

Building Open Container Initiative images based on Freedesktop SDK

Valentin David

Codethink Ltd.

foss-north 2020

March 30th

Table of Contents

- 1 Introduction
- 2 Very quick introduction to BuildStream
- 3 OCI images
- 4 Design your image with BuildStream
- 5 Conclusion

What is Freedesktop SDK?

Runtime of Flatpak applications.

Basic runtime and SDK to build containers for desktop applications.

Flatpak is not related to OCI.

On top of it, two Flatpak runtimes are built: KDE SDK and GNOME SDK.

Freedesktop SDK features

- Release every year
- Bug and security updates for 2 years
- ABI stability
- Automatic scan for CVEs
- Bootstrapped

Architectures

- x86-64 and i686
- aarch64 and armv7
- powerpc64le (experimental)

What does Freedesktop SDK contain?

Basic glibc, bash, coreutils, util-linux, findutils, diffutils, gawk...

Archive tar, cpio, zlib, bzip2, xz, zip...

Security openssl, gnutls, gnupg, nss...

Graphics X.org (x11 and xcb), Wayland, Cairo, GTK+3, SDL2...

Acceleration OpenGL, Vulkan, OpenCL dispatchers, Mesa drivers

Sound Pulseaudio, Alsa w/ pulse plugin

Media gstreamer, mpg123, ffmpeg, vorbis, theora, giflib, libpng...

Programming GCC, LLVM, gperf, flex, bison, ccache...

Build make, autotools, meson, ninja, cmake...

Interpreters Perl, Python, Ruby

Documentation gtk-doc, asciidoc, docbook, man-db...

Fonts DejaVu, Liberation, GNU Free, EmojiOne....

Font rendering Pango, Fontconfig, HarfBuzz

Debugging Strace, GDB

Spelling hunspell, aspell, LibreOffice dictionaries

Web curl, libsoup

What does Freedesktop SDK contain?

Basic glibc, bash, coreutils, util-linux, findutils, diffutils, gawk...

Archive tar, cpio, zlib, bzip2, xz, zip...

Security openssl, gnutls, gnupg, nss...

Graphics X.org (x11 and xcb), Wayland, Cairo, GTK+3, SDL2...

Acceleration OpenGL, Vulkan, OpenGL dispatchers, Mesa drivers

So

M

Programs

B

Interpreters Perl, Python, Ruby

Documentation gtk-doc, asciidoc, docbook, man-db...

Fonts DejaVu, Liberation, GNU Free, EmojiOne....

Font rendering Pango, Fontconfig, HarfBuzz

Debugging Strace, GDB

Spelling hunspell, aspell, LibreOffice dictionaries

Web curl, libsoup

The choice of technologies used by Freedesktop SDK is not an endorsement by the Freedesktop organisation.

Freedesktop SDK is built with BuildStream.

Build and integrate artifacts.

- Separate sandbox per element
- Reproducible build environment
- Cached
- Parallel builds

Why building OCI images?

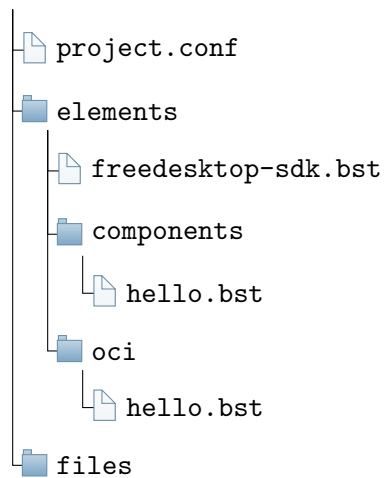
- For our own infrastructure.
- Helping existing continuous integration of applications building for Freedesktop SDK.
- Some applications may have daemon and desktop frontend components.
- Freedesktop SDK is the main project using entirely built with BuildStream.

Table of Contents

- 1 Introduction
- 2 Very quick introduction to BuildStream
- 3 OCI images
- 4 Design your image with BuildStream
- 5 Conclusion

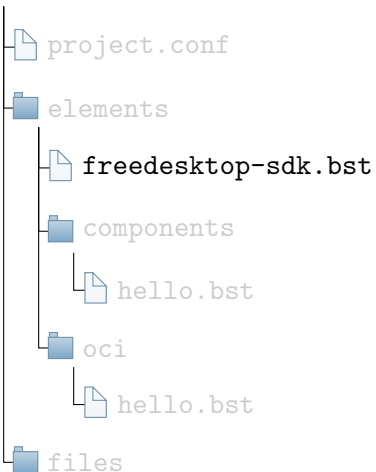
Example: GNU Hello

hello



Example: GNU Hello

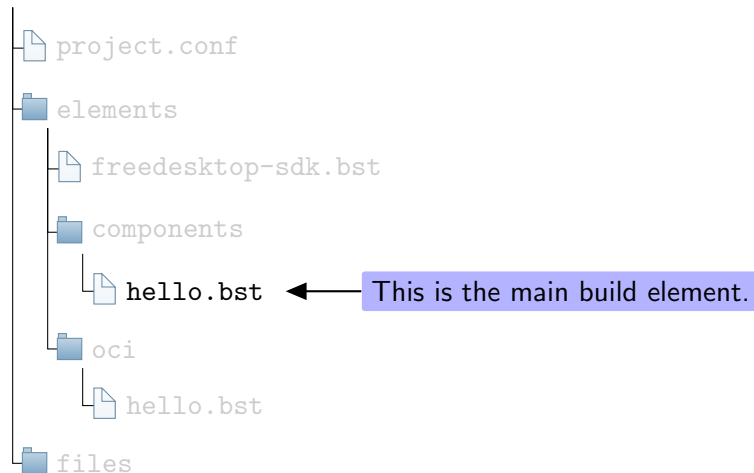
hello



This is a junction element. It refers to an upstream Build-Stream project.

Example: GNU Hello

hello




Anatomy of a BuildStream element

```
kind: autotools
build-depends:
- freedesktop-sdk.bst:public-stacks/buildsystem-autotools.bst
- freedesktop-sdk.bst:components/texinfo.bst
- freedesktop-sdk.bst:components/man.bst
depends:
- freedesktop-sdk.bst:bootstrap-import.bst
variables:
  autogen: |
    ./bootstrap --no-git \
      --gnulib-srcdir=gnulib \
      --skip-po
sources:
- kind: git_tag
  url: https://git.savannah.gnu.org/git/hello.git
  track: master
  ref: v2.10-0-gdc7dc56a00e48fe6f231a58f6537139fe2908fb9
```

The “kind” selects the plugin

Anatomy of a BuildStream element


```
kind: autotools
build-depends:
- freedesktop-sdk.bst:public-stacks/buildsystem-autotools.bst
- freedesktop-sdk.bst:components/texinfo.bst
- freedesktop-sdk.bst:components/help2man.bst
depends:
- freedesktop-sdk.bst:bootstrap-import.bst
variables:
  autogen: |
    ./bootstrap --no-git \
    List of build and runtime dependencies \
    --skip-po
sources:
- kind: git_tag
  url: https://git.savannah.gnu.org/git/hello.git
  track: master
  ref: v2.10-0-gdc7dc56a00e48fe6f231a58f6537139fe2908fb9
```



Anatomy of a BuildStream element

```
kind: autotools
build-depends:
- freedesktop-sdk.bst:public-stacks/buildsystem-autotools.bst
- freedesktop-sdk.bst:components/texinfo.bst
- freedesktop-sdk.bst:bootstrap-import.bst
depends:
- freedesktop-sdk.bst:bootstrap-import.bst
variables:
  autogen: |
    ./bootstrap --no-git \
               --gnulib-srcdir=gnulib \
               --skip-po
sources:
- kind: git_tag
  url: https://git.savannah.gnu.org/git/hello.git
  track: master
  ref: v2.10-0-gdc7dc56a00e48fe6f231a58f6537139fe2908fb9
```

Some customization for the plugin



Anatomy of a BuildStream element

```
kind: autotools
build-depends:
- freedesktop-sdk.bst:public-stacks/buildsystem-autotools.bst
- freedesktop-sdk.bst:components/texinfo.bst
- freedesktop-sdk.bst:components/help2man.bst
depends:
- freedesktop-sdk.bst:bootstrap-import.bst
variables:
  autogen: |
    ./bootstrap --no-  

    --gnu  

    --skip-po
sources:
- kind: git_tag
  url: https://git.savannah.gnu.org/git/hello.git
  track: master
  ref: v2.10-0-gdc7dc56a00e48fe6f231a58f6537139fe2908fb9
```

Description of sources



What to do from there?

```
$ bst build components/hello.bst
```

```
⋮
```

```
$ bst shell components/hello.bst /usr/bin/hello
```

```
⋮
```

```
Hello , world!
```

```
$ bst checkout components/hello.bst hello-rootfs
```

```
⋮
```

```
$
```

Table of Contents

- 1 Introduction
- 2 Very quick introduction to BuildStream
- 3 OCI images**
- 4 Design your image with BuildStream
- 5 Conclusion

Typical build of OCI images

- A base image probably from a distribution
- A Dockerfile
 - Eventually use package manager to add dependencies
 - Sequentially build some other dependencies
 - Build your main project
 - Optionally, extract runtime files to a new image to remove development files (multi-staged)
 - Configure

Hello container image w/ Dockerfile

```
FROM debian AS build
RUN apt-get update
RUN apt-get install -y git autoconf automake autopoint \
gcc make texinfo help2man
RUN mkdir /build
WORKDIR /build
RUN git clone https://git.savannah.gnu.org/git/hello.git
WORKDIR /build/hello
RUN git checkout dc7dc56a00e48fe6f231a58f6537139fe2908fb9
RUN ./bootstrap --skip-po
RUN ./configure --prefix=/usr --disable-dependency-tracking
RUN make -j16
RUN mkdir /install
RUN make -j1 install DESTDIR=/install

FROM debian
COPY --from=build /install /
ENTRYPOINT /usr/bin/hello
```

Hello container image w/ Dockerfile

```
FROM debian AS build
RUN apt-get update
RUN apt-get install -y git autoconf automake autopoint \
gcc make texinfo help2man
RUN mkdir /build
WORKDIR /build
RUN git clone https://git.savannah.gnu.org/git/hello.git
WORKDIR /build/hello
RUN git checkout dc7dc56a00e48fe6f231a58f6537139fe2908fb9
RUN ./bootstrap --skip-po
RUN ./configure --prefix=/usr --disable-dependency-tracking
RUN make -j16
RUN mkdir /install
RUN make install
```

Some commands may download from external sources. They break reproducibility.

```
FROM debian
COPY --from=build /install /
ENTRYPOINT /usr/bin/hello
```

What if you require libraries not shipped in distribution?

- Either make a package for the used distribution
 - Package manager deal with more complex situations: upgrade, uninstall, services, configuration files, user and permission managements
 - Sandbox is optional
 - Rebuild is not automatic
- Build directly as Dockerfile
 - No support for common build systems
 - No build dependencies between Dockerfiles
 - Cached: if one command is modified, all following commands have to be run again
 - Build sandbox has network by default, no reproducibility

OCI images are based on Dockefiles.

1 Dockerfile command = 1 image layer.

Download and storage can reuse common layers.

Table of Contents

- 1 Introduction
- 2 Very quick introduction to BuildStream
- 3 OCI images
- 4 Design your image with BuildStream**
- 5 Conclusion

Graph to layers - Non solution

- Translate every element to a layer.
- Topologically sort all layers.
- Each image is a subsequence.

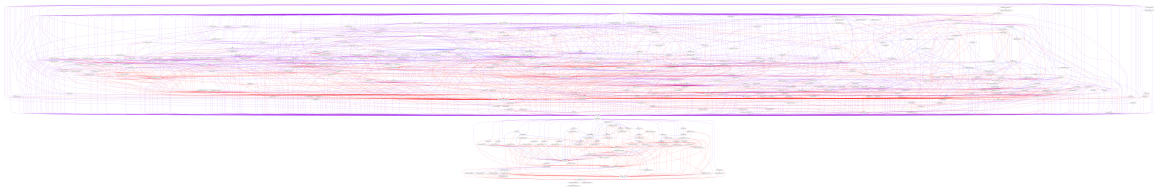
Issues:

- OCI implementations index layers by stack hash (ChainID) rather than layer hash (DiffID), so no subsequence.
- Some implementations or filesystem backends might not scale with hundreds of layers.

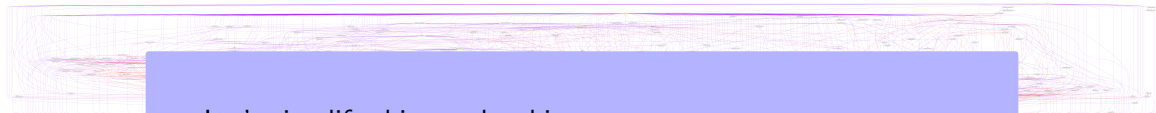
Our approach

- Developer decides of sensible layers.
- One BuildStream element per layer.
- Each layer makes also an image.
- Elements use dependencies to copy layers from other OCI images

Freedesktop SDK full dependency graph

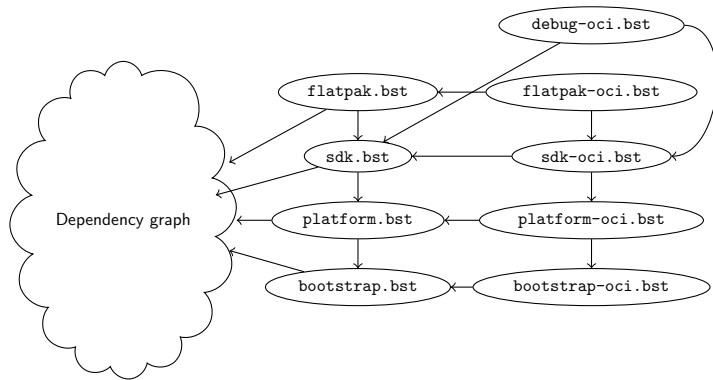


Freedesktop SDK full dependency graph

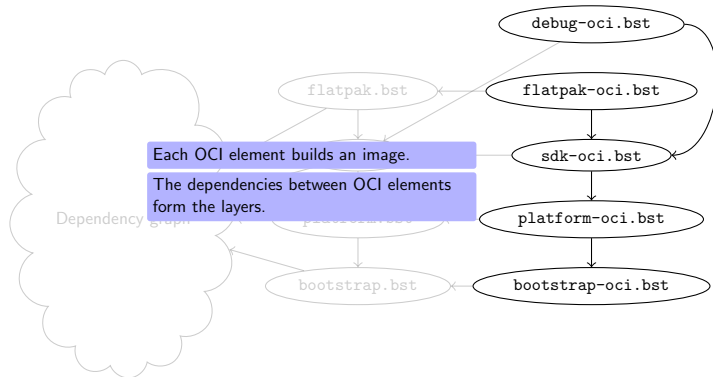


Let's simplify this graph a bit...

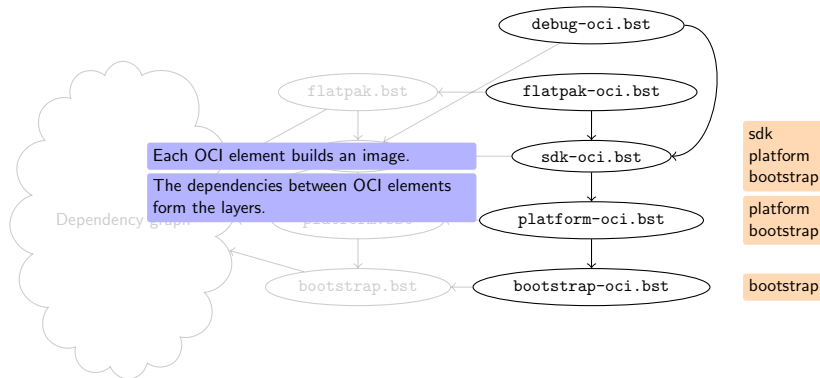
OCI image layering in Freedesktop SDK



OCI image layering in Freedesktop SDK



OCI image layering in Freedesktop SDK

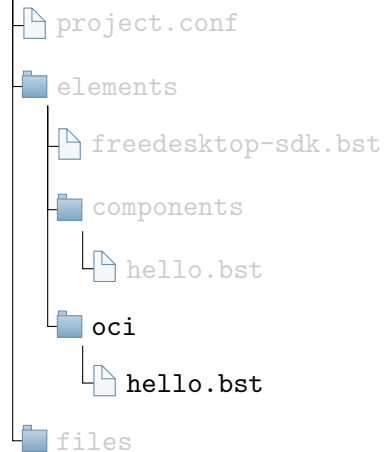


The BuildStream OCI plugin

- OCI or Docker 1.2 (with legacy compatibility)
- Enable/disable layer compression
- Configuration, annotations, history comments.
- Multi-image

Let's go back to our example

hello

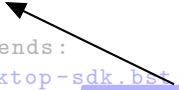


Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
    architecture: amd64
  parent:
    element: oci/bootstrap-oci.bst
  layer:
  - components/hello.bst
  comment: "Import GNU hello"
  config:
    Entrypoint: ["/usr/bin/hello"]
```



Select the OCI plugin

Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  image: freedesktop-sdk:oci/bootstrap-oci
  - We need the base image one which we
    build the layer.
    And the elements to build the current
    layer.
  layer:
  - components/hello.bst
  comment: "Import GNU hello"
  config:
    Entrypoint: ["/usr/bin/hello"]
```




Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
    architecture: amd64
  parent:
    element: oci/bootstrap-oci.bst
  layer:
  - components/hello.bst
  comment: "Import GNU hello"
  config:
    Entrypoint: ["/usr/bin/hello"]
```

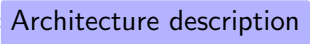


A black arrow points from a blue callout box containing the text "Use OCI specifications" to the "oci" value under the "mode:" key in the "config:" section of the code block.

Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
    architecture: amd64
  parent:
    element: oci/bootstrap-oci.bst
  layer:
  - 
  comment: "Import GNU hello"
  config:
    Entrypoint: ["/usr/bin/hello"]
```

Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
    architecture: amd64
  parent:
    element: oci/bootstrap-oci.bst
  layer:
  - components/hello.bst
  comment: "Import_GNU_hello"
  config:
    Entrypoint: ["/usr/bin/hello"]
```

The base image which we build the layer on



Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
```

The elements included in the layer

```
parent:
  element: oci/bootstrap-oci.bst
layer:
- components/hello.bst
comment: "Import_GNU_hello"
config:
  Entrypoint: ["/usr/bin/hello"]
```



Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
    architecture: amd64
    bootstrap-oci.bst
  layer:
  - components/hello.bst
  comment: "Import_GNU_hello"
  config:
    Entrypoint: ["/usr/bin/hello"]
```

Comment for the history




Layer element

```
kind: oci

build-depends:
- freedesktop-sdk.bst:oci/bootstrap-oci.bst
- components/hello.bst

config:
  mode: oci
  images:
  - os: linux
    architecture: amd64
    parent:
      element: oci/bootstrap-oci.bst
  layer:
    comment: "Import GNU hello"
    config:
      Entrypoint: ["/usr/bin/hello"]
```

Configuration of the image



Building the image

```
$ bst build oci/hello.bst
```

```
⋮
```

```
$ bst checkout oci/hello.bst --tar hello.tar
```

```
⋮
```

```
$ podman load -i hello.tar
```

Table of Contents

- 1 Introduction
- 2 Very quick introduction to BuildStream
- 3 OCI images
- 4 Design your image with BuildStream
- 5 Conclusion

- Fully build OCI images with one tool
- Cached, reproducible, parallel
- Customizable layers to optimize storage and network
- Freedesktop SDK provides a basic SDK with the most common system dependencies

Freedesktop SDK <https://gitlab.com/freedesktop-sdk/freedesktop-sdk>

BuildStream <https://buildstream.build/>

OCI plugin doc <https://buildstream.gitlab.io/bst-external/elements/oci.html>

Docker images <https://hub.docker.com/u/freedesktopsdk>

This work was sponsored by Codethink.