



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

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'flatpak-build-init.1' command

\$ man flatpak-build-init.1

FLATPAK BUILD-INIT(1) flatpak build-init FLATPAK BUILD-INIT(1)

NAME

flatpak-build-init - Initialize a build directory

SYNOPSIS

flatpak build-init [OPTION...] DIRECTORY APPNAME SDK RUNTIME [BRANCH]

DESCRIPTION

Initializes a separate build directory. DIRECTORY is the name of the directory. APPNAME is the application id of the app that will be built. SDK and RUNTIME specify the sdk and runtime that the application should be built against and run in. BRANCH specify the version of sdk and runtime

Initializes a directory as build directory which can be used as target directory of flatpak build. It creates a metadata inside the given directory. Additionally, empty files and var subdirectories are created.

It is an error to run build-init on a directory that has already been initialized as a build directory.

OPTIONS

The following options are understood:

-h, --help

Show help options and exit.

--arch=ARCH

The architecture to use. See flatpak --supported-arches for

architectures supported by the host.

`-v, --var=RUNTIME`

Initialize var from the named runtime.

`-w, --writable-sdk`

Initialize /usr with a copy of the sdk, which is writable during flatpak build. This can be used if you need to install build tools in /usr during the build. This is stored in the usr subdirectory of the app dir, but will not be part of the final app.

`--tag=TAG`

Add a tag to the metadata file. This option can be used multiple times.

`--sdk-extension=EXTENSION`

When using `--writable-sdk`, in addition to the sdk, also install the specified extension. This option can be used multiple times.

`--extension=NAME=VARIABLE[=VALUE]`

Add extension point info.

`--sdk-dir`

Specify a custom subdirectory to use instead of usr for `--writable-sdk`.

`--update`

Re-initialize the sdk and var, don't fail if already initialized.

`--base=APP`

Initialize the application with files from another specified application.

`--base-version=VERSION`

Specify the version to use for `--base`. If not specified, will default to "master".

`--base-extension=EXTENSION`

When using `--base`, also install the specified extension from the app. This option can be used multiple times.

`--type=TYPE`

This can be used to build different types of things. The default is "app" which is a regular app, but "runtime" creates a runtime based

on an existing runtime, and "extension" creates an extension for an app or runtime.

`--extension-tag=EXTENSION_TAG`

If building an extension, the tag to use when searching for the mount point of the extension.

`--verbose`

Print debug information during command processing.

`--ostree-verbose`

Print OSTree debug information during command processing.

EXAMPLES

```
$ flatpak build-init /build/my-app org.example.myapp org.gnome.Sdk  
org.gnome.Platform 3.36
```

SEE ALSO

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

[flatpak](#)

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