Rocky Enterprise Linux 9.2 Manual Pages on command 'flatpak-build-finish.1'

$ man flatpak-build-finish.1

FLATPAK BUILD-FINIS(1) flatpak build-finish FLATPAK BUILD-FINIS(1)

NAME

flatpak-build-finish - Finalize a build directory

SYNOPSIS

flatpak build-finish [OPTION...] DIRECTORY

DESCRIPTION

Finalizes a build directory, to prepare it for exporting. DIRECTORY is the name of the directory.

The result of this command is that desktop files, icons and D-Bus service files from the files subdirectory are copied to a new export subdirectory. In the metadata file, the command key is set in the [Application] group, and the supported keys in the [Environment] group are set according to the options.

As part of finalization you can also specify permissions that the app needs, using the various options specified below. Additionally during finalization the permissions from the runtime are inherited into the app unless you specify --no-inherit-permissions

You should review the exported files and the application metadata

before creating and distributing an application bundle.

It is an error to run build-finish on a directory that has not been initialized as a build directory, or has already been finalized.

OPTIONS

The following options are understood:

-h, --help

Show help options and exit.

--command=COMMAND

The command to use. If this option is not specified, the first executable found in files/bin is used.

Note that the command is used when the application is run via flatpak run, and does not affect what gets executed when the application is run in other ways, e.g. via the desktop file or D-Bus activation.

--require-version=MAJOR.MINOR.MICRO

Require this version or later of flatpak to install/update to this build.

--share=SUBSYSTEM

Share a subsystem with the host session. This updates the [Context] group in the metadata. SUBSYSTEM must be one of: network, ipc. This option can be used multiple times.

--unshare=SUBSYSTEM

Don't share a subsystem with the host session. This updates the [Context] group in the metadata. SUBSYSTEM must be one of: network, ipc. This option can be used multiple times.

--socket=SOCKET

Expose a well-known socket to the application. This updates the [Context] group in the metadata. SOCKET must be one of: x11, wayland, fallback-x11, pulseaudio, system-bus, session-bus, ssh-auth, pcsc, cups. This option can be used multiple times.

The fallback-x11 option makes the X11 socket available only if there is no Wayland socket. This option was introduced in 0.11.3.

To support older Flatpak releases, specify both x11 and

fallback-x11. The fallback-x11 option takes precedence when both are supported.

--nosocket=SOCKET

Don't expose a well known socket to the application. This updates the [Context] group in the metadata. SOCKET must be one of: x11, wayland, fallback-x11, pulseaudio, system-bus, session-bus, ssh-auth, pcsc, cups. This option can be used multiple times.

--device=DEVICE

Expose a device to the application. This updates the [Context] group in the metadata. DEVICE must be one of: dri, kvm, shm, all. This option can be used multiple times.

--nodevice=DEVICE

Don't expose a device to the application. This updates the [Context] group in the metadata. DEVICE must be one of: dri, kvm, shm, all. This option can be used multiple times.

--allow=FEATURE

Allow access to a specific feature. This updates the [Context] group in the metadata. FEATURE must be one of: devel, multiarch, bluetooth, canbus, per-app-dev-shm. This option can be used multiple times.

The devel feature allows the application to access certain syscalls such as ptrace(), and perf_event_open().

The multiarch feature allows the application to execute programs compiled for an ABI other than the one supported natively by the system. For example, for the x86_64 architecture, 32-bit x86 binaries will be allowed as well.

The bluetooth feature allows the application to use bluetooth (AF_BLUETOOTH) sockets. Note, for bluetooth to fully work you must also have network access.

The canbus feature allows the application to use canbus (AF_CAN) sockets. Note, for this work you must also have network access.

The per-app-dev-shm feature shares a single instance of /dev/shm between the application, any unrestricted subsandboxes that it

creates, and any other instances of the application that are launched while it is running.

--disallow=FEATURE

Disallow access to a specific feature. This updates the [Context] group in the metadata. FEATURE must be one of: devel, multiarch, bluetooth, canbus, per-app-dev-shm. This option can be used multiple times.

--filesystem=FS

Allow the application access to a subset of the filesystem. This updates the [Context] group in the metadata. FS can be one of: home, host, host-os, host-etc, xdg-desktop, xdg-documents, xdg-download, xdg-music, xdg-pictures, xdg-public-share, xdg-templates, xdg-videos, xdg-run, xdg-config, xdg-cache, xdg-data, an absolute path, or a homedir-relative path like ~/dir or paths relative to the xdg dirs, like xdg-download/subdir. The optional :ro suffix indicates that the location will be read-only. The optional :create suffix indicates that the location will be read-write and created if it doesn't exist. This option can be used multiple times. See the "[Context] filesystems" list in flatpak-metadata(5) for details of the meanings of these filesystems.

--nofilesystem=FILESYSTEM

Remove access to the specified subset of the filesystem from the application. This overrides to the Context section from the application metadata. FILESYSTEM can be one of: home, host, host-os, host-etc, xdg-desktop, xdg-documents, xdg-download, xdg-music, xdg-pictures, xdg-public-share, xdg-templates, xdg-videos, an absolute path, or a homedir-relative path like ~/dir. This option can be used multiple times.

--add-policy=SUBSYSTEM.KEY=VALUE

Add generic policy option. For example, "--add-policy=subsystem.key=v1 --add-policy=subsystem.key=v2" would map to this metadata:

[Policy subsystem]

key=v1;v2;

This option can be used multiple times.

--remove-policy=SUBSYSTEM.KEY=VALUE

Remove generic policy option. This option can be used multiple times.

--env=VAR=VALUE

Set an environment variable in the application. This updates the [Environment] group in the metadata. This overrides to the Context section from the application metadata. This option can be used multiple times.

--unset-env=VAR

Unset an environment variable in the application. This updates the unset-environment entry in the [Context] group of the metadata. This option can be used multiple times.

--env-fd=FD

Read environment variables from the file descriptor FD, and set them as if via --env. This can be used to avoid environment variables and their values becoming visible to other users. Each environment variable is in the form VAR=VALUE followed by a zero byte. This is the same format used by env -0 and /proc/*/environ.

--own-name=NAME

Allow the application to own the well known name NAME on the session bus. If NAME ends with .*, it allows the application to own all matching names. This updates the [Session Bus Policy] group in the metadata. This option can be used multiple times.

--talk-name=NAME

Allow the application to talk to the well known name NAME on the session bus. If NAME ends with .*, it allows the application to talk to all matching names. This updates the [Session Bus Policy] group in the metadata. This option can be used multiple times.

--system-own-name=NAME

Allow the application to own the well known name NAME on the system

bus. If NAME ends with .*, it allows the application to own all matching names. This updates the [System Bus Policy] group in the metadata. This option can be used multiple times.

--system-talk-name=NAME

Allow the application to talk to the well known name NAME on the system bus. If NAME ends with .*, it allows the application to talk to all matching names. This updates the [System Bus Policy] group in the metadata. This option can be used multiple times.

--persist=FILENAME

If the application doesn't have access to the real homedir, make the (homedir-relative) path FILENAME a bind mount to the corresponding path in the per-application directory, allowing that location to be used for persistent data. This updates the [Context] group in the metadata. This option can be used multiple times.

--runtime=RUNTIME, --sdk=SDK

Change the runtime or sdk used by the app to the specified partial ref. Unspecified parts of the ref are taken from the old values or defaults.

--metadata=GROUP=KEY[=VALUE]

Set a generic key in the metadata file. If value is left out it will be set to "true".

--extension=NAME=VARIABLE[=VALUE]

Add extension point info. See the documentation for flatpak-metadata(5) for the possible values of VARIABLE and VALUE.

--remove-extension=NAME

Remove extension point info.

--extension-priority=VALUE

Set the priority (library override order) of the extension point. Only useful for extensions. 0 is the default, and higher value means higher priority.

--extra-data=NAME:SHA256:DOWNLOAD-SIZE:INSTALL-SIZE:URL

Adds information about extra data uris to the app. These will be downloaded and verified by the client when the app is installed and

placed in the /app/extra directory. You can also supply an

/app/bin/apply_extra script that will be run after the files are

downloaded.

--no-exports

Don't look for exports in the build.

--no-inherit-permissions

Don't inherit runtime permissions in the app.

-v, --verbose

Print debug information during command processing.

--ostree-verbose

Print OSTree debug information during command processing.

EXAMPLES

$ flatpak build-finish /build/my-app --socket=x11 --share=ipc

Exporting share/applications/gnome-calculator.desktop

Exporting share/dbus-1/services/org.gnome.Calculator.SearchProvider.service

More than one executable

Using gcalccmd as command

Please review the exported files and the metadata

SEE ALSO

flatpak(1), flatpak-build-init(1), flatpak-build(1), flatpak-build-

export(1)

flatpak FLATPAK BUILD-FINIS(1)