

ADA USER JOURNAL

Volume 44
Number 2
June 2023

Contents

	<i>Page</i>
Editorial Policy for Ada User Journal	94
Editorial	95
Quarterly News Digest	96
Conference Calendar	117
Forthcoming Events	123
Special Contribution	
F. Singhoff et al.	
<i>“ADEPT 2022 Workshop: A Summary of Strengths and Weaknesses of the AADL Ecosystem”</i>	125
Articles from the AEiC 2023 Work-in-Progress Session	
J. Ferreira, A. Oliveira, A. Souto, J. Cecílio	
<i>“Software-Based Security Approach for Networked Embedded Devices”</i>	129
G. Costa, J. Cecílio, A. Casimiro	
<i>“Cooperative Autonomous Driving in Simulation”</i>	133
F. Lucchetti, M. Voelp	
<i>“Achieving Crash Fault Tolerance in Autonomous Vehicle Autopilot Software Stacks Through Safety-Critical Module Rejuvenation”</i>	137
D. de Niz, L. Wrage	
<i>“Symbolic Refinement for CPS”</i>	141
F. L. Malaquias, G. Giantamidis, S. Basagiannis, S. F. Rollini, I. Amundson	
<i>“Towards a Methodology to Design Provably Secure Cyber-Physical Systems”</i>	146
L. Creuse, M. Eyraud, V. Garèse	
<i>“Automatic Test Value Generation for Ada”</i>	152
J. Hughes	
<i>“Mechanization of the Ravenscar Profile in Coq”</i>	157
I. Rodríguez-Ferrandez, A. Jover Alvarez, M. M. Trompouki, L. Kosmidis, F. J. Cazorla	
<i>“Worst Case Execution Time and Power Estimation of Multicore and GPU Software: A Pedestrian Detection Use Case”</i>	161
Ada-Europe Associate Members (National Ada Organizations)	168
Ada-Europe Sponsors	Inside Back Cover

Quarterly News Digest

Alejandro R. Mosteo

Centro Universitario de la Defensa de Zaragoza, 50090, Zaragoza, Spain; Instituto de Investigación en Ingeniería de Aragón, Mariano Esquillor s/n, 50018, Zaragoza, Spain; email: amosteo@unizar.es

Contents

Preface by the News Editor	96
Ada-related Events	96
Ada and Education	98
Ada-related Resources	98
Ada-related Tools	99
Ada and Operating Systems	104
References to Publications	106
Ada and Other Languages	107
Ada Practice	110
Ada in Jest	116

[Messages without subject/newsgroups are replies from the same thread. Messages may have been edited for minor proofreading fixes. Quotations are trimmed where deemed too broad. Sender's signatures are omitted as a general rule. —arm]

Preface by the News Editor

Dear Reader,

I place the single focus of this issue on the momentous event all of us Ada enthusiasts have long been anticipating: Ada 2022 is officially here [1][2].

Sincerely,

Alejandro R. Mosteo.

[1] "Ada 2022 LRM by Springer", in References to Publications.

[2] "Ada 2022 at Last!", in Ada Practice.

Ada-related Events

Ada-Belgium Spring 2023 Event

From: Dirk Craeynest

<dirk@orka.cs.kuleuven.be>

Subject: Ada-Belgium Spring 2023 Event, Sun 28 May 2023

Date: Thu, 11 May 2023 14:31:54 -0000

Newsgroups: comp.lang.ada,

fr.comp.lang.ada,be.comp.programming

Ada-Belgium Spring 2023 Event

Sunday, May 28, 2023, 12:00-19:00

Leuven, Belgium

including at 15:00
2023 Ada-Belgium General Assembly
and at 16:00

Ada Round-Table Discussion

<http://www.cs.kuleuven.be/~dirk/ada-belgium/events/23/230528-abga.html>

*** Announcement

The next Ada-Belgium event will take place on Sunday, May 28, 2023 in Leuven.

After an interruption of 3 years due to the COVID-19 pandemic, and for the 13th time, Ada-Belgium organizes their "Spring Event", which starts at noon, runs until 7pm, and includes an informal lunch, the 30th General Assembly of the organization, and a round-table discussion on Ada-related topics the participants would like to bring up.

*** Schedule

- * 12:00 welcome and getting started (please be there!)
- * 12:15 informal lunch
- * 15:00 Ada-Belgium General Assembly
- * 16:00 Ada round-table + informal discussions
- * 19:00 end

*** Participation

Everyone interested (members and non-members alike) is welcome at any or all parts of this event.

For practical reasons registration is required. If you would like to attend, please send an email before Thursday, May 25, 18:00, to Dirk Craeynest <Dirk.Craeynest@cs.kuleuven.be> with the subject "Ada-Belgium Spring 2023 Event", so you can get precise directions to the place of the meeting. Even if you already responded to the preliminary announcement, please reconfirm your participation ASAP.

If you are interested to join Ada-Belgium, please register by filling out the 2023 membership application form[1] and by paying the appropriate fee before the General Assembly. After payment you will receive a receipt from our treasurer and you are considered a member of the organization for the year 2023 with all

member benefits[2]. Early enrollment ensures you receive the full Ada-Belgium membership benefits (including the Ada-Europe indirect membership benefits package).

As mentioned at earlier occasions, we have a limited stock of documentation sets and Ada related CD-ROMs that were distributed at previous events, as well as some back issues of the Ada User Journal[3]. These will be available on a first-come first-serve basis at the General Assembly for current and new members. (Please indicate in the above-mentioned registration e-mail that you're interested, so we can bring enough copies.)

[1] <http://www.cs.kuleuven.be/~dirk/ada-belgium/forms/member-form23.html>

[2] <http://www.cs.kuleuven.be/~dirk/ada-belgium/member-benefit.html>

[3] <http://www.ada-europe.org/auj/home/>

*** Informal lunch

The organization will provide food and beverage to all Ada-Belgium members. Non-members who want to participate at the lunch are also welcome: they can choose to join the organization or pay the sum of 20 Euros per person to the Treasurer of the organization.

*** General Assembly

All Ada-Belgium members have a vote at the General Assembly, can add items to the agenda, and can be a candidate for a position on the Board[4]. See the separate official convocation[5] for all details.

[4] <http://www.cs.kuleuven.be/~dirk/ada-belgium/board/>

[5] <http://www.cs.kuleuven.be/~dirk/ada-belgium/events/23/230528-abga-conv.html>

*** Ada Round-Table Discussion

As in recent years, we plan to keep the technical part of the Spring event informal as well. We will have a round-table discussion on Ada-related topics the participants would like to bring up. We invite everyone to briefly mention how they are using Ada in their work or non-work environment, and/or what kind of Ada-related activities they would like to embark on. We hope this might spark some concrete ideas for new activities and collaborations.

*** Directions

To permit this more interactive and social format, the event takes place at private premises in Leuven. As instructed above, please inform us by e-mail if you would like to attend, and we'll provide you precise directions to the place of the meeting. Obviously, the number of participants we can accommodate is not unlimited, so don't delay...

Looking forward to meet many of you!

Dirk Craeynest

President Ada-Belgium

Dirk.Craeynest@cs.kuleuven.be

Acknowledgements

We would like to thank our sponsors for their continued support of our activities: AdaCore, and KU Leuven (University of Leuven).

If you would also like to support Ada-Belgium, find out about the extra Ada-Belgium sponsorship benefits:

<http://www.cs.kuleuven.be/~dirk/ada-belgium/member-benefit.html#sponsor>

(V20230511.1)

AEiC 2023 Final Call

From: Dirk Craeynest

<dirk@orka.cs.kuleuven.be>

Subject: Press Release - AEiC 2023, Ada-Europe Reliable Softw. Technol.

Date: Fri, 9 Jun 2023 10:56:57 -0000

Newsgroups: comp.lang.ada,

fr.comp.lang.ada,comp.lang.misc

FINAL Call for Participation

*** UPDATED Program Summary ***

27th Ada-Europe International
Conference on

Reliable Software Technologies (AEiC
2023)

13-16 June 2023, Lisbon, Portugal

www.ada-europe.org/conference2023

Organized by Ada-Europe

in cooperation with ACM SIGAda,
SIGBED, SIGPLAN,

the Ada Resource Association (ARA),
and the University of Lisbon

#AEiC2023 #AdaEurope
#AdaProgramming

*** Final Program available on the
conference web site ***

*** Add tutorials and/or a workshop to
your conference registration ***

www.ada-europe.org/conference2023/tutorials.html

*** Welcome Event on Tuesday evening

Press release:

27th Ada-Europe Int'l Conference on
Reliable Software Technologies

International experts meet in Lisbon

Lisbon, Portugal (9 June 2023) - Ada-Europe together with the University of Lisbon organizes from 13 to 16 June 2023 the 27th Ada-Europe International Conference on Reliable Software Technologies (AEiC 2023), in cooperation with the Ada Resource Association (ARA), and with ACM's Special Interest Groups on Ada (SIGAda), on Embedded Systems (SIGBED) and on Programming Languages (SIGPLAN).

The Ada-Europe series of conferences is an established international forum for providers, practitioners and researchers in reliable software technologies. These events highlight the increased relevance of Ada in general and in safety- and security-critical systems in particular, and provide a unique opportunity for interaction and collaboration between academics and industrial practitioners.

This year's conference offers 4 tutorials, a keynote and a panel discussion, a technical program of 6 sessions with peer-reviewed papers, industrial and work-in-progress presentations, posters, social events, and 2 workshops. Presentations are given by authors from 15 countries.

Six tutorials are scheduled on Tuesday, targeting different audiences:

- "The HAC Ada Compiler",
- "Controlling I/O Devices with Ada and the Linux Simple I/O Library",
- "Everything you Always Wanted to Know about Characters and Strings",
- "Introduction to the Development of Safety Critical Software",
- "Rust Fundamentals",
- "Concurrency and Parallelism in Rust".

On Wednesday and Thursday, the networking area features WiP posters, as well as an Ada-Europe booth.

Eminent speakers have been invited on each of the core conference days:

- on Wednesday, a keynote talk by Alcides Fonseca, from LASIGE, University of Lisbon Faculty of Sciences, who will talk about "Applications of liquid types for more reliable software";
- on Thursday, a panel on "Promises and Challenges of AI-enabled Software Development Tools for Safety-Critical

Applications" with Douglas Schmidt (Vanderbilt University, USA), Jochen Quante (Robert Bosch GmbH, Germany), and Jon Pérez Cerrolaza (IKERLAN, Spain).

The technical program on Wednesday and Thursday includes 6 journal-track refereed technical papers, 7 industrial, and 15 work-in-progress presentations, in sessions on: Verification and Validation 1, Advanced Systems, Reliability and Performance, Verification and Validation 2, Reliable Programming, Real-Time Systems.

On Friday the conference hosts for the 8th year the workshop on "Challenges and New Approaches for Dependable and Cyber-Physical Systems Engineering" (DeCPS 2023), as well as the workshop "AADL by its Practitioners (ADEPT)".

Peer-reviewed papers have been submitted to a special issue of the Journal of Systems Architecture and are heading towards final acceptance as open-access publications. Industrial and work-in-progress presentations, together with tutorial abstracts, and workshop papers, will appear in issues of the Ada User Journal, the quarterly magazine of Ada-Europe.

The social program includes on Tuesday evening a Welcome Reception in the gardens of the National Museum of Science & Natural History, and on Wednesday evening the Conference Banquet in the "Casa do Alentejo" restaurant, an old palace in downtown Lisbon with several exquisite rooms, that served as a casino in the 20th century.

The Best Presentation Award will be offered during the Closing session.

The full program is available on the conference web site.

Online registration is still possible.

Latest updates:

The 16-page "Final Program" is available at www.ada-europe.org/conference2023/media/AEiC_2023_Final_Program.pdf

Check out the tutorials in the PDF program, or in the schedule at

www.ada-europe.org/conference2023/tutorials.html.

Registration is done on-line. For all details, go to

www.ada-europe.org/conference2023/registration.html.

A printed Conference Booklet with abstracts of all technical papers and industrial presentations will be included in every conference handout, and will be available on the conference web site.

AEiC 2023 is sponsored by Ada-Europe (www.ada-europe.org), AdaCore (www.adacore.com), and GMV (www.gmv.com).

Help promote the conference by advertising it.

Recommended Twitter hashtags:
#AEiC2023 #AdaEurope
#AdaProgramming.

Our apologies if you receive multiple copies of this announcement.

Please circulate widely.

Dirk Craeynest, AEiC 2023 Publicity Chair

Dirk.Craeynest@cs.kuleuven.be

* 27th Ada-Europe Int. Conf. Reliable Software Technologies (AEiC 2023)

* June 13-16, 2023, Lisbon, Portugal, www.ada-europe.org/conference2023

(V7.1)

Ada Monthly Meetup 2023

[see also "Ada Monthly Meeting Proposal" in this AUJ issue, pp.114-115 —arm]

From: *Fernando Oleo Blanco*
<irvise_ml@irvise.xyz>

Subject: *Ada Monthly Meetup 2023*

Date: *Wed, 31 May 2023 14:27:07 +0200*
Newsgroups: *comp.lang.ada*

Hi all,

This message contains the final time of the meeting, connection details and other info.

The (first!) Ada Monthly Meetup will take place this Saturday 3rd of June at 13:00 UTC Time. That corresponds to 15:00 CET (Central European Time: Madrid, Paris, Berlin, Rome...).

The meetup will take place over at Jitsi, a conferencing software that runs on any modern browser. The link is <https://meet.jit.si/2023AdaMonthlyMeetupJune> The room name is "2023AdaMonthlyMeetupJune" and in case it asks for a password, it will be set to "first". I do not want to set up a password, but in case it is needed, it will be the one above without the quotes. The room name is generally not needed as the link should take you directly there, but I want to write it down just in case someone needs it.

Talks:

No one proposed any topics, but that is fine as **this first meeting will not be recorded.** I will record it for internal testing and to see how it works, but it will not be published.

Having no talks will allow us, the community, to discuss any technical issues and comments that may help improve the experience of the monthly meetup. However, I will give a short introduction and share my ideas at the beginning :) Someone could also propose a topic for the next meetup too.

If I forgot something, please, point it out so that any issues can get patched out.

Best regards,

Fer

P.S: I, Fer, will post this over at the C.L.A and Ada-Lang.io. Feel free to repost this to Reddit, Gitter/Matrix, Telegram or any other channels! The more people know about this, the better (I hope).

P.P.S: this is for C.L.A only. The main thread was named "Ada Monthly Meetup Proposal". However, as this is no longer a proposal, but the actual thing, I am creating a new thread. For more information, please, refer to the aforementioned thread!

From: *Dirk Craeynest*

<dirk@orka.cs.kuleuven.be>

Date: *Thu, 1 Jun 2023 09:58:34 -0000*

Fernando Oleo Blanco

<irvise_ml@irvise.xyz> wrote:

>The more people know about this, the better (I hope).

Reposted to all Ada-Belgium members.
HTH

Ada and Education

New Project: Alice

From: *francesc...@gmail.com*

<francesc.rocher@gmail.com>

Subject: *New project: Alice*

Date: *Tue, 16 May 2023 11:22:12 -0700*

Newsgroups: *comp.lang.ada*

After months of dedicated work, I'm thrilled to introduce my project: Alice!

Alice, "Adventures for Learning and Inspiring Coding Excellence", is a collaborative Ada framework that allows programmers to enhance and share their solutions to various problem sources (e.g. Project Euler, CodinGame and Advent of Code), fostering collaboration, learning and creativity.

While it's currently in the proof of concept stage, and only Project Euler is supported, I believe it holds immense potential.

The wiki pages, <https://github.com/alice-adventures/Alice/wiki>, offer a glimpse into Alice's concept, participation opportunities, and development ideas.

I warmly invite all members of the Ada community, as well as beginners and

students exploring Ada, to read across the wiki pages and share your valuable feedback. Your insights and input will be instrumental in shaping Alice's future. Together, let's unlock the possibilities and make a significant impact.

Stay tuned for the upcoming public release, as we embark on this exciting journey together!

"Ada Computer Science" at RaspberryPi.org

From: *Ingo M.* <it.marks.info@gmail.com>

Subject: *"Ada Computer Science" at raspberrypi.org*

Date: *Sun, 28 May 2023 07:06:51 -0700*

Newsgroups: *comp.lang.ada*

The Raspberry Pi Foundation announces an "Ada Computer Science" project which has nothing to do with the Ada programming language.

<https://www.raspberrypi.org/blog/ada-computer-science/>

"We are excited to launch Ada Computer Science, the new online learning platform for teachers, students, and anyone interested in learning about computer science."

So far the focus is set on the current ChatGPT hype, and code examples in Python, Java, VB, and C#. It could be a good opportunity to promote the Ada language by providing similar courses. Otherwise there could be a risk that newcomers associate Ada with this project rather than the language.

From: *Dirk Craeynest*

<dirk@orka.cs.kuleuven.be>

Date: *Mon, 29 May 2023 07:36:04 -0000*

FWIW, I just posted the following comment on that page:

"Will you also be using the Ada programming language, a modern language with a long track record of successful projects and ideally suited to develop reliable and trustworthy software?"

It is currently marked as "This comment is awaiting moderation."

Ada-related Resources

[Delta counts are from February 12th to July 28th. —arm]

Ada on Social Media

From: *Alejandro R. Mosteo*

<amosteo@unizar.es>

Subject: *Ada on Social Media*

Date: *28 Jul 2023 14:35 CET*

To: *Ada User Journal readership*

Ada groups on various social media:

- Reddit: 8_371 (+22) members [1]

- LinkedIn: 3_448 (+12) members [2]
- Stack Overflow: 2_345 (+22) questions [3]
- Gitter: 230 (+11) people [4]
- Telegram: 159 (-1) users [5]
- Ada-lang.io: 133 (+26) users [6]
- Libera.Chat: 73 (-1) concurrent users [7]
- Twitter: Discontinued due to free API being removed.
- [1] <http://www.reddit.com/r/ada/>
- [2] <https://www.linkedin.com/groups/114211/>
- [3] <http://stackoverflow.com/questions/tagged/ada>
- [4] https://app.gitter.im/#/room/#ada-lang_Lobby:gitter.im
- [5] https://t.me/ada_lang
- [6] <https://forum.ada-lang.io/u>
- [7] <https://netsplit.de/channels/details.php?room=%23ada&net=Libera.Chat>

Repositories of Open Source Software

*From: Alejandro R. Mosteo
<amosteo@unizar.es>
Subject: Repositories of Open Source software
Date: 28 Jul 2023 14:41 CET
To: Ada User Journal readership*

- Rosetta Code: 941 (+15) examples [1]
41 (+1) developers [2]
- GitHub: 987* (+224) developers [3]
- Alire: 363 (+26) crates [4]
- Sourceforge: 243 (+3) projects [5]
- Open Hub: 214 (=) projects [6]
- Codelabs: 57 (+3) repositories [7]
- Bitbucket: 31 (=) repositories [8]

*This number is an unreliable lower bound due to GitHub search limitations.

- [1] <http://rosettacode.org/wiki/Category:Ada>
- [2] http://rosettacode.org/wiki/Category:Ada_User
- [3] <https://github.com/search?q=language%3AAda&type=Users>
- [4] <https://alire.ada.dev/crates.html>
- [5] <https://sourceforge.net/directory/language:ada/>
- [6] <https://www.openhub.net/tags?names=ada>
- [7] https://git.codelabs.ch/?a=project_index
- [8] <https://bitbucket.org/repo/all?name=ada&language=ada>

Language Popularity Rankings

*From: Alejandro R. Mosteo
<amosteo@unizar.es>
Subject: Ada in language popularity rankings
Date: 28 Jul 2023 14:53 CET
To: Ada User Journal readership*

[Positive ranking changes mean to go up in the ranking. —arm]

- TIOBE Index: 23 (+5) 0.77% (+0.35%) [1]
- PYPL Index: 16 (+3) 1.06% (+0.23%) [2]
- Stack Overflow Survey 42 (new) 0.77% (new) [3]
- IEEE Spectrum (general): 35 (=) Score: 1.16 [4]
- IEEE Spectrum (jobs): 33 (=) Score: 0.79 [4]
- IEEE Spectrum (trending): 32 (=) Score: 3.95 [4]

- [1] <https://www.tiobe.com/tiobe-index/>
- [2] <http://pypl.github.io/PYPL.html>
- [3] <https://survey.stackoverflow.co/2023/>
- [4] <https://spectrum.ieee.org/top-programming-languages/>

GnatStudio Cookbook

*From: Rod Kay <rodakay5@gmail.com>
Subject: [Ann] GnatStudio Cookbook
Date: Wed, 28 Jun 2023 06:53:49 +1000
Newsgroups: comp.lang.ada*

Hello again all,

In the hope it might help other people building or OS packaging GnatStudio, I've prepared a 'cookbook' of sorts.

It provides build instructions for the entire GnatStudio project stack, beginning with gprbuild-bootstrap and culminating in the build of gnatstudio. The individual 'recipes' take the form of pacman PKGBUILDS with tarballs and patches.

Here is the link ...

<https://github.com/charlie5/archlinux-gnatstudio-support/tree/main/gnatstudio-cookbook>

Ada-related Tools

AdaStudio 2023 Release 03/04/2023 Free Edition

*From: Leonid Dulman
<leonid.dulman@gmail.com>
Subject: Announce: AdaStudio-2023 release 03/04/2023 free edition
Date: Sun, 2 Apr 2023 21:12:06 -0700
Newsgroups: comp.lang.ada*

It based on Qt-6.5.0-everywhere opensource (expanded with modules from Qt-5.15: qtgamepad, qtx11extras,qtwinextras),VTK-9.2.0,FFMPEG-5.2.1,OpenCV-4.7.0,SDL2-2.24.0,QtAV-1.13 MDK-SDK(wang-bin)

Qt6ada version 6.5.0 open source and qt6base.dll ,qt6ext.dll (win64),libqt6base.so,libqt6txt.so(x86-64) built with Microsoft Visual Studio 2023 x64 Windows, GCC amd64 in Linux.

Package tested with GNAT gpl 2020 Ada compiler in Windows 64bit , Linux amd64 Debian 11.2

AdaStudio-2023 includes next modules : qt6ada,vtkada,qt6mdkada,qt6cvada(face recognition, face detection,face identification,objects detectection,QRcode detector,BARcode detection and others) and voice recognizer.

Qt6Ada is built under GNU LGPLv3 license <https://www.gnu.org/licenses/lgpl-3.0.html>.

Qt6Ada modules for Windows, Linux (Unix) are available from

Google drive <https://drive.google.com/drive/folders/0B2QuZL0e-yiPbmNQR183M1dTRVE?resourcekey=0-b-M35gZhynB6-LOQww33Tg&usp=sharing>

WebPage is <https://r3fowwcolhrzycn2yzzw-on.driv.tw/AdaStudio/index.html>

Directories tree is

[...]

The full list of released classes is in "Qt6 classes to Qt6Ada packages relation table.pdf"

The simple manual how to build Qt6Ada application can be read in "How to use Qt6ada.pdf"

If you have any problems or questions, tell me know.

Leonid(leonid.dulman@gmail.com)

VisualAda 1.0.0.12

*From: Alex Gamper
<alby.gamper@gmail.com>
Subject: ANN: VisualAda (Ada Integration for Visual Studio 2022) release 1.0.0.12
Date: Mon, 10 Apr 2023 18:43:56 -0700
Newsgroups: comp.lang.ada*

Dear Ada Community

VisualAda version 1.0.0.12 for Visual Studio 2022 has been released

Enhancements include the following

- Bug fixes in Intellisense (Statement completion)

Please feel free to download the free plugin from the following URL

<https://marketplace.visualstudio.com/items?itemName=AlexGamper.VisualStudioAda-2022>

Currency Library for Ada?

*From: A.J. <ianozia@gmail.com>
Subject: Currency Library for Ada?
Date: Thu, 13 Apr 2023 07:17:27 -0700
Newsgroups: comp.lang.ada*

Does anyone know if Ada has a currency library? Ideally one that includes the ISO 4217 currency standard?

I've seen currency referenced as examples in the Style Guide[1] and other documentation[2] but I'm having trouble searching for anything concrete (and googling "ada" and "currency" has not helped due to some unfortunately named crypto stuff).

I'm also interested in if there's any Ada libraries for iso 3166 (country codes).

If none of this exists, that's fine, it just gives me a reason to build it out myself, but I don't want duplication of effort :)

[1] <https://ada-lang.io/docs/style-guide/Reusability/#guideline-16>

[2] https://docs.adacore.com/live/wave/aunit/html/aunit_cb/aunit_cb/fixture.html

*From: J-P. Rosen <rosen@adalog.fr>
Date: Thu, 13 Apr 2023 19:37:19 +0200*

Le 13/04/2023 à 16:17, A.J. a écrit :

> I'm also interested in if there's any Ada libraries for iso 3166 (country codes).

It's a standard package, Ada.Locales

*From: A.J. <ianozia@gmail.com>
Date: Thu, 13 Apr 2023 11:12:19 -0700*

On Thursday, April 13, 2023 at 1:37:22 PM UTC-4, J-P. Rosen wrote:

> It's a standard package, Ada.Locales

Thanks for finding that! This looks like a good foundation for validating county codes, though it doesn't appear to contain an index of them, or expand into the 3-letter codes (e.g. USA vs US). I was looking into the implementation, and the GNAT[1] runtime seems to be true to spec, while the Drake runtime[2] looks like it's expanding into closer to what I'm looking for with its iso639 tables [3]. I may be able to build on this set, though and use the existing structures.

[1] <https://github.com/gcc-mirror/gcc/blob/master/gcc/ada/libgnat/a-locale.ads> & <https://github.com/gcc-mirror/gcc/blob/master/gcc/ada/libgnat/a-locale.adb>

[2] <https://github.com/ytomino/drake/blob/master/source/environment/a-locale.ads>

[3] <https://github.com/ytomino/drake/blob/master/source/environment/a-locale.adb#L60>

*From: Devin Rozsas <devinrozsas@gmail.com>
Date: Thu, 4 May 2023 10:59:11 -0700*

Em quinta-feira, 13 de abril de 2023 às 11:17:28 UTC-3, A.J. escreveu:

> Does anyone know if Ada has a currency library? Ideally one that includes the ISO 4217 currency standard?

I'm actually making something like this, but it isn't complete, and uses Lua scripts to handle different currencies (and formatting). It also has location support (country, state, city) and language support, including message translation (so the program can output stuff in the user's language). For this, TOML is used.

It uses Glottolog codes to identify languages, and FIFA codes for the countries.

I paused the development because I'm focusing on another project that has been causing me some headaches lately. It's broken and cannot deliver what you want - as of now.

*From: Devin Rozsas <devinrozsas@gmail.com>
Date: Thu, 4 May 2023 11:27:47 -0700*

By the way, the code is here: <https://sr.ht/~devin/Azurite-Ada/>

Again, it's incomplete, and probably isn't exactly what you're looking for.

Ada Interface to Excel File

*From: Adamagica <christ-usch.grein@t-online.de>
Subject: Ada interface to Excel file
Date: Wed, 19 Apr 2023 10:36:10 -0700
Newsgroups: comp.lang.ada*

I create Ada code from an Excel file. For this, I first manually export the file to csv format. The code generator works on the csv file. I'd like to automate this first step by including the export into the code generator.

I guess there is a C interface for Excel. I only just need the export functionality, not a full interface. However, being illiterate in C, I'd further welcome help on the way to define an Ada interface to this C code.

Can anyone help, please? Thanx a lot.

*From: Jeffrey R. Carter <spam.jrcarter.not@spam.acm.org.not>
Date: Wed, 19 Apr 2023 20:22:34 +0200*

G. de Montmollin has an Ada Excel writer, an Ada pkg for writing Excel files (<https://sourceforge.net/projects/excel-writer/>). Presumably it could be modified to read them.

*From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Thu, 20 Apr 2023 11:18:53 +0200*

AFAIK, Excel has an ODBC driver. So you can simply read/write an Excel table directly from Ada using ODBC SQL statements.

Units of Measurement for Ada 3.13

*From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Subject: ANN: Units of measurement for Ada v3.13
Date: Sun, 23 Apr 2023 09:20:44 +0200
Newsgroups: comp.lang.ada*

The library provides means for handling measurement units in Ada.

<http://www.dmitry-kazakov.de/ada/units.htm>

Changes to the previous version:

- The package Generic_Complex_Measures was added to provide dimensioned complex values;
- The package Complex_Measures added as an instance of Generic_Complex_Measures with the type Float.

*From: Simon Wright <simon@pushface.org>
Date: Sun, 23 Apr 2023 11:14:35 +0100*

Thanks for this.

The link in "You also may wish to visit this site devoted to the problem of dimensioned values in Ada."

(<http://www.christ-usch-grein.homepage.t-online.de/Ada/Dimension/SI.html>)

results in "Host not found".

*From: Adamagica <christ-usch.grein@t-online.de>
Date: Sun, 23 Apr 2023 03:48:50 -0700*
> The link [...] results in "Host not found".

This can be found there:

<https://www.adaic.org/ada-resources/tools-libraries/> see "Christoph Grein's Essentials"

or more directly:

<http://archive.adaic.com/tools/CKWG/Dimension/Dimension.html>

Simple Components 4.66

*From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Subject: ANN: Simple components for Ada v 4.66
Date: Sun, 23 Apr 2023 09:25:10 +0200
Newsgroups: comp.lang.ada*

The current version provides implementations of smart pointers, directed graphs, sets, maps, B-trees, stacks, tables, string editing, unbounded arrays, expression analyzers, lock-free data structures, synchronization primitives (events, race condition free pulse events, arrays of events, reentrant mutexes, deadlock-free arrays of mutexes), pseudo-random non-repeating numbers, symmetric encoding and decoding, IEEE 754 representations support, streams, persistent storage, multiple connections server/client designing tools and protocols implementations.

<http://www.dmitry-kazakov.de/ada/components.htm>

Changes to the previous version:

- The ambiguities in the ODBC.API package implementation are fixed.

GCC 13.1.0 for MacOS Monterey++

From: Simon Wright
<simon@pushface.org>
Subject: ANN: GCC 13.1.0 for macOS Monterey++
Date: Thu, 27 Apr 2023 16:22:37 +0100
Newsgroups: comp.lang.ada

Find this release, built on Intel but runs on Apple silicon under Rosetta, at https://github.com/simonjwright/distributing-gcc/releases/tag/gcc-13.1.0-x86_64

NB, previous builds were for macOS El Capitan or later, but that machine was elderly.

SparForte 2.6

From: Ken Burtch <koburtch@gmail.com>
Subject: ANN: SparForte 2.6
Date: Tue, 2 May 2023 05:07:35 -0700
Newsgroups: comp.lang.ada

SparForte is a scripting language, template language and shell based on Ada and Bourne shell. It has been in development for 22 years and has about 129,000 lines of code.

This release includes

New features: 11

Changes: 7

Fixes: 12

New features include case procedures, named shell sessions, and Alire support (experimental).

The details of the release are at

https://sparforte.com/news/2023/news_may_2023.html

A summary of the new features is at

https://www.pegasoft.ca/coder/coder_march_2023.html

SparForte can be downloaded from the home page at

<https://sparforte.com>

From: 196...@googlemail.com
<1963bib@googlemail.com>
Date: Sat, 6 May 2023 06:08:45 -0700

Can't be built without sound...

```
gcc -c -O1 -march=athlon64 -gnat12 -gnatfaoN -gnatVaep -gnateEeEeF -fstack-protector -I./adacgi-1.6/ -I./apq-2.1/ -I./pegasock/ -I./areadline/ -lpcr parser_sound.adb
```

parser_sound.adb:93:06: statement expected

gnatmake: "parser_sound.adb" compilation error

If you try and build it without readline...

```
gcc -c -O1 -march=athlon64 -gnat12 -gnatfaoN -gnatVaep -gnateEeEeF -fstack-protector -I./adacgi-1.6/ -I./pegasock/ pegasoft-user_io-getline.adb
```

pegasoft-user_io-getline.adb:286:26: "optional_bold" is undefined

gnatmake: "pegasoft-user_io-getline.adb" compilation error

From: Ken Burtch <koburtch@gmail.com>
Date: Sat, 6 May 2023 06:19:09 -0700

Thank you for sharing this issue.

My automated release tests build SparForte without sound and without readline each night and they have been successful.

I will attempt to diagnose what the problem is. Do you have any further information about your build environment that might be related to this issue?

From: Ken Burtch <koburtch@gmail.com>
Date: Sat, 6 May 2023 06:34:50 -0700

I have pushed fixes for these two issues to the GitHub master branch.

I will do a full set of build tests manually to ensure nothing else is missing.

I will also investigate what the errors were not caught by the automated tests.

Thank you for reporting this issue.

UXStrings 0.5.0

From: Blady <p.p11@orange.fr>
Subject: [ANN] Release of UXStrings 0.5.0
Date: Fri, 5 May 2023 05:36:42 +0200
Newsgroups: comp.lang.ada

This Ada library, providing Unicode character strings of dynamic length, is enriched by a third implementation: UXStrings3 [1] also available on Alire [2]. With this latter implementation, the characters are stored in Unicode form and the management of dynamic size uses the

standard Wide_Wide_Unbounded strings library.

Performance with Gnoga [3] is better. UXStrings2 already brought better performance in the case of strings only made up of ASCII characters (improvement by a factor 2 to 3 compared to UXStrings1). With UXStrings3 performance in the latter case is still improved (factor 6 to 7 compared to UXStrings1) moreover in the case of strings accentuated in French and strings containing emojis the process times are also improved (factor 7 to 8 by compared to UXStrings1 or even more in the case of emojis).

For all cases, the global memory occupation of the Gnoga application is generally similar (9 to 10 Mb). The memory occupation due to UXStrings3 is negligible compared to the memory occupation of the server engine implemented in Gnoga.

Study case: AdaEdit application using the Gnoga graphics library with

UTF-8 files:

English 315 kb

French: 447 kb

Emojis: 439 kb

Process: read all lines of the given file and display the full text

Regardless of the implementation chosen, the appealing of a library is mainly based on the capabilities it offers (API). So far in UXStrings, these are similar to those of the strings Ada standard libraries. If you find some missing, make your proposals on Github [4].

Pascal.

[1] <https://github.com/Blady-Com/UXStrings/blob/master/src/uxstrings3.ads>

[2] <https://alire.ada.dev/crates/uxstrings.html>

[3] <https://sourceforge.net/projects/gnoga>

[4] <https://github.com/Blady-Com/UXStrings/issues>

From: Vincent D.
<vincent.diemunsch@gmail.com>
Date: Thu, 29 Jun 2023 01:49:12 -0700

Hello Pascal,

Thank you for this contribution. Here are some comments:

- since UTFString is a class ("a tagged record type"), why don't you create an abstract root "UXString" and then derive specialized object types ? Like UTF_8_XString, UTF_16_XString, ASCII_XString, Win_1252_XString, Latin_XString, etc.

- The default format to convert between different encodings should be UTF-8 as it is now ubiquitous.

> [...] moreover in the case of strings accentuated in French and strings containing emojis the process times are also improved (factor 7 to 8 by compared to UXStrings1

- I find quite astonishing to have a factor 8 compared to UTF-8 encoding. Do you have an explanation ? This looks like a poor implementation because UTF-8 encoding is fast and allows direct manipulation in most cases. Maybe because random access is treated as sequential access for UTF-8 encoded strings but this again is poor implementation.

GnatStudio 20230501

*From: 196...@googlemail.com
<1963bib@googlemail.com>*

*Subject: GnatStudio 20230501 released
Date: Sun, 14 May 2023 14:31:05 -0700
Newsgroups: comp.lang.ada*

And for Linux it's an appimage. Why? I mean? Its...?

I just wish they could get it into shape where the build was doable without so much hassle - I've never been able to manage it.

*From: Rod Kay <rodakay5@gmail.com>
Date: Mon, 15 May 2023 20:41:14 +1000*

On 15/5/23 07:31, 196...@googlemail.com wrote:

> I just wish [...] the build was doable without so much hassle

The build *has* been getting easier. I maintain the Archlinux gnatStudio package and have nearly got it to build. Currently, I'm waiting on a new/matching release of the AdaCore spawn project. I could, I suppose, use the latest commit version but would prefer to use a formal release.

Also, in the new binary, 'Find all references' appears to be broken (it finds no references). I guess, the same would apply for the refactoring tool. I've been advised to report the issue and will do so tomorrow. It might help to know if other people also experience the same problem(s), before reporting ?

*From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Mon, 15 May 2023 15:58:43 +0200*

On 2023-05-15 12:41, Rod Kay wrote:

> It might help to know if other people also experience the same problem(s)

After typing "with Ada.Strings." I received a use clause as a suggested completion. After completing the subprogram name in a subprogram call, I was shown something other than the

subprogram specification. After typing the '(' for the parameter list, I was shown something other than the formal parameters.

*From: Maxim Reznik
<reznikmm@gmail.com>
Date: Thu, 1 Jun 2023 02:21:55 -0700*

понеделник, 15 мая 2023 г. в 13:43:31 UTC+3, Rod Kay:

> On 15/5/23 07:31, 196...@ wrote:

>> And for Linux it's an appimage. Why?

Why not? It's compact. It doesn't require any installation, so it's handy. You can extract content with --appimage-extract and install GS with ./squashfs-root/usr/doinstall as before.

> Currently, I'm waiting on a new/matching release of the AdaCore spawn project.

All sources are in release assets, like gnatstudio-sources-x86_64-linux.tar.gz. It has spawn-24.0w-20230428-162D4-src.tar.gz for example.

> Also, in the new binary, 'Find all references' appears to be broken

It looks like your ada_language_server doesn't work. Take a look in GS log files (in ~/.gnatstudio/ folder).

*From: Rod Kay <rodakay5@gmail.com>
Date: Fri, 2 Jun 2023 04:27:02 +1000*

On 1/6/23 19:21, Maxim Reznik wrote:

> All sources are in release assets

Ah, great. I will try to rebuild with this.

> It looks like your ada_language_server doesn't work.

I've just re-tested and 'Find all references' works perfectly. How embarrassing!

All i can think of is that I may have had an old gnatstudio version running when I did the GS update and so was still using the old version when I initially tested.

Thanks very much Reznik, very helpful.

*From: Maxim Reznik
<reznikmm@gmail.com>
Date: Mon, 5 Jun 2023 03:15:13 -0700*

Great! Waiting for GNAT Studio in Arch Linux :)

Speaking about AppImage. If you want installed version of GNAT Studio (for instance to have an access to the gnatdoc/gnatdoc4), then you can extract AppImage as an old .tag.gz archive and run doinstall:

```
chmod +x ./GNAT_Studio-x86_64.AppImage
```

```
./GNAT_Studio-x86_64.AppImage --appimage-extract
```

```
./squashfs-root/usr/doinstall
```

From: Rod Kay <rodakay5@gmail.com>

Date: Mon, 5 Jun 2023 22:55:51 +1000

On 5/6/23 20:15, Maxim Reznik wrote:

> Great! Waiting for GNAT Studio in Arch Linux :)

Heh, I've just this minute finished the build/install of GNAT Studio for Arch Linux. The build of GS (and all of it's dependencies) went very well, largely due to using all of the sources provided in the recent GS sources tarball release. So thank you again for suggesting that.

I still have one problem to solve. When I run GS, i get the

following Python error ...

```
Fatal Python error: init_fs_encoding:
failed to get the Python codec of the
filesystem encoding
```

```
Python runtime state: core initialized
```

```
ModuleNotFoundError: No module
named 'encodings'
```

A quick google did not yield any promising solutions but I will look again tomorrow. If anyone can suggest possible reasons/solutions I'd be very grateful. I know little about that pesky snake and less about how to treat one constricted by the beast :).

*From: Maxim Reznik
<reznikmm@gmail.com>
Date: Sat, 10 Jun 2023 03:25:24 -0700*

Probably something wrong with your Python installation. I've tried in GNAT Studio console:

```
>>> import encodings
>>> print(encodings.__file__)
/tmp/gs/share/gnatstudio/python/lib/python3.9/encodings/__init__.py
```

While if I run system packaged Python in my Ubuntu:

```
$ python3
>>> import encodings
>>> print(encodings.__file__)
/usr/lib/python3.10/encodings/__init__.py
$ dpkg-query -S
/usr/lib/python3.10/encodings/__init__.py
libpython3.10-minimal:amd64:
/usr/lib/python3.10/encodings/__init__.py
```

So, it's part of libpython3.10-minimal

*From: Rod Kay <rodakay5@gmail.com>
Date: Wed, 28 Jun 2023 06:47:39 +1000*

It turns out that gnatstudio expects '/usr/share/gnatstudio/python' to contain or point to the root of an OS's Python installation. So a simple soft link to '/usr' fixed this problem.

The only other problem was a deprecated Python module, which was very easy to patch/fix.

So now gnatstudio builds/runs on Archlinux with all the bells/whistles.

A final thanks, Maxim, for your help.

GCC 13.1.0 for Apple Silicon

From: Simon Wright
<simon@pushface.org>
Subject: [ANN] GCC 13.1.0 for Apple silicon
Date: Wed, 17 May 2023 20:23:04 +0100
Newsgroups: comp.lang.ada

See new GCC 13.1.0 releases for aarch64-apple-darwin (i.e. Apple silicon), both native and cross compilation to arm-eabi, at <https://github.com/simonjwright/distributing-gcc/releases>

GWindows 29-May-2023

From: Gautier Write-Only Address
<gautier_niouzes@hotmail.com>
Subject: Ann: GWindows release, 29-May-2023
Date: Mon, 29 May 2023 09:19:11 -0700
Newsgroups: comp.lang.ada

GWindows is a full Microsoft Windows Rapid Application Development framework for programming GUIs (Graphical User Interfaces) with Ada. GWindows works only with the GNAT development system, but with some effort, GWindows could be made pure Ada. GWindows is free and open-source!

Changes to the framework are detailed in [gwindows/changes.txt](#) or in the News forum on the project site.

In a nutshell (since last announcement here):

GWindows release, 29-May-2023 [revision 480]

* Fixes: color picker dialog, mouse wheel methods

478: Contribution: added package GWindows.Pipes

477: Contribution: added package GWindows.Timers

476: Contribution: added package GWindows.Persistence_IO

466: Contribution: initial release of package Office_Applications for helping creating office-like applications.

GWindows release, 13-Nov-2022 [revision 459]

458: GWindows.Common_Controls.Ex_List_View: added `Using_Payloads` to the enumerated type `Comparison_Technique_Type`.

With this choice, sorting runs 100x faster.

451: GWindows.Common_Controls.Ex_List_View: added `As_Strings_Default` to the enumerated type `Comparison_Technique_Type` (sorting runs faster if default alphabetical sorting is desired).

449: GWindows.Application: added procedure `Add_To_Recent_Documents`.

Windows Explorer & Desktop puts the name on top of various "recent documents" lists, for instance in the task bar.

447: GWindows.Common_Controls.Ex_List_View: massive speedup on sorting of large lists (e.g. 6x faster for 20,000 items).

GWindows release, 18-Jun-2022 [revision 440]

* Installer: ResEdit.xml configuration file for the ResEdit

Resource Editor is automatically created and set up for current GNAT installation(s), GWindows and GWenerator.

* Fixed a few 32/64 bit incompatibilities in GWindows.Windows and GWindows.Common_Controls.Ex_List_View.

* Fixed various GNATCOM issues.

GWindows Project site:

<https://sf.net/projects/gnavi/>

GWindows GitHub clone:

<https://github.com/zertovitch/gwindows>

From: Drpi <314@drpi.fr>
Date: Mon, 29 May 2023 21:55:27 +0200

What do you mean by "pure Ada" ?

From: Gautier Write-Only Address
<gautier_niouzes@hotmail.com>
Date: Mon, 29 May 2023 16:59:39 -0700

IIRC, there are a few GNAT-only attributes, like `Unrestricted_Access`, used. No big deal.

But good point, I could check "purity" with the ObjectAda compiler.

From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Tue, 30 May 2023 09:56:13 +0200

I took a quick look. Wouldn't all of Gnatcom need to be replaced?

From: Gautier Write-Only Address
<gautier_niouzes@hotmail.com>
Date: Sat, 3 Jun 2023 23:09:18 -0700

Good question.

When I compile a project using GWindows, GNAT uses 10 of the 53 GNATCOM packages.

One GNATism is 4x `Unrestricted_Access` in GNATCOM.Types, for accesses such as:

```
VARIANT_MISSING : aliased constant
VARIANT := (
  VT_ERROR, 0, 0, 0, u => (Which => 8,
  scode => DISP_E_PARAMNOTFOUND));
PVARIANT_MISSING :
  Pointer_To_VARIANT :=
  VARIANT_MISSING'Unrestricted_Access;
```

that could be either resolved into a standard Ada form or exiled into another package (GWindows doesn't need them).

Something tougher is a couple of intrinsic imports (`sync_add_and_fetch`, `sync_sub_and_fetch`):

```
function sync_add_and_fetch
  (Ref : access Interfaces.Unsigned_32;
  Add : Interfaces.Unsigned_32)
  return Interfaces.Unsigned_32
with Import,
  Convention => Intrinsic,
  External_Name =>
  "__sync_add_and_fetch_4";
```

which seems to be specific to GCC (and actually, not even all versions of GCC...)

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Sat, 17 Jun 2023 02:18:05 -0500

This looks like an atomic operation. A portable Ada definition of such operations is found in C.6.1-C.6.4 of Ada 2022. Probably those could be used to replace the operation (of course, that would limit one to compilers supporting that part of Ada 2022; dunno if anyone is doing that yet).

LEA 0.87

From: Gautier Write-Only Address
<gautier_niouzes@hotmail.com>
Subject: Ann: LEA v.0.87
Date: Mon, 29 May 2023 09:29:20 -0700
Newsgroups: comp.lang.ada

LEA is a Lightweight Editor for Ada

Web site: <http://l-e-a.sf.net/>

Source repository #1:
<https://sf.net/p/l-e-a/code/HEAD/tree/>

Source repository #2:
<https://github.com/zertovitch/lea>

Changes since last announcement here:

- Added auto insert feature: e.g. typing ``(` inserts `)``.
- Added color theme Solarized Light.
- Added a "stealth mode" in which LEA doesn't leave traces in the registry.
- Editor adds `--`` if the cursor is within a comment when the Return key is pressed (consequence: a comment is split into two comments).
- If the cursor is within a string literal when the Return key is pressed, the string literal is split into two valid string literals with a ``&`` between them.

- Added unhandled exception information to message list
- Tabs with the various file names
- LEA doesn't write scilexer.dll as a file; thus, it runs as a portable application (in the sense: you can run it from a read-only drive directly, without installation)
- Added a Build & Run button (for the HAC compiler).

Features:

- multi-document
- multiple undo's & redo's
- multi-line & multi-point edit, rectangular selections
- color themes, easy to switch
- duplication of lines and selections
- syntax highlighting
- parenthesis matching
- bookmarks

Currently available on Windows.

Gtk or other implementations are possible: the LEA_Common[.*] packages are pure Ada, as well as HAC.

*From: Drpi <314@drpi.fr>
Date: Mon, 29 May 2023 21:56:35 +0200*

Just missing Alire and ALS compatibility ;)

*From: Gautier Write-Only Address <gautier_niouzes@hotmail.com>
Date: Mon, 29 May 2023 22:27:14 -0700*

Alire: are you missing a LEA crate?

ALS: = Ada language server?

*From: Drpi <314@drpi.fr>
Date: Tue, 30 May 2023 08:03:19 +0200*

> Alire: are you missing a LEA crate?

Why not but I was thinking about compiling/running Alire projects from LEA.

> ALS: = Ada language server?

That's it. Auto-completion and mouse-over documentation in LEA.

*From: Gautier Write-Only Address <gautier_niouzes@hotmail.com>
Date: Wed, 31 May 2023 20:18:57 -0700*

> Why not but I was thinking compiling/running Alire projects from LEA.

Good idea! For instance the "Build & Run" command (the green button) would launch "alr run" in that context.

Added to the to-do list.

Side note: a cool project would be a graphical tool, "Alire Explorer" (good name to be found) with buttons for the key Alire commands, a box displaying the contents of "alr show", ...

Perhaps something to be made with GNOGA.

>> ALS: = Ada language server?

> That's it. Auto-completion and mouse-over documentation in LEA.

Also added to the to-do list.

*From: Gautier Write-Only Address <gautier_niouzes@hotmail.com>
Date: Fri, 9 Jun 2023 14:41:27 -0700*

LEA is now available on Alire (<https://alire.ada.dev/>) !

alr index --update-all

alr get lea

cd lea <-- here you press the Tab key to complete

alr build

lea

Ayacc and Aflex 2023

*From: Gautier Write-Only Address <gautier_niouzes@hotmail.com>
Subject: Re: Status of ayacc and aflex?
Date: Wed, 31 May 2023 13:42:18 -0700
Newsgroups: comp.lang.ada*

Old thread, but since some search engines point to here as top hit when searching for "ayacc" and "aflex", it is worth mentioning that the new developments (as of mid 2023) of ayacc and aflex are located here:

<https://github.com/Ada-France/ayacc>

<https://github.com/Ada-France/aflex>

PragmAda Reusable Components

*From: Pragmada Software Engineering <pragmada@pragmada.x10hosting.com>
Subject: [Reminder] The PragmAda Reusable Components
Date: Thu, 1 Jun 2023 10:38:17 +0200
Newsgroups: comp.lang.ada*

The PragmARCs are a library of (mostly) useful Ada reusable components provided as source code under the GMGPL or BSD 3-Clause license at <https://github.com/jrcarter/PragmARC>.

This reminder will be posted about every six months so that newcomers become aware of the PragmARCs. I presume that those who want notification when the PragmARCs are updated have used Github's notification mechanism to receive them, so I no longer post update announcements. Anyone who wants to receive notifications without using Github's mechanism should contact me directly.

Qplt

*From: Jeffrey R. Carter <spam.jrcarter.not@spam.acm.org.not>
Subject: Ann: Qplt
Date: Fri, 2 Jun 2023 17:49:20 +0200
Newsgroups: comp.lang.ada*

I have created Qplt (Quick Plot), and Ada-GUI program to quickly produce a plot of a data set, and make it publicly available in hopes that it might prove useful. The program automatically selects axis ranges and tick intervals. The user may select whether points, lines, or both are plotted, and supply a title and axis labels.

Qplt is available at

<https://github.com/jrcarter/Qplt>

Ada and Operating Systems

GCC 13.1.0 (x86_64) on Ventura 13.3.1

*From: Bill Findlay <findlaybill@blueyonder.co.uk>
Subject: Trying GCC 13.1.0 (x86_64) on Ventura 13.3.1
Date: Sat, 29 Apr 2023 00:55:53 +0100
Newsgroups: comp.lang.ada*

Hi Simon,

Many thanks for the x86 macOS build of GNAT. Does it incorporate front-end updates since the Sep 30 build of gnat-12.2.0-1?

> which gnat

> /opt/gcc-13.1.0/bin/gnat

Using the command:

```
> GCC -c -I./-I./Source -funwind-tables -gnatl12j96 -gnatw.e -gnatwD -gnatwH -gnatwP -gnatwT -gnatw.W -gnatw.B -gnatwC -gnatw.u -gnatyO -gnatw.Y -gnatw.N -fdata-sections -ffunction-sections -gnatfn -mtune=native -Ofast -fno-stack-check -fomit-frame-pointer -flto -I/Users/wf/KDF9/emulation/Source/ee9.adb
```

I got:

```
> clang (LLVM option parsing): Unknown command line argument '-x86-pad-for-align=false'. Try: 'clang (LLVM option parsing) --help'
```

```
> clang (LLVM option parsing): Did you mean '--x86-slh-loads=false'?
```

```
> gnatmake: "/Users/wf/KDF9/emulation/Source/ee9.a db" compilation error
```

*From: Simon Wright <simon@pushface.org>
Date: Sat, 29 Apr 2023 16:08:04 +0100*

Bill Findlay
<findlaybill@blueyonder.co.uk> writes:
> Does it incorporate front-end updates
since the Sep 30 build of gnat-12.2.0-1?

It includes whatever changes AdaCore & fellow maintainers have made! From here
<<https://gcc.gnu.org/gcc-13/changes.html>>,

Ada

Traceback support added in RTEMS for the PPC ELF and ARM architectures.

Support for versions older than VxWorks 7 has been removed.

General improvements to the contracts in the standard libraries.

Addition of GNAT.Binary_Search.

Further additions and fixes for the Ada 2022 specification.

The Pragma SPARK_Mode=>Auto is now accepted. Contract analysis has been further improved.

Documentation improvements.

Ada FreeDos/DOS Experiences

*From: Hou Van Boere
<houvanboere@gmail.com>
Subject: Please Share Ada -Freedos - Dos experiences
Date: Sat, 27 May 2023 08:44:55 -0700
Newsgroups: comp.lang.ada*

Hi Everyone

I am thinking about using FreeDos as a kind of RTOS. The application is to control scientific instruments so portability is a non-issue.

Can you please share bits and pieces about running Ada on FreeDos (or MS DOS)

*From: Joakim Strandberg
<joakimds@kth.se>
Date: Sat, 27 May 2023 12:49:36 -0700*

I wrote about how to get DJGPP compiler on DOS:
https://www.reddit.com/r/ada/comments/vrhsv5/how_to_install_gnat_314b_on_freedos_13/

I recommend installing a recent version of DJGPP, you will be able to use a lot of the Ada language except for tasking which DJGPP does not support on FreeDos.

I have been looking for an Ada83 or Ada95 compiler for DOS which compiles real-mode executables but the ones I found are still proprietary and can be bought. It indicates there are still old systems on old hardware still in use.

I haven't built something on DOS, just toying with the idea. I've successfully been able to execute my Advent of code solutions for 2022 on FreeDOS. I've also

successfully switched to VGA mode and putting pixels on the screen and switching back to text mode from an Ada application. I did it by interfacing with C code that had assembler embedded, if I remember correctly.

*From: Joakim Strandberg
<joakimds@kth.se>
Date: Sat, 27 May 2023 13:02:58 -0700*

Another idea is to use the ObjectAda 7.0 compiler (free version) from 1996 that can be downloaded here:

<https://archive.org/details/ObjectAdaSE7>

It runs on Windows 95/98 but looking at the documentation for the ObjectAda compiler it says it is possible to use the compiler to create executables for DOS by using a DOS Extender. I haven't tried it but it should be possible to get working. Unfortunately there are limitations with the free version. One good thing is that it is possible to use tasks freely for creating a FreeDOS application but one must restrict one-self to Ada95 since the compiler is from 1996.

There is a professional version of ObjectAda from 2002 that can be downloaded here:
<https://vetusware.com/download/ObjectAda%207.2.2%20Enterprise%207.2.2/?id=17315>

I've tested it and it works but the documentation no longer talks about being able to create executables for FreeDOS. Maybe it can still be used to make executables for FreeDOS?

*From: Joakim Strandberg
<joakimds@kth.se>
Date: Sat, 27 May 2023 13:07:30 -0700*

However, the biggest obstacle for using FreeDOS is hardware support. FreeDOS depends upon BIOS and all motherboards since 2020 no longer support BIOS. Does anybody know of any hardware produced today that supports FreeDOS?

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 27 May 2023 14:54:28 -0700*

Thanks Joakim! this is super helpful.

I downloaded the compiler cd.

I have tried this:

<https://github.com/andrewwutw/build-djgpp>

It looks helpful to build dljgpp but it does not work well enough on Trisquel Linux. I find that building GCC on current or old Slackware versions seems to work well and I am going to re-try this project. I know I will have to re-run with --enable-languages=c,Ada later but at least it should set up most of what is needed.

Your Freedos environment tips will help a lot.

I just bought my son a new computer and I am kind of depressed after. The store was huge but completely geared towards gaming. It seems like today's computers are not well suited for hardware interfacing and hacking with electronics. There was way more expansion in the past and I hate having to configure for legacy bios. I think this will be dropped soon too and then we will be stuck

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 27 May 2023 14:55:28 -0700*

P.S I use less than half of Ada 95 so this compiler could help a lot.

*From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.net>
Date: Sun, 28 May 2023 01:00:21 +0200*

On 2023-05-27 17:44, Hou Van Boere wrote:

> Can you please share bits and pieces about running Ada on FreeDos(or MS DOS)

I used Ada (83) (Janus/Ada and Meridian Ada) on DOS PCs in the 80s and 90s. It was much like writing command-line applications for Linux or Windows today. I also did some low-level stuff, trapping key strokes and doing graphics. But if you're thinking of using DOS as an RTOS then that's probably not very helpful for you.

RR Software (rrsoftware.com) continues to sell an Ada-83 DOS compiler, and may be able to provide an Ada-95 DOS compiler on request. Their prices are reasonable: the personal edition of their Ada-95 Windows compiler is \$195.

Or you could look at the MaRTE OS RTOS (<https://marte.unican.es/>) which is written mostly in Ada and supports GNAT compilers. I don't know how that would compare in terms of ease of getting things set up or developing S/W for it.

*From: Keith Thompson
<keith.s.thompson+u@gmail.com>
Date: Sat, 27 May 2023 16:31:39 -0700*

Joakim Strandberg <joakimds@kth.se> writes:

> There is a professional version of ObjectAda from 2002 that can be downloaded

I wonder if those are authorized copies. I suspect they aren't.

Aonix no longer exists, but apparently its assets are now owned by PTC, which still sells (a much newer version of) ObjectAda.
<https://www.ptc.com/en/products/developer-tools/objectada>

The copy on archive.org is of a CD whose label says "This edition of ObjectAda is not licensed for development of commercial software. This CD may not

be re-sold. It does have an "All rights reserved" copyright message.

(I worked for Aonix many years ago, but I have no current connection with them or their successors.)

From: Drpi <314@drpi.fr>

Date: Sun, 28 May 2023 13:01:17 +0200

> However, the biggest obstacle for using FreeDOS is hardware support.

Do you really need to use old PC hardware ?

On a PC (and ARM), you can also run QNX which is a real-time micro-kernel OS. It is a commercial product but is free for education and research.

On PCs it is currently easy to use PCIe extension boards. Like FPGA boards.

Also, there are very powerful non x86 (mostly ARM) hardware today. Most of these boards have PCIe ports to easily add extension boards.

From: Drpi <314@drpi.fr>

Date: Sun, 28 May 2023 19:42:08 +0200

I forgot to say that AdaCore sells a Ada compiler for some QNX versions (7.x +) but I don't know if there is a free version for education/research.

ARM 64-bit Binary Support

From: Dmitry A. Kazakov

<mailbox@dmitry-kazakov.de>

Subject: ARM 64-bit binary support

Date: Sat, 24 Jun 2023 13:22:09 +0200

Newsgroups: comp.lang.ada

Recently Linux Fedora and Ubuntu distributions stopped ARMv7 support (32-bit).

I added 64-bit architecture to the repositories of the following libraries for Debian, Fedora and Ubuntu:

- Ada industrial control widget library

<http://www.dmitry-kazakov.de/ada/aicwl.htm>

- Fuzzy machine learning framework

http://www.dmitry-kazakov.de/ada/fuzzy_ml.htm

- Fuzzy sets, logic, numbers

<http://www.dmitry-kazakov.de/ada/fuzzy.htm>

- GtkAda (pre-built)

<http://www.dmitry-kazakov.de/ada/gtkada.htm>

- GtkAda contributions

http://www.dmitry-kazakov.de/ada/gtkada_contributions.htm

- MAX! cube GUI for management of indoor radiator thermostats

http://www.dmitry-kazakov.de/ada/max_home_automation.htm

- Interval arithmetic

<http://www.dmitry-kazakov.de/ada/intervals.htm>

- Measurement units

<http://www.dmitry-kazakov.de/ada/units.htm>

- Simple component

<http://www.dmitry-kazakov.de/ada/components.htm>

- String editing, UTF-8 issues

http://www.dmitry-kazakov.de/ada/strings_edit.htm

- Table management

<http://www.dmitry-kazakov.de/ada/tables.htm>

ARMv7 builds are continued for the last official releases of the corresponding OSes.

References to Publications

Ada 2022 LRM by Springer

From: Dirk Craeynest

<dirk@orka.cs.kuleuven.be>

Subject: Ada 2022 Language Reference Manual to be Published by Springer

Date: Wed, 14 Jun 2023 06:49:45 -0000

Newsgroups: comp.lang.ada, fr.comp.lang.ada, comp.lang.misc

FOR IMMEDIATE RELEASE

Ada 2022 Language Reference Manual to be Published by Springer

Lisbon, Portugal, June 14, 2023 - Ada-Europe today announced, at its 27th International Conference on Reliable Software Technologies (AEiC 2023), that the Ada 2022 Language Reference Manual (LRM) will be published by Springer in its LNCS series later this year.

Ada 2022 is the latest edition of the Ada programming language standard, technically denominated ISO/IEC 8652:2023, which was formally approved and officially published by ISO, the Geneva-based International Organization for Standardization, on May 2, 2023.

The Ada 2022 LRM is available online:

www.ada-auth.org/standards/ada22.html.

An overview of Ada 2022 is at:

www.ada-auth.org/standards/overview22.html.

To mark this official milestone, and in continuation of its established practice, Ada-Europe undertook to support the production of the new LRM as a dedicated issue of the Springer-published LNCS series.

About Ada-Europe

Ada-Europe is the international non-profit organization that promotes the knowledge and use of the Ada programming language in academia, research and industry. Its flagship event is the annual International Conference on Reliable Software Technologies, a high-quality technical and scientific event that has been successfully running in the current format for the last 27 years. Ada-Europe has member organizations in Belgium, Denmark, France, Germany, Spain, and Switzerland, as well as individual members in many other countries. For information about Ada-Europe, its charter, activities and sponsors, please visit: www.ada-europe.org. Ada-Europe is headquartered in Brussels, Belgium.

A PDF version of this press release is available at www.ada-europe.org.

Organization Contacts

Ada-Europe

Tullio Vardanega, Ada-Europe President

president@ada-europe.org

Press Contacts

Ada-Europe

Dirk Craeynest, Ada-Europe Vice-president

c/o KU Leuven, Department of Computer Science

dirk.craeynest@cs.kuleuven.be

(VAda2022.1)

From: Adamagica <christ-usch.grein@t-online.de>

Date: Wed, 14 Jun 2023 01:20:19 -0700

> The Ada 2022 LRM is available online:

> www.ada-auth.org/standards/ada22.html.

This is still Draft 35. The final version is not yet available. See also <https://groups.google.com/g/comp.lang.ada/c/P26SS3L7kA0> - Ada 23 at Last!

From: Dirk Craeynest

<dirk@orka.cs.kuleuven.be>

Date: Wed, 14 Jun 2023 14:13:33 -0000

AdaMagica <christ-usch.grein@t-online.de> wrote:

>This ist still Draft 35. The final version is not yet available.

Note that the page at the above URL mentions:

"This is draft 35. This draft contains all ARG-approved AI12s. This is the draft that has been submitted to complete the standardization process."

So draft 35 *is* what was submitted to ISO.

Randy, the RM editor, is aware that this and a few other web pages still have to be updated now ISO published the new RM, and he assured me after the WG9 meeting yesterday that this is on his "to do list".

>See also

<https://groups.google.com/g/comp.lang.ada/c/P26SS3L7kA0> - Ada 23 at Last!

That message claimed about the ISO document: "The ToC is very different from Draft 35."

While draft 35 is what was submitted to ISO, the documents indeed are not identical. Though I would not say the ToC's are "very different".

Yes, the introductory chapters in the ISO document are slightly different from those in the RM on ada-auth.org, and there's no Annex on "Obsolescent Features" nor a "Glossary" (that was removed in draft 35 anyway). All this is due to specific requirements that ISO has for its standards. There are more differences, such as the ISO document not having any paragraph numbers as those are not allowed in ISO standards.

But the bulk of the ToC is identical, apart from those differences required by ISO. Most importantly: the described language in both documents is identical.

*From: Egil H H <ehh.public@gmail.com>
Date: Wed, 14 Jun 2023 09:11:05 -0700*

On Wednesday, June 14, 2023 at 3:13:36 PM UTC+1, Dirk Craeynest wrote:

> But the bulk of the ToC is identical, apart from those differences required by ISO. Most importantly: the described language in both documents is identical.

The clause numbering is not the same, as clause 1 has been split into 4 clauses in the ISO version, so clause `2 Lexical Elements` in the Draft corresponds to `5 Lexical Elements` in the ISO version.

And (at least) one bug in the ISO ToC, `7.3.4 Delta Aggregates` and `7.3.5 Container Aggregates` are collapsed beneath `7.3.3. Array Aggregates`, even though the subclause level is the same.

*From: Adamagica <christ-usch.grein@t-online.de>
Date: Wed, 14 Jun 2023 09:15:37 -0700*

Dirk Craeynest schrieb am Mittwoch, 14. Juni 2023 um 16:13:36 UTC+2:

> So draft 35 *is* what was submitted to ISO.

Yes; I know...

> That message claimed about the ISO document: "The ToC is very different from Draft 35."

Funny, when I first opened the preview, the complete table of contents with page

numbers could be read. The ISO document had far fewer pages than Draft 35 (951 pages). I wondered how this could be...

Now the ToC is without page numbers, so I cannot compare.

If you compare the ISO ToC and the Draft 35 one, you'll see that clause and subclause numbers differ. So old references like RM 3.5 will lead astray.

--- ISO locuta, causa finita. ---

From: Adamagica <christ-usch.grein@t-online.de>

Date: Wed, 14 Jun 2023 10:03:26 -0700

Egil H H schrieb am Mittwoch, 14. Juni 2023 um 18:11:07 UTC+2:

> And (at least) one bug in the ISO ToC,

Not only this. The whole of 7.4 to 7.10 is collapsed under 7.3.3.

From: Keith Thompson

<keith.s.thompson+u@gmail.com>

Date: Wed, 14 Jun 2023 12:47:29 -0700

dirk@orka.cs.kuleuven.be. (Dirk Craeynest) writes:

> the ISO document not having any paragraph numbers as those are not allowed in ISO standards.

Is disallowing paragraph numbers a recent change I have a copy of the 2011 ISO C standard, ISO/IEC 9899:2011 (E), and it definitely has paragraph numbers. (Which are extremely useful, BTW; it seems silly for ISO to disallow them.)

From: Simon Wright

<simon@pushface.org>

Date: Thu, 15 Jun 2023 20:36:35 +0100

Egil H H <ehh.public@gmail.com> writes:

> And (at least) one bug in the ISO ToC,

From my point of view, never mind the bug, this makes the ISO document a white elephant.

The stability of the clause numbering, and the hyperlinking, make the RM the valuable document that it is.

From: Randy Brukaradt

<randy@rrsoftware.com>

Date: Sat, 17 Jun 2023 02:49:12 -0500

Actually, paragraph numbers weren't allowed back in the Ada 83/Ada 95 days. So the original ISO versions didn't have them. You can use them in ISO documents now (I don't know when this changed), but you have to get a special waiver to do so - for *every* individual standard that you want to have them (that's a recent change, for the worse). And if we added them to the ISO version (after getting the appropriate waiver -- which I didn't know about for this last round of standardization), they'd be different than the ones in the RM

(because they wouldn't allow versioning or inserted numbers). That doesn't seem helpful to me, YMMV.

ISO no longer lets us be compatible with the clause numbering of previous versions -- ALL standards have to follow their numbering for initial stuff. They've also changed from requiring not using Annexes I and O (since they're easily confused with chapters (nope, now sections (nope, now clauses)) -- to requiring having Annexes I and O.

Bob Duff explained it best: The people maintaining the "standards for standards" have made no attempt to keep upward compatibility in their work (unlike us Ada people). Every standard in existence has to be changed substantially with each new edition in order to meet the ever-changing requirements. It's hard to believe that these people don't understand (or don't care) that these standards are used for a very long time.

Randy Brukaradt, Project Editor, ISO/IEC 8652

Ada and Other Languages

Java and Python get "record" Type after 40 Years.

*From: Nasser M. Abbasi
<nma@12000.org>*

Subject: Java and Python have just discovered "record" type finally after 40 years.

Date: Fri, 12 May 2023 12:50:14 -0500

Newsgroups: comp.lang.ada

Java 14 now has "Record" !

"records are meant to be data carriers"

<https://www.digitalocean.com/community/tutorials/java-records-class>

And Python 3.7 now has records, they call it "data class"

<https://realpython.com/python-data-classes/>

"One new and exciting feature coming in Python 3.7 is the data class. A data class is a class typically containing mainly data"

What took them so long? Pascal and Ada had records from day one, only 40 years ago or so.

From: Richardthiebaud

<thiebauddick2@aol.com>

Date: Fri, 12 May 2023 14:58:52 -0400

And Cobol had them 63 years ago.

From: Jeffrey R. Carter

<spam.jrcarter.not@spam.acm.org.not>

Date: Fri, 12 May 2023 23:33:54 +0200

Pascal had them in 1970. Algol, I think, had them in 1960.

From: Niklas Holsti

<niklas.holsti@tidorum.invalid>

Date: Sat, 13 May 2023 10:13:22 +0300

On 2023-05-12 19:50, Nasser M. Abbasi wrote:

> What took them so long?

Java and Python have classes, which have records as a special case, if the term "record" is understood as in most other languages, including Ada.

But it seems that the Java 14 "record" is not quite the same as an Ada record, because Java 14 records are meant to be immutable data carriers, not mutable data structures. Still, Java 14 records are described as a (very) special case of classes.

> Pascal had them in 1970. Algol, I think, had them in 1960.

Algol 60 did not have records, only arrays.

Algol W, a precursor to Pascal, had them in 1966.

Simula had them in 1967. (Wikipedia says "In 1966 C. A. R. Hoare introduced the concept of record class construct".)

Algol 68 had them in 1968.

Pascal had them in 1970, as you say.

From: Luke A. Guest

<laguest@archeia.com>

Date: Sat, 13 May 2023 12:18:04 +0100

What about COBOL and LISP?

From: Niklas Holsti

<niklas.holsti@tidorum.invalid>

Date: Sat, 13 May 2023 19:53:45 +0300

On 2023-05-13 14:18, Luke A. Guest wrote:

> What about COBOL and LISP?

As I understand it (but I don't claim to be expert), the early COBOL languages could describe the structure of file records, and of working-storage objects, as nested sequences of components and sub-records, but each such description defined a `_single_` "record" object, not a "record" data-type that could have many instances. So if you wanted to have two record objects with the same structure, you had to duplicate the whole record description.

However, Wikipedia says that the COBOL record structure inspired records for Pascal.

Early LISP languages did not have record types, AFAIK. But you could of course use lists to program record-like data structures.

From: J-P. Rosen <rosen@adalog.fr>

Date: Sun, 14 May 2023 08:46:15 +0200

Le 13/05/2023 à 18:53, Niklas Holsti a écrit :

> So if you wanted to have two record objects with the same structure, you had to duplicate the whole record description.

AFAIR, COBOL didn't have types, but you could define a variable LIKE another one.

> Early LISP languages did not have record types, AFAIK. But you could of course use lists to program record-like data structures.

Of course, in LISP there is only one structure, for data and programs alike: the list!

From: Nasser M. Abbasi

<nma@12000.org>

Date: Sun, 14 May 2023 02:20:42 -0500

On 5/14/2023 1:46 AM, J-P. Rosen wrote:

> Of course, in LISP there is only one structure, for data and programs alike: the list!

This is similar to Mathematica. I programmed a little in Lisp, and it was kinda fun.

In Mathematica, its main data struct is also the list and list of lists and list of list of lists and so on.

```
a={1,2,3};
```

```
a={{1,2,3},{4,5,6}};
```

Everything in Mathematica is pretty much build using lists.

Few years ago, Wolfram introduced Association, which acts like a RECORD. It is really like a dictionary. It has key->value pairs so one can do:

```
myData = <| "name"->"me", "age"->99 |>
```

To read the value of a field one uses `myData["name"]` or `myData["age"]`.

It is amazing how people can program so much code using only just a list as the main basic data structure and be able to get away with it :)

I think RECORD is the most important data structure myself.

Without a RECORD (called struct in C), programming is much harder. This is what Java and Python have discovered just now. I guess the language designers of these languages never bothered to look at Pascal or Ada before.

But better late than never I guess.

From: Luke A. Guest

<laguest@archeia.com>

Date: Sun, 14 May 2023 10:45:36 +0100

On 14/05/2023 07:46, J-P. Rosen wrote:

> Of course, in LISP there is only one structure, for data and programs alike: the list!

Well, that's not true anymore, especially not in common lisp which has a variety of data structures including records, I was quite surprised to see that when I was looking at it last year.

From: Luke A. Guest

<laguest@archeia.com>

Date: Sun, 14 May 2023 10:49:17 +0100

On 14/05/2023 08:20, Nasser M. Abbasi wrote:

> This is what Java and Python have discovered just now.

I think people might finally be realising that you can't do everything with only one programming paradigm.

From: Ben Bacarisse

<ben.usenet@bsb.me.uk>

Date: Sun, 14 May 2023 11:37:21 +0100

"J-P. Rosen" <rosen@adalog.fr> writes:

> Of course, in LISP there is only one structure, for data and programs alike: the list!

LISP had S-expressions -- pairs of atoms or other S-expressions. A list was just a special case. Many other structures could be built using S-expressions. An important one was that association list -- a list of (key . value) pairs that was often used very much like a record type (though it's quite a different beast).

From: Jeffrey R. Carter

<spam.jrcarter.not@spam.acm.org.not>

Date: Sun, 14 May 2023 12:39:02 +0200

On 2023-05-14 08:46, J-P. Rosen wrote:

> Of course, in LISP there is only one structure, for data and programs alike: the list!

In the LISP I learned, there were only S-expressions (SEXes). A SEX is either an atom or a list of SEXes. Another way of putting it was there were atoms and lists of atoms or lists. Either way, there were also atoms.

From: J-P. Rosen <rosen@adalog.fr>

Date: Sun, 14 May 2023 17:10:23 +0200

Le 14/05/2023 à 12:39, Jeffrey R. Carter a écrit :

> A SEX is either an atom or a list of SEXes.

Right, but I would define atoms as the basic data, not a data /structure/. Oh well, just a matter of definition...

From: Ben Bacarisse

<ben.usenet@bsb.me.uk>

Date: Sun, 14 May 2023 16:14:33 +0100

"Jeffrey R. Carter"

<spam.jrcarter.not@spam.acm.org.not> writes:

> A SEX is either an atom or a list of SEXes.

I never saw a LISP S-expressions defined that way. Did this list really not have a "dotted pair" as the basic structure with lists being simply a special case?

From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Sun, 14 May 2023 18:56:27 +0200

On 2023-05-14 17:14, Ben Bacarisse wrote:

> Did this list really not have a "dotted pair" as the basic structure with lists being simply a special case?

The book I learned from (/Let's Talk LISP/ by Laurent Siklóssy, 1976) introduces dotted pairs in chapter 10.7.1 (out of 12 chapters) on page 145 (out of 213, excluding appendices and index). Chapter 10 deals with the internal representation of data in LISP. The implication is that they were not considered part of the normal use of the language.

S-expressions, on the other hand, are introduced in chapter 1.1 on page 2. The book also presents the grammar

S-expression ::= atom | list

list ::= '(' inside ')'

inside ::= empty | S-expression | S-expression inside

empty ::=

From: Ben Bacarisse
<ben.usenet@bsb.me.uk>
Date: Mon, 15 May 2023 02:11:55 +0100

"Jeffrey R. Carter"
<spam.jrcarter.not@spam.acm.org.not>
writes:

> The book I learned from (/Let's Talk LISP/ by Laurent Siklóssy, 1976)

Do you still have it? Does it discuss association lists? I'd call them a normal part of LISP and it would be odd to force the associations to be lists rather than pairs. Does Siklóssy imply that an ASSOC list is a list of lists of length 2, or does he not discuss them until the very end?

> The book also presents the grammar

That's an interesting way to simplify things for the learner though I would not have chosen to use a term that already had another meaning by 1976. The author could have used something like L-expression and avoided any future confusion.

From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Mon, 15 May 2023 12:44:13 +0200

On 2023-05-15 03:11, Ben Bacarisse wrote:

> Do you still have it? Does it discuss association lists?

The Function ASSOC is discussed in Chapter 9 as an auxiliary function used by EVAL (Chapter 9 discusses the working of EVAL). It says

"ASSOC finds the value of a variable in the ALIST. The ALIST is a list of sublists of two SEXes each of the form (variable value-of-the-variable)."

In a footnote he notes that the ALIST could be a list of dotted pairs, which are defined in the next chapter.

I never did much with LISP after learning it, and never looked at any other textbooks, so he might have an idiosyncratic approach. Seems rather OT for c.l.a.

From: Ben Bacarisse
<ben.usenet@bsb.me.uk>
Date: Wed, 17 May 2023 01:24:32 +0100

"Jeffrey R. Carter"
<spam.jrcarter.not@spam.acm.org.not>
writes:

> ASSOC finds the value of a variable in the ALIST. The ALIST is a list of sublists of two SEXes each of the form (variable value-of-the-variable).

Thanks. Is this a dialect made up for pedagogic purposes? I don't know of any practical LISP that went down this route.

> Seems rather OT for c.l.a.

Yes, it is. Happy to stop. I was just curious about where your use of terms originated and that now explained.

Ada Scales Down!

From: Hou Van Boere
<houvanboere@gmail.com>
Subject: Ada Scales Down!
Date: Sat, 13 May 2023 17:17:50 -0700
Newsgroups: comp.lang.ada

Hi Everyone

Just a little cross post:

<https://sourceforge.net/p/gnucobol/discussion/cobol/thread/5f771109ad/>

I am having so much fun with Ada again. I think the foreign binding examples on the net are horrible. With little wimpy inline packages, you can bring foreign code in easily.

Everyone complains about C but a teenager can fool around with it on a weekend and end up a C programmer a few years later. Ada does not present this way but in fact it is easy to write little wimpy programs in it just for fun and even wimpy programs will often need non-Ada libraries.

From: Luke A. Guest
<laguest@archeia.com>
Date: Sun, 14 May 2023 10:53:29 +0100

On 14/05/2023 01:17, Hou Van Boere wrote:

> Hi Everyone

>

> Just a little cross post:

> <https://sourceforge.net/p/gnucobol/discussion/cobol/thread/5f771109ad/>

It's not 1979 anymore, you can use unicode in Ada and even lowercase letters. This is not Oberon where the language is stuck in the 70's where there was a limited character set available on keyboards. I think even COBOL can now accept lowercase keywords now, but I'm not sure about this.

From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sun, 14 May 2023 06:59:25 -0700

Hi Luke

I knew someone would mention this :) Most people program in lowercase with COBOL now. It is a personal preference. I use a smaller font and have more code on the screen with uppercase and I am just kinda retro about a lot of things. Think Amish using a computer :)

Does Safer Mean Slower?

From: Nasser M. Abbasi
<nma@12000.org>
Subject: does a safer language mean it is slower to run?
Date: Wed, 7 Jun 2023 22:55:51 -0500
Newsgroups: comp.lang.ada

Some folks in this thread

<https://discourse.julialang.org/t/comparison-of-rust-to-julia-for-scientific-computing/78508>

"I'm not an expert, but my feeling is that Rust is a "safer" language, which to me means it must be slower."

etc..

Some in that thread seem to argue that a safer language will/could be slower than otherwise.

Since Ada is known to be one of the safest languages, do others here feel there is any truth to this?

I thought that by having more type information in the language, the compiler will be able to make more optimizations (because it knows more), and hence the generated code should actually be faster, not slower with a language that is less safe?

I am not a compiler expert but what do others here think?

From: Niklas Holsti
<niklas.holsti@tidorum.invalid>
Date: Thu, 8 Jun 2023 09:57:14 +0300

If a language needs run-time checks to ensure safety, those checks usually take some time, making for slower execution.

If a language has a type system and compilation-time (legality) rules such that the compiler can prove that some run-time checks are not needed, that reduces or eliminates the slow-down. This is the case for Ada.

The effect of type information on optimization is harder (at least for me) to understand. If the type information lets the compiler assume that some objects are not aliased, that can help optimization because more computation can be done in registers alone, without using main memory. This applies to Ada, but also applies to standard C, for example, although some people use non-standard C features (compiler options) to negate this.

However, when comparing the "speed" of two languages and their two implementations I think that the implementations usually matter more than the languages.

*From: Dmitry A. Kazakov
<mailbox@dmitry-kazakov.de>
Date: Thu, 8 Jun 2023 10:00:52 +0200*

On 2023-06-08 05:55, Nasser M. Abbasi wrote:

> "I'm not an expert, but my feeling is that Rust is a "safer" language, which to me means it must be slower."

I think the comparison is misplaced. Julia is an interpreted language, very slow, on par with Python. It has memory mapped arrays like Ada does, but lacks Python's precompiled modules. The syntax is wonderfully arbitrary and unpredictable...

If safety is prevention of logical errors (bugs) you and your team and people deploying the software could make, then techniques and processes determine the outcome. The language can only support certain techniques. Of these techniques and processes some may require run-time overhead. When people compare languages, they frequently do programming techniques instead. As it was observed many decades ago:

"Besides, the determined Real Programmer can write Fortran programs in any language."

And finally, if you are determined to use some technique, then lack of language support makes the language less safe. E.g. if you are in some agile programming league then semantic constraints imposed by Ada would make things only worse.

Even Brainf*ck might be the safest language under circumstances... (-:-)

*From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Thu, 8 Jun 2023 10:50:44 +0200*

On 2023-06-08 05:55, Nasser M. Abbasi wrote:

> Since Ada is known to be one of the safest languages, do others here feel there is any truth to this?

Equivalent programs in compiled, non-GC languages have equivalent execution times. Robert Dewar famously had a set of equivalent Ada and C programs that produced identical machine code when compiled with gcc. So this is false.

The problem is getting equivalent programs. If the safe language includes run-time checks, then equivalent checks must be manually added to the unsafe language. Ada.Text_IO is not equivalent to C's I/O facilities. And so on.

One consequence of this is that both programs will be equally correct. What is usually compared is a correct (run-time checks) program in the safe language to an incorrect (no run-time checks) program in the unsafe language.

About optimization, Tartan made its living selling highly optimizing C compilers for TI chips, which came with a free C compiler. They also made highly optimizing Ada compilers, which did a better job of optimization than their C compilers. This was documented in

C vs Ada: arguing performance religion

(<https://dl.acm.org/doi/10.1145/216578.216583>)

which discusses four advantages Ada (83) has over C for optimization.

See also

Ada Outperforms Assembly: A Case Study

(<https://www2.seas.gwu.edu/~adagroup/sigada-website/lawlis.html>)

TI bought Tartan and sold its Ada compilers to DDC-I.

Ada Practice

Working around -ffreestanding Limitations

*From: Hou Van Boere
<houvanboere@gmail.com>
Subject: Working around -freestanding limitations?
Date: Sat, 1 Apr 2023 05:26:37 -0700
Newsgroups: comp.lang.ada*

Hi Everyone.

I know there are several floss RTOS options for us but I don't really need all of the support they offer and they just make things more complex.

Here are my goals:

1)I want to build my own circuit board with a microprocessor not microcontroller.

2)I want to run with gcc/gnatmake ... -freestanding

3)I only need the Ada 83 subset, which I guess is pretty close to Ravenscar

What options do I have? I like to keep things small and simple when possible.

Thanks for reading

*From: Simon Wright
<simon@pushface.org>
Date: Sat, 01 Apr 2023 14:35:23 +0100*

Hou Van Boere
<houvanboere@gmail.com> writes:

> 2)I want to run with gcc/gnatmake ... -freestanding

-freestanding isn't an option for gnatmake; where does it come from?

> 3)I only need the Ada 83 subset, which I guess is pretty close to Ravenscaler

The Ada 83 subset is going to be larger than Ravenscar.

If you don't want an RTOS you could use one of the light runtimes, e.g. light-cortex-m0.

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 1 Apr 2023 09:12:54 -0700*

Could you tell me where to find the light runtimes? I have only worked with Ada on full desktops. Does the FSF version ship with light runtimes?

*From: Niklas Holsti
<niklas.holsti@tidorum.invalid>
Date: Sat, 1 Apr 2023 19:14:28 +0300*

Certainly using most RTOS from Ada is more complex than using an Ada RTS from Ada.

Do you want to use tasking at all? Or just a single thread?

> 1)I want to build my own circuit board with a microprocessor not microcontroller.

Can you explain why? Input/output is often more complex with a microprocessor (I assume you mean something that could run a PC or a tablet) than with a microcontroller. A microprocessor may need a lot of complex initialization and driver SW which you can get in some RTOS but not in an Ada RTS. And I believe that circuit-board design is more complex for microprocessors than for microcontrollers, however I have no experience with either case.

> 3)I only need the Ada 83 subset, which I guess is pretty close to Ravenscaler

I see the Ada 83 tasking features as almost orthogonal to Ravenscar. Ada 83 has no protected objects, and all inter-task

communication must be done with rendez-vous using task entries. Ravenscar forbids task entries and rendez-vous and substitutes protected objects. Both work, but Ravenscar is perhaps more resistant to deadlock errors.

*From: Drpi <314@drpi.fr>
Date: Sat, 1 Apr 2023 18:29:45 +0200*

Le 01/04/2023 à 18:12, Hou Van Boere a écrit :

> "-freestanding isn't an option for gnatmake; where does it come from?"

I guess it is -ffreestanding :

<https://stackoverflow.com/questions/17692428/what-is-ffreestanding-option-in-gcc>

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 1 Apr 2023 09:42:26 -0700*

Thanks for your feedback Niklas. I am new to Ravenscar (just spelled it wrong today), this is very helpful.

It would be nice to have threads but I am not sure I actually need them.

I have serviced scientific instruments for 24 years now. I want to start fabricating them. I will have some bumps along the way with PCB design but I am confident that it will work out.

I have been playing around with Ada since 2012 but I still have lots to learn and I don't program in the day so it is not my strong suit. The hardware side should work out but I am worried about the software end of things. Ada is lovely but massive. There are so many features, so many libraries (some of which are abandoned). There are only so many hours in a day.

I have a subset of the language I like and if I can just control CPU address and data lines, I shouldn't need a RTOS. Trying several of them out could take a great deal of time.

I don't seem to have any extra runtimes with my install:

gnatls -v

GNATLS 11.2.0

Copyright (C) 1997-2021, Free Software Foundation, Inc.

Source Search Path:

<Current_Directory>

/usr/lib64/gcc/x86_64-slackware-linux/11.2.0/adainclude

Object Search Path:

<Current_Directory>

/usr/lib64/gcc/x86_64-slackware-linux/11.2.0/adalib

Project Search Path:

<Current_Directory>

/usr/x86_64-slackware-linux/lib/gnat

/usr/x86_64-slackware-linux/share/gpr

/usr/share/gpr

/usr/lib/gnat

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 1 Apr 2023 09:50:28 -0700*

"Could you tell me where to find the light runtimes?"

I am just answering my own question to avoid wasting people's time. I found this:

<https://github.com/AdaCore/bb-runtimes>

*From: Drpi <314@drpi.fr>
Date: Sat, 1 Apr 2023 18:54:20 +0200*
> 1) I want to build my own circuit board with a microprocessor not microcontroller.

Nowadays, microprocessors are rare. Even x86 microprocessors could be named microcontrollers since they integrate many (not all) peripherals.

High end microcontrollers are very complex to initialize. Especially since they integrate security functionalities (like secure boot), SDRAM controllers, PCIe controllers, Gigabit Ethernet controllers, 3D GPU, video encoders/decoders, camera interface, LCD interface, HDMI interface...

Even middle range microcontrollers are (very) complex.

Manufacturers provide drivers source code (in C) for all peripherals. They also provide tools to graphically set chip configuration and output C code to help the programmer.

Complexity depends on the chip you choose.

*From: Drpi <314@drpi.fr>
Date: Sat, 1 Apr 2023 18:55:36 +0200*

Le 01/04/2023 à 18:50, Hou Van Boere a écrit :

> I am just answering my own question to avoid wasting people's time. I found this:

> <https://github.com/AdaCore/bb-runtimes>

The best way is to use Alire
<https://alire.ada.dev/>

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 1 Apr 2023 09:58:49 -0700*

Thanks DrPi

I will probably stick with what I know. Most of the instruments I work on have Motorola chips and parallel buses. I don't think I will use SPI, IC2 or dozens of other protocols/features found in most modern circuit boards.

From: Drpi <314@drpi.fr>

Date: Sat, 1 Apr 2023 18:59:42 +0200

Le 01/04/2023 à 18:42, Hou Van Boere a écrit :

> I don't seem to have any extra runtimes with my install:

Today, the easiest route is to use ARM based chips as there are maintained runtimes for them (through Alire and bbruntimes).

*From: Drpi <314@drpi.fr>
Date: Sat, 1 Apr 2023 19:02:15 +0200*

Le 01/04/2023 à 18:58, Hou Van Boere a écrit :

> I will probably stick with what I know. Most of the instruments I work on have Motorola chips

Great chips at their time but I'm afraid you'll have hard time compiling a dedicated GNAT compiler.

*From: Hou Van Boere
<houvanboere@gmail.com>
Date: Sat, 1 Apr 2023 10:24:28 -0700*

I am sure you are right but still, you get the general idea.

Thermo Electron has pretty much bought most of the industry out. I will copy and paste, mix and match old stuff to re-implement instruments they don't care about anymore. I don't need to make anything cutting edge. The old stuff was more than good enough

*From: Drpi <314@drpi.fr>
Date: Sat, 1 Apr 2023 20:33:36 +0200*

Le 01/04/2023 à 19:24, Hou Van Boere a écrit :

> The old stuff was more than good enough

Indeed, an interesting project.

You first need an Ada cross-compiler. Here is a link about this :
https://wiki.osdev.org/GNAT_Cross-Compiler

You also need a runtime. This is up to you to code it. You can use bbruntimes as a template. This can request modifications on your hardware. For example, the runtime needs a timer to track time. If your microprocessor does not have an embedded timer, you'll have to add one on your board.

Other links of interest :

<https://forum.ada-lang.io/>

<https://github.com/ohenley/awesome-ada>

Matrix rooms (<https://matrix.org/clients/>) :

Ada news : <https://matrix.to/#/ada-lang:matrix.org>

Ada language : <https://matrix.to/#/ada-lang:matrix.org>

Alire : https://matrix.to/#/ada-lang_Alire:gitter.im

Many other resources exist.

From: philip...@gmail.com
<philip.munts@gmail.com>
Date: Wed, 5 Apr 2023 09:21:44 -0700

I would suggest you look at my Linux Simple I/O Library:
<https://github.com/pmunts/libsimpleio>

The Ada binding makes it pretty easy to build test fixtures, control devices, etc. I even used an Ada program to replace a multizone sprinkler controller.

Next, take a look at MuntsOS Embedded Linux:
<https://github.com/pmunts/muntsos>

Together, they make it possible to replace many microcontroller applications with a Raspberry Pi or a BeagleBone or a PocketBeagle. With the Raspberry Pi family, it is very easy to fabricate custom boards using a Raspberry Pi Zero, CM3, or CM4 (least to most complex) as a CPU module. If you run Raspberry Pi OS instead of MuntsOS, it is even self hosting.

Currently I don't have any support for IEEE-488, though I have a USB interface and an old CalComp plotter on the shelf I've been meaning to play around with.

I'll be teaching a workshop at AdaEurope 2023 in Lisbon in June showing how all this works.

Constancy of X'Address

From: Niklas Holsti
<niklas.holsti@tidorum.invalid>
Subject: Constancy of X'Address?
Date: Wed, 5 Apr 2023 12:24:39 +0300
Newsgroups: comp.lang.ada

A discussion in comp.arch (on the new C23 standard for C) brought up these questions, which I could not answer with confidence:

- Is the address of an object constant in Ada? That is, if I have some object X in an Ada program, do repeated applications of X'Address always return the same value?
- Does the answer depend on how X is allocated (created): on the library level, on the stack, or in a pool ("new")?

The issue behind this question is whether an Ada program could use garbage collection that moves objects around, for example a compacting collector.

From: Maxim Reznik
<reznikmm@gmail.com>
Date: Fri, 7 Apr 2023 10:04:24 -0700

If the type of the object is limited, then the object address is a constant. For other objects there is no such guarantee I guess.

From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Fri, 7 Apr 2023 19:19:14 +0200

'Address is discussed in ARM 13.3 (http://www.ada-auth.org/standards/aarm12_w_tc1/html/A-A-13-3.html). (12.c) says "The validity of a given address depends on the run-time model; thus, in order to use Address clauses correctly, one needs intimate knowledge of the run-time model."

Under Implementation Advice, (15-16) say "The recommended level of support for the Address attribute is:

"X'Address should produce a useful result if X is an object that is aliased or of a by-reference type, or is an entity whose Address has been specified."

There is nothing specific about whether the value can change.

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Sat, 8 Apr 2023 04:03:02 -0500

I agree with the other answers (for the most part). Not much is guaranteed about 'Address. But by-reference and aliased objects have to *work* like they are not moved. If the runtime can deal with moving such things, Ada doesn't care.

As a practical matter, most Ada implementations assume objects don't move. Janus/Ada assigns everything at compile-time, so the only time anything moves is when it is created or destroyed.

The big problem with garbage collection in Ada is that early finalization is not allowed (other than a few tiny exceptions in failure cases and [in post-Ada 22] certain function results.) So any object that might have a controlled part can never be garbage collected, even if there is no other use or access to it.

Changing that is a very hard problem, as you cannot allow finalization to happen at any instant or by any arbitrary task (if you did, every finalization would be a race scenario, and every Finalize routine would need dedicated locking). I've suggested allowing it for "unreachable objects" (not a useful definition by itself, it would need to be defined) at places where masters are being exited anyway (so finalization should be expected at those locations). But it's unclear if you can build a useful garbage collector that way (and what the overhead would be).

Contracts in Generic Formal Subprogram

From: Mockturtle
<framefritti@gmail.com>
Subject: Contracts in generic formal subprogram
Date: Sat, 8 Apr 2023 00:00:38 -0700
Newsgroups: comp.lang.ada

Dear.all,

this is something that looked like a natural and nice idea to me, but the compiler disagreed :-): specifying contracts in formal subprograms in generic declarations. Actually, RM 12.6 does not prohibit this on a syntactic level (an aspect_specification part is included), but the compiler complains.

To understand what I mean, please check the following real code toy-zed (can you hear the grammar screaming?)

```
-----
generic
  type Ring is private;
  with function Divides (Num, Den : Ring)
    return Boolean;
  with function Is_Invertible (X : Ring)
    return Boolean;
  with function Inv (X : Ring) return Ring
  with Pre => Is_Invertible (X);

  with function Gcd (X, Y : Ring)
    return Ring
  with Post => Divides (X, Gcd'Result)
  and Divides (Y, Gcd'Result);
package Pippo is
  -- stuff
end Pippo;
-----
```

The meaning I have in mind is something like

* For "Pre" aspect: whoever writes the function Inv can assume that X is invertible since Inv will never be called (by the package code, at least) with X not invertible. Also Inv cannot have a stricter pre-condition. In a sense, the package expects Inv to work correctly if and only if the pre-condition is true.

* For "Post" aspect: I expect that the result of GCD satisfies the post condition. Post conditions for the actual subprogram can be stricter, as long as the post condition of the formal parameter is satisfied. For example, if Ring is Integer, GCD could always return a positive value that divides both X and Y. The fact that the result is positive does not hurt.

Should the actual subprogram specify the same contract? I am not sure (and I guess this could be a stumbling block for the adoption of this idea). One could say that the actual subprogram gets a contract that is the AND of the actual subprogram and the contract specified in the generic declaration, it is up to the programmer to check that they are compatible. I guess the compatibility could be verified by the compiler itself in simple cases, but I expect that this could not be feasible in some cases (maybe of academic interest?).

From: Dmitry A. Kazakov
<mailbox@dmitry-kazakov.de>
Date: Sat, 8 Apr 2023 10:02:31 +0200

On 2023-04-08 09:00, mockturtle wrote:

> Should the actual subprogram specify the same contract? I am not sure (and I guess this could be a stumbling block for the adoption of this idea).

The general principle of substitutability is that the preconditions can be weakened, the postconditions can be strengthened.

From: Randy Brukardt
<randy@rrsoftware.com>

Date: Sat, 8 Apr 2023 04:09:38 -0500

Ada 2022 allows such contracts; Ada 2012 did not. (See 6.1.1, and specifically 6.1.1(1/5)). Whether your compiler has caught up, who knows.

Logically the contracts should "match" (with the weakening/strengthening that Dmitry mentioned), but that was too hard for Ada, so they're just additive. (A proper matching mechanism is more the sort of thing that SPARK does, Ada only enforces these contracts at runtime) That is, when you call through a generic formal subprogram, you enforce the preconditions of both the formal and the actual subprogram, and similarly for the postconditions. If they mismatch, you might not be able to make a successful call. If it hurts, don't do that. ;-)

From: Simon Wright
<simon@pushface.org>

Date: Sat, 08 Apr 2023 17:48:11 +0100

GCC 12.2.0 accepts this code with -gnat2022.

From: Mockturtle
<framefritti@gmail.com>

Date: Sat, 8 Apr 2023 10:27:16 -0700

On Saturday, April 8, 2023 at 6:48:14 PM UTC+2, Simon Wright wrote:

> GCC 12.2.0 accepts this code with -gnat2022.

True! Cool... In my opinion, contracts are among the coolest (and maybe more exclusive) features of Ada

From: G.B.
<bauhaus@notmyhomepage.invalid>

Date: Tue, 11 Apr 2023 07:56:45 +0200

On 08.04.23 10:02, Dmitry A. Kazakov wrote:

> The general principle of substitutability is that the preconditions can be weakened, the postconditions can be strengthened.

Side track: "weak" and "strong" alone sounding like a valuation to the uninitiated, but neither technical nor precise; and the "objects" of comparison of sets of conditions being implicit; and the ARM not defining a technical term for these adjectives unless weak ordering helps.

If these adjectives induce confusion, can they be avoided? E.g., by instead mentioning the sets of Pre- and Post-

conditions of those actual/formal/overriding subprograms. I guess that super- and subset relations will permit helpfully defining an ordering to be understood (in general, if not in the ARM).

From: Dmitry A. Kazakov
<mailbox@dmitry-kazakov.de>

Date: Tue, 11 Apr 2023 14:03:27 +0200

On 2023-04-11 07:56, G.B. wrote:

> Side track: "weak" and "strong" alone sounding like a valuation to the uninitiated [...]

The formal meaning of weaker/stronger relation on predicates P and Q:

weaker $P \Rightarrow Q$

stronger $Q \Rightarrow P$

The formal rationale is that if you have a proof

$P1 \Rightarrow P2 \Rightarrow P3$

Then weakening $P1$ to $P1' \Rightarrow P1$ and strengthening $P3 \Rightarrow P3'$ keeps it:

$P1' \Rightarrow P2 \Rightarrow P3'$

As for ARM.

Regarding dynamic checks all the above is irrelevant because dynamic checks are no contracts. Furthermore, since the proper contracts include Constraint_Error or Storage_Error raised, do you really care how are you going to bomb your program while keeping proper contracts? (-:) Sincere advice: forget about this mess.

For static checks a proof of implication is rather straightforward since we assume that all static predicates must be decidable anyway.

Of course, with generics you might run into troubles as you would have both proper contracts, i.e. the instantiated ones, and the generic ones expressed in generic terms. Instantiated contracts are easy to check, but what one would actually wish is checking generic contracts, which might turn out to be impossible. The glimpse of the problem is what any Ada programmer knows: generic instantiations may fail to compile even if the actual parameters match...

From: Spiros Bousbouras
<spibou@gmail.com>

Date: Wed, 12 Apr 2023 02:18:45 -0000

On Tue, 11 Apr 2023 14:03:27 +0200

"Dmitry A. Kazakov" <mailbox@dmitry-kazakov.de> wrote:

> $P1' \Rightarrow P2 \Rightarrow P3'$

You have it backwards ; if $P1$ implies P then $P1$ is stronger than $P1$.

From: Niklas Holsti
<niklas.holsti@tdorun.invalid>

Date: Wed, 12 Apr 2023 09:49:35 +0300

Speaking of logic in general, rather than Ada contracts in particular, I would say that you got it right, and Dmitry did not.

Suppose we have a theorem about geometrical figures F , and at first we can prove the theorem only if we assume (precondition) that the figure F is a square. Later we manage to improve the proof so that it holds also for rectangles. I would say, and I think mathematicians would say, that we /weakened/ the assumptions from " F is a square" to " F is a rectangle", and indeed the former (stronger) implies the latter (weaker), which is not as Dmitry defined "stronger".

From: Dmitry A. Kazakov
<mailbox@dmitry-kazakov.de>

Date: Thu, 13 Apr 2023 08:27:30 +0200

On 2023-04-12 04:18, Spiros Bousbouras wrote:

> On Tue, 11 Apr 2023 14:03:27 +0200

> "Dmitry A. Kazakov"
<mailbox@dmitry-kazakov.de> wrote:

>> $P1' \Rightarrow P2 \Rightarrow P3'$

>

> You have it backwards ; if $P1'$ implies $P1$ then $P1'$ is stronger than $P1$.

Yes, you are right. Inclusion is an inverse of implication.

A weaker predicate is true on a set that contains the set where the stronger predicate is.

Looking for Feedback: ISO 3166-1 Country Code Reference in Ada

From: A.J. <ianozia@gmail.com>
Subject: Looking for feedback: ISO 3166-1 country Country Code Reference in Ada
Date: Sat, 15 Apr 2023 11:52:08 -0700
Newsgroups: comp.lang.ada

I just created a library for accessing ISO 3166-1 records in Ada compatible with Ada.Locales. Before I try to publish it to Alire, I'm hoping to get some feedback if anyone has some. It's possible that feedback will result in the function calls, naming convention, or structure being set up differently, so please let me know what you think.

https://github.com/AJ-Ianozi/iso_3166

I also posted this on the subreddit, so apologies for any redundancy for those viewing both!

From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>

Date: Mon, 17 Apr 2023 11:36:53 +0200

Some initial thoughts on what you have:

It seems likely that your clients will use the alpha codes for input and display. It will be more convenient for that if the

alpha codes are subtypes of String rather than distinct types.

Since you have already enumerated all 250 possible alpha codes, your predicates could look like

```
subtype Alpha_Code_2 is String (1 .. 2)
  with Dynamic_Predicate => Alpha_Code_2
    in "AF" | "AL" | ...;
```

and similar for the 3-letter codes.

Since you have already enumerated all 250 possible numeric codes, you could use a restricted range for your numeric (sub)type, with a predicate restricting it to valid values.

These use the language to do validity checking for you.

Regarding the design of such a pkg, my initial instinct was to use enumeration types for the alpha codes, but a little investigation shows that some of the codes are Ada reserved words, so that doesn't work. So I would stick with the String subtypes and provide functions such that, given one of the values, the client can obtain the others, as well as the name. Alternatively, one could have functions to return a record such as you provide. Which is preferable depends on how such a pkg is typically used.

There are various possible implementations, with different tradeoffs.

2023 Stack Overflow: Ada in the Programming Options for the First Time

From: Fabien Chouteau
<fabien.chouteau@gmail.com>

Subject: 2023 Stack Overflow: Ada in the programming options for the first time
Date: Tue, 9 May 2023 02:39:56 -0700
Newsgroups: comp.lang.ada

The 2023 Stack Overflow survey is live: <https://stackoverflow.blog/2023/05/08/the-2023-developer-survey-is-now-live/>

And for the first time Ada is listed in the options for "programming, scripting, and markup languages"!

Don't hesitate to fill the survey and show that the Ada community is alive.

Ada 2022 at Last!

From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Subject: Ada 23 at Last!
Date: Wed, 10 May 2023 11:45:01 +0200
Newsgroups: comp.lang.ada

<https://www.iso.org/standard/83621.html>

From: Nasser M. Abbasi
<nma@I2000.org>
Date: Wed, 10 May 2023 09:27:18 -0500

Is there a site that gives summary of new features/changes/improvements in Ada 2023?

From: Adamagica <christ-usch.grein@t-online.de>

Date: Wed, 10 May 2023 07:32:25 -0700

Yes, Ada now has an ARM and a LEG (language enhancement guide), see:

<http://www.ada-auth.org/standards/overview22.html>

Ada Monthly Meeting Proposal

[see also "Ada Monthly Meetup 2023" in this AUJ issue, p.98 —arm]

From: Fernando Oleo Blanco

<irvise_ml@irvise.xyz>

Subject: Re: Ada Monthly Meeting proposal

Date: Wed, 10 May 2023 17:39:45 +0200

Newsgroups: comp.lang.ada

* Reboot of the Ada Monthly Meeting

Dear all. Once again, after a long pause, I want to revive the idea of a monthly meeting to discuss the latest Ada events, projects, releases or just have a chat about a topic.

I will not repeat what I said in the original message as all points still stand.

I was happy with the reception that the proposal gathered, alas it did not take place. However, I was thinking about having one at the beginning of each month. There would be a pause during summer (August for most people and potentially September) and FOSDEM.

* When do "we" start?

As I would not like to postpone it much more, I would like to kickstart it this June. So the first one would be either Saturday 3 or Sunday 4 of July.

I know this sounds a bit rushed. However, if I do not set a date for me and other people, we will just keep pushing it further and further. This first meetup would just be to test the waters and receive feedback. There would be another one in July and then summer, after which I hope to get a serious and continuous stream of meetups.

I was thinking that we could have a meetup at around 12PM UTC time. It is early but not crazy early for those in the USA and late for those in far east Asia such as Australia. Here in Europe it falls close to the meal time, which is not ideal... If a lot of people do not like this time, it can be easily moved a bit earlier or later... I WOULD LIKE TO RECEIVE SOME FEEDBACK ON THIS.

* What to expect?

I would like to keep these meetups sweet and short. I was thinking maybe 45 minutes long, maybe an hour. That would

allow for a quick round of news, topics and introductions (something like what Maxim Reznik does but a lot shorter). Then 2 to 4 topics (depending on the time needed by each one). The topics would be what other and I already proposed. This would give between 20 to 10 minutes for each topic.

Once again, this is the starting proposal. Adjustments will be needed.

* What do I need?

Feedback:

- What is your opinion?
- Do you have a topic/project that you would like to show to the community?
- Do you like the chosen time?
- Do you like the week of the month?
- Is Jitsi [1] a good enough platform to do the meetings?
- Do you think that 45 min / 1 h is a good enough duration?
- Would you like to participate on Saturday 3 or Sunday 4?

[1] <https://meet.jit.si/>

From: Simon Wright

<simon@pushface.org>

Date: Wed, 10 May 2023 20:26:36 +0100

Fernando Oleo Blanco
<irvise_ml@irvise.xyz> writes:

> - What is your opinion?

Good idea.

> - Do you have a topic/project that you would like to show to the community?

Will have to think about that! Mac issues? Alire vs Mac?

> - Do you like the chosen time?

Fine by me.

> - Do you like the week of the month?

No problem

> - Is Jitsi [1] a good enough platform to do the meetings?

Will have to see!

> - Do you think that 45 min / 1 h is a good enough duration?

Certainly OK for the first meeting

> - Would you like to participate on Saturday 3 or Sunday 4?

Either could be managed! Slight preference for Saturday

From: Jeffrey R. Carter

<spam.jrcarter.not@spam.acm.org.not>

Date: Wed, 10 May 2023 21:41:33 +0200

On 2023-05-10 17:39, Fernando Oleo Blanco wrote:

> - Do you like the chosen time?

Any time will be inconvenient for some, but one must be chosen. 12:00 UTC is fine with me, but during January, the time on the west coast of the US is UTC -08, and on the east coast of Australia, UTC +11. 12:00 UTC corresponds to 04:00 in California and 23:00 in Sydney. 04:00 is rather painful. It might be better to choose 13:00 UTC (05:00 and 00:00).

*From: Ben Bacarisse
<ben.usenet@bsb.me.uk>
Date: Wed, 10 May 2023 20:47:02 +0100*

Fernando Oleo Blanco
<irvise_ml@irvise.xyz> writes:
> * When do "we" start?

Presumably 3rd or 4th June.

*From: frances...@gmail.com
<francesc.rocher@gmail.com>
Date: Wed, 10 May 2023 23:55:48 -0700*

El dia dimecres, 10 de maig de 2023 a les 17:39:50 UTC+2, Fernando Oleo Blanco va escriure:

> - What is your opinion?

Great initiative!

> - Do you have a topic/project that you would like to show to the community?

Not yet, but for sure I'd like to show a couple of projects I'm working on.

> - Do you like the chosen time?

No problem.

> - Do you like the week of the month?

Good enough, easy to remember.

> - Is Jitsi [1] a good enough platform to do the meetings?

Let's see how it works.

> - Do you think that 45 min / 1 h is a good enough duration?

It could be flexible as it depends on the schedule and Q&A, so let's see.

> - Would you like to participate on Saturday 3 or Sunday 4?

Both are ok, but preferably Saturday.

Thanks Fernando for leading this proposal,

*From: amo...@unizar.es
<amosteo@unizar.es>
Date: Thu, 11 May 2023 04:05:44 -0700*

On Wed, May 10, 2023 at 5:39 PM 'Fernando Oleo Blanco' via comp.lang.ada <comp.lang.ada@googlegroups.com> wrote:

- What is your opinion?

Great initiative!

- Do you have a topic/project that you would like to show to the community?

Not right now, happy to just meet people.

- Do you like the chosen time?

Works for me.

- Do you like the week of the month?

No opinion.

- Is Jitsi [1] a good enough platform to do the meetings?

It's worked for me in the past.

- Do you think that 45 min / 1 h is a good enough duration?

Yes, no more than that.

- Would you like to participate on Saturday 3 or Sunday 4?

For this instance, I can only on the 3rd. It should be indifferent normally.

Thanks Fer for leading.

*From: A.J. <ianozia@gmail.com>
Date: Mon, 15 May 2023 18:19:07 -0700*
> Feedback:

> - What is your opinion?

I'm absolutely up for this.

> - Do you have a topic/project that you would like to show to the community?

I recently released an Ada ISO Library for country and currency codes[1], I could talk about that if anyone is interested. I also use Ada with Alire on a mac, so I'm interested in listening to that discussion.

> - Do you like the chosen time?

It looks like 12pm UTC is 8am EDT. I normally get up around 6AM, so I can make this work.

> - Do you like the week of the month?

That should be fine.

> - Is Jitsi [1] a good enough platform to do the meetings?

If it works in a browser, I have no issues with it.

> - Do you think that 45 min / 1 h is a good enough duration?

This is good for the first such meeting. We can see how it goes and adjust in later meetings.

> - Would you like to participate on Saturday 3 or Sunday 4?

I prefer Saturday over Sunday, but either one works for me.

[1] https://github.com/ada-iso/ada_iso/tree/v2.0.0

*From: Fernando Oleo Blanco
<irvise_ml@irvise.xyz>
Date: Thu, 25 May 2023 19:35:12 +0200*

Hi all and especially A.J.

I will try to make the meeting happen. I made the announcement here [1].

If you would like to participate, save the date! If you have any ideas or proposals, they are welcome (this goes specially to

you A.J., I assume you would like to present :).

There is a bit more info in the link if anybody else is interested. I will select the exact time next week.

[1] <https://forum.ada-lang.io/t/ada-monthly-meeting/384/2?u=irvise>

*From: Keith Thompson
<keith.s.thompson+u@gmail.com>
Date: Thu, 25 May 2023 13:22:56 -0700*

The proposed time is 12:00 or 13:00 UTC on Sat 2023-06-03.

That's 05:00 or 06:00 in the US Pacific time zone (California et al).

I understand that scheduling meetings for an international audience is hard. I might join if it's later in the day in my time zone -- but I wouldn't have much to contribute anyway, so please don't base your decision on that. But I suspect a lot of people in the US won't join if it's that early.

Is a Boolean Type Inherently Atomic?

*From: Rod Kay <rodakay5@gmail.com>
Subject: Is a Boolean type inherently atomic?*

*Date: Fri, 12 May 2023 22:17:48 +1000
Newsgroups: comp.lang.ada*

Surely only the least significant bit of the least significant byte is relevant and so the value cannot be garbled by one task writing and another reading at the same time ?

*From: Jeffrey R. Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Fri, 12 May 2023 14:53:24 +0200*

Boolean types with other representations using multiple bits are possible, so your assumption doesn't hold.

*From: J-P. Rosen <rosen@adalog.fr>
Date: Fri, 12 May 2023 18:56:54 +0200*

Le 12/05/2023 à 14:53, Jeffrey R.Carter a écrit :

> Boolean types with other representations using multiple bits are possible, so your assumption doesn't hold.

True, especially considering the special exception for boolean types in 13.4(8)

Anyway, if you intend to access a variable from multiple tasks, it doesn't cost much to add an Atomic aspect to the declaration, at least to inform the reader!

*From: Niklas Holsti
<niklas.holsti@tidorum.invalid>
Date: Fri, 12 May 2023 20:38:29 +0300*

On 2023-05-12 15:17, Rod Kay wrote:

> Surely only the least significant bit of the least significant byte is relevant and so the value cannot be garbled by one

task writing and another reading at the same time ?

That seems very likely indeed, unless (as others have commented) the representation has been specified to use more bits. However, the Ada RM states in C.6(8/3) that "every atomic type or object is also defined to be volatile", and of course Boolean variables are not considered volatile unless they are specified to be Atomic or Volatile. So a Boolean type is not inherently atomic in the Ada RM sense of "atomic".

And of course if you use a shared variable to communicate data between tasks, that variable should be marked as Volatile, and there should also be some Atomic

accesses to ensure that actions are "sequential", so marking the variable as Atomic is best.

From: Adamagica <christ-usch.grein@t-online.de>

Date: Fri, 12 May 2023 11:02:15 -0700

AARM 3.5.3(1.a), 13.4(8.b, 10/5) has some information about boolean representations.

Ada in Jest

Doggerel

From: Rod Kay <rodakay5@gmail.com>

Subject: Doggerel

Date: Sat, 3 Jun 2023 12:33:30 +1000

Newsgroups: comp.lang.ada

I've been holding off posting this for fear of rotten tomatoes ... but here goes ...

"Tis no uncertain adage,

That that balmy beggar Babbage,

Was to antsy Aunty Ada,

No uncertain ennui saviour!"

... just putting on my hazmat suit now, so fire away :).