Alire: Ada Has a Package Manager

Alejandro Mosteo, professor at Centro Universitario de la Defensa de Zaragoza

Fabien Chouteau, embedded software engineer at AdaCore

Pierre-Marie de Rodat, software engineer at AdaCore
OUTLINE

- INTRODUCTION & MOTIVATION
  - Project Status
  - Design Basics

- DEMO 1: QUICKSTARTING A PROJECT

- DEMO 2: CONTRIBUTING A PROJECT TO ALIRE
MOTIVATION

- EXPECTATIONS
- REAL ADVANTAGES
MOTIVATION

● EXPECTATIONS ➔ Of the younger generations
➔ From other contexts

● REAL ADVANTAGES

$ sudo apt install libgtkada16.1.0-dev     # Linux systems

dependencies {
    compile 'com.example.android:lib-magic:1.3'
}     // Java’s Gradle
Ada world

Out there
MOTIVATION

● EXPECTATIONS

➔ Code reuse
➔ Simplicity
➔ Library awareness
➔ Portability
➔ Reproducibility
➔ Security

● REAL ADVANTAGES
PROJECT STATUS

- TIMELINE

- DEMO 0: testing some Ada software
PROJECT STATUS

● TIMELINE

● DEMO 0: testing some Ada software

→ Feb 2016: first repo & discussions
→ Feb 2018: Ada-Europe 2018
→ Jun 2018: AdaCore ROS2 stay
→ Apr 2019: AdaCore support
→ Aug 2019: website goes live
→ Nov 2019: internal beta
→ Today! 🎉🎉🎉 Public beta
PROJECT STATUS

Right now

● Starting to come together
  ○ Many rough spots
● Known missing features

2020 projections

● Polish existing workflows
  ○ Publishing
● Missing features
  ○ Pinned versions
● First stable binary release
  ○ Linux/MacOS/Windows
  ○ Help needed!
PROJECT STATUS

- TIMELINE

- DEMO 0: testing some Ada software
$ alr get --build eagle_lander
$ alr get --build hangman

$ cd hangman_1.0.0_a5790492
$ alr run

****** WELCOME TO HANGMAN ******
By: Jon Hollan, Mark Hoffman, & Brandon Ball

$ alr run --list

Project hangman builds these executables:
   hangmain (found at /tmp/demo/hangman_1.0.0_a5790492/bin/hangmain)

$ alr list

ada_lua         An Ada binding for Lua
adacurses       Wrapper on different packagings of NcursesAda
adayaml         Experimental YAML 1.3 implementation in Ada

$ alr search x

<table>
<thead>
<tr>
<th>NAME</th>
<th>VERSION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>rxada</td>
<td>0.1.0</td>
<td>RxAda port of the Rx framework</td>
</tr>
<tr>
<td>xml_ez_out</td>
<td>1.6.0</td>
<td>Creation of XML-formatted output from Ada programs</td>
</tr>
<tr>
<td>xstrings</td>
<td>1.0.0</td>
<td>Renaming of gnatcoll.strings w/o other dependencies</td>
</tr>
</tbody>
</table>
DESIGN BASICS

- Repositories
- Classification
- Underlying data types
- On-disk structure
REPOSITORIES @ https://github.com/alire-project
### alire-index vs alire vs alr

<table>
<thead>
<tr>
<th>alire-index</th>
<th>alire</th>
<th>alr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community index</td>
<td>Library project</td>
<td>Command-line tool</td>
</tr>
<tr>
<td>Describes “crates”</td>
<td>Common functionality</td>
<td>Interactive part</td>
</tr>
<tr>
<td>TOML files</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://.../alire-project/alire-index

https://.../alire-project/alire
Types of Package Managers

SYSTEM vs SANDBOX

PLATFORM vs LANGUAGE

BINARIES vs SOURCES

OFFICIAL vs COMMUNITY
Ada package
GPR project file
Not all are a library
Too general bundle
Crate
version 1.2.3-prerelease+anything

major . minor . patch

- Major changes **break** compatibility
- Minor changes **add** functionality
- Patch changes **fix** bugs

Minor/Patch upgrades “should” be safe.

- Meaningful only when offering an API
- Can assimilate other versioning methods

Calendar versioning: 20180501.0.0
$ alr show --solve pygamer_simulator

pygamer_simulator=0.1.0: SDL simulator of the AdaFruit PyGamer

Dependencies (direct):
  hal^0.1.0
  sdlada^2.3.1

Dependencies (solution):
  hal=0.1.0
  libsdl2=2.0.0
  libsdl2_image=2.0.0
  libsdl2_ttf=2.0.0
  sdlada=2.3.1
$ alr get hello
$ tree hello*

```
hello_1.0.1_dcc36a2f/
├── alire
│   ├── cache
│   │   └── dependencies
│   │       └── libhello_1.0.0_7bc7dcfd
│   │           ├── libhello.gpr
│   │           └── LICENSE
│   │               └── README.md
│   └── hello.toml
├── hello.gpr
├── LICENSE
├── README.md
└── src
    └── hello.adb
```
Contributing to the index (1/2)

- Clone the index repository
  [https://github.com/alire-project/alire-index/](https://github.com/alire-project/alire-index/)

- Let alr know about it
  `$ git index --add`

- Describe the crate in a text file: `index/XX/XXX.toml`
Contributing to the index (2/2)

- Crate format spec:  
  https://github.com/alire-project/alire/blob/master/doc/catalog-format-spec.rst

- TOML syntax, contains:
  - General information about the crate: description, website, authors, ...
  - Release-specific information: where to get sources, dependencies, ...

- Demo time!
ALR
ALIRE: Ada Library REpository.

A catalog of ready-to-use Ada libraries plus a command-line tool (\texttt{alr}) to obtain, compile, and incorporate them into your own projects. It aims to fulfill a similar role to Rust's \texttt{cargo} or OCaml's \texttt{opam}.

Caveat emptor
Documentation at this time is minimal. Expect further efforts in this direction until this warning is removed.

TL;DR
Available for Debian stable / Ubuntu >=17.10 / macOS

\texttt{alr} is undergoing frequent changes in preparation for a first publicly announced beta. Hence, the current recommendation is to run the latest \texttt{master} branch version, and double-check in case of problems that no new PRs have been merged since your last compiled version.

To install run the following as user in a terminal, or see below for more details:

\begin{enumerate}
\item If running on macOS, \texttt{export OS=macOS}
\item \texttt{git clone --recursive https://github.com/alire-project/alire.git}
\item \texttt{cd alire}
\item \texttt{cppbuild -j0 -p -P alir.env}
\end{enumerate}

After a successful build, you will have a \texttt{bin/alr} executable that you can put in your path if desired.

To see available crates per platform/compiler, see the \texttt{alire-crates-ci} companion repository.

Design principles
• Ada_taml
• Ada_voxel_space_demo
• Adayaml
• Ado
• Ado_postgresql
• Ado_sqlite
• Agpl
• Ajunitgen
• Apdf
• Asis
• Aws

• Bar_codes
• Basalt

• C_strings
• Cortex_m

• Eagle_lander
• Eloda

• Geste
• Geste_examples
• Gid
• Glut
• Gnat
• Gnucci
Geste
GEneric Sprite and Tile Engine

Website
Author: Fabien Chouteau
Maintainer: fabien.chouteau@gmail.com
License: BSD 3-Clauses
Version: 1.0
Dependencies: No dependency.
Tags: game, nsad, rendering, sprite
Badge: ![giste](https://giste.org)

GESTE is a sprite and tile 2D render engine designed to run on micro-controllers low performance systems. GESTE also provides a basic math and physic engine using fixed point arithmetic.

Creating maps
GESTE is developed in parallel with tiled-code-gen, a tool that generates code from the Tiled Map Editor.

Examples
The crate giste_example contains 3 examples of different game genre:
  - platformer
  - RPG
  - racing

Design
Layers
Virtapu
/crates/virtapu.html
A virtual Audio Processing Unit to simulate 8-bit era game sounds
fabien.chouteau@gmail.com MIT ...

Geste_examples
/crates/geste_examples.html
Example for the GESTE game engine fabien.chouteau@gmail.com BSD 3-Clauses ...

Hangman
/crates/hangman.html
Hangman game for the console alejandro@mosteo.com ...

Eagle_lander
/crates/eagle_lander.html
Apollo 11 lunar lander simulator fabien.chouteau@gmail.com GPL 3.0 ...

Geste
/crates/geste.html
GEneric Sprite and Tile Engine fabien.chouteau@gmail.com BSD 3-Clauses ...

Pygamer_bsp
/crates/pygamer_bsp.html