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-metadata.5' command

$ man flatpak-metadata.5

FLATPAK METADATA(5) flatpak metadata FLATPAK METADATA(5)

NAME

flatpak-metadata - Information about an application or runtime

DESCRIPTION

Flatpak uses metadata files to describe applications and runtimes. The metadata file for a deployed application or runtime is placed in the toplevel deploy directory. For example, the metadata for the locally installed application org.gnome.Calculator is in ~/.local/share/flatpak/app/org.gnome.Calculator/current/active/metadata.

Most aspects of the metadata configuration can be overridden when launching applications, either temporarily via options of the flatpak run command, or permanently with the flatpak override command.

A metadata file describing the effective configuration is available inside the running sandbox at /.flatpak-info. For compatibility with older Flatpak versions, /run/user/$UID/flatpak-info is a symbolic link to the same file.

FILE FORMAT

The metadata file is using the same .ini file format that is used for systemd unit files or application .desktop files.

[Application] or [Runtime]

Metadata for applications starts with an [Application] group, metadata for runtimes with a [Runtime] group.

The following keys can be present in these groups:

name (string)

The name of the application or runtime. This key is mandatory.

runtime (string)

The fully qualified name of the runtime that is used by the application. This key is mandatory for applications.

sdk (string)

The fully qualified name of the sdk that matches the runtime. Available since 0.1.

command (string)

The command to run. Only relevant for applications. Available since 0.1.

required-flatpak (string list)

The required version of Flatpak to run this application or runtime. For applications, this was available since 0.8.0. For runtimes, this was available since 0.9.1, and backported to 0.8.3 for the 0.8.x branch.

Flatpak after version 1.4.3 and 1.2.5 support multiple versions here. This can be useful if you need to support features that are backported to a previous stable series. For example if you want to use a feature added in 1.6.0 that was also backported to 1.4.4 you would use 1.6.0;1.4.4;. Note that older versions of flatpak will just use the first element in the list, so make that the largest version.

tags (string list)

Tags to include in AppStream XML. Typical values in use on Flathub include beta, stable, proprietary and upstream-maintained. Available since 0.4.12.

[Context]

This group determines various system resources that may be shared with the application when it is run in a flatpak sandbox.

All keys in this group (and the group itself) are optional.

shared (list)

List of subsystems to share with the host system. Possible

subsystems: network, ipc. Available since 0.3.

sockets (list)

List of well-known sockets to make available in the sandbox.

Possible sockets: x11, wayland, fallback-x11, pulseaudio, session-bus, system-bus, ssh-auth, pcsc, cups. When making a socket available, flatpak also sets well-known environment variables like DISPLAY or DBUS_SYSTEM_BUS_ADDRESS to let the application find sockets that are not in a fixed location. Available since 0.3.

devices (list)

List of devices to make available in the sandbox. Possible values:

dri

Graphics direct rendering (/dev/dri). Available since 0.3.

kvm

Virtualization (/dev/kvm). Available since 0.6.12.

all

All device nodes in /dev, but not /dev/shm (which is separately specified). Available since 0.6.6.

shm

Access to the host /dev/shm (/dev/shm). Available since 1.6.1.

filesystems (list)

List of filesystem subsets to make available to the application.

Possible values:

home

The entire home directory. Available since 0.3.

home/path

Alias for ~/path Available since 1.10. For better compatibility with older Flatpak versions, prefer to write this as ~/path.

host

The entire host file system, except for directories that are handled specially by Flatpak. In particular, this shares /home, /media, /opt, /run/media and /srv if they exist.

/dev is not shared: use devices=all; instead.

Parts of /sys are always shared. This option does not make

additional files in /sys available.

Additionally, this keyword provides all of the same directories in /run/host as the host-os and host-etc keywords. If this keyword is used in conjunction with one of the host- keywords, whichever access level is higher (more permissive) will be used for the directories in /run/host: for example, host:rw;host-os:ro; is equivalent to host:rw;.

These other reserved directories are currently excluded: /app, /bin, /boot, /etc, /lib, /lib32, /lib64, /proc, /root, /run, /sbin, /tmp, /usr, /var.

Available since 0.3.

host-os

The host operating system's libraries, executables and static data from /usr and the related directories /bin, /lib, /lib32, /lib64, /sbin. Additionally, this keyword provides access to a subset of /etc that is associated with packaged libraries and executables, even if the host-etc keyword was not used: /etc/ld.so.cache, (used by the dynamic linker) and /etc/alternatives (on operating systems that use it, such as Debian).

To avoid conflicting with the Flatpak runtime, these are mounted in the sandbox at /run/host/usr, /run/host/etc/ld.so.cache and so on.

Available since 1.7.

host-etc

The host operating system's configuration from /etc.

To avoid conflicting with the Flatpak runtime, this is mounted in the sandbox at /run/host/etc.

Available since 1.7.

xdg-desktop, xdg-documents, xdg-download, xdg-music, xdg-pictures, xdg-public-share, xdg-videos, xdg-templates

freedesktop.org special directories[1]. Available since 0.3.

xdg-desktop/path, xdg-documents/path, etc.

Subdirectories of freedesktop.org special directories.

Available since 0.4.13.

xdg-cache, xdg-config, xdg-data

Directories defined by the freedesktop.org Base Directory Specification[2]. Available since 0.6.14.

xdg-cache/path, xdg-config/path, xdg-data/path

Subdirectories of directories defined by the freedesktop.org Base Directory Specification. Available since 0.6.14.

xdg-run/path

Subdirectories of the XDG_RUNTIME_DIR defined by the freedesktop.org Base Directory Specification. Note that xdg-run on its own is not supported. Available since 0.4.13.

/path

An arbitrary absolute path. Available since 0.3.

~/path

An arbitrary path relative to the home directory. Available since 0.3.

~

The same as home. Available since 1.10. For better compatibility with older Flatpak versions, prefer to write this as home.

One of the above followed by :ro

Make the given directory available read-only.

One of the above followed by :rw

Make the given directory available read/write. This is the default.

One of the above followed by :create

Make the given directory available read/write, and create it if it does not already exist.

persistent (list)

List of homedir-relative paths to make available at the corresponding path in the per-application home directory, allowing the locations to be used for persistent data when the application

does not have access to the real homedir. For instance making ".myapp" persistent would make "~/.myapp" in the sandbox a bind mount to "~/.var/app/org.my.App/.myapp", thus allowing an unmodified application to save data in the per-application location. Available since 0.3.

features (list)

List of features available or unavailable to the application, currently from the following list:

devel

Allow system calls used by development-oriented tools such as perf, strace and gdb. Available since 0.6.10.

multiarch

Allow running multilib/multiarch binaries, for example i386 binaries in an x86_64 environment. Available since 0.6.12.

bluetooth

Allow the application to use bluetooth (AF_BLUETOOTH) sockets. Note, for bluetooth to fully work you must also have network access. Available since 0.11.8.

canbus

Allow the application to use canbus (AF_CAN) sockets. Note, for this work you must also have network access. Available since 1.0.3.

per-app-dev-shm

Share a single instance of /dev/shm between all instances of this application run by the same user ID, including sub-sandboxes. If the application has the shm device permission in its devices list, then this feature flag is ignored. Available since 1.12.0.

A feature can be prefixed with ! to indicate the absence of that feature, for example !devel if development and debugging are not allowed.

unset-environment (list)

A list of names of environment variables to unset. Note that

environment variables to set to a value (possibly empty) appear in the [Environment] group instead.

[Instance]

This group only appears in /.flatpak-info for a running app, and not in the metadata files written by application authors. It is filled in by Flatpak itself.

instance-id (string)

The ID of the running instance. This number is used as the name of the directory in XDG_RUNTIME_DIR/.flatpak where Flatpak stores information about this instance.

instance-path (string)

The absolute path on the host system of the app's persistent storage area in $HOME/.var.

app-path (string)

The absolute path on the host system of the app's app files, as mounted at /app inside the container. Available since 0.6.10. Since 1.12.0, if flatpak run was run with the --app-path option, this key gives the absolute path of whatever files were mounted on /app, even if that differs from the app's normal app files. If flatpak run was run with --app-path= (resulting in an empty directory being mounted on /app), the value is set to the empty string.

original-app-path (string)

If flatpak run was run with the --app-path option, this key gives the absolute path of the app's original files, as mounted at /run/parent/app inside the container. Available since 1.12.0. If this key is missing, the app files are given by app-path.

app-commit (string)

The commit ID of the application that is running. The filename of a deployment of this commit can be found in original-app-path if present, or app-path otherwise.

app-extensions (list of strings)

A list of app extensions that are mounted into the running

instance. The format for each list item is EXTENSION_ID=COMMIT. If original-app-path is present, the extensions are mounted below /run/parent/app; otherwise, they are mounted below /app.

branch (string)

The branch of the app, for example stable. Available since 0.6.10.

arch (string)

The architecture of the running instance.

flatpak-version (string)

The version number of the Flatpak version that ran this app. Available since 0.6.11.

runtime-path (string)

The absolute path on the host system of the app's runtime files, as mounted at /usr inside the container. Available since 0.6.10. Since 1.12.0, if flatpak run was run with the --usr-path option, this key gives the absolute path of whatever files were mounted on /usr, even if that differs from the app's normal runtime files.

original-runtime-path (string)

If flatpak run was run with the --runtime-path option, this key gives the absolute path of the app's original runtime, as mounted at /run/parent/usr inside the container. Available since 1.12.0. If this key is missing, the runtime files are given by runtime-path.

runtime-commit (string)

The commit ID of the runtime that is used. The filename of a deployment of this commit can be found in original-runtime-path if present, or runtime-path otherwise.

runtime-extensions (list of strings)

A list of runtime extensions that are mounted into the running instance. The format for each list item is EXTENSION_ID=COMMIT. If original-app-path is present, the extensions are mounted below /run/parent/usr; otherwise, they are mounted below /usr.

extra-args (string)

Extra arguments that were passed to flatpak run.

sandbox (boolean)

Whether the --sandbox option was passed to flatpak run.

build (boolean)

Whether this instance was created by flatpak build.

session-bus-proxy (boolean)

True if this app cannot access the D-Bus session bus directly (either it goes via a proxy, or it cannot access the session bus at all). Available since 0.8.0.

system-bus-proxy (boolean)

True if this app cannot access the D-Bus system bus directly (either it goes via a proxy, or it cannot access the system bus at all). Available since 0.8.0.

[Session Bus Policy]

If the sockets key is not allowing full access to the D-Bus session bus, then flatpak provides filtered access.

The default policy for the session bus only allows the application to own its own application ID and subnames. For instance if the app is called "org.my.App", it can only own "org.my.App" and "org.my.App.*". Its also only allowed to talk to the bus itself (org.freedesktop.DBus) and the portal APIs APIs (bus names of the form org.freedesktop.portal.*).

Additionally the app is always allowed to reply to messages sent to it, and emit broadcast signals (but these will not reach other sandboxed apps unless they are allowed to talk to your app.

If the [Session Bus Policy] group is present, it provides policy for session bus access.

Each key in this group has the form of a D-Bus bus name or prefix thereof, for example org.gnome.SessionManager or org.freedesktop.portal.*

The possible values for entry are, in increasing order or access:

none

The bus name or names in question is invisible to the application. Available since 0.2.

see

The bus name or names can be enumerated by the application.

Available since 0.2.

talk

The application can send messages/ and receive replies and signals from the bus name or names. Available since 0.2.

own

The application can own the bus name or names (as well as all the above). Available since 0.2.

[System Bus Policy]

If the sockets key is not allowing full access to the D-Bus system bus, then flatpak does not make the system bus available unless the [System Bus Policy] group is present and provides a policy for filtered access. Available since 0.2.

Entries in this group have the same form as for the [Session Bus Policy] group. However, the app has no permissions by default.

[Environment]

The [Environment] group specifies environment variables to set when running the application. Available since 0.3.

Entries in this group have the form VAR=VALUE where VAR is the name of an environment variable to set.

Note that environment variables can also be unset (removed from the environment) by listing them in the unset-environment entry of the [Context] group.

[Extension NAME]

Runtimes and applications can define extension points, which allow optional, additional runtimes to be mounted at a specified location inside the sandbox when they are present on the system. Typical uses for extension points include translations for applications, or debuginfo for sdks. The name of the extension point is specified as part of the group heading. Since 0.11.4, the name may optionally include a tag in the NAME in the name@tag ref syntax if you wish to use different configurations (eg, versions) of the same extension

concurrently. The "tag" is effectively ignored, but is necessary in order to allow the same extension name to be specified more than once.

directory (string)

The relative path at which the extension will be mounted in the sandbox. If the extension point is for an application, the path is relative to /app, otherwise it is relative to /usr. This key is mandatory. Available since 0.1.

version (string)

The branch to use when looking for the extension. If this is not specified, it defaults to the branch of the application or runtime that the extension point is for. Available since 0.4.1.

versions (string)

The branches to use when looking for the extension. If this is not specified, it defaults to the branch of the application or runtime that the extension point is for. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.4.

add-ld-path (string)

A path relative to the extension point directory that will be appended to LD_LIBRARY_PATH. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.3.

merge-dirs (string)

A list of relative paths of directories below the extension point directory that will be merged. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.3.

download-if (string)

A condition that must be true for the extension to be auto-downloaded. As of 1.1.1 this supports multiple conditions separated by semi-colons.

These are the supported conditions:

active-gl-driver

Is true if the name of the active GL driver matches the extension point basename. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.3.

active-gtk-theme

Is true if the name of the current GTK theme (via org.gnome.desktop.interface GSetting) matches the extension point basename. Added 0.10.1.

have-intel-gpu

Is true if the i915 kernel module is loaded. Added 0.10.1.

on-xdg-desktop-*

Is true if the suffix (case-insensitively) is in the XDG_CURRENT_DESKTOP env var. For example on-xdg-desktop-GNOME-classic. Added 1.1.1.

autoprune-unless (string)

A condition that must be false for the extension to be considered unused when pruning. For example, flatpak uninstall --unused uses this information. The only currently recognized value is active-gl-driver, which is true if the name of the active GL driver matches the extension point basename. Available since 0.11.8.

enable-if (string)

A condition that must be true for the extension to be enabled. As of 1.1.1 this supports multiple conditions separated by semi-colons. See download-if for available conditions.

subdirectory-suffix (string)

A suffix that gets appended to the directory name. This is very useful when the extension point naming scheme is "reversed". For example, an extension point for GTK+ themes would be /usr/share/themes/$NAME/gtk-3.0, which could be achieved using subdirectory-suffix=gtk-3.0. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.3.

subdirectories (boolean)

If this key is set to true, then flatpak will look for extensions whose name is a prefix of the extension point name, and mount them at the corresponding name below the subdirectory. Available since 0.1.

no-autodownload (boolean)

Whether to automatically download extensions matching this extension point when updating or installing a 'related' application or runtime. Available since 0.6.7.

locale-subset (boolean)

If set, then the extensions are partially downloaded by default, based on the currently configured locales. This means that the extension contents should be a set of directories with the language code as name. Available since 0.9.13 (and 0.6.6 for any extensions called *.Locale)

autodelete (boolean)

Whether to automatically delete extensions matching this extension point when deleting a 'related' application or runtime. Available since 0.6.7.

collection-id (string)

The ID of the collection that this extension point belongs to. If this is unspecified, it defaults to the collection ID of the application or runtime that the extension point is for. Currently, extension points must be in the same collection as the application or runtime that they are for. Available since 0.99.1.

[ExtensionOf]

This optional group may be present if the runtime is an extension.

ref (string)

The ref of the runtime or application that this extension belongs to. Available since 0.9.1.

runtime (string)

The runtime this extension will be inside of. If it is an app extension, this is the app's runtime; otherwise, this is identical to ref, without the runtime/ prefix. Available since 1.5.0.

priority (integer)

The priority to give this extension when looking for the best match. Default is 0. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.3.

tag (string)

The tag name to use when searching for this extension's mount point in the parent flatpak. Available since 0.11.4.

[Extra Data]

This optional group may be present if the runtime or application uses extra data that gets downloaded separately. The data in this group gets merged into the repository summary, with the xa.extra-data-sources key. If multiple extra data sources are present, their uri, size and checksum keys are grouped together by using the same suffix. If only one extra data source is present, the suffix can be omitted.

NoRuntime (boolean)

Whether to mount the runtime while running the /app/bin/apply_extra script. Defaults to true, i.e. not mounting the runtime. Available since 0.9.1, and backported to the 0.8.x branch in 0.8.4.

uriX (string)

The uri for extra data source X. The only supported uri schemes are http and https. Available since 0.6.13.

sizeX (integer)

The size for extra data source X. Available since 0.6.13.

checksumX (string)

The sha256 sum for extra data source X. Available since 0.6.13.

[Policy SUBSYSTEM]

Subsystems can define their own policies to be placed in a group whose name has this form. Their values are treated as lists, in which items can have their meaning negated by prepending ! to the value. They are not otherwise parsed by Flatpak. Available since 0.6.13.

EXAMPLE

```
[Application]
name=org.gnome.Calculator
runtime=org.gnome.Platform/x86_64/3.20
sdk=org.gnome.Sdk/x86_64/3.20
command=gnome-calculator
[Context]
shared=network;ipc;
```

sockets=x11;wayland;

filesystems=xdg-run/dconf;~/.config/dconf:ro;

[Session Bus Policy]

ca.desrt.dconf=talk

[Environment]

DCONF_USER_CONFIG_DIR=.config/dconf

[Extension org.gnome.Calculator.Locale]

directory=share/runtime/locale

subdirectories=true

[Extension org.gnome.Calculator.Debug]

directory=lib/debug

SEE ALSO

flatpak(1), flatpak-run(1), flatpak-override(1)

NOTES

1. freedesktop.org special directories

   https://www.freedesktop.org/wiki/Software/xdg-user-dirs/

2. freedesktop.org Base Directory Specification

   https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html

flatpak FLATPAK METADATA(5)