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

$ man flatpak-build.1

FLATPAK BUILD(1) flatpak build FLATPAK BUILD(1)

NAME

flatpak-build - Build in a directory

SYNOPSIS

flatpak build [OPTION...] DIRECTORY [COMMAND [ARG...]]

DESCRIPTION

Runs a build command in a directory. DIRECTORY must have been initialized with flatpak build-init.

The sdk that is specified in the metadata file in the directory is mounted at /usr and the files and var subdirectories are mounted at /app and /var, respectively. They are writable, and their contents are preserved between build commands, to allow accumulating build artifacts there.

OPTIONS

The following options are understood:

-h, --help

Show help options and exit.

-v, --verbose

Print debug information during command processing.

--ostree-verbose

Print OSTree debug information during command processing.

-r, --runtime

Use the non-devel runtime that is specified in the application metadata instead of the devel runtime.

-p, --die-with-parent

Kill the build process and all children when the launching process dies.

--bind-mount=DEST=SOURCE

Add a custom bind mount in the build namespace. Can be specified multiple times.

--build-dir=PATH

Start the build in this directory (default is in the current directory).

--share=SUBSYSTEM

Share a subsystem with the host session. This overrides the Context section from the application 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 overrides the Context section from the application 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 overrides to the Context section from the application 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.

--nosocket=SOCKET

Don't expose a well-known socket to the application. This overrides to the Context section from the application 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 overrides to the Context section from the application 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 overrides to the Context section from the application 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.

See flatpak-build-finish(1) for the meaning of the various features.

--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=FILESYSTEM[:ro|:create]

Allow the application access to a subset of the filesystem. 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, 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.

--with-appdir

Expose and configure access to the per-app storage directory in $HOME/.var/app. This is not normally useful when building, but helps when testing built apps.

--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 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 overrides the unset-environment entry in the [Context] group of the metadata, and the [Environment] group. 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. This overrides to the Context section from the application 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. This overrides to the Context section from the application 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. This overrides to the Context section from the application 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. This overrides to the Context section from the application 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 overrides to the Context section from the application metadata. This option can be used multiple times.

--sdk-dir=DIR

Normally if there is a usr directory in the build dir, this is used

for the runtime files (this can be created by --writable-sdk or --type=runtime arguments to build-init). If you specify --sdk-dir, this directory will be used instead. Use this if you passed --sdk-dir to build-init.

--readonly

Mount the normally writable destination directories read-only. This can be useful if you want to run something in the sandbox but guarantee that it doesn't affect the build results. For example tests.

--metadata=FILE

Use the specified filename as metadata in the exported app instead of the default file (called metadata). This is useful if you build multiple things from a single build tree (such as both a platform and a sdk).

--log-session-bus

Log session bus traffic. This can be useful to see what access you need to allow in your D-Bus policy.

--log-system-bus

Log system bus traffic. This can be useful to see what access you need to allow in your D-Bus policy.

EXAMPLES

$ flatpak build /build/my-app rpmbuild my-app.src.rpm

SEE ALSO

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