NAME

bzr − Bazaar next−generation distributed version control

SYNOPSIS

bzr *command* [*command_options*]
bzr help
bzr help *command*

DESCRIPTION

Bazaar (or bzr) is a distributed version control system that is powerful, friendly, and scalable. Bazaar is a project of Canonical Ltd and part of the GNU Project to develop a free operating system.

Bazaar keeps track of changes to software source code (or similar information); lets you explore who changed it, when, and why; merges concurrent changes; and helps people work together in a team.

COMMAND OVERVIEW

bzr add [FILE...]
Add specified files or directories.

bzr alias [NAME]
Set/unset and display aliases.

bzr annotate FILENAME
Show the origin of each line in a file.

bzr bind [LOCATION]
Convert the current branch into a checkout of the supplied branch.

bzr branch FROM_LOCATION [TO_LOCATION]
Create a new branch that is a copy of an existing branch.

bzr branches [LOCATION]
List the branches available at the current location.

bzr break−lock [LOCATION]
Break a dead lock.

bzr cat FILENAME
Write the contents of a file as of a given revision to standard output.

bzr check [PATH]
Validate working tree structure, branch consistency and repository history.

bzr checkout [BRANCH_LOCATION] [TO_LOCATION]
Create a new checkout of an existing branch.

bzr clean−tree
Remove unwanted files from working tree.

bzr commit [SELECTED...]
Commit changes into a new revision.

bzr config [NAME]
Display, set or remove a configuration option.

bzr conflicts
List files with conflicts.

bzr deleted
List files deleted in the working tree.

bzr diff [FILE...]
Show differences in the working tree, between revisions or branches.

bzr dpush [LOCATION]
Push into a different VCS without any custom bzr metadata.

bzr export DEST [BRANCH_OR_SUBDIR]
Export current or past revision to a destination directory or archive.

bzr help [TOPIC]
Show help on a command or other topic.

bzr ignore [NAME_PATTERN...]
Ignore specified files or patterns.

bzr ignored
List ignored files and the patterns that matched them.

bzr info [LOCATION]
Show information about a working tree, branch or repository.

bzr init [LOCATION]
Make a directory into a versioned branch.

bzr init−repository LOCATION
Create a shared repository for branches to share storage space.

bzr join TREE
Combine a tree into its containing tree.

bzr launchpad−login [NAME]
Show or set the Launchpad user ID.

bzr launchpad−mirror [LOCATION]
Ask Launchpad to mirror a branch now.

bzr launchpad−open [LOCATION]
Open a Launchpad branch page in your web browser.

bzr log [FILE...]
Show historical log for a branch or subset of a branch.

bzr lp−find−proposal
Find the proposal to merge this revision.

bzr lp−propose−merge [SUBMIT_BRANCH]
Propose merging a branch on Launchpad.

bzr ls [PATH]
List files in a tree.

bzr merge [LOCATION]
Perform a three−way merge.

bzr missing [OTHER_BRANCH]
Show unmerged/unpulled revisions between two branches.

bzr mkdir DIR...
Create a new versioned directory.

bzr mv [NAMES...]
Move or rename a file.

bzr nick [NICKNAME]
Print or set the branch nickname.

bzr pack [BRANCH_OR_REPO]
Compress the data within a repository.

bzr ping LOCATION
Pings a Bazaar smart server.

bzr plugins
List the installed plugins.

bzr pull [LOCATION]
Turn this branch into a mirror of another branch.

bzr push [LOCATION]
Update a mirror of this branch.

bzr reconcile [BRANCH]
Reconcile bzr metadata in a branch.

bzr reconfigure [LOCATION]
Reconfigure the type of a bzr directory.

bzr register−branch [PUBLIC_URL]
Register a branch with launchpad.net.

bzr remerge [FILE...]
Redo a merge.

bzr remove [FILE...]
Remove files or directories.

bzr remove−branch [LOCATION]
Remove a branch.

bzr remove−tree [LOCATION...]
Remove the working tree from a given branch/checkout.

bzr renames [DIR]
Show list of renamed files.

bzr resolve [FILE...]
Mark a conflict as resolved.

bzr revert [FILE...]
Set files in the working tree back to the contents of a previous revision.

bzr revno [LOCATION]
Show current revision number.

bzr root [FILENAME]
Show the tree root directory.

bzr send [SUBMIT_BRANCH] [PUBLIC_BRANCH]
Mail or create a merge−directive for submitting changes.

bzr serve
Run the bzr server.

bzr shelve [FILE...]
Temporarily set aside some changes from the current tree.

bzr sign−my−commits [LOCATION] [COMMITTER]
Sign all commits by a given committer.

bzr split TREE
Split a subdirectory of a tree into a separate tree.

bzr status [FILE...]
Display status summary.

bzr switch [TO_LOCATION]
Set the branch of a checkout and update.

bzr tag [TAG_NAME]
Create, remove or modify a tag naming a revision.

bzr tags
List tags.

bzr testament [BRANCH]
Show testament (signing−form) of a revision.

bzr unbind
Convert the current checkout into a regular branch.

bzr uncommit [LOCATION]
Remove the last committed revision.

bzr unshelve [SHELF_ID]
Restore shelved changes.

bzr update [DIR]
Update a working tree to a new revision.

bzr upgrade [URL]
Upgrade a repository, branch or working tree to a newer format.

bzr verify−signatures [LOCATION]
Verify all commit signatures.

bzr version
Show version of bzr.

bzr version−info [LOCATION]
Show version information about this tree.

bzr view [FILE...]
Manage filtered views.

bzr whoami [NAME]
Show or set bzr user id.

COMMAND REFERENCE

bzr −−help
Alias for "help", see "bzr help".

bzr −?
Alias for "help", see "bzr help".

bzr −h
Alias for "help", see "bzr help".

bzr ?
Alias for "help", see "bzr help".

bzr add [FILE...]
Options:
−−dry−run Show what would be done, but don't actually do anything.
−−file−ids−from ARG Lookup file ids from this tree.
−−help, −h Show help message.
−−no−recurse, −N Don't recursively add the contents of directories.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.

−−verbose, −v Display more information.

See also: ignore, remove

Add specified files or directories.

In non−recursive mode, all the named items are added, regardless of whether they were previously ignored. A warning is given if any of the named files are already versioned.

In recursive mode (the default), files are treated the same way but the behaviour for directories is different. Directories that are already versioned do not give a warning. All directories, whether already versioned or not, are searched for files or subdirectories that are neither versioned or ignored, and these are added. This search proceeds recursively into versioned directories. If no names are given '.' is assumed.

A warning will be printed when nested trees are encountered, unless they are explicitly ignored.

Therefore simply saying 'bzr add' will version all files that are currently unknown.

Adding a file whose parent directory is not versioned will implicitly add the parent, and so on up to the root. This means you should never need to explicitly add a directory, they'll just get added when you add a file in the directory.

−−dry−run will show which files would be added, but not actually add them.

−−file−ids−from will try to use the file ids from the supplied path. It looks up ids trying to find a matching parent directory with the same filename, and then by pure path. This option is rarely needed but can be useful when adding the same logical file into two branches that will be merged later (without showing the two different adds as a conflict). It is also useful when merging another project into a subdirectory of this one.

Any files matching patterns in the ignore list will not be added unless they are explicitly mentioned.

In recursive mode, files larger than the configuration option add.maximum_file_size will be skipped. Named items are never skipped due to file size.

bzr alias [NAME]

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−remove Remove the alias.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Set/unset and display aliases.

Examples:

Show the current aliases:

```
bzr alias
```

Show the alias specified for 'll':

```
bzr alias ll
```

Set an alias for 'll':

bzr alias ll="log −−line −r−10..−1"

To remove an alias for 'll':

bzr alias −−remove ll

bzr ann

Alias for "annotate", see "bzr annotate".

bzr annotate FILENAME

Options:

−−all Show annotations on all lines.
−−directory ARG, −d Branch to operate on, instead of working directory.
−−help, −h Show help message.
−−long Show commit date in annotations.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−show−ids Show internal object ids.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Aliases: ann, blame, praise

Show the origin of each line in a file.

This prints out the given file with an annotation on the left side indicating which revision, author and date introduced the change.

If the origin is the same for a run of consecutive lines, it is shown only at the top, unless the −−all option is given.

bzr bind [LOCATION]

Options:

−−directory ARG, −d Branch to operate on, instead of working directory.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: checkouts, unbind

Convert the current branch into a checkout of the supplied branch. If no branch is supplied, rebind to the last bound location.

Once converted into a checkout, commits must succeed on the master branch before they will be applied to the local branch.

Bound branches use the nickname of its master branch unless it is set locally, in which case binding will update the local nickname to be that of the master.

bzr blame

Alias for "annotate", see "bzr annotate".

bzr branch FROM_LOCATION [TO_LOCATION]

Options:

```
--bind                  Bind new branch to from location.
--files-from ARG        Get file contents from this tree.
--hardlink              Hard-link working tree files where
                        possible.
--help, -h              Show help message.
--no-tree               Create a branch without a working-tree.
--quiet, -q             Only display errors and warnings.
--revision ARG, -r      See "help revisionspec" for details.
--stacked               Create a stacked branch referring to the
                        source branch. The new branch will
                        depend on the availability of the source
                        branch for all operations.
--standalone            Do not use a shared repository, even if
                        available.
--switch                Switch the checkout in the current
                        directory to the new branch.
--usage                 Show usage message and options.
--use-existing-dir      By default branch will fail if the
                        target directory exists, but does not
                        already have a control directory.  This
                        flag will allow branch to proceed.
--verbose, -v           Display more information.
```

Aliases: get, clone

See also: checkout

Create a new branch that is a copy of an existing branch.

If the TO_LOCATION is omitted, the last component of the FROM_LOCATION will be used. In other words, "branch ../foo/bar" will attempt to create ./bar. If the FROM_LOCATION has no / or path separator embedded, the TO_LOCATION is derived from the FROM_LOCATION by stripping a leading scheme or drive identifier, if any. For example, "branch lp:foo-bar" will attempt to create ./foo-bar.

To retrieve the branch as of a particular revision, supply the --revision parameter, as in "branch foo/bar -r 5".

The synonyms 'clone' and 'get' for this command are deprecated.

bzr branches [LOCATION]

Options:

```
--help, -h              Show help message.
--quiet, -q             Only display errors and warnings.
--recursive, -R         Recursively scan for branches rather
                        than just looking in the specified
                        location.
--usage                 Show usage message and options.
--verbose, -v           Display more information.
```

List the branches available at the current location.

This command will print the names of all the branches at the current location.

bzr break−lock [LOCATION]

Options:

−−config LOCATION is the directory where the config lock is.
−−force Do not ask for confirmation before breaking the lock.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Break a dead lock.

This command breaks a lock on a repository, branch, working directory or config file.

CAUTION: Locks should only be broken when you are sure that the process holding the lock has been stopped.

You can get information on what locks are open via the 'bzr info [location]' command.

Examples:

```
bzr break−lock
bzr break−lock bzr+ssh://example.com/bzr/foo
bzr break−lock −−conf ~/.bazaar
```

bzr cat FILENAME

Options:

−−directory ARG, −d Branch to operate on, instead of working directory.
−−filters Apply content filters to display the convenience form.
−−help, −h Show help message.
−−name−from−revision The path name in the old tree.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: ls

Write the contents of a file as of a given revision to standard output.

If no revision is nominated, the last revision is used.

Note: Take care to redirect standard output when using this command on a binary file.

bzr check [PATH]

Options:

−−branch Check the branch related to the current directory.
−−help, −h Show help message.

−−quiet, −q	Only display errors and warnings.
−−repo	Check the repository related to the current directory.
−−tree	Check the working tree related to the current directory.
−−usage	Show usage message and options.
−−verbose, −v	Display more information.

See also: reconcile

Validate working tree structure, branch consistency and repository history.

This command checks various invariants about branch and repository storage to detect data corruption or bzr bugs.

The working tree and branch checks will only give output if a problem is detected. The output fields of the repository check are:

revisions
: This is just the number of revisions checked. It doesn't indicate a problem.

versionedfiles
: This is just the number of versionedfiles checked. It doesn't indicate a problem.

unreferenced ancestors
: Texts that are ancestors of other texts, but are not properly referenced by the revision ancestry. This is a subtle problem that Bazaar can work around.

unique file texts
: This is the total number of unique file contents seen in the checked revisions. It does not indicate a problem.

repeated file texts
: This is the total number of repeated texts seen in the checked revisions. Texts can be repeated when their file entries are modified, but the file contents are not. It does not indicate a problem.

If no restrictions are specified, all Bazaar data that is found at the given location will be checked.

Examples:

Check the tree and branch at 'foo':

```
bzr check −−tree −−branch foo
```

Check only the repository at 'bar':

```
bzr check −−repo bar
```

Check everything at 'baz':

bzr check baz

bzr checkin

Alias for "commit", see "bzr commit".

bzr checkout [BRANCH_LOCATION] [TO_LOCATION]

Options:

```
--files-from ARG        Get file contents from this tree.
--hardlink              Hard-link working tree files where
                        possible.
--help, -h              Show help message.
--lightweight           Perform a lightweight checkout.
                        Lightweight checkouts depend on access
                        to the branch for every operation.
                        Normal checkouts can perform common
                        operations like diff and status without
                        such access, and also support local
                        commits.
--quiet, -q             Only display errors and warnings.
--revision ARG, -r      See "help revisionspec" for details.
--usage                 Show usage message and options.
--verbose, -v           Display more information.
```

Alias: co

See also: branch, checkouts, remove-tree, working-trees

Create a new checkout of an existing branch.

If BRANCH_LOCATION is omitted, checkout will reconstitute a working tree for the branch found in '.'. This is useful if you have removed the working tree or if it was never created - i.e. if you pushed the branch to its current location using SFTP.

If the TO_LOCATION is omitted, the last component of the BRANCH_LOCATION will be used. In other words, "checkout ../foo/bar" will attempt to create ./bar. If the BRANCH_LOCATION has no / or path separator embedded, the TO_LOCATION is derived from the BRANCH_LOCATION by stripping a leading scheme or drive identifier, if any. For example, "checkout lp:foo-bar" will attempt to create ./foo-bar.

To retrieve the branch as of a particular revision, supply the --revision parameter, as in "checkout foo/bar -r 5". Note that this will be immediately out of date [so you cannot commit] but it may be useful (i.e. to examine old code.)

bzr ci

Alias for "commit", see "bzr commit".

bzr clean-tree

Options:

```
--detritus              Delete conflict files, merge and revert
                        backups, and failed selftest dirs.
--directory ARG, -d     Branch to operate on, instead of working
                        directory.
--dry-run               Show files to delete instead of deleting
                        them.
--force                 Do not prompt before deleting.
--help, -h              Show help message.
```

--ignored Delete all ignored files.
--quiet, -q Only display errors and warnings.
--unknown Delete files unknown to bzr (default).
--usage Show usage message and options.
--verbose, -v Display more information.

Remove unwanted files from working tree.

By default, only unknown files, not ignored files, are deleted. Versioned files are never deleted.

Another class is 'detritus', which includes files emitted by bzr during normal operations and selftests. (The value of these files decreases with time.)

If no options are specified, unknown files are deleted. Otherwise, option flags are respected, and may be combined.

To check what clean-tree will do, use --dry-run.

bzr clone
Alias for "branch", see "bzr branch".

bzr co
Alias for "checkout", see "bzr checkout".

bzr commit [SELECTED...]
Options:
--author ARG Set the author's name, if it's different from the committer.
--commit-time ARG Manually set a commit time using commit date format, e.g. '2009-10-10 08:00:00 +0100'.
--exclude ARG, -x Do not consider changes made to a given path.
--file MSGFILE, -F Take commit message from this file.
--fixes ARG Mark a bug as being fixed by this revision (see "bzr help bugs").
--help, -h Show help message.
--local Perform a local commit in a bound branch. Local commits are not pushed to the master branch until a normal commit is performed.
--lossy When committing to a foreign version control system do not push data that can not be natively represented.
--message ARG, -m Description of the new revision.
--quiet, -q Only display errors and warnings.
--show-diff, -p When no message is supplied, show the diff along with the status summary in the message editor.
--strict Refuse to commit if there are unknown files in the working tree.
--unchanged Commit even if nothing has changed.
--usage Show usage message and options.
--verbose, -v Display more information.

Aliases: ci, checkin

See also: add, bugs, hooks, uncommit

Commit changes into a new revision.

An explanatory message needs to be given for each commit. This is often done by using the −−message option (getting the message from the command line) or by using the −−file option (getting the message from a file). If neither of these options is given, an editor is opened for the user to enter the message. To see the changed files in the boilerplate text loaded into the editor, use the −−show−diff option.

By default, the entire tree is committed and the person doing the commit is assumed to be the author. These defaults can be overridden as explained below.

Selective commits:

If selected files are specified, only changes to those files are committed. If a directory is specified then the directory and everything within it is committed.

When excludes are given, they take precedence over selected files. For example, to commit only changes within foo, but not changes within foo/bar:

bzr commit foo −x foo/bar

A selective commit after a merge is not yet supported.

Custom authors:

If the author of the change is not the same person as the committer, you can specify the author's name using the −−author option. The name should be in the same format as a committer−id, e.g. "John Doe <jdoe@example.com>". If there is more than one author of the change you can specify the option multiple times, once for each author.

Checks:

A common mistake is to forget to add a new file or directory before running the commit command. The −−strict option checks for unknown files and aborts the commit if any are found. More advanced pre−commit checks can be implemented by defining hooks. See *(Aq*(Aqbzr help hooks*(Aq*(Aq for details.

Things to note:

If you accidentially commit the wrong changes or make a spelling mistake in the commit message say, you can use the uncommit command to undo it. See *(Aq*(Aqbzr help uncommit*(Aq*(Aq for details.

Hooks can also be configured to run after a commit. This allows you to trigger updates to external systems like bug trackers. The −−fixes option can be used to record the association between a revision and

one or more bugs. See *(Aq*(Aqbzr help bugs*(Aq*(Aq for details.

bzr config [NAME]

Options:

```
--all                 Display all the defined values for the
                      matching options.
--directory ARG, -d   Branch to operate on, instead of working
                      directory.
--help, -h            Show help message.
--quiet, -q           Only display errors and warnings.
--remove              Remove the option from the configuration
                      file.
--scope ARG           Reduce the scope to the specified
                      configuration file.
--usage               Show usage message and options.
--verbose, -v         Display more information.
```

See also: configuration

Display, set or remove a configuration option.

Display the active value for option NAME.

If --all is specified, NAME is interpreted as a regular expression and all matching options are displayed mentioning their scope and without resolving option references in the value). The active value that bzr will take into account is the first one displayed for each option.

If NAME is not given, --all .* is implied (all options are displayed for the current scope).

Setting a value is achieved by using NAME=value without spaces. The value is set in the most relevant scope and can be checked by displaying the option again.

Removing a value is achieved by using --remove NAME.

bzr conflicts

Options:

```
--directory ARG, -d   Branch to operate on, instead of working
                      directory.
--help, -h            Show help message.
--quiet, -q           Only display errors and warnings.
--text                List paths of files with text conflicts.
--usage               Show usage message and options.
--verbose, -v         Display more information.
```

See also: conflict-types, resolve

List files with conflicts.

Merge will do its best to combine the changes in two branches, but there are some kinds of problems only a human can fix. When it encounters those, it will mark a conflict. A conflict means that you need to fix something, before you can commit.

Conflicts normally are listed as short, human-readable messages. If --text is supplied, the pathnames of files with text conflicts are listed, instead. (This is useful for editing all files with text conflicts.)

Use bzr resolve when you have fixed a problem.

bzr del
Alias for "remove", see "bzr remove".

bzr deleted
Options:
--directory ARG, -d Branch to operate on, instead of working directory.
--help, -h Show help message.
--quiet, -q Only display errors and warnings.
--show-ids Show internal object ids.
--usage Show usage message and options.
--verbose, -v Display more information.

See also: ls, status

List files deleted in the working tree.

bzr di
Alias for "diff", see "bzr diff".

bzr dif
Alias for "diff", see "bzr diff".

bzr diff [FILE...]
Options:
--change ARG, -c Select changes introduced by the specified revision. See also "help revisionspec".
--context ARG How many lines of context to show.
--diff-options ARG Pass these options to the external diff program.
--format ARG, -F Diff format to use.
--help, -h Show help message.
--new ARG Branch/tree to compare to.
--old ARG Branch/tree to compare from.
--prefix ARG, -p Set prefixes added to old and new filenames, as two values separated by a colon. (eg "old/:new/").
--quiet, -q Only display errors and warnings.
--revision ARG, -r See "help revisionspec" for details.
--usage Show usage message and options.
--using ARG Use this command to compare files.
--verbose, -v Display more information.

Aliases: di, dif

See also: status

Show differences in the working tree, between revisions or branches.

If no arguments are given, all changes for the current tree are listed. If files are given, only the changes in those files are listed. Remote and multiple branches can be compared by using the --old and --new

options. If not provided, the default for both is derived from the first argument, if any, or the current tree if no arguments are given.

"bzr diff −p1" is equivalent to "bzr diff −−prefix old/:new/", and produces patches suitable for "patch −p1".

Note that when using the −r argument with a range of revisions, the differences are computed between the two specified revisions. That is, the command does not show the changes introduced by the first revision in the range. This differs from the interpretation of revision ranges used by "bzr log" which includes the first revision in the range.

Exit values:
1 − changed
2 − unrepresentable changes
3 − error
0 − no change

Examples:
Shows the difference in the working tree versus the last commit:

bzr diff

Difference between the working tree and revision 1:

bzr diff −r1

Difference between revision 3 and revision 1:

bzr diff −r1..3

Difference between revision 3 and revision 1 for branch xxx:

bzr diff −r1..3 xxx

The changes introduced by revision 2 (equivalent to −r1..2):

bzr diff −c2

To see the changes introduced by revision X:

bzr diff −cX

Note that in the case of a merge, the −c option shows the changes compared to the left hand parent. To see the changes against another parent, use:

bzr diff −r<chosen_parent>..X

The changes between the current revision and the previous revision (equivalent to −c−1 and −r−2..−1)

bzr diff −r−2..

Show just the differences for file NEWS:

bzr diff NEWS

Show the differences in working tree xxx for file NEWS:

bzr diff xxx/NEWS

Show the differences from branch xxx to this working tree:

bzr diff −−old xxx

Show the differences between two branches for file NEWS:

bzr diff −−old xxx −−new yyy NEWS

Same as 'bzr diff' but prefix paths with old/ and new/:

bzr diff −−prefix old/:new/

Show the differences using a custom diff program with options:

bzr diff −−using /usr/bin/diff −−diff−options −wu

bzr dpush [LOCATION]

Options:

−−directory ARG, −d Branch to push from, rather than the one containing the working directory.
−−help, −h Show help message.
−−no−rebase Do not rebase after push.
−−quiet, −q Only display errors and warnings.
−−remember Remember the specified location as a default.
−−strict Refuse to push if there are uncommitted changes in the working tree, −−no−strict disables the check.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Push into a different VCS without any custom bzr metadata.

This will afterwards rebase the local branch on the remote branch unless the −−no−rebase option is used, in which case the two branches will be out of sync after the push.

bzr export DEST [BRANCH_OR_SUBDIR]

Options:

−−directory ARG, −d Branch to operate on, instead of working directory.
−−filters Apply content filters to export the convenient form.
−−format ARG Type of file to export to.
−−help, −h Show help message.
−−per−file−timestamps Set modification time of files to that of the last revision in which it was changed.
−−quiet, −q Only display errors and warnings.

```
--revision ARG, -r       See "help revisionspec" for details.
--root ARG               Name of the root directory inside the
                         exported file.
--uncommitted            Export the working tree contents rather
                         than that of the last revision.
--usage                  Show usage message and options.
--verbose, -v            Display more information.
```

Export current or past revision to a destination directory or archive.

If no revision is specified this exports the last committed revision.

Format may be an "exporter" name, such as tar, tgz, tbz2. If none is given, try to find the format with the extension. If no extension is found exports to a directory (equivalent to --format=dir).

If root is supplied, it will be used as the root directory inside container formats (tar, zip, etc). If it is not supplied it will default to the exported filename. The root option has no effect for 'dir' format.

If branch is omitted then the branch containing the current working directory will be used.

Note: Export of tree with non-ASCII filenames to zip is not supported.

Supported formats	Autodetected by extension
dir	(none)
tar	.tar
tbz2	.tar.bz2, .tbz2
tgz	.tar.gz, .tgz
zip	.zip

bzr get

Alias for "branch", see "bzr branch".

bzr help [TOPIC]

Options:

```
--help, -h               Show help message.
--long                   Show help on all commands.
--quiet, -q              Only display errors and warnings.
--usage                  Show usage message and options.
--verbose, -v            Display more information.
```

Aliases: ?, --help, -?, -h

See also: topics

Show help on a command or other topic.

bzr ignore [NAME_PATTERN...]

Options:

```
--default-rules          Display the default ignore rules that
                         bzr uses.
```

−−directory ARG, −d Branch to operate on, instead of working directory.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: ignored, patterns, status

Ignore specified files or patterns.

See *(Aq*(Aqbzr help patterns*(Aq*(Aq for details on the syntax of patterns.

If a .bzrignore file does not exist, the ignore command will create one and add the specified files or patterns to the newly created file. The ignore command will also automatically add the .bzrignore file to be versioned. Creating a .bzrignore file without the use of the ignore command will require an explicit add command.

To remove patterns from the ignore list, edit the .bzrignore file. After adding, editing or deleting that file either indirectly by using this command or directly by using an editor, be sure to commit it.

Bazaar also supports a global ignore file ~/.bazaar/ignore. On Windows the global ignore file can be found in the application data directory as C:\Documents and Settings\<user>\Application Data\Bazaar\2.0\ignore. Global ignores are not touched by this command. The global ignore file can be edited directly using an editor.

Patterns prefixed with '!' are exceptions to ignore patterns and take precedence over regular ignores. Such exceptions are used to specify files that should be versioned which would otherwise be ignored.

Patterns prefixed with '!!' act as regular ignore patterns, but have precedence over the '!' exception patterns.

Notes:

* Ignore patterns containing shell wildcards must be quoted from the shell on Unix.

* Ignore patterns starting with "#" act as comments in the ignore file. To ignore patterns that begin with that character, use the "RE:" prefix.

Examples:
Ignore the top level Makefile:

bzr ignore ./Makefile

Ignore .class files in all directories...:

bzr ignore "*.class"

...but do not ignore "special.class":

bzr ignore "!special.class"

Ignore files whose name begins with the "#" character:

```
bzr ignore "RE:^#"
```

Ignore .o files under the lib directory:

```
bzr ignore "lib/**/*.o"
```

Ignore .o files under the lib directory:

```
bzr ignore "RE:lib/.*\.o"
```

Ignore everything but the "debian" toplevel directory:

```
bzr ignore "RE:(?!debian/).*"
```

Ignore everything except the "local" toplevel directory,
but always ignore autosave files ending in ~, even under local/:

```
bzr ignore "*"
bzr ignore "!./local"
bzr ignore "!!*~"
```

bzr ignored

Options:

```
--directory ARG, -d      Branch to operate on, instead of working
                         directory.
--help, -h               Show help message.
--quiet, -q              Only display errors and warnings.
--usage                  Show usage message and options.
--verbose, -v            Display more information.
```

See also: ignore, ls

List ignored files and the patterns that matched them.

List all the ignored files and the ignore pattern that caused the file to be ignored.

Alternatively, to list just the files:

```
bzr ls --ignored
```

bzr info [LOCATION]

Options:

```
--help, -h               Show help message.
--quiet, -q              Only display errors and warnings.
--usage                  Show usage message and options.
--verbose, -v            Display more information.
```

See also: repositories, revno, working-trees

Show information about a working tree, branch or repository.

This command will show all known locations and formats associated to the tree, branch or repository.

In verbose mode, statistical information is included with each report. To see extended statistic information,

use a verbosity level of 2 or higher by specifying the verbose option multiple times, e.g. −vv.

Branches and working trees will also report any missing revisions.

Examples:

Display information on the format and related locations:

bzr info

Display the above together with extended format information and basic statistics (like the number of files in the working tree and number of revisions in the branch and repository):

bzr info −v

Display the above together with number of committers to the branch:

bzr info −vv

bzr init [LOCATION]

Options:

−−append−revisions−only Never change revnos or the existing log. Append revisions to it only.
−−create−prefix Create the path leading up to the branch if it does not already exist.
−−format ARG Specify a format for this branch. See "help formats".
−−2a Format for the bzr 2.0 series. Uses group−compress storage. Provides rich roots which are a one−way transition.
−−default Format for the bzr 2.0 series. Uses group−compress storage. Provides rich roots which are a one−way transition.
−−development−colo The 2a format with experimental support for colocated branches.
−−pack−0.92 Pack−based format used in 1.x series. Introduced in 0.92. Interoperates with bzr repositories before 0.92 but cannot be read by bzr < 0.92.
−−help, −h Show help message.
−−no−tree Create a branch without a working tree.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: branch, checkout, init−repository

Make a directory into a versioned branch.

Use this to create an empty branch, or before importing an existing project.

If there is a repository in a parent directory of the location, then the history of the branch will be stored in the repository. Otherwise init creates a standalone branch which carries its own history in the .bzr

directory.

If there is already a branch at the location but it has no working tree, the tree can be populated with 'bzr checkout'.

Recipe for importing a tree of files:

```
cd ~/project
bzr init
bzr add .
bzr status
bzr commit -m "imported project"
```

bzr init-repo

Alias for "init-repository", see "bzr init-repository".

bzr init-repository LOCATION

Options:

```
--format ARG            Specify a format for this repository.
                    See "bzr help formats" for details.
--2a                Format for the bzr 2.0 series. Uses
                    group-compress storage. Provides rich
                    roots which are a one-way transition.
--default           Format for the bzr 2.0 series. Uses
                    group-compress storage. Provides rich
                    roots which are a one-way transition.
--development-colo      The 2a format with experimental support
                    for colocated branches.
--pack-0.92             Pack-based format used in 1.x series.
                    Introduced in 0.92. Interoperates with
                    bzr repositories before 0.92 but cannot
                    be read by bzr < 0.92.
--help, -h          Show help message.
--no-trees          Branches in the repository will default
                    to not having a working tree.
--quiet, -q         Only display errors and warnings.
--usage             Show usage message and options.
--verbose, -v           Display more information.
```

Alias: init-repo

See also: branch, checkout, init, repositories

Create a shared repository for branches to share storage space.

New branches created under the repository directory will store their revisions in the repository, not in the branch directory. For branches with shared history, this reduces the amount of storage needed and speeds up the creation of new branches.

If the --no-trees option is given then the branches in the repository will not have working trees by default. They will still exist as directories on disk, but they will not have separate copies of the files at a certain revision. This can be useful for repositories that store branches which are interacted with through checkouts or remote branches, such as on a server.

Examples:

Create a shared repository holding just branches:

```
bzr init-repo --no-trees repo
bzr init repo/trunk
```

Make a lightweight checkout elsewhere:

```
bzr checkout --lightweight repo/trunk trunk-checkout
cd trunk-checkout
(add files here)
```

bzr join TREE

Options:

--help, -h	Show help message.
--quiet, -q	Only display errors and warnings.
--usage	Show usage message and options.
--verbose, -v	Display more information.

See also: split

Combine a tree into its containing tree.

This command requires the target tree to be in a rich-root format.

The TREE argument should be an independent tree, inside another tree, but not part of it. (Such trees can be produced by "bzr split", but also by running "bzr branch" with the target inside a tree.)

The result is a combined tree, with the subtree no longer an independent part. This is marked as a merge of the subtree into the containing tree, and all history is preserved.

bzr launchpad-login [NAME]

Options:

--help, -h	Show help message.
--no-check	Don't check that the user name is valid.
--quiet, -q	Only display errors and warnings.
--usage	Show usage message and options.
--verbose, -v	Display more information.

Alias: lp-login

Show or set the Launchpad user ID.

When communicating with Launchpad, some commands need to know your Launchpad user ID. This command can be used to set or show the user ID that Bazaar will use for such communication.

Examples:

Show the Launchpad ID of the current user:

```
bzr launchpad-login
```

Set the Launchpad ID of the current user to 'bob':

```
bzr launchpad-login bob
```

bzr launchpad−mirror [LOCATION]

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Alias: lp−mirror

Ask Launchpad to mirror a branch now.

bzr launchpad−open [LOCATION]

Options:

−−dry−run Do not actually open the browser. Just say the URL we would use.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Alias: lp−open

Open a Launchpad branch page in your web browser.

bzr log [FILE...]

Options:

−−authors ARG What names to list as authors − first, all or committer.
−−change ARG, −c Show just the specified revision. See also "help revisionspec".
−−exclude−common−ancestry Display only the revisions that are not part of both ancestries (require −rX..Y).
−−forward Show from oldest to newest.
−−help, −h Show help message.
−−include−merged Show merged revisions like −−levels 0 does.
−−levels N, −n Number of levels to display − 0 for all, 1 for flat.
−−limit N, −l Limit the output to the first N revisions.
−−log−format ARG Use specified log format.
−−gnu−changelog Format used by GNU ChangeLog files.
−−line Log format with one line per revision.
−−long Detailed log format.
−−short Moderately short log format.
−−match ARG, −m Show revisions whose properties match this expression.
−−match−author ARG Show revisions whose authors match this expression.
−−match−bugs ARG Show revisions whose bugs match this expression.
−−match−committer ARG Show revisions whose committer matches this expression.

−−match−message ARG Show revisions whose message matches this expression.
−−omit−merges Do not report commits with more than one parent.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−show−diff, −p Show changes made in each revision as a patch.
−−show−ids Show internal object ids.
−−signatures Show digital signature validity.
−−timezone ARG Display timezone as local, original, or utc.
−−usage Show usage message and options.
−−verbose, −v Show files changed in each revision.

See also: log−formats, revisionspec

Show historical log for a branch or subset of a branch.

log is bzr's default tool for exploring the history of a branch. The branch to use is taken from the first parameter. If no parameters are given, the branch containing the working directory is logged. Here are some simple examples:

bzr log log the current branch
bzr log foo.py log a file in its branch
bzr log http://server/branch log a branch on a server

The filtering, ordering and information shown for each revision can be controlled as explained below. By default, all revisions are shown sorted (topologically) so that newer revisions appear before older ones and descendants always appear before ancestors. If displayed, merged revisions are shown indented under the revision in which they were merged.

Output control:

The log format controls how information about each revision is displayed. The standard log formats are called *(Aq*(Aqlong*(Aq*(Aq, *(Aq*(Aqshort*(Aq*(Aq*(Aq and *(Aq*(Aqline*(Aq*(Aq. The default is long. See *(Aq*(Aqbzr help log−formats*(Aq*(Aq*(Aq for more details on log formats.

The following options can be used to control what information is displayed:

−l N display a maximum of N revisions
−n N display N levels of revisions (0 for all, 1 for collapsed)
−v display a status summary (delta) for each revision
−p display a diff (patch) for each revision
−−show−ids display revision−ids (and file−ids), not just revnos

Note that the default number of levels to display is a function of the log format. If the −n option is not used, the standard log formats show just the top level (mainline).

Status summaries are shown using status flags like A, M, etc. To see the changes explained using words like *(Aq*(Aqadded*(Aq*(Aq*(Aq and *(Aq*(Aqmodified*(Aq*(Aq*(Aq

instead, use the −vv option.

Ordering control:

To display revisions from oldest to newest, use the −−forward option. In most cases, using this option will have little impact on the total time taken to produce a log, though −−forward does not incrementally display revisions like −−reverse does when it can.

Revision filtering:

The −r option can be used to specify what revision or range of revisions to filter against. The various forms are shown below:

−rX display revision X
−rX.. display revision X and later
−r..Y display up to and including revision Y
−rX..Y display from X to Y inclusive

See *(Aq*(Aqbzr help revisionspec*(Aq*(Aq for details on how to specify X and Y. Some common examples are given below:

−r−1 show just the tip
−r−10.. show the last 10 mainline revisions
−rsubmit:.. show what's new on this branch
−rancestor:path.. show changes since the common ancestor of this branch and the one at location path
−rdate:yesterday.. show changes since yesterday

When logging a range of revisions using −rX..Y, log starts at revision Y and searches back in history through the primary ("left−hand") parents until it finds X. When logging just the top level (using −n1), an error is reported if X is not found along the way. If multi−level logging is used (−n0), X may be a nested merge revision and the log will be truncated accordingly.

Path filtering:

If parameters are given and the first one is not a branch, the log will be filtered to show only those revisions that changed the nominated files or directories.

Filenames are interpreted within their historical context. To log a deleted file, specify a revision range so that the file existed at the end or start of the range.

Historical context is also important when interpreting pathnames of renamed files/directories. Consider the following example:

* revision 1: add tutorial.txt
* revision 2: modify tutorial.txt
* revision 3: rename tutorial.txt to guide.txt; add tutorial.txt

In this case:

* *(Aq*(Aqbzr log guide.txt*(Aq*(Aq will log the file added in revision 1

* *(Aq*(Aqbzr log tutorial.txt*(Aq*(Aq will log the new file added in revision 3

* *(Aq*(Aqbzr log −r2 −p tutorial.txt*(Aq*(Aq will show the changes made to
the original file in revision 2.

* *(Aq*(Aqbzr log −r2 −p guide.txt*(Aq*(Aq will display an error message as there
was no file called guide.txt in revision 2.

Renames are always followed by log. By design, there is no need to
explicitly ask for this (and no way to stop logging a file back
until it was last renamed).

Other filtering:

The −−match option can be used for finding revisions that match a
regular expression in a commit message, committer, author or bug.
Specifying the option several times will match any of the supplied
expressions. −−match−author, −−match−bugs, −−match−committer and
−−match−message can be used to only match a specific field.

Tips & tricks:

GUI tools and IDEs are often better at exploring history than command
line tools: you may prefer qlog or viz from qbzr or bzr−gtk, the
bzr−explorer shell, or the Loggerhead web interface. See the Plugin
Guide <http://doc.bazaar.canonical.com/plugins/en/> and
<http://wiki.bazaar.canonical.com/IDEIntegration>.

You may find it useful to add the aliases below to *(Aq*(Aqbazaar.conf*(Aq*(Aq:

```
[ALIASES]
tip = log −r−1
top = log −l10 −−line
show = log −v −p
```

(Aq(Aqbzr tip*(Aq*(Aq will then show the latest revision while *(Aq*(Aqbzr top*(Aq*(Aq
will show the last 10 mainline revisions. To see the details of a
particular revision X, *(Aq*(Aqbzr show −rX*(Aq*(Aq.

If you are interested in looking deeper into a particular merge X,
use *(Aq*(Aqbzr log −n0 −rX*(Aq*(Aq.

(Aq(Aqbzr log −v*(Aq*(Aq on a branch with lots of history is currently
very slow. A fix for this issue is currently under development.
With or without that fix, it is recommended that a revision range
be given when using the −v option.

bzr has a generic full−text matching plugin, bzr−search, that can be
used to find revisions matching user names, commit messages, etc.
Among other features, this plugin can find all revisions containing
a list of words but not others.

When exploring non−mainline history on large projects with deep history, the performance of log can be greatly improved by installing the historycache plugin. This plugin buffers historical information trading disk space for faster speed.

bzr lp−find−proposal

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Find the proposal to merge this revision.

Finds the merge proposal(s) that discussed landing the specified revision. This works only if the if the merged_revno was recorded for the merge proposal. The proposal(s) are opened in a web browser.

Only the revision specified is searched for. To find the mainline revision that merged it into mainline, use the "mainline" revision spec.

So, to find the merge proposal that reviewed line 1 of README:

bzr lp−find−proposal −r mainline:annotate:README:1

bzr lp−login

Alias for "launchpad−login", see "bzr launchpad−login".

bzr lp−mirror

Alias for "launchpad−mirror", see "bzr launchpad−mirror".

bzr lp−open

Alias for "launchpad−open", see "bzr launchpad−open".

bzr lp−propose

Alias for "lp−propose−merge", see "bzr lp−propose−merge".

bzr lp−propose−merge [SUBMIT_BRANCH]

Options:

−−approve Mark the proposal as approved immediately, setting the approved revision to tip.
−−fixes ARG The bug this proposal fixes.
−−help, −h Show help message.
−−message ARG, −m Commit message.
−−quiet, −q Only display errors and warnings.
−−review ARG, −R Requested reviewer and optional type.
−−staging Propose the merge on staging.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Aliases: lp−submit, lp−propose

Propose merging a branch on Launchpad.

This will open your usual editor to provide the initial comment. When it has created the proposal, it will

open it in your default web browser.

The branch will be proposed to merge into SUBMIT_BRANCH. If SUBMIT_BRANCH is not supplied, the remembered submit branch will be used. If no submit branch is remembered, the development focus will be used.

By default, the SUBMIT_BRANCH's review team will be requested to review the merge proposal. This can be overriden by specifying −−review (−R). The parameter the launchpad account name of the desired reviewer. This may optionally be followed by '=' and the review type. For example:

bzr lp−propose−merge −−review jrandom −−review review−team=qa

This will propose a merge, request "jrandom" to perform a review of unspecified type, and request "review−team" to perform a "qa" review.

bzr lp−submit

Alias for "lp−propose−merge", see "bzr lp−propose−merge".

bzr ls [PATH]

Options:

−−directory ARG, −d Branch to operate on, instead of working directory.
−−from−root Print paths relative to the root of the branch.
−−help, −h Show help message.
−−ignored, −i Print ignored files.
−−kind ARG, −k List entries of a particular kind: file, directory, symlink.
−−null, −0 Use an ASCII NUL (\0) separator rather than a newline.
−−quiet, −q Only display errors and warnings.
−−recursive, −R Recurse into subdirectories.
−−revision ARG, −r See "help revisionspec" for details.
−−show−ids Show internal object ids.
−−unknown, −u Print unknown files.
−−usage Show usage message and options.
−−verbose, −v Display more information.
−−versioned, −V Print versioned files.

See also: cat, status

List files in a tree.

bzr merge [LOCATION]

Options:

−−change ARG, −c Select changes introduced by the specified revision. See also "help revisionspec".
−−directory ARG, −d Branch to merge into, rather than the one containing the working directory.
−−force Merge even if the destination tree has uncommitted changes.
−−help, −h Show help message.

−−interactive, −i Select changes interactively.
−−merge−type ARG Select a particular merge algorithm.
−−diff3 Merge using external diff3.
−−lca LCA−newness merge.
−−merge3 Native diff3−style merge.
−−weave Weave−based merge.
−−preview Instead of merging, show a diff of the merge.
−−pull If the destination is already completely merged into the source, pull from the source rather than merging. When this happens, you do not need to commit the result.
−−quiet, −q Only display errors and warnings.
−−remember Remember the specified location as a default.
−−reprocess Reprocess to reduce spurious conflicts.
−−revision ARG, −r See "help revisionspec" for details.
−−show−base Show base revision text in conflicts.
−−uncommitted Apply uncommitted changes from a working copy, instead of branch changes.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: remerge, send, status−flags, update

Perform a three−way merge.

The source of the merge can be specified either in the form of a branch, or in the form of a path to a file containing a merge directive generated with bzr send. If neither is specified, the default is the upstream branch or the branch most recently merged using −−remember. The source of the merge may also be specified in the form of a path to a file in another branch: in this case, only the modifications to that file are merged into the current working tree.

When merging from a branch, by default bzr will try to merge in all new work from the other branch, automatically determining an appropriate base revision. If this fails, you may need to give an explicit base.

To pick a different ending revision, pass "−−revision OTHER". bzr will try to merge in all new work up to and including revision OTHER.

If you specify two values, "−−revision BASE..OTHER", only revisions BASE through OTHER, excluding BASE but including OTHER, will be merged. If this causes some revisions to be skipped, i.e. if the destination branch does not already contain revision BASE, such a merge is commonly referred to as a "cherrypick". Unlike a normal merge, Bazaar does not currently track cherrypicks. The changes look like a normal commit, and the history of the changes from the other branch is not stored in the commit.

Revision numbers are always relative to the source branch.

Merge will do its best to combine the changes in two branches, but there are some kinds of problems only a human can fix. When it encounters those, it will mark a conflict. A conflict means that you need to fix something, before you can commit.

Use bzr resolve when you have fixed a problem. See also bzr conflicts.

If there is no default branch set, the first merge will set it (use −−no−remember to avoid setting it). After that, you can omit the branch to use the default. To change the default, use −−remember. The value will only be saved if the remote location can be accessed.

The results of the merge are placed into the destination working directory, where they can be reviewed (with bzr diff), tested, and then committed to record the result of the merge.

merge refuses to run if there are any uncommitted changes, unless −−force is given. If −−force is given, then the changes from the source will be merged with the current working tree, including any uncommitted changes in the tree. The −−force option can also be used to create a merge revision which has more than two parents.

If one would like to merge changes from the working tree of the other branch without merging any committed revisions, the −−uncommitted option can be given.

To select only some changes to merge, use "merge −i", which will prompt you to apply each diff hunk and file change, similar to "shelve".

Examples:
To merge all new revisions from bzr.dev:

```
bzr merge ../bzr.dev
```

To merge changes up to and including revision 82 from bzr.dev:

```
bzr merge −r 82 ../bzr.dev
```

To merge the changes introduced by 82, without previous changes:

```
bzr merge −r 81..82 ../bzr.dev
```

To apply a merge directive contained in /tmp/merge:

```
bzr merge /tmp/merge
```

To create a merge revision with three parents from two branches feature1a and feature1b:

```
bzr merge ../feature1a
bzr merge ../feature1b −−force
bzr commit −m 'revision with three parents'
```

bzr missing [OTHER_BRANCH]

Options:

−−directory ARG, −d Branch to operate on, instead of working directory.
−−help, −h Show help message.
−−include−merged Show all revisions in addition to the mainline ones.
−−log−format ARG Use specified log format.
−−gnu−changelog Format used by GNU ChangeLog files.
−−line Log format with one line per revision.
−−long Detailed log format.
−−short Moderately short log format.

−−mine−only Display changes in the local branch only.
−−my−revision ARG Filter on local branch revisions (inclusive). See "help revisionspec" for details.
−−other Same as −−theirs−only.
−−quiet, −q Only display errors and warnings.
−−reverse Reverse the order of revisions.
−−revision ARG, −r Filter on other branch revisions (inclusive). See "help revisionspec" for details.
−−show−ids Show internal object ids.
−−theirs−only Display changes in the remote branch only.
−−this Same as −−mine−only.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: merge, pull

Show unmerged/unpulled revisions between two branches.

OTHER_BRANCH may be local or remote.

To filter on a range of revisions, you can use the command −r begin..end −r revision requests a specific revision, −r ..end or −r begin.. are also valid.

Exit values:
1 − some missing revisions
0 − no missing revisions

Examples:

Determine the missing revisions between this and the branch at the remembered pull location:

bzr missing

Determine the missing revisions between this and another branch:

bzr missing http://server/branch

Determine the missing revisions up to a specific revision on the other branch:

bzr missing −r ..−10

Determine the missing revisions up to a specific revision on this branch:

bzr missing −−my−revision ..−10

bzr mkdir DIR...

Options:

```
--help, -h          Show help message.
--parents, -p       No error if existing, make parent
                    directories as needed.
--quiet, -q         Only display errors and warnings.
--usage             Show usage message and options.
--verbose, -v       Display more information.
```

Create a new versioned directory.

This is equivalent to creating the directory and then adding it.

bzr move

Alias for "mv", see "bzr mv".

bzr mv [NAMES...]

Options:

```
--after             Move only the bzr identifier of the
                    file, because the file has already been
                    moved.
--auto              Automatically guess renames.
--dry-run           Avoid making changes when guessing
                    renames.
--help, -h          Show help message.
--quiet, -q         Only display errors and warnings.
--usage             Show usage message and options.
--verbose, -v       Display more information.
```

Aliases: move, rename

Move or rename a file.

Usage:

```
bzr mv OLDNAME NEWNAME

bzr mv SOURCE... DESTINATION
```

If the last argument is a versioned directory, all the other names are moved into it. Otherwise, there must be exactly two arguments and the file is changed to a new name.

If OLDNAME does not exist on the filesystem but is versioned and NEWNAME does exist on the filesystem but is not versioned, mv assumes that the file has been manually moved and only updates its internal inventory to reflect that change. The same is valid when moving many SOURCE files to a DESTINATION.

Files cannot be moved between branches.

bzr nick [NICKNAME]

Options:

```
--directory ARG, -d  Branch to operate on, instead of working
                     directory.
--help, -h           Show help message.
--quiet, -q          Only display errors and warnings.
--usage              Show usage message and options.
```

−−verbose, −v Display more information.

See also: info

Print or set the branch nickname.

If unset, the colocated branch name is used for colocated branches, and the branch directory name is used for other branches. To print the current nickname, execute with no argument.

Bound branches use the nickname of its master branch unless it is set locally.

bzr pack [BRANCH_OR_REPO]

Options:

−−clean−obsolete−packs Delete obsolete packs to save disk space.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: repositories

Compress the data within a repository.

This operation compresses the data within a bazaar repository. As bazaar supports automatic packing of repository, this operation is normally not required to be done manually.

During the pack operation, bazaar takes a backup of existing repository data, i.e. pack files. This backup is eventually removed by bazaar automatically when it is safe to do so. To save disk space by removing the backed up pack files, the −−clean−obsolete−packs option may be used.

Warning: If you use −−clean−obsolete−packs and your machine crashes during or immediately after repacking, you may be left with a state where the deletion has been written to disk but the new packs have not been. In this case the repository may be unusable.

bzr ping LOCATION

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Pings a Bazaar smart server.

This command sends a 'hello' request to the given location using the bzr smart protocol, and reports the response.

bzr plugins

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

List the installed plugins.

This command displays the list of installed plugins including version of plugin and a short description of each.

−−verbose shows the path where each plugin is located.

A plugin is an external component for Bazaar that extends the revision control system, by adding or replacing code in Bazaar. Plugins can do a variety of things, including overriding commands, adding new commands, providing additional network transports and customizing log output.

See the Bazaar Plugin Guide <http://doc.bazaar.canonical.com/plugins/en/> for further information on plugins including where to find them and how to install them. Instructions are also provided there on how to write new plugins using the Python programming language.

bzr praise

Alias for "annotate", see "bzr annotate".

bzr pull [LOCATION]

Options:

−−directory ARG, −d Branch to pull into, rather than the one containing the working directory.
−−help, −h Show help message.
−−local Perform a local pull in a bound branch. Local pulls are not applied to the master branch.
−−overwrite Ignore differences between branches and overwrite unconditionally.
−−overwrite−tags Overwrite tags only.
−−quiet, −q Only display errors and warnings.
−−remember Remember the specified location as a default.
−−revision ARG, −r See "help revisionspec" for details.
−−show−base Show base revision text in conflicts.
−−usage Show usage message and options.
−−verbose, −v Show logs of pulled revisions.

See also: push, send, status−flags, update

Turn this branch into a mirror of another branch.

By default, this command only works on branches that have not diverged. Branches are considered diverged if the destination branch's most recent commit is one that has not been merged (directly or indirectly) into the parent.

If branches have diverged, you can use 'bzr merge' to integrate the changes from one into the other. Once one branch has merged, the other should be able to pull it again.

If you want to replace your local changes and just want your branch to match the remote one, use pull −−overwrite. This will work even if the two branches have diverged.

If there is no default location set, the first pull will set it (use −−no−remember to avoid setting it). After that, you can omit the location to use the default. To change the default, use −−remember. The value will only be saved if the remote location can be accessed.

The −−verbose option will display the revisions pulled using the log_format configuration option. You can use a different format by overriding it with −Olog_format=<other_format>.

Note: The location can be specified either in the form of a branch, or in the form of a path to a file containing a merge directive generated with bzr send.

bzr push [LOCATION]

Options:

−−create−prefix Create the path leading up to the branch if it does not already exist.
−−directory ARG, −d Branch to push from, rather than the one containing the working directory.
−−help, −h Show help message.
−−no−tree Don't populate the working tree, even for protocols that support it.
−−overwrite Ignore differences between branches and overwrite unconditionally.
−−overwrite−tags Overwrite tags only.
−−quiet, −q Only display errors and warnings.
−−remember Remember the specified location as a default.
−−revision ARG, −r See "help revisionspec" for details.
−−stacked Create a stacked branch that references the public location of the parent branch.
−−stacked−on ARG Create a stacked branch that refers to another branch for the commit history. Only the work not present in the referenced branch is included in the branch created.
−−strict Refuse to push if there are uncommitted changes in the working tree, −−no−strict disables the check.
−−usage Show usage message and options.
−−use−existing−dir By default push will fail if the target directory exists, but does not already have a control directory. This flag will allow push to proceed.
−−verbose, −v Display more information.

See also: pull, update, working−trees

Update a mirror of this branch.

The target branch will not have its working tree populated because this is both expensive, and is not supported on remote file systems.

Some smart servers or protocols *may* put the working tree in place in the future.

This command only works on branches that have not diverged. Branches are considered diverged if the destination branch's most recent commit is one that has not been merged (directly or indirectly) by the source branch.

If branches have diverged, you can use 'bzr push −−overwrite' to replace the other branch completely,

discarding its unmerged changes.

If you want to ensure you have the different changes in the other branch, do a merge (see bzr help merge) from the other branch, and commit that. After that you will be able to do a push without '−−overwrite'.

If there is no default push location set, the first push will set it (use −−no−remember to avoid setting it). After that, you can omit the location to use the default. To change the default, use −−remember. The value will only be saved if the remote location can be accessed.

The −−verbose option will display the revisions pushed using the log_format configuration option. You can use a different format by overriding it with −Olog_format=<other_format>.

bzr reconcile [BRANCH]

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: check

Reconcile bzr metadata in a branch.

This can correct data mismatches that may have been caused by previous ghost operations or bzr upgrades. You should only need to run this command if 'bzr check' or a bzr developer advises you to run it.

If a second branch is provided, cross−branch reconciliation is also attempted, which will check that data like the tree root id which was not present in very early bzr versions is represented correctly in both branches.

At the same time it is run it may recompress data resulting in a potential saving in disk space or performance gain.

The branch *MUST* be on a listable system such as local disk or sftp.

bzr reconfigure [LOCATION]

Options:

−−bind−to ARG Branch to bind checkout to.
−−force Perform reconfiguration even if local changes will be lost.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−repository_trees ARG Whether new branches in the repository have trees.
−−with−no−trees Reconfigure repository to not create working trees on branches by default.
−−with−trees Reconfigure repository to create working trees on branches by default.
−−repository_type ARG Location fo the repository.
−−standalone Reconfigure to be a standalone branch (i.e. stop using shared repository).
−−use−shared Reconfigure to use a shared repository.
−−stacked−on ARG Reconfigure a branch to be stacked on another branch.

−−tree_type ARG The relation between branch and tree.
−−branch Reconfigure to be an unbound branch with no working tree.
−−checkout Reconfigure to be a bound branch with a working tree.
−−lightweight−checkout Reconfigure to be a lightweight checkout (with no local history).
−−tree Reconfigure to be an unbound branch with a working tree.
−−unstacked Reconfigure a branch to be unstacked. This may require copying substantial data into it.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: branches, checkouts, standalone−trees, working−trees

Reconfigure the type of a bzr directory.

A target configuration must be specified.

For checkouts, the bind−to location will be auto−detected if not specified. The order of preference is 1. For a lightweight checkout, the current bound location. 2. For branches that used to be checkouts, the previously−bound location. 3. The push location. 4. The parent location. If none of these is available, −−bind−to must be specified.

bzr register−branch [PUBLIC_URL]

Options:
−−author ARG Branch author's email address, if not yourself.
−−branch−description ARG Longer description of the purpose or contents of the branch.
−−branch−name ARG Short name for the branch; by default taken from the last component of the url.
−−branch−title ARG One−sentence description of the branch.
−−dry−run Prepare the request but don't actually send it.
−−help, −h Show help message.
−−link−bug ARG The bug this branch fixes.
−−project ARG Launchpad project short name to associate with the branch.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Register a branch with launchpad.net.

This command lists a bzr branch in the directory of branches on launchpad.net. Registration allows the branch to be associated with bugs or specifications.

Before using this command you must register the project to which the branch belongs, and create an account for yourself on launchpad.net.

arguments:
public_url: The publicly visible url for the branch to register.
This must be an http or https url (which Launchpad can read from to access the branch). Local file urls, SFTP urls, and bzr+ssh urls will not work.
If no public_url is provided, bzr will use the configured public_url if there is one for the current branch, and otherwise error.

example:
bzr register−branch http://foo.com/bzr/fooproject.mine \
−−project fooproject

bzr remerge [FILE...]

Options:
−−help, −h Show help message.
−−merge−type ARG Select a particular merge algorithm.
−−diff3 Merge using external diff3.
−−lca LCA−newness merge.
−−merge3 Native diff3−style merge.
−−weave Weave−based merge.
−−quiet, −q Only display errors and warnings.
−−reprocess Reprocess to reduce spurious conflicts.
−−show−base Show base revision text in conflicts.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Redo a merge.

Use this if you want to try a different merge technique while resolving conflicts. Some merge techniques are better than others, and remerge lets you try different ones on different files.

The options for remerge have the same meaning and defaults as the ones for merge. The difference is that remerge can (only) be run when there is a pending merge, and it lets you specify particular files.

Examples:
Re−do the merge of all conflicted files, and show the base text in conflict regions, in addition to the usual THIS and OTHER texts:

bzr remerge −−show−base

Re−do the merge of "foobar", using the weave merge algorithm, with additional processing to reduce the size of conflict regions:

bzr remerge −−merge−type weave −−reprocess foobar

bzr remove [FILE...]

Options:
−−file−deletion−strategy ARGThe file deletion mode to be used.
−−keep Delete from bzr but leave the working copy.
−−no−backup Don't backup changed files.
−−safe Backup changed files (default).
−−help, −h Show help message.

--new Only remove files that have never been committed.
--quiet, -q Only display errors and warnings.
--usage Show usage message and options.
--verbose, -v Display more information.

Aliases: rm, del

Remove files or directories.

This makes Bazaar stop tracking changes to the specified files. Bazaar will delete them if they can easily be recovered using revert otherwise they will be backed up (adding an extension of the form .~#~). If no options or parameters are given Bazaar will scan for files that are being tracked by Bazaar but missing in your tree and stop tracking them for you.

bzr remove-branch [LOCATION]

Options:
--directory ARG, -d Branch to operate on, instead of working directory.
--force Remove branch even if it is the active branch.
--help, -h Show help message.
--quiet, -q Only display errors and warnings.
--usage Show usage message and options.
--verbose, -v Display more information.

Alias: rmbranch

Remove a branch.

This will remove the branch from the specified location but will keep any working tree or repository in place.

Examples:

Remove the branch at repo/trunk:

bzr remove-branch repo/trunk

bzr remove-tree [LOCATION...]

Options:
--force Remove the working tree even if it has uncommitted or shelved changes.
--help, -h Show help message.
--quiet, -q Only display errors and warnings.
--usage Show usage message and options.
--verbose, -v Display more information.

See also: checkout, working-trees

Remove the working tree from a given branch/checkout.

Since a lightweight checkout is little more than a working tree this will refuse to run against one.

To re−create the working tree, use "bzr checkout".

bzr rename
Alias for "mv", see "bzr mv".

bzr renames [DIR]
Options:
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: status

Show list of renamed files.

bzr resolve [FILE...]
Options:
−−action ARG How to resolve the conflict.
−−done Marks the conflict as resolved.
−−take−other Resolve the conflict taking the merged version into account.
−−take−this Resolve the conflict preserving the version in the working tree.
−−all Resolve all conflicts in this tree.
−−directory ARG, −d Branch to operate on, instead of working directory.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Alias: resolved

See also: conflicts

Mark a conflict as resolved.

Merge will do its best to combine the changes in two branches, but there are some kinds of problems only a human can fix. When it encounters those, it will mark a conflict. A conflict means that you need to fix something, before you can commit.

Once you have fixed a problem, use "bzr resolve" to automatically mark text conflicts as fixed, "bzr resolve FILE" to mark a specific conflict as resolved, or "bzr resolve −−all" to mark all conflicts as resolved.

bzr resolved
Alias for "resolve", see "bzr resolve".

bzr revert [FILE...]
Options:
−−forget−merges Remove pending merge marker, without changing any files.
−−help, −h Show help message.

−−no−backup Do not save backups of reverted files.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: cat, export, merge, shelve

Set files in the working tree back to the contents of a previous revision.

Giving a list of files will revert only those files. Otherwise, all files will be reverted. If the revision is not specified with '−−revision', the working tree basis revision is used. A revert operation affects only the working tree, not any revision history like the branch and repository or the working tree basis revision.

To remove only some changes, without reverting to a prior version, use merge instead. For example, "merge . −r −2..−3" (don't forget the ".") will remove the changes introduced by the second last commit (−2), without affecting the changes introduced by the last commit (−1). To remove certain changes on a hunk−by−hunk basis, see the shelve command. To update the branch to a specific revision or the latest revision and update the working tree accordingly while preserving local changes, see the update command.

Uncommitted changes to files that are reverted will be discarded. Howver, by default, any files that have been manually changed will be backed up first. (Files changed only by merge are not backed up.) Backup files have '.˜#˜' appended to their name, where # is a number.

When you provide files, you can use their current pathname or the pathname from the target revision. So you can use revert to "undelete" a file by name. If you name a directory, all the contents of that directory will be reverted.

If you have newly added files since the target revision, they will be removed. If the files to be removed have been changed, backups will be created as above. Directories containing unknown files will not be deleted.

The working tree contains a list of revisions that have been merged but not yet committed. These revisions will be included as additional parents of the next commit. Normally, using revert clears that list as well as reverting the files. If any files are specified, revert leaves the list of uncommitted merges alone and reverts only the files. Use *(Aq*(Aqbzr revert.*(Aq*(Aq in the tree root to revert all files but keep the recorded merges, and *(Aq*(Aqbzr revert −−forget−merges*(Aq*(Aq to clear the pending merge list without reverting any files.

Using "bzr revert −−forget−merges", it is possible to apply all of the changes from a branch in a single revision. To do this, perform the merge as desired. Then doing revert with the "−−forget−merges" option will keep the content of the tree as it was, but it will clear the list of pending merges. The next commit will then contain all of the changes that are present in the other branch, but without any other parent revisions. Because this technique forgets where these changes originated, it may cause additional conflicts on later merges involving the same source and target branches.

bzr revno [LOCATION]

Options:
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−tree Show revno of working tree.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: info

Show current revision number.

This is equal to the number of revisions on this branch.

bzr rm
Alias for "remove", see "bzr remove".

bzr rmbranch
Alias for "remove−branch", see "bzr remove−branch".

bzr root [FILENAME]
Options:
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Show the tree root directory.

The root is the nearest enclosing directory with a .bzr control directory.

bzr send [SUBMIT_BRANCH] [PUBLIC_BRANCH]
Options:
−−body ARG Body for the email.
−−format ARG Use the specified output format.
−−from ARG, −f Branch to generate the submission from, rather than the one containing the working directory.
−−help, −h Show help message.
−−mail−to ARG Mail the request to this address.
−−message ARG, −m Message string.
−−no−bundle Do not include a bundle in the merge directive.
−−no−patch Do not include a preview patch in the merge directive.
−−output ARG, −o Write merge directive to this file or directory; use − for stdout.
−−quiet, −q Only display errors and warnings.
−−remember Remember submit and public branch.
−−revision ARG, −r See "help revisionspec" for details.
−−strict Refuse to send if there are uncommitted changes in the working tree, −−no−strict disables the check.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: merge, pull

Mail or create a merge−directive for submitting changes.

A merge directive provides many things needed for requesting merges:

* A machine−readable description of the merge to perform

* An optional patch that is a preview of the changes requested

* An optional bundle of revision data, so that the changes can be applied directly from the merge directive, without retrieving data from a branch.

(Aqbzr send(Aq creates a compact data set that, when applied using bzr merge, has the same effect as merging from the source branch.

By default the merge directive is self−contained and can be applied to any branch containing submit_branch in its ancestory without needing access to the source branch.

If −−no−bundle is specified, then Bazaar doesn't send the contents of the revisions, but only a structured request to merge from the public_location. In that case the public_branch is needed and it must be up−to−date and accessible to the recipient. The public_branch is always included if known, so that people can check it later.

The submit branch defaults to the parent of the source branch, but can be overridden. Both submit branch and public branch will be remembered in branch.conf the first time they are used for a particular branch. The source branch defaults to that containing the working directory, but can be changed using −−from.

Both the submit branch and the public branch follow the usual behavior with respect to −−remember: If there is no default location set, the first send will set it (use −−no−remember to avoid setting it). After that, you can omit the location to use the default. To change the default, use −−remember. The value will only be saved if the location can be accessed.

In order to calculate those changes, bzr must analyse the submit branch. Therefore it is most efficient for the submit branch to be a local mirror. If a public location is known for the submit_branch, that location is used in the merge directive.

The default behaviour is to send the merge directive by mail, unless −o is given, in which case it is sent to a file.

Mail is sent using your preferred mail program. This should be transparent on Windows (it uses MAPI). On Unix, it requires the xdg−email utility. If the preferred client can't be found (or used), your editor will be used.

To use a specific mail program, set the mail_client configuration option. (For Thunderbird 1.5, this works around some bugs.) Supported values for specific clients are "claws", "evolution", "kmail", "mail.app" (MacOS X's Mail.app), "mutt", and "thunderbird"; generic options are "default", "editor", "emacsclient", "mapi", and "xdg−email". Plugins may also add supported clients.

If mail is being sent, a to address is required. This can be supplied either on the commandline, by setting the submit_to configuration option in the branch itself or the child_submit_to configuration option in the submit branch.

Two formats are currently supported: "4" uses revision bundle format 4 and merge directive format 2. It is significantly faster and smaller than older formats. It is compatible with Bazaar 0.19 and later. It is the default. "0.9" uses revision bundle format 0.9 and merge directive format 1. It is compatible with Bazaar 0.12 − 0.18.

The merge directives created by bzr send may be applied using bzr merge or bzr pull by specifying a file containing a merge directive as the location.

bzr send makes extensive use of public locations to map local locations into URLs that can be used by other people. See *(Aqbzr help configuration*(Aq to set them, and use *(Aqbzr info*(Aq to display them.

bzr serve

Options:

```
--allow-writes          By default the server is a readonly
                        server. Supplying --allow-writes
                        enables write access to the contents of
                        the served directory and below. Note
                        that \*(Aq\*(Aqbzr serve\*(Aq\*(Aq does not perform
                        authentication, so unless some form of
                        external authentication is arranged
                        supplying this option leads to global
                        uncontrolled write access to your file
                        system.
--client-timeout ARG    Override the default idle client timeout
                        (5min).
--directory ARG, -d     Serve contents of this directory.
--help, -h              Show help message.
--inet                  Serve on stdin/out for use from inetd or
                        sshd.
--listen ARG            Listen for connections on nominated
                        address.
--port ARG              Listen for connections on nominated
                        port. Passing 0 as the port number will
                        result in a dynamically allocated port.
                        The default port depends on the
                        protocol.
--protocol ARG          Protocol to serve.
--bzr                   The Bazaar smart server protocol over
                        TCP. (default port: 4155)
--quiet, -q             Only display errors and warnings.
--usage                 Show usage message and options.
--verbose, -v           Display more information.
```

Alias: server

Run the bzr server.

bzr server

Alias for "serve", see "bzr serve".

bzr shelve [FILE...]

Options:

```
--all                   Shelve all changes.
--destroy               Destroy removed changes instead of
                        shelving them.
--directory ARG, -d     Branch to operate on, instead of working
                        directory.
--help, -h              Show help message.
--list                  List shelved changes.
--message ARG, -m       Message string.
--quiet, -q             Only display errors and warnings.
--revision ARG, -r      See "help revisionspec" for details.
```

−−usage Show usage message and options.
−−verbose, −v Display more information.
−−writer ARG Method to use for writing diffs.
−−plain Plaintext diff output.

See also: configuration, unshelve

Temporarily set aside some changes from the current tree.

Shelve allows you to temporarily put changes you've made "on the shelf", ie. out of the way, until a later time when you can bring them back from the shelf with the 'unshelve' command. The changes are stored alongside your working tree, and so they aren't propagated along with your branch nor will they survive its deletion.

If shelve −−list is specified, previously−shelved changes are listed.

Shelve is intended to help separate several sets of changes that have been inappropriately mingled. If you just want to get rid of all changes and you don't need to restore them later, use revert. If you want to shelve all text changes at once, use shelve −−all.

If filenames are specified, only the changes to those files will be shelved. Other files will be left untouched.

If a revision is specified, changes since that revision will be shelved.

You can put multiple items on the shelf, and by default, 'unshelve' will restore the most recently shelved changes.

For complicated changes, it is possible to edit the changes in a separate editor program to decide what the file remaining in the working copy should look like. To do this, add the configuration option

change_editor = PROGRAM @new_path @old_path

where @new_path is replaced with the path of the new version of the file and @old_path is replaced with the path of the old version of the file. The PROGRAM should save the new file with the desired contents of the file in the working tree.

bzr sign−my−commits [LOCATION] [COMMITTER]

Options:
−−dry−run Don't actually sign anything, just print the revisions that would be signed.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Sign all commits by a given committer.

If location is not specified the local tree is used. If committer is not specified the default committer is used.

This does not sign commits that already have signatures.

bzr split TREE

Options:

```
--help, -h          Show help message.
--quiet, -q         Only display errors and warnings.
--usage             Show usage message and options.
--verbose, -v       Display more information.
```

See also: join

Split a subdirectory of a tree into a separate tree.

This command will produce a target tree in a format that supports rich roots, like 'rich-root' or 'rich-root-pack'. These formats cannot be converted into earlier formats like 'dirstate-tags'.

The TREE argument should be a subdirectory of a working tree. That subdirectory will be converted into an independent tree, with its own branch. Commits in the top-level tree will not apply to the new subtree.

bzr st

Alias for "status", see "bzr status".

bzr stat

Alias for "status", see "bzr status".

bzr status [FILE...]

Options:

```
--change ARG, -c    Select changes introduced by the
                    specified revision. See also "help
                    revisionspec".
--help, -h          Show help message.
--no-classify       Do not mark object type using indicator.
--no-pending        Don't show pending merges.
--quiet, -q         Only display errors and warnings.
--revision ARG, -r  See "help revisionspec" for details.
--short, -S         Use short status indicators.
--show-ids          Show internal object ids.
--usage             Show usage message and options.
--verbose, -v       Display more information.
--versioned, -V     Only show versioned files.
```

Aliases: st, stat

See also: diff, revert, status-flags

Display status summary.

This reports on versioned and unknown files, reporting them grouped by state. Possible states are:

added
: Versioned in the working copy but not in the previous revision.

removed
: Versioned in the previous revision but removed or deleted in the working copy.

renamed

Path of this file changed from the previous revision;
the text may also have changed. This includes files whose
parent directory was renamed.

modified
Text has changed since the previous revision.

kind changed
File kind has been changed (e.g. from file to directory).

unknown
Not versioned and not matching an ignore pattern.

Additionally for directories, symlinks and files with a changed executable bit, Bazaar indicates their type using a trailing character: '/', '@' or '*' respectively. These decorations can be disabled using the '−−no−classify' option.

To see ignored files use 'bzr ignored'. For details on the changes to file texts, use 'bzr diff'.

Note that −−short or −S gives status flags for each item, similar to Subversion's status command. To get output similar to svn −q, use bzr status −SV.

If no arguments are specified, the status of the entire working directory is shown. Otherwise, only the status of the specified files or directories is reported. If a directory is given, status is reported for everything inside that directory.

Before merges are committed, the pending merge tip revisions are shown. To see all pending merge revisions, use the −v option. To skip the display of pending merge information altogether, use the no−pending option or specify a file/directory.

To compare the working directory to a specific revision, pass a single revision to the revision argument.

To see which files have changed in a specific revision, or between two revisions, pass a revision range to the revision argument. This will produce the same results as calling 'bzr diff −−summarize'.

bzr switch [TO_LOCATION]

Options:

```
−−create−branch, −b      Create the target branch from this one
                         before switching to it.
−−directory ARG, −d      Branch to operate on, instead of working
                         directory.
−−force                  Switch even if local commits will be
                         lost.
−−help, −h               Show help message.
−−quiet, −q              Only display errors and warnings.
−−revision ARG, −r       See "help revisionspec" for details.
−−store                  Store and restore uncommitted changes in
                         the branch.
−−usage                  Show usage message and options.
−−verbose, −v            Display more information.
```

Set the branch of a checkout and update.

For lightweight checkouts, this changes the branch being referenced. For heavyweight checkouts, this

checks that there are no local commits versus the current bound branch, then it makes the local branch a mirror of the new location and binds to it.

In both cases, the working tree is updated and uncommitted changes are merged. The user can commit or revert these as they desire.

Pending merges need to be committed or reverted before using switch.

The path to the branch to switch to can be specified relative to the parent directory of the current branch. For example, if you are currently in a checkout of /path/to/branch, specifying 'newbranch' will find a branch at /path/to/newbranch.

Bound branches use the nickname of its master branch unless it is set locally, in which case switching will update the local nickname to be that of the master.

bzr tag [TAG_NAME]

Options:

−−delete Delete this tag rather than placing it.
−−directory ARG, −d Branch in which to place the tag.
−−force Replace existing tags.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: commit, tags

Create, remove or modify a tag naming a revision.

Tags give human−meaningful names to revisions. Commands that take a −r (−−revision) option can be given −rtag:X, where X is any previously created tag.

Tags are stored in the branch. Tags are copied from one branch to another along when you branch, push, pull or merge.

It is an error to give a tag name that already exists unless you pass −−force, in which case the tag is moved to point to the new revision.

To rename a tag (change the name but keep it on the same revsion), run *(Aq*(Aqbzr tag new−name −r tag:old−name*(Aq*(Aq and then *(Aq*(Aqbzr tag −−delete oldname*(Aq*(Aq.

If no tag name is specified it will be determined through the 'automatic_tag_name' hook. This can e.g. be used to automatically tag upstream releases by reading configure.ac. See *(Aq*(Aqbzr help hooks*(Aq*(Aq for details.

bzr tags

Options:

−−directory ARG, −d Branch whose tags should be displayed.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−show−ids Show internal object ids.
−−sort ARG Sort tags by different criteria.

--usage Show usage message and options.
--verbose, -v Display more information.

See also: tag

List tags.

This command shows a table of tag names and the revisions they reference.

bzr testament [BRANCH]

Options:
--help, -h Show help message.
--long Produce long-format testament.
--quiet, -q Only display errors and warnings.
--revision ARG, -r See "help revisionspec" for details.
--strict Produce a strict-format testament.
--usage Show usage message and options.
--verbose, -v Display more information.

Show testament (signing-form) of a revision.

bzr unbind

Options:
--directory ARG, -d Branch to operate on, instead of working directory.
--help, -h Show help message.
--quiet, -q Only display errors and warnings.
--usage Show usage message and options.
--verbose, -v Display more information.

See also: bind, checkouts

Convert the current checkout into a regular branch.

After unbinding, the local branch is considered independent and subsequent commits will be local only.

bzr uncommit [LOCATION]

Options:
--dry-run Don't actually make changes.
--force Say yes to all questions.
--help, -h Show help message.
--keep-tags Keep tags that point to removed revisions.
--local Only remove the commits from the local branch when in a checkout.
--quiet, -q Only display errors and warnings.
--revision ARG, -r See "help revisionspec" for details.
--usage Show usage message and options.
--verbose, -v Display more information.

See also: commit

Remove the last committed revision.

−−verbose will print out what is being removed. −−dry−run will go through all the motions, but not actually remove anything.

If −−revision is specified, uncommit revisions to leave the branch at the specified revision. For example, "bzr uncommit −r 15" will leave the branch at revision 15.

Uncommit leaves the working tree ready for a new commit. The only change it may make is to restore any pending merges that were present before the commit.

bzr unshelve [SHELF_ID]

Options:

−−action ARG The action to perform.
−−apply Apply changes and remove from the shelf.
−−delete−only Delete changes without applying them.
−−dry−run Show changes, but do not apply or remove them.
−−keep Apply changes but don't delete them.
−−preview Instead of unshelving the changes, show the diff that would result from unshelving.
−−directory ARG, −d Branch to operate on, instead of working directory.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

See also: shelve

Restore shelved changes.

By default, the most recently shelved changes are restored. However if you specify a shelf by id those changes will be restored instead. This works best when the changes don't depend on each other.

bzr up

Alias for "update", see "bzr update".

bzr update [DIR]

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−show−base Show base revision text in conflicts.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Alias: up

See also: pull, status−flags, working−trees

Update a working tree to a new revision.

This will perform a merge of the destination revision (the tip of the branch, or the specified revision) into the working tree, and then make that revision the basis revision for the working tree.

You can use this to visit an older revision, or to update a working tree that is out of date from its branch.

If there are any uncommitted changes in the tree, they will be carried across and remain as uncommitted changes after the update. To discard these changes, use 'bzr revert'. The uncommitted changes may conflict with the changes brought in by the change in basis revision.

If the tree's branch is bound to a master branch, bzr will also update the branch from the master.

You cannot update just a single file or directory, because each Bazaar working tree has just a single basis revision. If you want to restore a file that has been removed locally, use 'bzr revert' instead of 'bzr update'. If you want to restore a file to its state in a previous revision, use 'bzr revert' with a '-r' option, or use 'bzr cat' to write out the old content of that file to a new location.

The 'dir' argument, if given, must be the location of the root of a working tree to update. By default, the working tree that contains the current working directory is used.

bzr upgrade [URL]

Options:

--clean	Remove the backup.bzr directory if successful.
--dry-run	Show what would be done, but don't actually do anything.
--format ARG	Upgrade to a specific format. See "bzr help formats" for details.
--2a	Format for the bzr 2.0 series. Uses group-compress storage. Provides rich roots which are a one-way transition.
--default	Format for the bzr 2.0 series. Uses group-compress storage. Provides rich roots which are a one-way transition.
--development-colo	The 2a format with experimental support for colocated branches.
--pack-0.92	Pack-based format used in 1.x series. Introduced in 0.92. Interoperates with bzr repositories before 0.92 but cannot be read by bzr < 0.92.
--help, -h	Show help message.
--quiet, -q	Only display errors and warnings.
--usage	Show usage message and options.
--verbose, -v	Display more information.

See also: check, formats, reconcile

Upgrade a repository, branch or working tree to a newer format.

When the default format has changed after a major new release of Bazaar, you may be informed during certain operations that you should upgrade. Upgrading to a newer format may improve performance or make new features available. It may however limit interoperability with older repositories or with older versions of Bazaar.

If you wish to upgrade to a particular format rather than the current default, that can be specified using the --format option. As a consequence, you can use the upgrade command this way to "downgrade" to an earlier format, though some conversions are a one way process (e.g. changing from the 1.x default to the 2.x default) so downgrading is not always possible.

A backup.bzr.~#~ directory is created at the start of the conversion process (where # is a number). By default, this is left there on completion. If the conversion fails, delete the new .bzr directory and rename this one back in its place. Use the −−clean option to ask for the backup.bzr directory to be removed on successful conversion. Alternatively, you can delete it by hand if everything looks good afterwards.

If the location given is a shared repository, dependent branches are also converted provided the repository converts successfully. If the conversion of a branch fails, remaining branches are still tried.

For more information on upgrades, see the Bazaar Upgrade Guide, http://doc.bazaar.canonical.com/latest/en/upgrade−guide/.

bzr verify−signatures [LOCATION]

Options:

−−acceptable−keys ARG, −k Comma separated list of GPG key patterns which are acceptable for verification.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Verify all commit signatures.

Verifies that all commits in the branch are signed by known GnuPG keys.

bzr version

Options:

−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−short Print just the version number.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Show version of bzr.

bzr version−info [LOCATION]

Options:

−−all Include all possible information.
−−check−clean Check if tree is clean.
−−format ARG Select the output format.
−−custom Version info in Custom template−based format.
−−python Version info in Python format.
−−rio Version info in RIO (simple text) format (default).
−−help, −h Show help message.
−−include−file−revisions Include the last revision for each file.
−−include−history Include the revision−history.
−−quiet, −q Only display errors and warnings.
−−revision ARG, −r See "help revisionspec" for details.
−−template ARG Template for the output.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Show version information about this tree.

You can use this command to add information about version into source code of an application. The output can be in one of the supported formats or in a custom format based on a template.

For example:

```
bzr version-info --custom \
  --template="#define VERSION_INFO \"Project 1.2.3 (r{revno})\"\n"
```

will produce a C header file with formatted string containing the current revision number. Other supported variables in templates are:

```
* {date} - date of the last revision
* {build_date} - current date
* {revno} - revision number
* {revision_id} - revision id
* {branch_nick} - branch nickname
* {clean} - 0 if the source tree contains uncommitted changes,
        otherwise 1
```

bzr view [FILE...]

Options:

--all	Apply list or delete action to all views.
--delete	Delete the view.
--help, -h	Show help message.
--name ARG	Name of the view to define, list or delete.
--quiet, -q	Only display errors and warnings.
--switch ARG	Name of the view to switch to.
--usage	Show usage message and options.
--verbose, -v	Display more information.

Manage filtered views.

Views provide a mask over the tree so that users can focus on a subset of a tree when doing their work. After creating a view, commands that support a list of files - status, diff, commit, etc - effectively have that list of files implicitly given each time. An explicit list of files can still be given but those files must be within the current view.

In most cases, a view has a short life-span: it is created to make a selected change and is deleted once that change is committed. At other times, you may wish to create one or more named views and switch between them.

To disable the current view without deleting it, you can switch to the pseudo view called *(Aq*(Aqoff*(Aq*(Aq. This can be useful when you need to see the whole tree for an operation or two (e.g. merge) but want to switch back to your view after that.

Examples:

To define the current view:

```
bzr view file1 dir1 ...
```

To list the current view:

bzr view

To delete the current view:

bzr view −−delete

To disable the current view without deleting it:

bzr view −−switch off

To define a named view and switch to it:

bzr view −−name view−name file1 dir1 ...

To list a named view:

bzr view −−name view−name

To delete a named view:

bzr view −−name view−name −−delete

To switch to a named view:

bzr view −−switch view−name

To list all views defined:

bzr view −−all

To delete all views:

bzr view −−delete −−all

bzr whoami [NAME]

Options:

−−branch Set identity for the current branch instead of globally.
−−directory ARG, −d Branch to operate on, instead of working directory.
−−email Display email address only.
−−help, −h Show help message.
−−quiet, −q Only display errors and warnings.
−−usage Show usage message and options.
−−verbose, −v Display more information.

Show or set bzr user id.

Examples:

Show the email of the current user:

bzr whoami −−email

Set the current user:

bzr whoami "Frank Chu <fchu@example.com>"

ENVIRONMENT

BZRPATH
Path where bzr is to look for shell plugin external commands.

BZR_EMAIL
E−Mail address of the user. Overrides EMAIL.

EMAIL E−Mail address of the user.

BZR_EDITOR
Editor for editing commit messages. Overrides EDITOR.

EDITOR
Editor for editing commit messages.

BZR_PLUGIN_PATH
Paths where bzr should look for plugins.

BZR_DISABLE_PLUGINS
Plugins that bzr should not load.

BZR_PLUGINS_AT
Plugins to load from a directory not in BZR_PLUGIN_PATH.

BZR_HOME
Directory holding .bazaar config dir. Overrides HOME.

BZR_HOME (Win32)
Directory holding bazaar config dir. Overrides APPDATA and HOME.

BZR_REMOTE_PATH
Full name of remote 'bzr' command (for bzr+ssh:// URLs).

BZR_SSH
Path to SSH client, or one of paramiko, openssh, sshcorp, plink or lsh.

BZR_LOG
Location of .bzr.log (use '/dev/null' to suppress log).

BZR_LOG (Win32)
Location of .bzr.log (use 'NUL' to suppress log).

BZR_COLUMNS
Override implicit terminal width.

BZR_CONCURRENCY
Number of processes that can be run concurrently (selftest)

BZR_PROGRESS_BAR
Override the progress display. Values are 'none' or 'text'.

BZR_PDB
Control whether to launch a debugger on error.

BZR_SIGQUIT_PDB
Control whether SIGQUIT behaves normally or invokes a breakin debugger.

BZR_TEXTUI_INPUT
Force console input mode for prompts to line−based (instead of char−based).

FILES

~/.bazaar/bazaar.conf

Contains the user's default configuration. The section **[DEFAULT]** is used to define general configuration that will be applied everywhere. The section **[ALIASES]** can be used to create command aliases for commonly used options.

A typical config file might look something like:

[DEFAULT]
email=John Doe <jdoe@isp.com>
[ALIASES]
commit = commit −−strict
log10 = log −−short −r −10..−1

SEE ALSO

http://bazaar.canonical.com/