



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'flatpak-build-commit-from.1'***

***\$ man flatpak-build-commit-from.1***

FLATPAK BUILD-COMMI(1) flatpak build-commit-from FLATPAK BUILD-COMMI(1)

#### NAME

flatpak-build-commit-from - Create new commits based on existing one  
(possibly from another repository)

#### SYNOPSIS

flatpak build-commit-from [OPTION...] DST-REPO DST-REF...

#### DESCRIPTION

Creates new commits on the DST-REF branch in the DST-REPO, with the contents (and most of the metadata) taken from another branch, either from another repo, or from another branch in the same repository.

The collection ID set on DST-REPO (if set) will be used for the newly created commits.

This command is very useful when you want to maintain a branch with a clean history that has no unsigned or broken commits. For instance, you can import the head from a different repository from an automatic builder when you've verified that it worked. The new commit will have no parents or signatures from the autobuilder, and can be properly signed with the official key.

Any deltas that affect the original commit and that match parent commits in the destination repository are copied and rewritten for the new commit id.

## OPTIONS

The following options are understood:

`-h, --help`

Show help options and exit.

`--src-repo=SRC-REPO`

The (local) repository to pull the source branch from. Defaults to the destination repository.

`--src-ref=SRC-REF`

The branch to use as the source for the new commit. Defaults to the same as the destination ref, which is useful only if a different source repo has been specified.

`--extra-collection-id=COLLECTION-ID`

Add an extra collection-ref binding for this collection, in addition to whatever would normally be added due to the destination repository collection id. This option can be used multiple times.

`--subset=SUBSET`

Mark the commit to be included in the named subset. This will cause the commit to be put in the named subset summary (in addition to the main one), allowing users to see only this subset instead of the whole repo.

`--untrusted`

The source repository is not trusted, all objects are copied (not hardlinked) and all checksums are verified.

`-s, --subject=SUBJECT`

One line subject for the commit message. If not specified, will be taken from the source commit.

`-b, --body=BODY`

Full description for the commit message. If not specified, will be taken from the source commit.

`--update-appstream`

Update the appstream branch after the build.

`--no-update-summary`

Don't update the summary file after the new commit is added. This means the repository will not be useful for serving over http until `build-update-repo` has been run. This is useful if you want to do multiple repo operations before finally updating the summary.

`--force`

Create new commit even if the content didn't change from the existing branch head.

`--disable-fsync`

Don't fsync when writing to the repository. This can result in data loss in exceptional situations, but can improve performance when working with temporary or test repositories.

`--gpg-sign=KEYID`

Sign the commit with this GPG key. This option can be used multiple times.

`--gpg-homedir=PATH`

GPG Homedir to use when looking for keyrings

`--end-of-life=REASON`

Mark build as end-of-life

`--end-of-life-rebase=OLDID=NEWID`

Mark new refs as end-of-life. Unlike `--end-of-life`, this one takes an ID that supersedes the current one. By the user's request, the application data may be preserved for the new application. Note, this is actually a prefix match, so if you say `org.the.app=org.new.app`, then something like `org.the.app.Locale` will be rebased to `org.new.app.Locale`.

`--timestamp=TIMESTAMP`

Override the timestamp of the commit. Use an ISO 8601 formatted date, or NOW for the current time

`--disable-fsync`

Don't fsync when writing to the repository. This can result in data loss in exceptional situations, but can improve performance when

working with temporary or test repositories.

`-v, --verbose`

Print debug information during command processing.

`--ostree-verbose`

Print OSTree debug information during command processing.

## EXAMPLES

To revert a commit to the commit before:

```
$ flatpak build-commit-from --timestamp=NOW
```

```
--src-ref=app/org.gnome.gedit/x86_64/master^ repo
```

```
app/org.gnome.gedit/x86_64/master
```

## SEE ALSO

[ostree\(1\)](#), [flatpak\(1\)](#), [flatpak-build-init\(1\)](#), [flatpak-build\(1\)](#),

[flatpak-build-finish\(1\)](#), [flatpak-build-sign\(1\)](#), [flatpak-build-update-](#)

[repo\(1\)](#)

[flatpak](#)

[FLATPAK BUILD-COMMI\(1\)](#)