# H A C

## HAC Ada Compiler

From an abandoned teaching project to a usable script-like Ada tool

G. de Montmollin, Ada-Europe 2022

# **HAC**'s history

**1990's**: Looking for a small, quick Ada compiler. Found SmallAda, an abandoned project with an operational compiler, including a programmable type system (arrays, records), nested subprograms, tasks.

**1999**: First attempt to translate SmallAda from Pascal to Ada, using **P2Ada**.

**2009**: Improved **P2Ada** to translate the full SmallAda sources.

**2013**: January 24th: **Day Zero** of **HAC**. Hello World, Fibonacci, sorting demos and few other tests work!

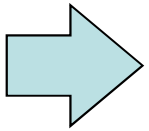
**2020**: After a long sleep, the real development of **HAC** begins!

# **HAC**'s project goals

- **General motivation**: provide a simple, quick compiler
  - users: beginners, non IT specialists, or even IT specialists
  - focus: teaching, small programs such as text parsers, file converters, shell scripts launching applications, code generators, numerical simulations, …

- **Former goal**: make it work. Have an Ada compiler that produces "Hello world!" with a tiny object code size.

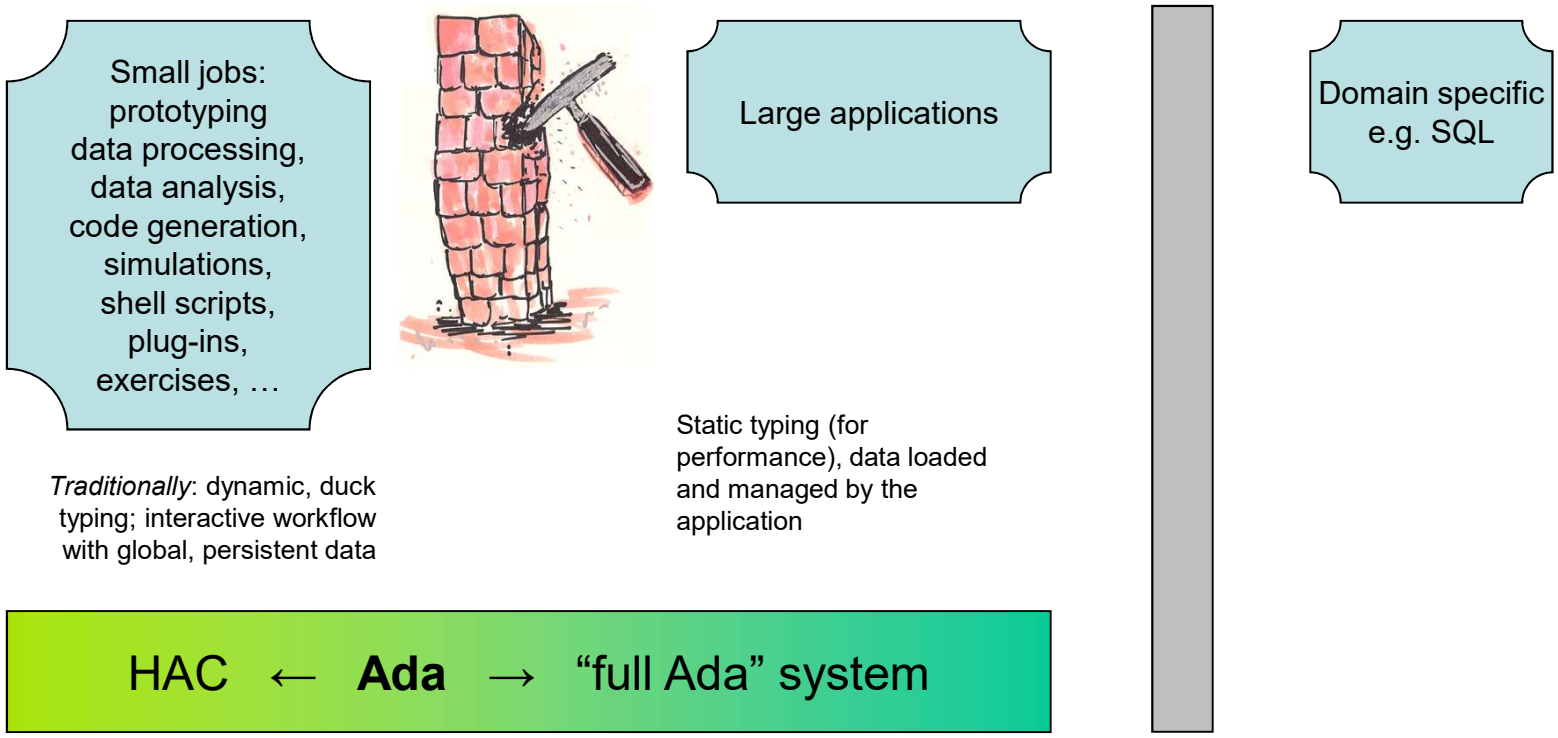
```
with HAL;
procedure Hello is
begin
  HAL.Put ("Hello world!");
end Hello;
```



```
Position: Opcode                          Approx source location
-------------------------------------------------
       0: K_PUSH_TWO_DISCRETE_LITERALS      4  Hello
       1: K_PUSH_TWO_DISCRETE_LITERALS      4  Hello
       2: K_FILE_I_O                        4  Hello; SP_PUT; "Hello world!"
       3: K_HALT_INTERPRETER                5  [-- The Universe --]
```


# **HAC**'s project goals

- **New goal**: break the language wall between the "script" world and the "compiled" world



# **HAC**'s development in 2020 .. 2022

- Ada compatible
- Exception trace-backs
- HAL (HAC Ada Library) package with lots of goodies
- Variable-length strings (type HAL.VString)
- Text files (type HAL.File_Type)
- Subtypes
- Range checks
- Modularity (library-level packages, procedures and functions)

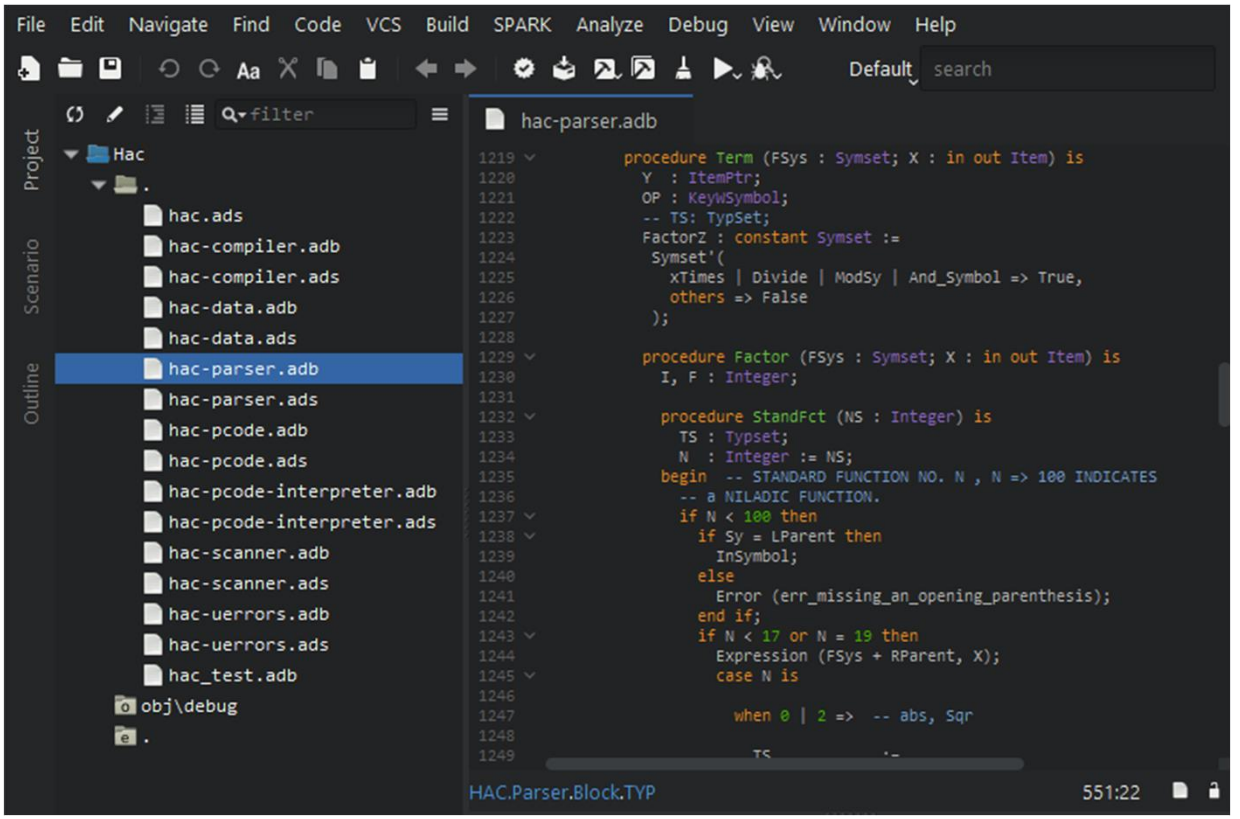
Behind the scenes:

- Regression test suite with 51 tests (36 thanks to Advent of Code)
- Compiler: global variables replaced by OO
- Parser, VM interpreter were split into child packages

# **HAC**'s development in 2020 .. 2022

Behind the scenes (continued):

Before (e.g. in 2014):

# **HAC**'s development in 2020 .. 2022

Behind the scenes (continued):

Even before (Pascal-S, 1973):

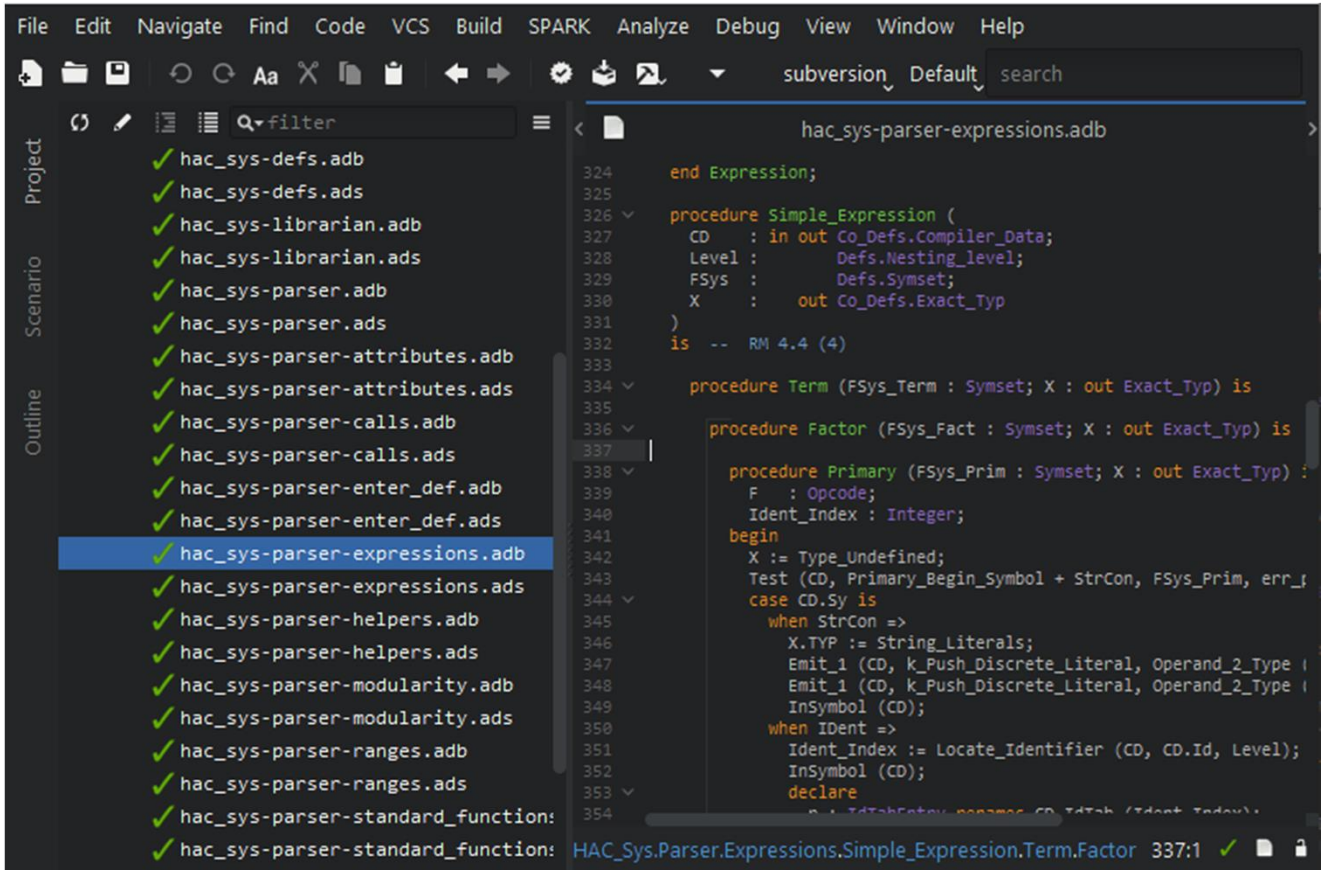



```
          var y:item; op:symbol;

          procedure term(fsys:symset; var x:item);
            var y:item; op:symbol; ts:typset;

            procedure factor(fsys:symset; var x:item);
              var i,f: integer;

              procedure standfct(n: integer);
                 var ts: typset;
              begin (*standard function no. n*)
                if sy = lparent then insymbol else error(9);
                if n < 17 then
                  begin expression(fsys+[rparent],x);
                    case n of
(*abs,sqr*)          0,2:  begin ts := [ints,reals];
                             tab[i].typ := x.typ;
                             if x.typ = reals then n := n+1
                           end ;
(*odd,chr*)          4,5:  ts := [ints];
(*ord*)              6:    ts := [ints,bools,chars];
(*succ,pred*)        7,8:  ts := [chars];
(*round,trunc*)      9,10,11,12,13,14,15,16:
```

# **HAC**'s development in 2020 .. 2022

Behind the scenes (continued):

Currently:

# **HAC**'s characteristics

- **Build time** of the full HAC compiler & VM interpreter, by GNAT: 7.8 seconds (i7-9700 CPU @ 3.00GHz, using 8 cores)

- **Build time** of test/floats.adb (346 lines), by HAC: 0.003 second; by GNAT: 1.2 second

- **System dependency**: none

- **Ada source input**: any stream (file, editor data, web stream, zip archive, …)

- **Target**: Virtual Machine (p-code with ad-hoc extensions)

# **HAC**'s future

- **More Ada features**
- **Interfacing (HAC ↔ native "full Ada")**
- **Built-in libraries:**
  - SQLite
  - Some portable graphical toolkit
  - Ada.*

# Where to find **HAC** ?

**HAC is free, open-source (MIT license)**

- **SourceForge:**
  - **Home page: https://hacadacompiler.sourceforge.io/**
  - **Project page: https://sourceforge.net/projects/hacadacompiler/**

- **Github: https://github.com/zertovitch/hac**

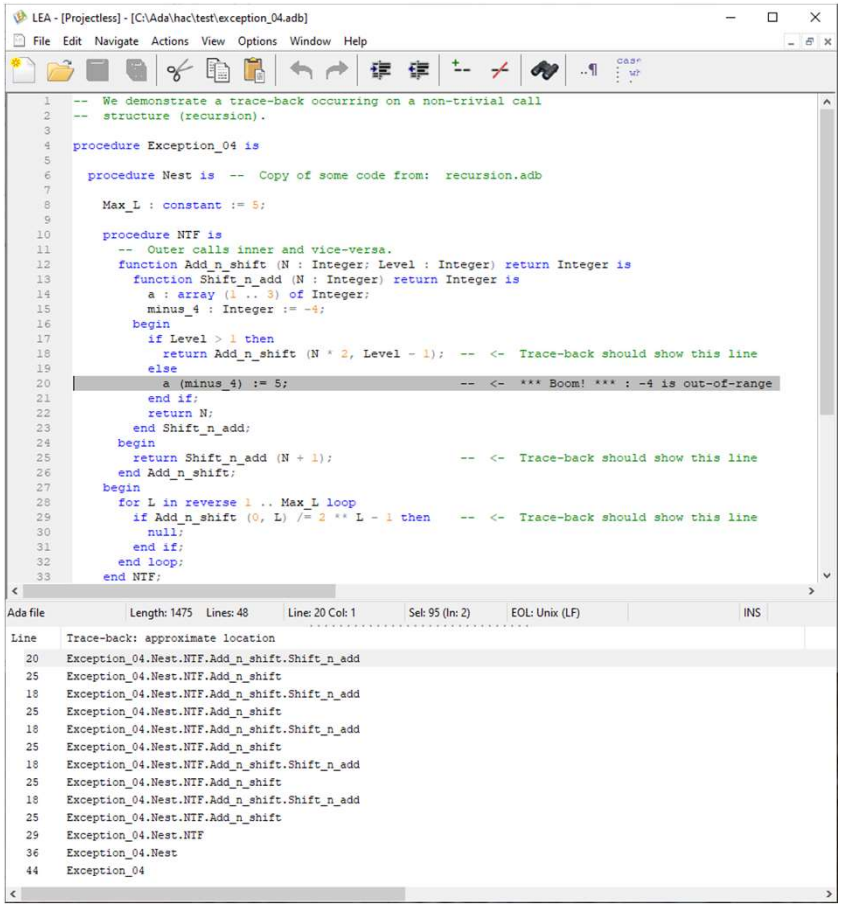
# Projects related to **HAC**

## LEA:

**L**ightweight **E**ditor for **A**da

- https://sourceforge.net/projects/l-e-a/
- https://github.com/zertovitch/lea

LEA embeds HAC.




## Pascal-to-Ada

- https://sourceforge.net/projects/p2ada/
- https://github.com/zertovitch/pascal-to-ada

# Thanks

- To Stéphane Rivière for the documentation, “real world” feedback (large production scripts) and numerous suggestions, especially for the HAL library.

- To users of HAC and LEA (Lightweight Editor for Ada ) for their feedback.

Drawings: Diane de Montmollin