".

changes : report about local changes in a project tree

-h, --help Display a help message and exit.

-H Display a verbose help message and exit.

-V, --version Display a release identifier string and exit.

-A, --archive Override `my-default-archive`.

-d, --dir DIR Change to DIR first.

-o, --output DIR Save changeset in DIR (implies --keep).

-v, --verbose Verbose changeset report.

-q, --quiet Suppress progress information

--diffs Include diffs in the output.

-k, --keep Don't remove the output directory on termination.

--link hardlink unchanged files to revision library

Generate a patch report describing the differences between the project tree containing DIR (or the current directory) and REVISION.

The default patch level for a given version is the latest level for which the project tree has a patch. The default archive and version is as printed by "tla tree-version".

file-diffs : show local changes to a file

-h, --help Display a help message and exit.

-H Display a verbose help message and exit.

-V, --version Display a release identifier string and exit.

-A, --archive Override `my-default-archive`

-N, --new-file Treat missing file as empty

Print diffs between FILE and the corresponding file in a cached copy of REVISION.

The default patch level for a given version is the latest level for which the project tree has a patch. The default archive and version is as printed by "tla tree-version".

Project Tree Inventory Commands

inventory : inventory a source tree

-h, --help display help

-H Display a verbose help message and exit.

-V, --version display version info

-s, --source list source files

-p, --precious list precious files

-b, --backups list backup files

-j, --junk list junk files

-u, --unrecognized list unrecognized files

-t, --trees list roots of nested trees

```

-d, --directories    list only directories
-f, --files         list only non-directories
-B, --both          list both dirs and files
--kind              indicate file kinds

--all               include arch control files
--nested            include nested trees

--ids               list with ids (source files only)
--untagged          include files that are missing ids

--explicit          use explicit file ids
--implicit          permit implicit file ids
--tagline           permit tagline file ids
--names             use name-based file ids

```

With no arguments, print a human-readable inventory report.

With category options (--source etc) limit the report to just those files. With no other options, the report includes all sections and files.

The options -d, -f, and -b cancel each other.

If a directory is precious, junk, or unrecognized, only the directory name itself is printed -- its contents are not searched.

Each command option implies the corresponding category option (e.g. "--source-command" implies "--source"). The exit status of the command is ignored.

tree-lint : audit a source tree

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-s, --broken-symlinks Just list broken symlinks
-u, --unrecognized-files Just list files violating naming conventions
-t, --untagged-files  Just list files lacking inventory ids
-m, --missing-files   Just list inventory ids lacking corresponding files
-d, --duplicate-ids   Just list duplicated ids
--strict            exit with non-0 status on _any_ oddity

```

Audit a source tree for missing files, untagged files, duplicate ids, and files not matching recognized naming conventions.

The default is to list files failing any of those tests. Enabling any of the `specific' tests disables the rest, unless they're explicitly enabled on the command line. Therefore, "tla tree-lint" is equivalent to "tla tree-lint -sutmd"

id : report the inventory id for a file

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
--implicit          Use the implicit id tagging method.
--tagline           Use the tagline id tagging method (default).
--explicit          Use the explicit id tagging method.
--names            Use the names id tagging method.
--silent           No output -- exit status only.

```

Print a file's inventory id.

id-tagging-method : print or change a project tree id tagging method

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-d, --dir DIR  cd to DIR first
--strict      exit with error if method not set
```

Print or change the method by which source files are identified in DIR (or the current directory).

When setting, METHOD must be one of:

```
names          -- use naming conventions only
implicit       -- use naming conventions but permit
               for inventory tags
explicit       -- require explicit designation of source
tagline        -- mix names, explicit and arch-tag: methods
```

When printing, if --strict is provided but no id tagging method is explicitly set, print an error.

add-id : add an explicit inventory id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-i, --id ID    Specify ID, instead of using auto-generated id.
```

Create an explicit inventory id for FILE (which may be a regular file, symbolic link, or directory).

add : (alias for add-id)

add an explicit inventory id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-i, --id ID    Specify ID, instead of using auto-generated id.
```

Create an explicit inventory id for FILE (which may be a regular file, symbolic link, or directory).

delete-id : remove an explicit inventory id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

Remove an explicit inventory id for FILE (which may be a regular file, symbolic link, or directory).

delete : (alias for delete-id)

remove an explicit inventory id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

Remove an explicit inventory id for FILE (which may be a regular file, symbolic link, or directory).

rm : remove a file (or dir, or symlink) and its explicit inventory tag (if any)

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

move-id : move an explicit inventory id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

Move an explicit inventory id for FILE (which may be a regular file or symbolic link but which must not be a directory).

move : (alias for move-id)

move an explicit inventory id

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

Move an explicit inventory id for FILE (which may be a regular file or symbolic link but which must not be a directory).

mv : move a file (or dir, or symlink) and it's explicit inventory tag (if any)

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
```

This command is suitable as a replacement for mv(1) within any tla tree for any move operation, regardless of id tagging method.

explicit-default : print or modify default ids

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-D, --dir DIR  cd to DIR first
-d, --delete   remove the default
-s, --strong   use the strong default (default)
-w, --weak     use the weak default
--dont-care    use the dont-care default
```

For files in this directory, use:

```
ID-PREFIX__BASENAME
```

as the default explicit id for all files in this directory that do not have an explicit explicit id.

With no arguments, print the previously set ID-PREFIX.

By default, this command sets, prints or deletes a "strong default" -- a default explicit id which overrides implicit ids.

With --weak, set (or print) a weak default id which is overridden by explicit ids.

The --dont-care option sets (or with -d, clears) a flag for that directory that causes unidged files not to be reported as such in "tla tree-lint" reports.

default-id : (alias for explicit-default)

print or modify default ids

```

-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.

-D, --dir DIR   cd to DIR first
-d, --delete    remove the default
-s, --strong    use the strong default (default)
-w, --weak      use the weak default
--dont-care     use the dont-care default

```

For files in this directory, use:

```
ID-PREFIX__BASENAME
```

as the default explicit id for all files in this directory that do not have an explicit explicit id.

With no arguments, print the previously set ID-PREFIX.

By default, this command sets, prints or deletes a "strong default" -- a default explicit id which overrides implicit ids.

With --weak, set (or print) a weak default id which is overridden by explicit ids.

The --dont-care option sets (or with -d, clears) a flag for that directory that causes unidged files not to be reported as such in "tla tree-lint" reports.

id-tagging-defaults : print the default =tagging-method contents

```

-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.

```

Print the boilerplate contents that are used to initialize new =tagging-method files.

Patch Set Commands

changeset : compute a whole-tree changeset

```

-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.

--file-list FILES record only diffs of selected files

```

Create the output directory DESTINATION (it must not already exist).

Compare source trees ORIGINAL and MODIFIED. Create a changeset tree in DESTINATION

See also "tla apply-changeset -H".

mkpatch : (alias for changeset)

compute a whole-tree changeset

```

-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.

--file-list FILES record only diffs of selected files

```

Create the output directory DESTINATION (it must not already exist).

Compare source trees ORIGINAL and MODIFIED. Create a changeset tree in DESTINATION

See also "tla apply-changeset -H".

apply-changeset : apply a whole-tree changeset

```

-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-N, --forward   pass the --forward option to `patch`
-r, --reverse   Apply the changeset in reverse

```

Apply the patches in PATCHDIR to the source tree SRCDIR.

See also "tla changeset -H".

dopatch : (alias for apply-changeset)

apply a whole-tree changeset

```

-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-N, --forward   pass the --forward option to `patch`
-r, --reverse   Apply the changeset in reverse

```

Apply the patches in PATCHDIR to the source tree SRCDIR.

See also "tla changeset -H".

show-changeset : generate a report from a changeset

```

-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
--diffs        include diff output

```

Produce a human-readable summary of a changeset.

Archive Transaction Commands

make-archive : create a new archive directory

```

-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-m, --mirror MASTER      create mirror of specified archive
-M, --mirror-from MASTER-SOURCE  create pull-based mirror of specified archive
-l, --listing            Keep .listing files up-to-date in this archive.
-s, --signed            GPG sign the contents of this archive.

```

NAME is the global name for the archive. It must be an email address with a fully qualified domain name, optionally followed by "--" and a string of letters, digits, periods and dashes.

With --mirror, label the new archive as a mirror of MASTER. Ordinary commits can not be made to a mirror archive, however the command "tla archive-mirror" can write to a mirror.

Two special forms of this command are available:

```
tla make-archive --mirror MASTER LOCATION
```

is equivalent to:

```
tla make-archive --mirror MASTER MASTER-MIRROR LOCATION
```

and thus "tla archive-mirror MASTER" will push changes from MASTER to the newly created mirror.

```
tla make-archive --mirror-from MASTER-SOURCE LOCATION
```

is equivalent to:

```
tla make-archive --mirror MASTER-SOURCE MASTER LOCATION
```

and thus "tla archive-mirror MASTER" will pull changes from MASTER-SOURCE to the newly created mirror.

The first of these two special forms is especially handy when creating a remote mirror in which to publish a local repository. The second special form is handy when creating a local mirror of a remote archive, locally bound to the remote archive's name.

archive-setup : create new categories, branches and versions

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-f, --file FILE read the list of desired versions from FILE
-b, --branches  make base-0 tags
-c, --cache     archive cache when creating base-0 tags
```

Given a list of categories, branches, and versions, ensure that these exist in the appropriate archive.

For example, starting with an empty archive,:

```
% tla archive-setup proja--devo--1.0 projb
```

creates a category, branch, and version for project A, and a category for project B

When the --file option is used, the list of items to create is read from a file (use "-" to indicate standard input).

With the --file option, the option --branches may be provided. In this case, the input should have two items per line: a target version (to create), and a source version to build a tag from. For example, given an input file containing:

```
projb--devo--1.0 his@foo.com/proja--devo--1.0
```

with --branches, archive-setup creates a projb version in your default archive, and adds a base-0 revision to that project which is a tag of the latest revision in his@foo.com's archive. (This can make it quite easy to set-up a new archive which contains branches from existing archives, for example.)

If --branches is used, the --cache option may also be used. It causes each new base-0 revision to be archive-cached (see "tla cacherev -H")

make-category : create a new archive category

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
```

Create CATEGORY as a category in the indicated archive. The category must not already exist.

make-branch : create a new archive branch

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
```

Create BRANCH as a branch in the indicated archive.

The category containing the branch must already exist.

make-version : create a new archive version

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
```

Create VERSION as a version in the indicated archive. The branch for this version must not already exist.

import : archive a full-source base-0 revision

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.

-A, --archive       Override `my-default-archive'.
-d, --dir DIR       Change to DIR first.
-l, --log FILE      Commit with log file FILE.
-s, --summary TEXT  log with summary TEXT plus log-for-merge output
-L, --log-message TEXT log with TEXT
-S, --setup         Use `archive-setup' if necessary.
```

Archive a from-scratch base revision of the project tree containing DIR (or the current directory). Use this command to create the first revision of a new project.

If --log-message is specified without --summary, then TEXT is used both as the summary and the first line of the log body.

commit : archive a changeset-based revision

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.

-A, --archive       Override `my-default-archive'
-d, --dir DIR       cd to DIR first
-l, --log FILE      commit with log file FILE
-s, --summary TEXT  log with summary TEXT plus log-for-merge output
-L, --log-message TEXT log with TEXT plus log-for-merge output
--strict           strict tree-lint
--seal             create a version-0 revision
--fix              create a versionfix revision
--out-of-date-ok   commit even if out of date
--file-list FILE   commit only changes to files listed in FILE
```

Archive a revision of the project tree containing DIR (or the current directory)

If there are any automated ChangeLog files, update them before computing the patch set.

If --log-message is specified without --summary, then TEXT is used both as the summary and the first line of the log body.

get : construct a project tree for a revision

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.

-A, --archive       Override `my-default-archive'
--cache DIR         cache root for trees with pristines
--no-pristine       don't save a pristine copy
--link              hardlink files to revision library instead of copying
--library           ensure the revision is in a revision library
--sparse            add library entries sparsely (--link, --library)
--non-sparse        add library entries densely (--link, --library)
-s, --silent        no output
--no-greedy-add     do not allow greedy libraries to add revisions
```

Extract REVISION from an archive, creating the new project tree DIR. If DIR is not specified, store the working copy in a subdirectory of the current directory, giving it the name of the revision. CAUTION: when using the links option be sure to use copy-on-write on your editor or the revision library will be corrupted. Either way, the project tree must not already exist.

get-changeset : retrieve a changeset from an archive

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
```

Retrieve the changeset for REVISION and store it in the directory DIR (or a directory named REVISION.patches). DIR must not already exist.

lock-revision : lock (or unlock) an archive revision

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-u, --unlock    release a lock owned by you
-b, --break     break any existing lock
```

Acquire the lock needed to create REVISION.

Revision must be the next patch-level in sequence, however, if no patch level is specified, the lock is taken unconditionally. If no version is specified, the latest version is locked.

archive-mirror : update an archive mirror

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
--no-cached     don't copy cached revisions
-s, --summary   print the summary of each patch
--cached-tags   copy only cachedrevs for tags to other archives
```

If no arguments are given, update your `my-default-archive'-MIRROR archive with the contents of `my-default-archive'.

If a [FROM] archive is given, update the [FROM]-MIRROR archive with the contents of the [FROM] archive

If both [FROM] and [TO] archives are specified, update [TO] with the contents of [FROM]

If LIMIT is provided, it should be a category, branch, version, or revision name. Only the indicated part of FROM will be copied to TO. (see "tla make-archive -H").

Archive Commands

abrowse : print an outline describing archive contents

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-r, --reverse   sort revisions from newest to oldest
-s, --summary   print a summary of each patch
-c, --creator   print the creator of each patch
-D, --date      print the date of each patch
-k, --kind      show each revision kind (import, changeset or id)
-f, --full      print full patch level names
--desc         implies -s -c -D -k
```

```

--local-merges      list merges from the same archive
--foreign-merges   list merges from other archives
--merges            list all merges
--categories        show category names only
--branches          show branch names only
--versions          show version names only
--omit-empty        omit empty or unchanged-since items
--since SNAP-FILE  show revisions after those listed in SNAP-FILE
--since-limits      limit output to items in the since file
--snap SNAP-FILE    record the highest revisions shown
--force            overwrite an existing snap-file

```

Describe the contents of an archive in outline format.

With LIMIT, look only that part of the archive.

LIMIT may be a fully qualified name and may be an archive name.

rbrowse : print an outline describing an archive's contents

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
--all              Search all archives
--since SNAP-FILE  Only show differences since SNAP-FILE
--snap SNAP-FILE   Snap an archive
-f, --snap-force    Force snap writing
-r                 Reverse patch logs
-s                 Print the summary of matching patches
-c                 Print the creator of matching patches
-D                 Print the date of matching patches
--patch-regex REGEX Only show revisions that contain [REGEX]
                   (implies -s)
-A, --archive       Use [archive] instead of default
--show-sealed      Do not hide sealed branches

```

an outline of an archive will be printed, showing the categories, branches and revisions of the specified archive if no archive is given then `my-default-archive` is used.

If [LIMIT REGEX] is specified, revisions will only be shown if the category, branch or version matches [LIMIT REGEX]. If

--patch-regex [REGEX] is given, then only patchlogs matching [REGEX] will be given

categories : list the categories in an archive

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                   and exit.
-A, --archive       Override `my-default-archive'

```

Print a list of the categories present in an archive.

branches : list the branches in an archive category

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                   and exit.
-A, --archive       Override `my-default-archive'

```

Print a list of the branches within CATEGORY

versions : list the versions in an archive branch

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-r, --reverse   sort from newest to oldest
```

Print a list of versions within an archive branch.

The list is ordinarily sorted from oldest to newest, but the order can be changed with -r (--reverse).

revisions : list the revisions in an archive version

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-r, --reverse   sort from newest to oldest
-f, --full      list fully qualified names
-s, --summary   print a summary of each patch
-c, --creator   print the creator of each patch
-D, --date      print the date of each patch
```

Print a list of revisions within an archive version.

The list is ordinarily sorted from oldest to newest, but the order can be changed with -r (--reverse).

With optional arguments specifying patches, list only those patches, if they exist. If a listed patch does not exist, exit with status 1. The -r (--reverse) flag has no effect with optional arguments.

ancestry : display the ancestry of a revision

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-d, --dir DIR   cd to DIR first
-m, --merges    show merges into this development line
-r, --reverse   list oldest to newest
-s, --summary   print a summary of each patch
-c, --creator   print the creator of each patch
-D, --date      print the date of each patch
```

Print the ancestry of a revision.

ancestry-graph : display the ancestry of a revision

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-d, --dir DIR   cd to DIR first
-m, --merges    show merges into this development line
-r, --reverse   list oldest to newest
-i, --immediate show only the immediate ancestor
-p, --previous  show the (namespace) previous revision
```

Print a list describing the ancestry of a revision.

cat-archive-log : print the contents of an archived log entry

```
-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-A, --archive   Override `my-default-archive'
--headers      show only log headers
```

Retrieve and print the log message for the indicated revision from its archive.

cacherev : cache a full source tree in an archive

```
-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-A, --archive   Override `my-default-archive'
--cache DIR     cache root for trees with pristines
```

Cache a full-text copy of the indicated revision in the archive. This can speed up subsequent calls to "tla get" for that and subsequent revisions.

If no revision is specified, but the command is run from within a project tree, cache the latest revision in the default version of that tree.

cachedrevs : list cached revisions in an archive

```
-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-A, --archive   Override `my-default-archive'
```

Report which revisions of VERSION have been cached as whole trees in the archive.

See also "tla cacherev -H".

uncacherev : remove a cached full source tree from an archive

```
-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-A, --archive   Override `my-default-archive'
```

Remove the cached form of REVISION from its archive.

If REVISION is not specified, but the command is run from within a project tree, uncache the latest revision in the default version of that tree.

Also see "tla cacherev -H".

archive-meta-info : report meta-info from an archive

```
-h, --help      Display a help message and exit.
-H              Display a verbose help message and exit.
-V, --version   Display a release identifier string
               and exit.
-A, --archive   Override `my-default-archive'
```

Print the contents of a meta-info file from an archive. For example,

```
% tla archive-meta-info name
```

prints the official name of your default archive.

archive-snapshot : update an archive snapshot

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
```

Update the directory DIR with a "snapshot" of of an archive (or the part of the archive indicated by LIMIT

For each archive snapshotted, DIR will contain a file and subdirectory (where \$ARCH is the name of the archive):

```
./$ARCH.added
./$ARCH/
```

Similarly, for each category snapshotted, DIR will contain:

```
./$ARCH/$CAT.added
./$ARCH/$CAT/
```

and so on, recursively, for branches and versions.

For each revision, the snapshot contains:

```
./$ARCH/$CAT/$BRANCH/$VERSION/$REVISION.added
```

and that file contains a copy of the patch log entry for that revision.

Snapshots can be used in combination with other tools ('make' is suggested) to trigger one-time events in response to new additions to an archive.

archive-version : list the archive-version in an archive

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
```

Print the archive format identifier string of an arch archive.

archive-fixup : fix ancillary files (e.g. .listing files) in an archive

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
```

This is a kind of "catch-all" command to fix the non-standard, non-transactional state of an archive. Currently its sole function is to repair .listing files in archives that use them.

Patch Log Commands

make-log : initialize a new log file entry

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
```

`-d, --dir DIR` cd to DIR first

Create (if needed) a log entry file in the root of the project tree containing DIR, for the indicated VERSION.

The name of the new log file is printed to standard output.

log-versions : list patch log versions in a project tree

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.
-d, --dir DIR      cd to DIR first
-a, --archive ARCHIVE list only logs for ARCHIVE
-c, --category CATEGORY list only logs for CATEGORY
-b, --branch BRANCH list only logs for BRANCH
-v, --vsn VERSION  list only logs for VERSION
-r, --reverse      reverse the sort order
```

Print the list of versions for which there are patch logs in the project tree containing DIR (or the current directory).

add-log-version : add a patch log version to a project tree

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.
-A, --archive ARCHIVE Override `my-default-archive'
-d, --dir DIR      cd to DIR first
```

Add an initially empty list of patch log for VERSION to the project tree containing DIR (or the current directory).

A project tree patch log is a pre-requisite for checking the directory in on that version, or updating with patches from that version.

Note that versions recorded for a project tree are recorded by fully qualified version names: an archive name plus the name of a version within that archive.

remove-log-version : remove a version's patch log from a project tree

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.
-A, --archive ARCHIVE Override `my-default-archive'
-d, --dir DIR      cd to DIR first
```

Remove the patch log for VERSION from the project tree containing DIR (or the current directory).

Use this command with caution -- it erases revision history from the project tree!

logs : list patch logs for a version in a project tree

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.
-A, --archive ARCHIVE Override `my-default-archive'
-d, --dir DIR      cd to DIR first
-r, --reverse      sort from newest to oldest
-s, --summary      print the summary of each patch
-c, --creator      print the creator of each patch
-D, --date         print the date of each patch
```

```

--local-merges      list merges from the same archive
--foreign-merges   list merges from other archives
--merges            list all merges
-f, --full          print full patch level names

```

Print the list of patches applied to a project tree DIR (or the current directory) from VERSION.

cat-log : print the contents of a project tree log entry

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
-d, --dir DIR       cd to DIR first

```

Retrieve and print the patch log for the indicated revision from a project tree.

changelog : generate a ChangeLog from a patch log

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
-d, --dir DIR       cd to DIR first
--no-files          exclude file lists from ChangeLog
--untagged          don't implicitly tag the output file
--new-entry PATCH,FILE make FILE the first (top) entry
                    for patch level PATCH

```

Generate a ChangeLog for VERSION from the patch log for DIR.

log-for-merge : generate a log entry body for a merge

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
-d, --dir DIR       cd to DIR first
-r, --reverse       sort from newest to oldest

```

Generate a log entry body describing a merge into VERSION.

merges : report where two branches have been merged

```

-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
-d, --dir DIR       cd to DIR first
-r, --reverse       sort from newest to oldest
-f, --full          print full patch level names

```

Print a list of pairs of patch level names:

```
%s %s INTO-RVN FROM-RVN
```

where each pair indicates that at patch level INTO-RVN of INTO, the patch log entry FROM-RVN was added.

FROM may be a branch name, version name, or revision name. If a branch or version name, all merges from that branch or version are reported. If a revision name, only the merge points for that specific revision are reported.

INTO may be a version name or revision name. If a version name, all merge points within that version are printed. If a revision name, all merge points at that revision or earlier are printed.

Output is sorted using patch-level ordering of the first column.

Included patches are listed by full name unless FROM is a revision name. If FROM is a revision, --full causes its full name to be printed.

new-merges : list tree patches new to a version

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-d, --dir DIR   cd to DIR first
-r, --reverse   sort from newest to oldest
```

List all patch log entries in the tree containing DIR that have not already been merged with VERSION.

Multi-project Configuration Commands

build-config : instantiate a multi-project config

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-d, --dir DIR   cd to DIR first
--no-pristines  don't create pristine copies
--link         hardlink files to revision library instead of copying
--library      ensure revisions are in the revision library
--sparse       add library entries sparsely (--link, --library)
-r, --release-id  overwrite ./=RELEASE-ID for this config
```

Build the named configuration. See also "tla cat-config -H".

cat-config : output information about a multi-project config

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-d, --dir DIR   cd to DIR first
-o, --output CFG  write the output as config CFG
-f, --force     overwrite an exiting config (with --output)
-s, --snap      Show current patch levels of subtree packages.
```

Parse and print the indicate config file from a project tree

A config file contains blank lines, comment lines starting with "#", and config specification lines. The config file called \$NAME is stored in a project tree as ./configs/\$NAME or as ./\$NAME.

A config specification line contains a relative path within a project tree, and a specification of the project or revision to store in that location.

For example, the line:

```
./src/arch lord@emf.net--2003b/arch--devo--1.0
```

means that, when building the configuration, the latest revision of arch--devo--1.0 should be created within the tree as ./src/arch.

The project specification can be a branch name, version name or the name of a specific revision.

The option --snap says to examine the project tree to find out which revisions of configured project are printed, and generate a new config specification that references those specific revisions. For example, the output for the line shown above might be:

```
./src/arch lord@emf.net--2003b/arch--devo--1.0--patch-21
```

The option `--output` causes the output from this command to be recorded as a new configuration file (or to replace an existing file if `--force` is provided).

Commands for Branching and Merging

tag : create a continuation revision (aka tag or branch)

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.

-A, --archive      Override `my-default-archive'
-l, --log FILE     commit with log file FILE
--no-cacherev     Do not cacherev tag even if different archive
--seal            create a version-0 revision
--fix             create a versionfix revision
-S, --setup       use `archive-setup' if necessary.
```

Create the continuation revision TAG-VERSION (branch point or tag) which is equivalent to SOURCE-REVISION (plus a log entry).

If no log entry is provided, a trivial log entry will be created.

update : update a project tree to reflect recent archived changes

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.

-A, --archive      Override `my-default-archive'
-d, --dir DIR      Update project tree in DIR (default `.`)
-N, --forward     pass the --forward option to `patch'
--dest DEST       Instead of modifying the project tree in-place,
                  make a copy of it to DEST and apply the result to that
```

Update a project tree by getting the latest revision of VERSION (or the default version of the project tree) and then applying a patch set of the differences between the project tree and the highest revision with which it is up-to-date.

If the merge involves conflicts, a warning message is printed, and the new project tree will contain ".rej" files.

replay : apply revision changesets to a project tree

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.

-A, --archive      Override `my-default-archive'
--list FILE       read a list of patches to apply
--new             replay only new patches
--reverse        reverse the named patch
-N, --forward     pass the --forward option to `patch'
-d, --dir DIR     Operate on project tree in DIR (default `.`)
--dest DEST       Instead of modifying the project tree in-place,
                  make a copy of it to DEST and apply the result to that
--skip-present    skip patches that contain 1 or more patch logs already in this tree
```

The result is formed by applying patches in the latest revision of VERSION (or the default version of the project tree), stopping after the first patch that causes conflicts. If multiple VERSIONs are specified, they are applied in turn.

If one or more specific REVISIONs (including patch-levels) is specified instead, only those patch sets, and no others, will be applied. With the --list option, read a list of patches to apply from FILE (- for standard input). Complete revision names should be listed, one per line. replay will stop at the first patch in the list that causes a merge conflict, leaving behind files with names of the form:

```
,,replay.conflicts-in -- the name of the patch that caused conflicts
,,replay.remaining   -- the list of patches not yet applied
```

star-merge : merge mutually merged branches

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.
-A, --archive      Override tree archive
-c, --changes OUTPUT Generate but don't apply the changeset.
-r, --reference VERSION Set reference version (default: project tree version)
-N, --forward      pass the --forward option to `patch'
-t, --three-way    Perform a 3-way (diff3-style) merge.
-d, --dir DIR      Operate on project tree in DIR (default `.`)
```

Merge changes from FROM into the project tree, considering common ancestry from the project tree reference version, REFERENCE (which defaults to the tree version of the project tree). If FROM is not given, it defaults to the tree revision.

The purpose of this command is to merge changes between development lines which may have merged in both directions. It uses merge history to attempt to minimize the spurious conflicts that can arise from `mutual merging'.

FROM indicates a revision (it may be specified as a version name, indicating the latest revision in that version).

TREE is the project tree into which the merge will occur.

Star-merge works by computing the most recent ANCESTOR revision of REFERENCE and FROM and then applying the changeset:

```
delta (ANCESTOR, FROM)
```

to TREE.

The "most recent ancestor" is defined as follows:

MAYBE_ANCESTOR_1 is defined as the highest patch level of FROM in REFERENCE for which both TREE and FROM have a patch log. In other words, it is the latest REFERENCE revision of FROM's version already merged into TREE.

MAYBE_ANCESTOR_2 is defined as the highest patch level in REFERENCE for which both FROM and REFERENCE have a patch log. In other words, it is the latest revision of REFERENCE already merged into FROM.

MAYBE_ANCESTOR_2, if it is not "nil", was merged into FROM at some revision of FROM's version, which we can call LAST_MERGE_INTO_FROM.

If both MAYBE_ANCESTOR_1 or MAYBE_ANCESTOR_2 are nil, star-merge can do nothing.

If just one of MAYBE_ANCESTOR_1 is MAYBE_ANCESTOR_2 is not nil, then that non-nil value is ANCESTOR.

If both MAYBE_ANCESTOR_1 and MAYBE_ANCESTOR_2 are not nil, then LAST_MERGE_INTO_FROM and MAYBE_ANCESTOR_1 are compared (both are revisions in FROM's version). If MAYBE_ANCESTOR_1 is the later revision, then MAYBE_ANCESTOR_1 is ANCESTOR, otherwise, MAYBE_ANCESTOR_2 is ANCESTOR.

apply-delta : Compute a changeset between any two trees or revisions and apply it to a project tree

```
-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.
-A, --archive      Override `my-default-archive'
-N, --forward      pass the --forward option to `patch'
--cache DIR        specify a cache root for pristine copies
-d, --dir DIR      Operate on project tree in DIR (default `.`)
--dest DEST        Instead of modifying the project tree in-place,
                  make a copy of it to DEST and apply the result to that
```

A delta between A and B (both of which may be either a full revision or a project tree) is computed, and then applied to the project tree.

Exit Status Codes:

- 0 No conflict during patch
- 1 Conflicts occurred during patch
- 3 Internal Error

missing : print patches missing from a project tree

- h, --help Display a help message and exit.
- H Display a verbose help message and exit.
- V, --version Display a release identifier string and exit.
- A, --archive Override `my-default-archive'
- d, --dir DIR cd to DIR first
- r, --reverse sort from newest to oldest
- s, --summary display a summary of each missing patch
- c, --creator display the creator of each missing patch
- D, --date display the date of each missing patch
- f, --full print full revision names
- merges print a merge list for each missing patch
- skip-present skip patches that contain 1 or more patch logs already in this tree

Print a list of patches missing in the project tree containing DIR (or the current directory) for VERSION (or the default version. of the project tree).

The flag --merges means, for each patch, to print the list of patches included in the patch in two columns. For example:

```
PATCH-A      PATCH-A
PATCH-A      PATCH-B
PATCH-A      PATCH-C
```

means that PATCH-A includes the changes from PATCH-B and PATCH-C. (Every patch includes at least itself.)

join-branch : construct a project tree for a version

- h, --help Display a help message and exit.
- H Display a verbose help message and exit.
- V, --version Display a release identifier string and exit.
- A, --archive Override `my-default-archive'
- d, --dir DIR Operate on project tree in DIR (default `'.')
- dest DEST Instead of modifying the project tree in-place, make a copy of it to DEST and apply the result to that

REVISION must be a continuation (e.g. tag) of an ancestor of the project tree. The command replays the changeset for REVISION in the project tree, which has the effect of adding the log for the branch (making the tag an ancestor of the resulting tree).

As an exception to the usual rules, REVISION may be abbreviated as a version name, in which case the base-0 revision of that version is indicated.

sync-tree : unify a project tree's patch-log with a given revision

- h, --help Display a help message and exit.
- H Display a verbose help message and exit.
- V, --version Display a release identifier string and exit.
- A, --archive Override `my-default-archive'
- d, --dir DIR Operate on project tree in DIR (default `'.')
- dest DEST Instead of modifying the project tree in-place, make a copy of it to DEST and apply the result to that

The new project tree is formed by getting the REVISION and adding all patch-log entries from REVISION. No actual merging is performed -- only the patch-log is changed.

delta : Compute a changeset (or diff) between any two trees or revisions

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
--cache DIR    specify a cache root for pristine copies
--diffs       print changeset report with diffs
```

Given (REVISION|TREE)-A and (REVISION|TREE)-B, tla will build a changeset that comprises the changes between REVISION-A and REVISION-B

Example:

```
tla delta tla--devo--1.1--patch-6 \
          tla--devo--1.1--patch-8 ,,changes
```

Will pull patch-6 and patch-8 from tla--devo--1.1 and compute a changeset, which will be saved in a newly created ,,changes directory. If you would like a report instead, append the --diffs option

Local Cache Commands

changes : report about local changes in a project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string and exit.
-A, --archive   Override `my-default-archive'.
-d, --dir DIR   Change to DIR first.
-o, --output DIR Save changeset in DIR (implies --keep).
-v, --verbose   Verbose changeset report.
-q, --quiet     Suppress progress information
--diffs       Include diffs in the output.
-k, --keep     Don't remove the output directory on termination.
--link       hardlink unchanged files to revision library
```

Generate a patch report describing the differences between the project tree containing DIR (or the current directory) and REVISION. The default patch level for a given version is the latest level for which the project tree has a patch. The default archive and version is as printed by "tla tree-version".

file-diffs : show local changes to a file

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version   Display a release identifier string
              and exit.
-A, --archive   Override `my-default-archive'
-N, --new-file  Treat missing file as empty
```

Print diffs between FILE and the corresponding file in a cached copy of REVISION.

The default patch level for a given version is the latest level for which the project tree has a patch. The default archive and version is as printed by "tla tree-version".

file-find : find given version of file

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-N, --new-file Print missing file as `/dev/null'
```

Print location of file corresponding to FILE in a cached copy of REVISION

The default patch level for a given version is the latest level for which the project tree has a patch. The default archive and version is as printed by "tla tree-version".

pristines : list pristine trees in a project tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-d, --dir DIR  cd to DIR first
-u, --unlocked return only unlocked pristines
-l, --locked   return only locked pristines
-r, --reverse  reverse sort order
```

Print the list of pristine revisions cached in project tree DIR (or the current directory).

lock-pristine : lock (or unlock) a pristine tree

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-d, --dir DIR  cd to DIR first
-u, --unlock   unlock, rather than lock
```

Lock the indicated pristine REVISION in the project tree containing DIR (or the current directory). This prevents the revision from being automatically recycled.

add-pristine : ensure that a project tree has a particular pristine revision

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-d, --dir DIR  cd to DIR first
```

Extract REVISION from an archive, creating a pristine copy. Store that pristine copy in the current directory's project tree

find-pristine : find and print the path to a pristine revision

```
-h, --help      Display a help message and exit.
-H             Display a verbose help message and exit.
-V, --version  Display a release identifier string
              and exit.
-A, --archive  Override `my-default-archive'
-d, --dir DIR  cd to DIR first
-u, --unlocked return only an unlocked pristine
-l, --locked   return only a locked pristine
```

-t, --tree-only search this tree only, not siblings
-s, --silent exit status only

Print the location of a pristine copy of the indicated revision

Revision Library Commands

my-revision-library : print or change your revision library path

-h, --help Display a help message and exit.
-H Display a verbose help message and exit.
-V, --version Display a release identifier string and exit.
-e, --errname specify program name for errors
-d, --delete unspecify your revision library
-s, --silent suppress reassuring messages
--search use the full search path
--add use the full add path
--search-only use the search-only path
--add-only use the add-only path
--first prepend to the path if setting (default appends)

With no argument, and without -d, print the path to your revision library.

With an argument, record DIR as the path to your revision library in ~/.arch-params/=revision-library

With the option -d (--delete) and no argument, ensure that you do not have a revision library path set in ~/.arch-params.

If no revision library is set, the program exits with status 1, printing an error message unless the -s (--silent) option is given.

library-config : configure parameters of a revision library

-h, --help Display a help message and exit.
-H Display a verbose help message and exit.
-V, --version Display a release identifier string and exit.
--greedy make the library greedy
--non-greedy make the library not greedy
--sparse make the library sparse (by default)
--non-sparse make the library not sparse (by default)

Set/show various parameters for a revision library.

library-find : find and print the location of a revision in the revision library

-h, --help Display a help message and exit.
-H Display a verbose help message and exit.
-V, --version Display a release identifier string and exit.
-A, --archive Override `my-default-archive'
-e, --errname specify program name for errors
-s, --silent suppress reassuring messages

Find REVISION in your revision library and print its path. If the revision is not present, print an error message (unless --silent is specified) and exit with non-0 status.

library-add : add a revision to the revision library

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
-s, --sparse        Don't fill in gaps in the library.
--non-sparse        Fill in gaps in the library.
-L, --library LIB   specify which library to add to
--for-links PATH    require a lib on the same device as PATH
```

Add REVISION to your revision library.

This command has no effect and exits with status 0 if the indicated revision is already in the library.

library-remove : remove a revision from the revision library

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
```

Remove REVISION from your revision library.

If the revision is not present, print an error message and exit with non-0 status.

library-archives : list the archives in your revision library

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
```

List all archives with records in the revision library.

library-categories : list the categories in your revision library

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
```

List all categories within a given archive in the revision library.

library-branches : list the branches in a library category

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
```

List all branches within CATEGORY in the revision library.

library-versions : list the versions in a library branch

```
-h, --help          Display a help message and exit.
-H                  Display a verbose help message and exit.
-V, --version       Display a release identifier string
                    and exit.
-A, --archive       Override `my-default-archive'
```

`-r, --reverse` sort from newest to oldest

List all versions within a particular archive/branch with records in the revision library.

library-revisions : list the revisions in a library version

`-h, --help` Display a help message and exit.

`-H` Display a verbose help message and exit.

`-V, --version` Display a release identifier string and exit.

`-A, --archive` Override ``my-default-archive'`

`-r, --reverse` sort from newest to oldest

`-f, --full` list fully qualified names

`-s, --summary` print a summary of each patch

`-c, --creator` print the creator of each patch

`-D, --date` print the date of each patch

List all revisions within a particular archive/version with records in the revision library.

library-log : output a log message from the revision library

`-h, --help` Display a help message and exit.

`-H` Display a verbose help message and exit.

`-V, --version` Display a release identifier string and exit.

`-A, --archive` Override ``my-default-archive'`

`-s, --silent` suppress reassuring messages

Print the log message for REVISION from the library.

library-file : find a file in a revision library

`-h, --help` Display a help message and exit.

`-H` Display a verbose help message and exit.

`-V, --version` Display a release identifier string and exit.

`-A, --archive` Override ``my-default-archive'`

`--id` interpret FILE as an inventory id

`--this` interpret FILE as a file within a project tree

Find FILE in REVISION in your revision library and print its path. If the revision or file is not present, print an error message and exit with non-0 status.

Published Revisions Commands

grab : grab a published revision

`-h, --help` Display a help message and exit.

`-H` Display a verbose help message and exit.

`-V, --version` Display a release identifier string and exit.

Grabs a published revision from LOCATION.

Miscellaneous Scripting Support

parse-package-name : parse a package name

```

-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.

-A, --archive      Override `my-default-archive'
-a, --arch         print the archive name
--non-arch         print the non-archive part of the name
-c, --category     print the category name
-b, --branch       print the branch name
-p, --package      print the package name
-v, --vsn          print the version id
--package-version print the category--branch--version
-l, --lvl          print the patch level
--patch-level     synonym for --lvl

```

Options -b, -B, -p, -v, and -l cancel each other.

For -b, -v, and -l, just the id string is printed without "--".

For -p, the output is "--".

valid-package-name : test a package name for validity

```

-h, --help          Display a help message and exit.
-H                 Display a verbose help message and exit.
-V, --version      Display a release identifier string
                  and exit.

-e, --errname      specify program name for errors
--archive          require and explicit archive
--no-archive       prohibit and explicit archive
-c, --category     require a category
-p, --package      require category, permit branch
-v, --vsn          require a version number
-l, --patch-level  require a patch level
--lvl             synonym for --patch-level
-t, --tolerant     tolerate more specific names

```

Exit with status 0 if PACKAGE is a valid package name, status 1 otherwise.

By default, require a basename or basename plus branch label.

Options -v and -l cancel -b and vice versa.

Option -l implies -v.

If an error name is specified (-e or --errname), then invalid names cause an error message on stdout. Otherwise, the exit status is the only output.

By default, validation is strict. For example, -b checks for a valid basename and does not permit a branch label or version number.

With -t, more specific names are permitted. For example, -b will permit a branch name, version number, and patch level.