Streamlining application development for Genode with Goa

Johannes Schlatow
<johannes.schlatow@genode-labs.com>
What is Genode?

**Genode OS Framework**

- component-based OS framework
- supports different (micro-)kernels and CPU architectures
- quarterly releases since 2008
What is Genode?

**Genode OS Framework**
- component-based OS framework
- supports different (micro-)kernels and CPU architectures
- quarterly releases since 2008

**Sculpt OS**
- OS showcase based on Genode
- used as daily driver by developers
- targets PC and PinePhone
- releases twice a year

Streamlining application development for Genode with Goa
Why Goa?

Genode’s build system

- useful and efficient for framework development
- cumbersome for app development (framework users)
- steep learning curve $\rightarrow$ off-putting for newcomers and hobbyists
Why Goa?

Genode’s build system
- useful and efficient for framework development
- cumbersome for app development (framework users)
- steep learning curve → off-putting for newcomers and hobbyists

Goa - goal, but reached a little bit sooner
- started in 2019 by Norman Feske as a side project
- moved under the umbrella of Genode Labs in 2023
- has seen quite a few feature additions since then
- streamlines development of individual applications
How are applications made available to other users?

Genode Package Management

- each user manages its own **depot**
- a depot contains different types of **archives**:
  - `api` header files
  - `src` source code
  - `bin` architecture-specific binary for `src` archive
  - `raw` raw data archive, architecture-independent
  - `pkg` runtime information, lists required `raw/src/pkg` archives
  - `index` curated list of a user’s `pkg` archives, shown in Sculpt OS

- archives are placed in corresponding **subdirectories** within depot
How are applications made available to other users?

Genode Package Management

- each user manages its own **depot**
- a depot contains different types of **archives**:  
  - api  header files  
  - src  source code  
  - bin  architecture-specific binary for src archive  
  - raw  raw data archive, architecture-independent  
  - pkg  runtime information, lists required raw/src/pkg archives  
  - index  curated list of a user’s pkg archives, shown in Sculpt OS
- archives are placed in corresponding **subdirectories** within depot

→ a Goa project resembles this structure
Goa workflow

Import  Build  Run  Export  Publish

$ goa [import, build, run, export, publish,...]

Streamlining application development for Genode with Goa
Goa workflow

Import | Build | Run | Export | Publish

$ goa [import, build, run, export, publish, ...]
Importing 3rd-party source code

$ goa import
- requires import file in project directory
- populates src and/or raw directory
Importing 3rd-party source code

$ goa import

- requires *import* file in project directory
- populates *src* and/or *raw* directory

```text
LICENSE := GPLv3
VERSION := 3.3.7
DOWNLOADS := calc.archive

BASE_URL := https://gitlab.com/ubports/development/apps/lomiri-calculator-app/-/archive/
URL(calc) := $(BASE_URL)/v$(VERSION)/lomiri-calculator-app-v$(VERSION).tar.gz
SHA(calc) := 821f045...
DIR(calc) := src
```

*See also:* goa help import
Importing 3rd-party source code

$ goa import

- requires import file in project directory
- populates src and/or raw directory

LICENSE := GPLv3
VERSION := 3.3.7
DOWNLOADS := calc.archive

BASE_URL := https://gitlab.com/ubports/development/apps/lomiri-calculator-app/-/archive/
URL(calc) := $(BASE_URL)/v$VERSION/lomiri-calculator-app-v$VERSION.tar.gz
SHA(calc) := 821f045...
DIR(calc) := src

See also: goa help import
$ goa import

- requires *import* file in project directory
- populates *src* and/or *raw* directory

**LICENSE**  := GPLv3
**VERSION**  := 3.3.7
**DOWNLOADS** := calc.archive

**BASE_URL**  := https://gitlab.com/ubportsdevelopment/apps/lomiri-calculator-app/-/archive/
**URL(calc)** := $(BASE_URL)/v$$(VERSION)/lomiri-calculator-app-v$$(VERSION).tar.gz
**SHA(calc)** := 821f045...
**DIR(calc)** := src

See also: goa help import
Building software using commodity build systems

$ goa build

- compiles code in src/ directory
- requires used_apis and artifacts file

See also: goa help [artifacts|build-systems]
Building software using commodity build systems

$ goa build
- compiles code in src/ directory
- requires used_apis and artifacts file

**supported build systems**
- plain GNU Make
- CMake
- autoconf
- qmake
- cargo (Rust)
- ...add your favorite build system

See also: goa help [artifacts|build-systems]
Test-Running a scenario

$ goa run

required files
- pkg/«name»/archives
- pkg/«name»/runtime

See also: goa help [runtime|targets]
Test-Running a scenario

$ goa run

required files
- pkg/«name»/archives
- pkg/«name»/runtime

supported targets (via --target)
- linux (host system, default)
- sculpt (file sync via HTTP PUT)
- ...add your own target

See also: goa help [runtime|targets]
Demo
Exporting and Publishing

$ goa publish *(implies export)*

- export: assembles depot archives
- publish: creates signed tar archives

See also: goa help add-depot-user
$ goa publish \textit{(implies export)}

- export: assembles depot archives
- publish: creates signed tar archives

\textbf{required files}

- \textit{version}
- \textit{LICENSE}
- \textit{pkg/\textbackslash name/\textbackslash README}

\textbf{See also:} goa help add-depot-user
Demo
Getting Started

User stories / Tutorials

https://genodians.org

Streamlining application development for Genode with Goa
User stories / Tutorials

Thank you!

$ git clone https://github.com/genodelabs/goa

https://genodians.org
References

Goa source code
https://github.com/genodelabs/goa

Goa project repositories
https://github.com/nfeske/goa-playground
https://github.com/{jschlatow,cnuke,ssumpf}/goa-projects

Genodians.org community blog
https://genodians.org

Genode OS Framework
https://genode.org

Sculpt OS download and manual
https://genode.org/download/sculpt

Streamlining application development for Genode with Goa