Rigorous Methods for Software Engineering (F21RS-F20RS)
Getting Started with SPARK 2014

Andrew Ireland
Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh

Overview

- Context and a little history.
- Accessing SPARK toolkit and an Ada compiler – Linux and Windows.
- How to statically analyze, compile and execute SPARK code, e.g. *Hello World!*

A Brief History of Ada

- In response to increased software development and maintenance costs the US Defence Department in 1975 published a set of strict safety criteria for programming languages that were to be used in their systems.
- None of the available languages met the criteria so a competition was set-up to design a new language.
- The winner was Ada (1983) and became a ISO Standard.
- A major revision of the language was completed in 1995, giving rise to Ada 95: revision included object-oriented constructs, optional annexes, *e.g.* systems programming, real-time systems, distributed systems, security, ...
- The last major revision came with Ada 2012, where the language extensions included *aspects* and *pragmas*. These allow a programmer to write both executable statements and assertions, i.e. properties about the intended behaviour of your code that can be verified statically.

Ada – Some Applications

- Ada is the most commonly used language for Air Traffic Control Systems worldwide.
- Ada is used increasingly in commercial "fly-by-wire" aircraft such as the Boeing 777 and Airbus.
- Ada is used within advanced avionics, *i.e.* fighter jets, and many military command and control applications.
- Ada is used for embedded systems in nuclear power plants, rail transportation systems, industrial process control, ...

An Advert for SPARK like Programming Language

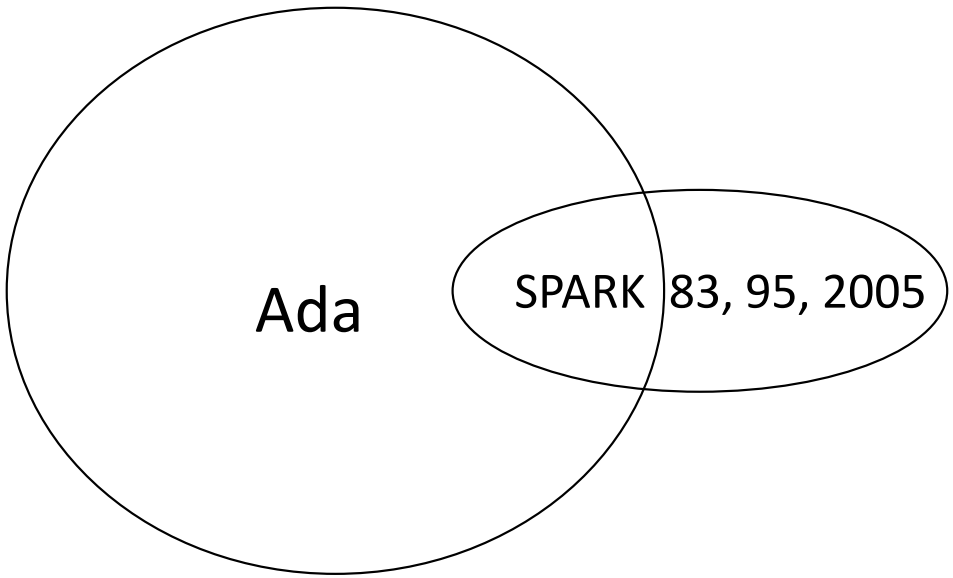
"It is not too late! I believe that by careful pruning of the Ada language, it is still possible to select a very powerful subset that would be reliable and efficient in implementation and safe and economic to use."

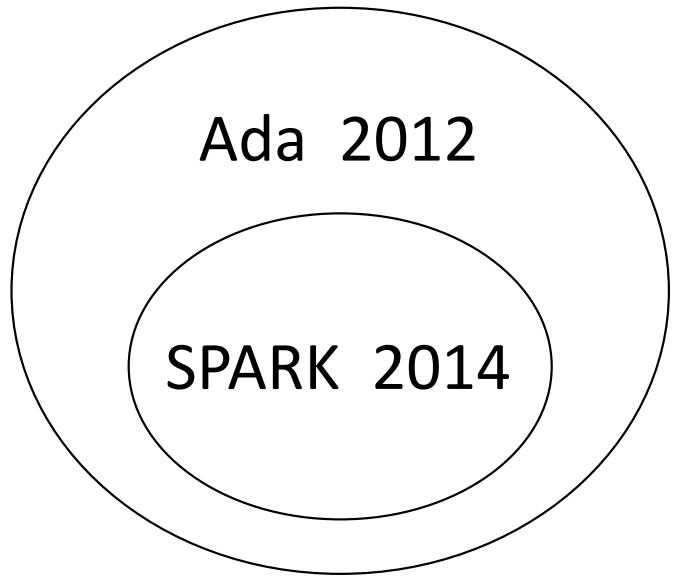
Professor Tony Hoare,
1980 ACM Turing Award Lecture

What is SPARK?

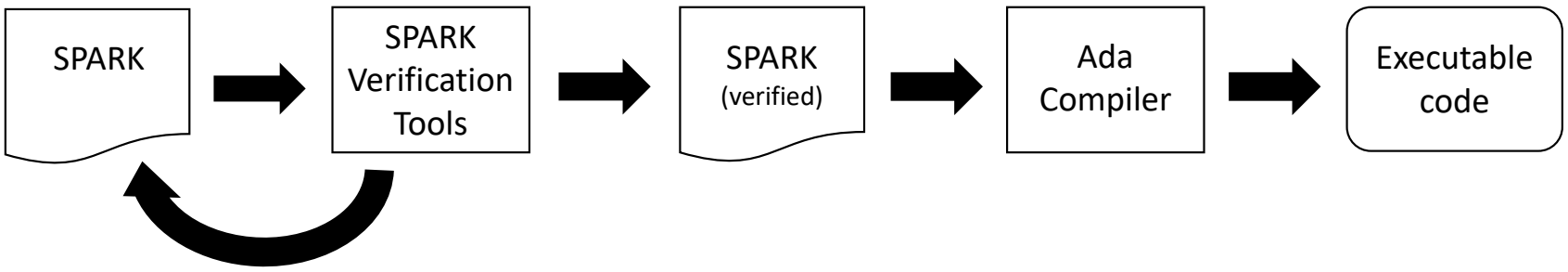
- SPARK is a high level programming language aimed at high integrity applications.
- SPARK was designed to exploit the strengths of Ada while eliminating the potential for ambiguities and insecurities, *e.g.*
  - functions in SPARK are true mathematical functions, *i.e.* can not have side-effects, so any ambiguity in terms of order of evaluation is eliminated.
  - Pointers are prohibited.
  - Aliasing is prohibited.
- SPARK was designed to support verification, both in terms of mainstream static analysis, *i.e.* flow analysis, and formal proof.
- The amount of space a SPARK program requires at run-time can be predicted via static analysis, *e.g.* guaranteed to have no memory leakage.

Ada-SPARK Relationships





The SPARK Approach



- The SPARK verification tools are applicable before coding is complete, *i.e.* the contracts should proceed the coding.
- SPARK approach advocates "correctness-by-construction".
- Note that adding contracts to existing Ada code, i.e. "Sparking the Ada" – is **not recommended!**

SPARK – Some Applications

SHOLIS: a sophisticated decision support system for helicopters landing on a Type 23 Frigate HMS Sutherland and Merlin HM2.

Eurofighter Typhoon: one of the most advanced multi-role combat aircraft

iFACTS: provides a range of decision support and management tools for air traffic controllers - including electronic flight strips and aircraft trajectory prediction

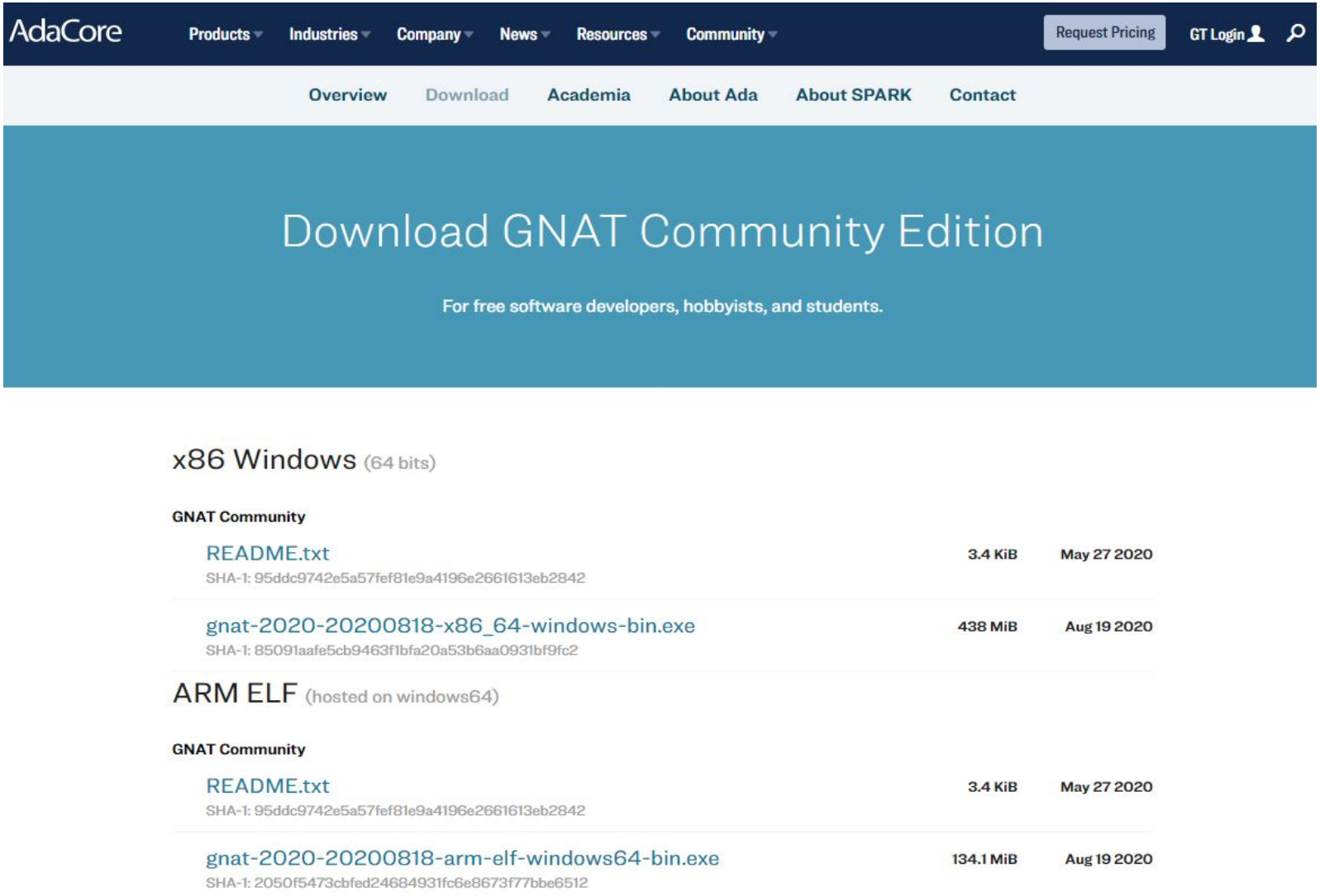
Accessing Ada and the SPARK Analysis Tools (Edinburgh)

- Linux:
  - **GNAT Studio** is available in the School Linux Lab (EMB 2.50), i.e., open a Linux terminal window and type `gnatstudio` at the command line.
  - Remote access via **X2GO** for details see:

    `https://www.macs.hw.ac.uk/cs/faq.html#Qnx`
- Windows:
  - **GNAT Studio** is available in the University Windows Lab (EMB 2.52), i.e., available as a Windows App.
  - Remote access via **KeyServer** for details see:

    `https://www.hw.ac.uk/uk/services/is/it-essentials/keyserver.htm`

Down loading Ada and the SPARK Toolkit



Use the **Community 2021** version, which you can download from `https://www.adacore.com/download`

Hello World!

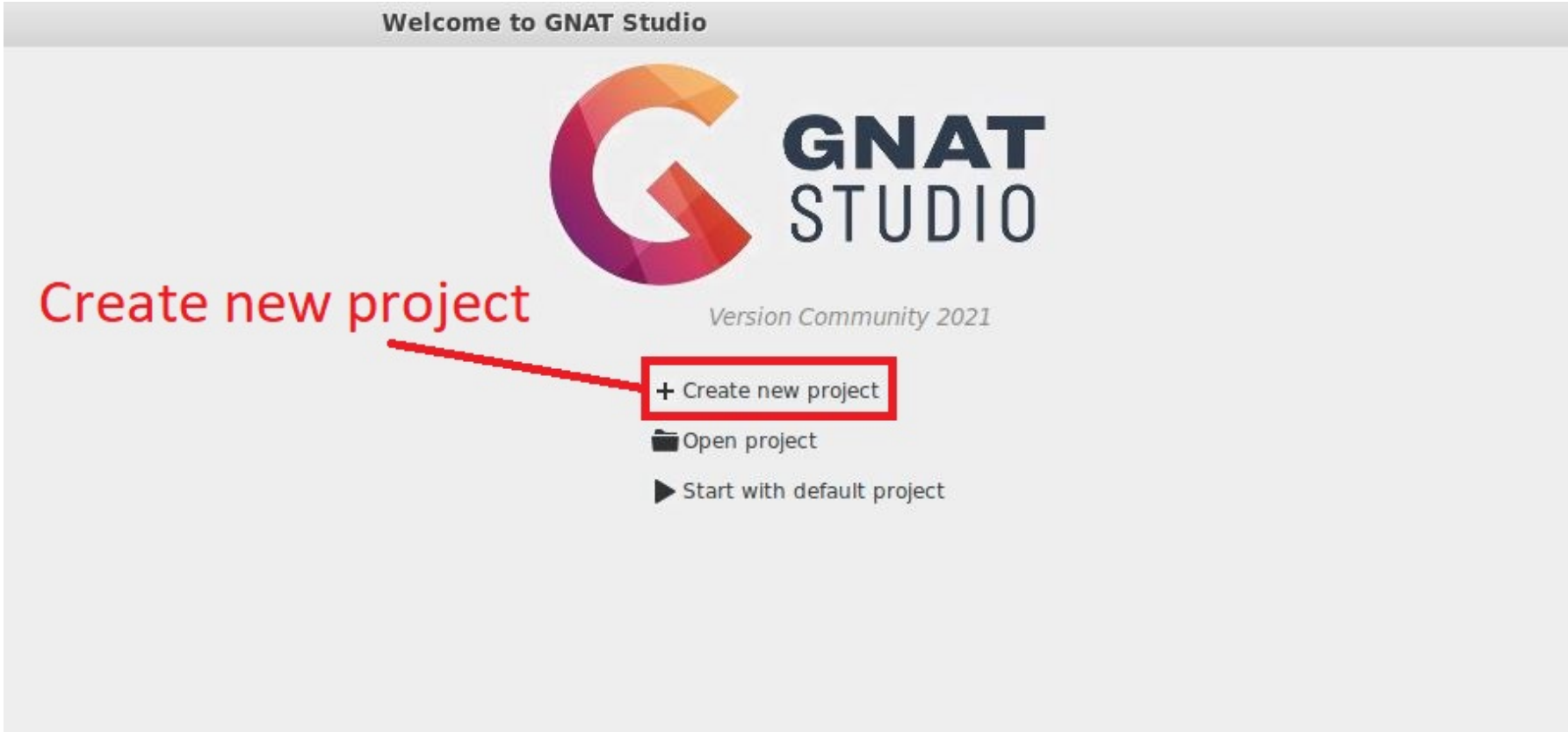
```
pragma SPARK_Mode (On);
with Text_IO;
-- My first program (this is a comment)!
procedure Hello is
begin
   Text_IO.Put_Line("Hello WORLD!");
end Hello;
```

A closer look at the language later, for now let's see how to analyze, compile and execute this code.

GNAT Studio – A Quick Tour

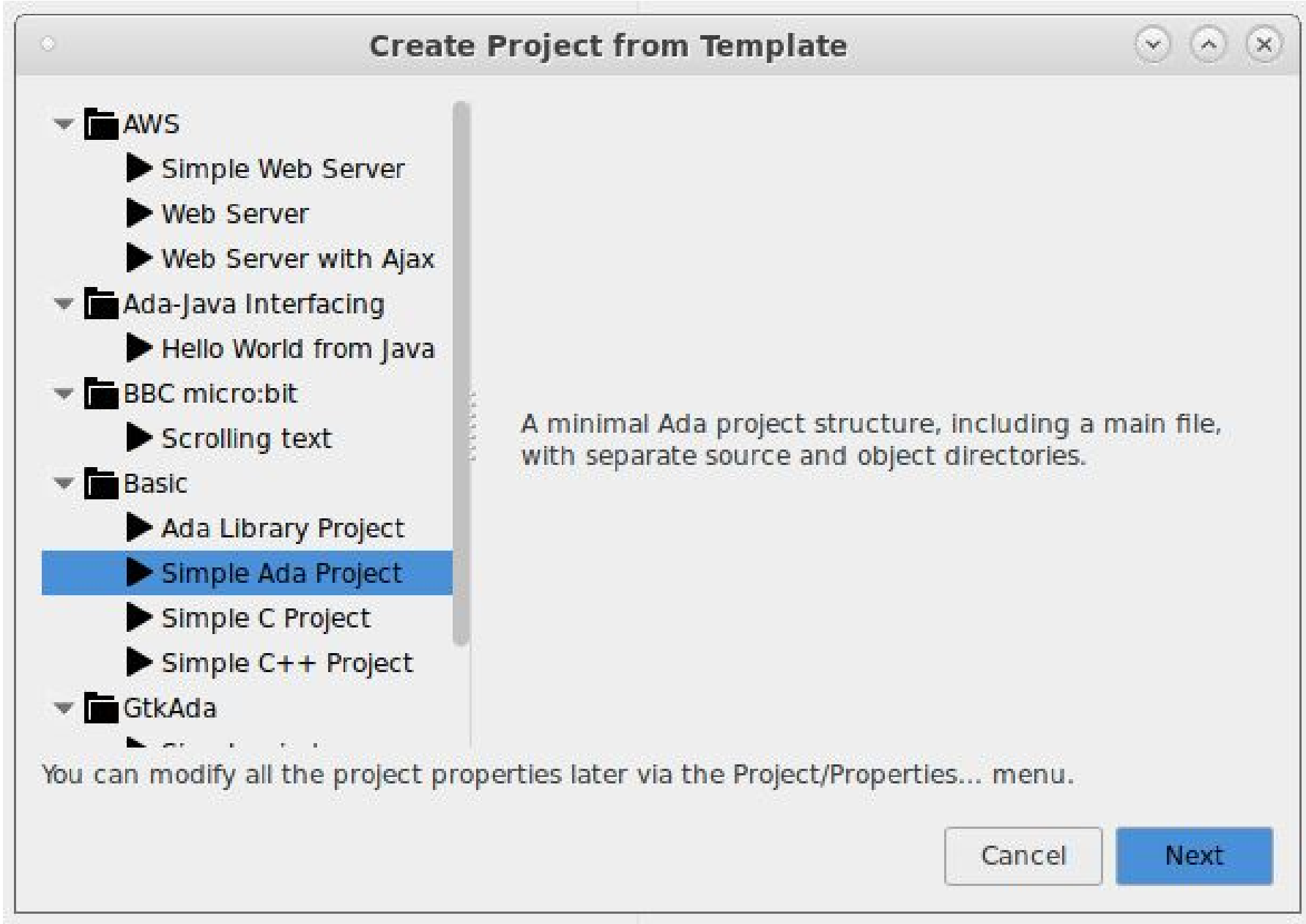
Creating a new project.

GNAT Studio – A Quick Tour

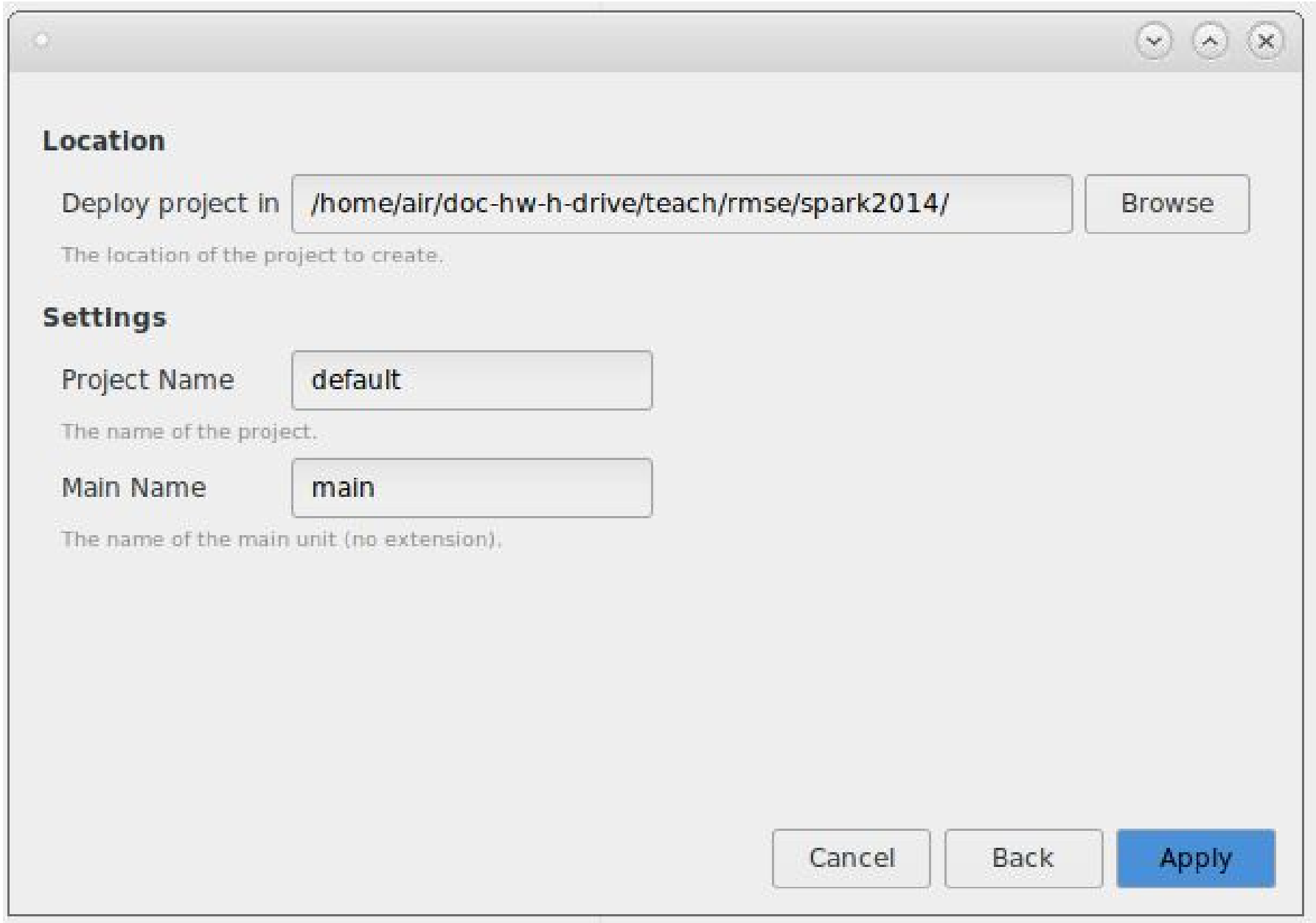


Select **Create new project.**

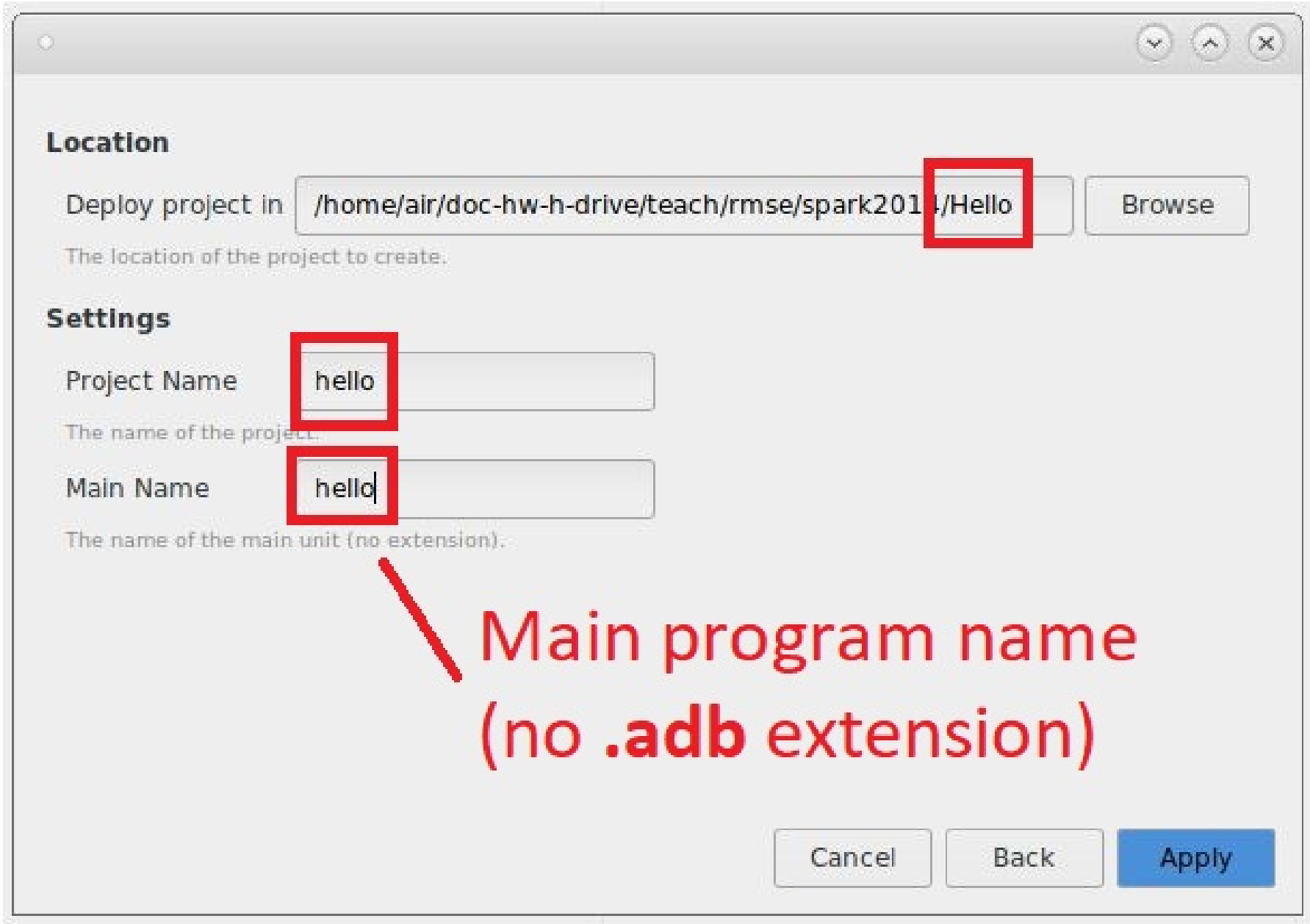
GNAT Studio – A Quick Tour



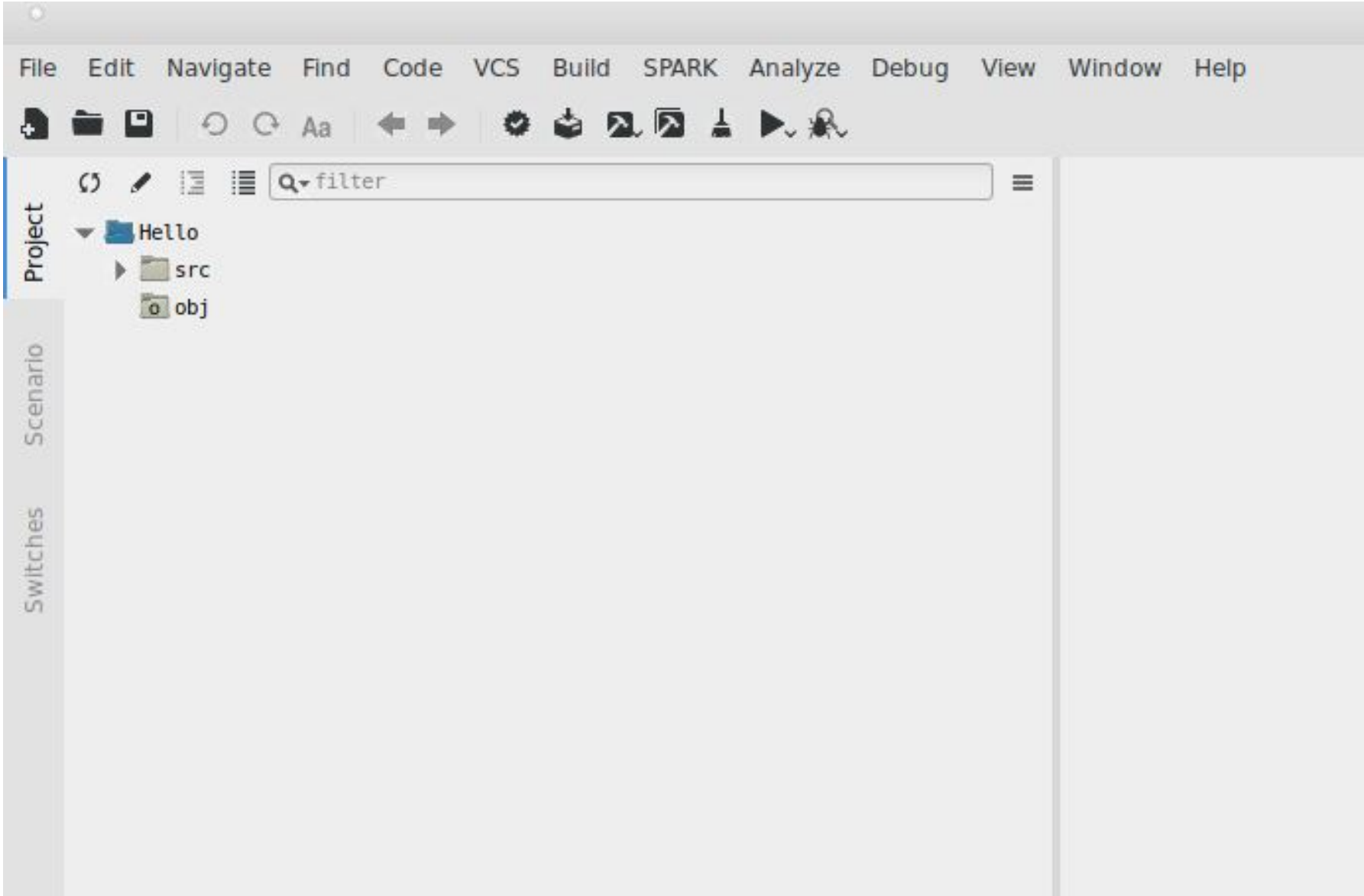
GNAT Studio – A Quick Tour



GNAT Studio – A Quick Tour

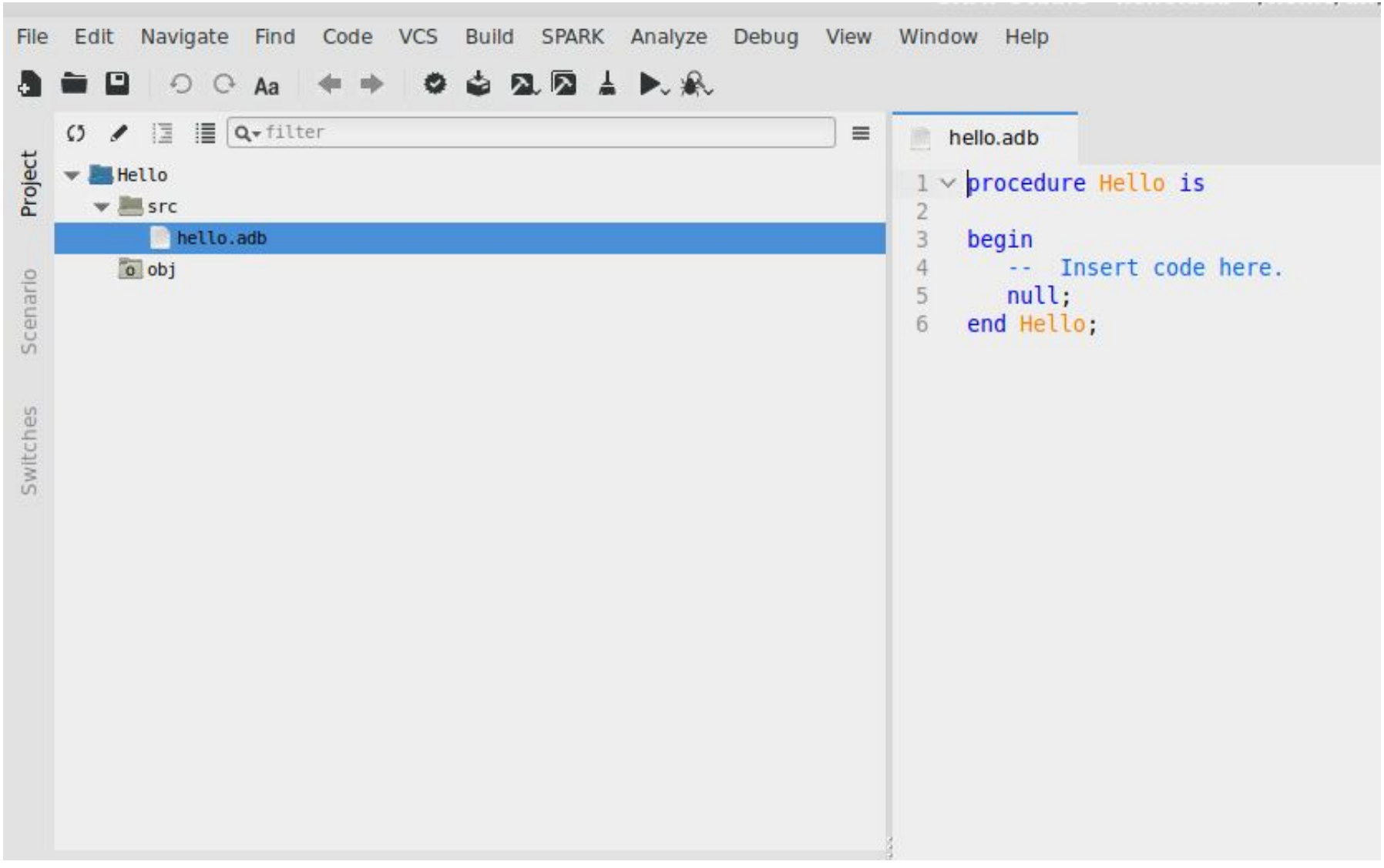


GNAT Studio – A Quick Tour



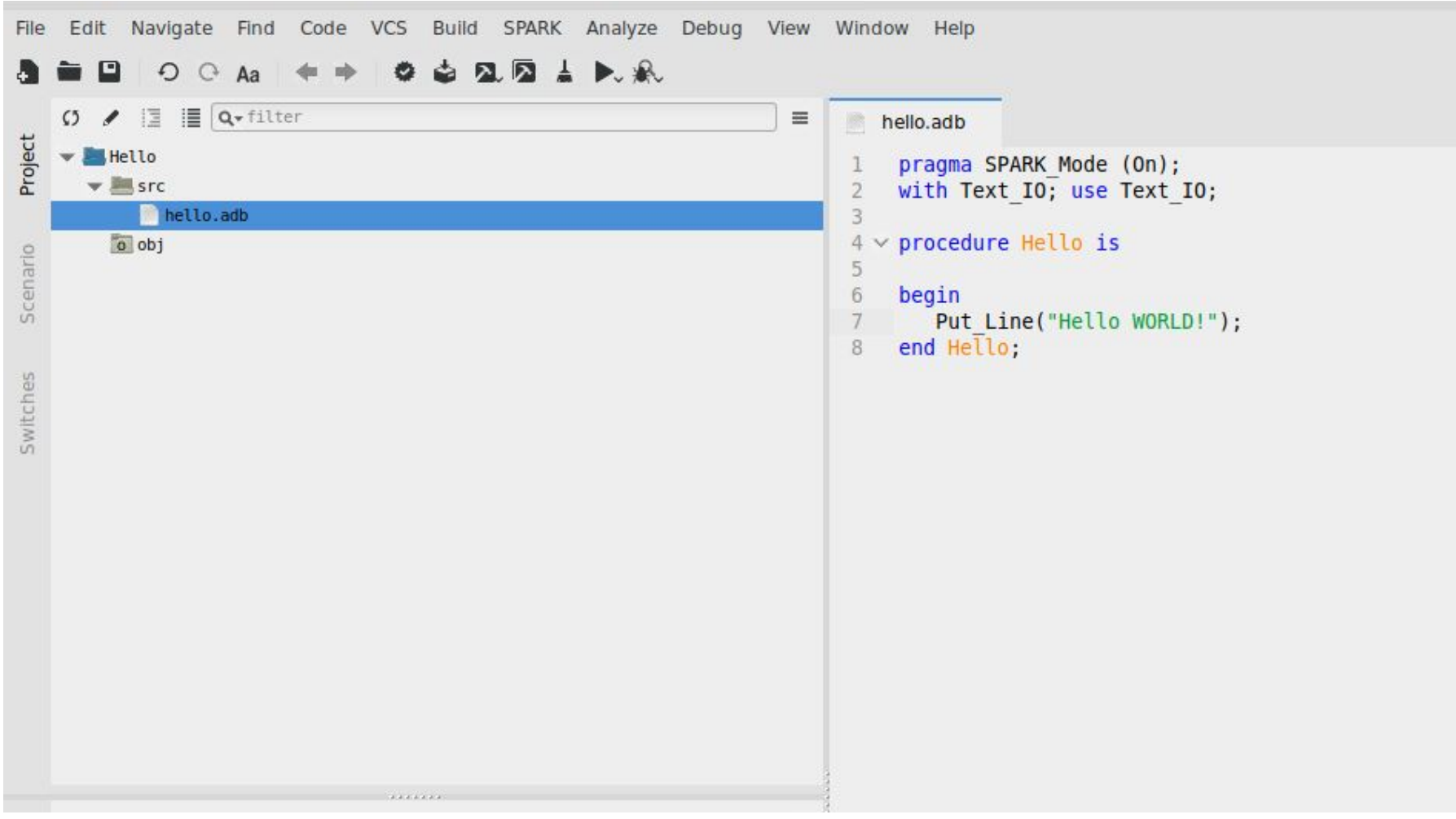
Note separate folders for source (**src**) and executable (**obj**) files.

GNAT Studio – A Quick Tour



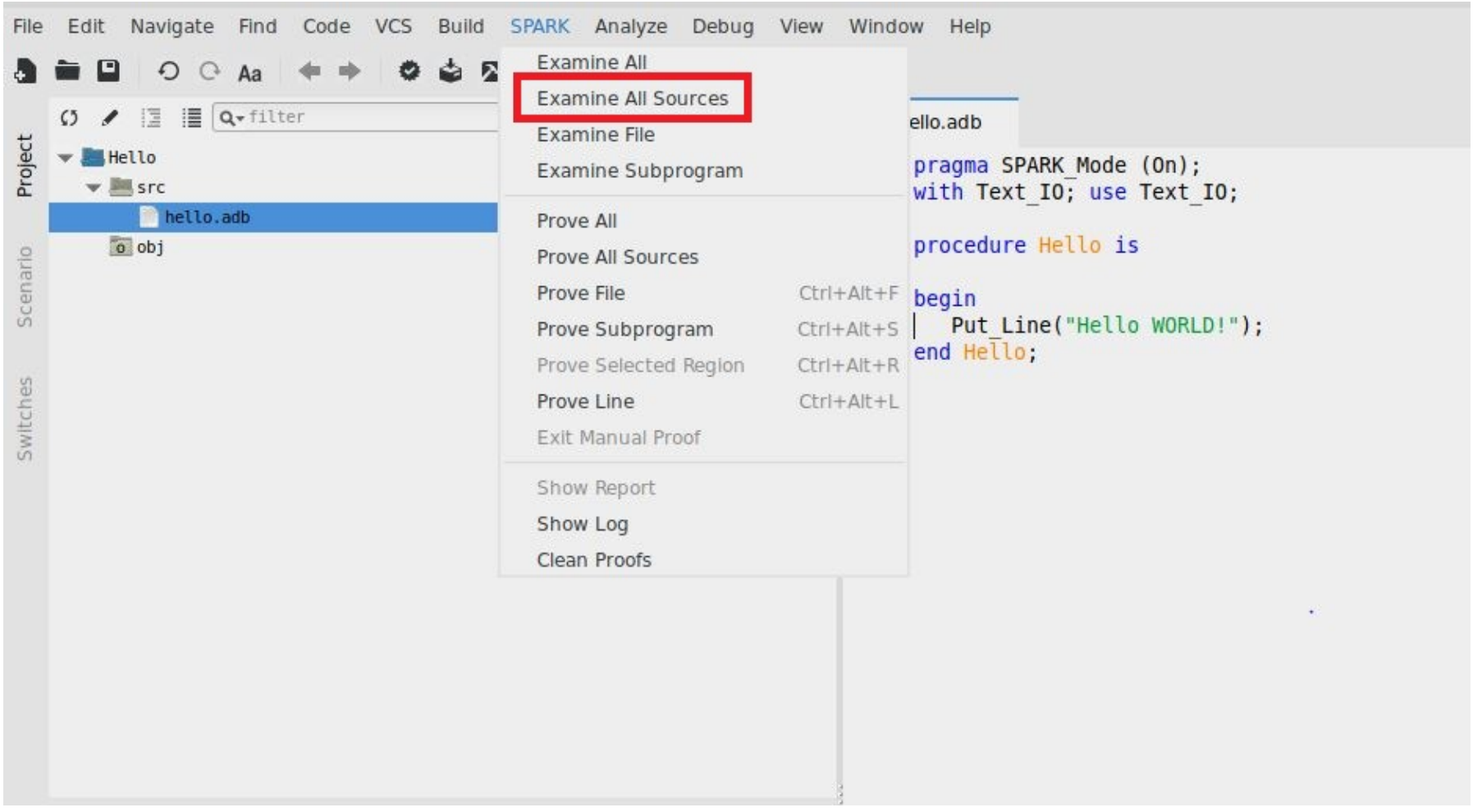
Note a **skeleton** main file is generated inside the **src** folder.

GNAT Studio – A Quick Tour



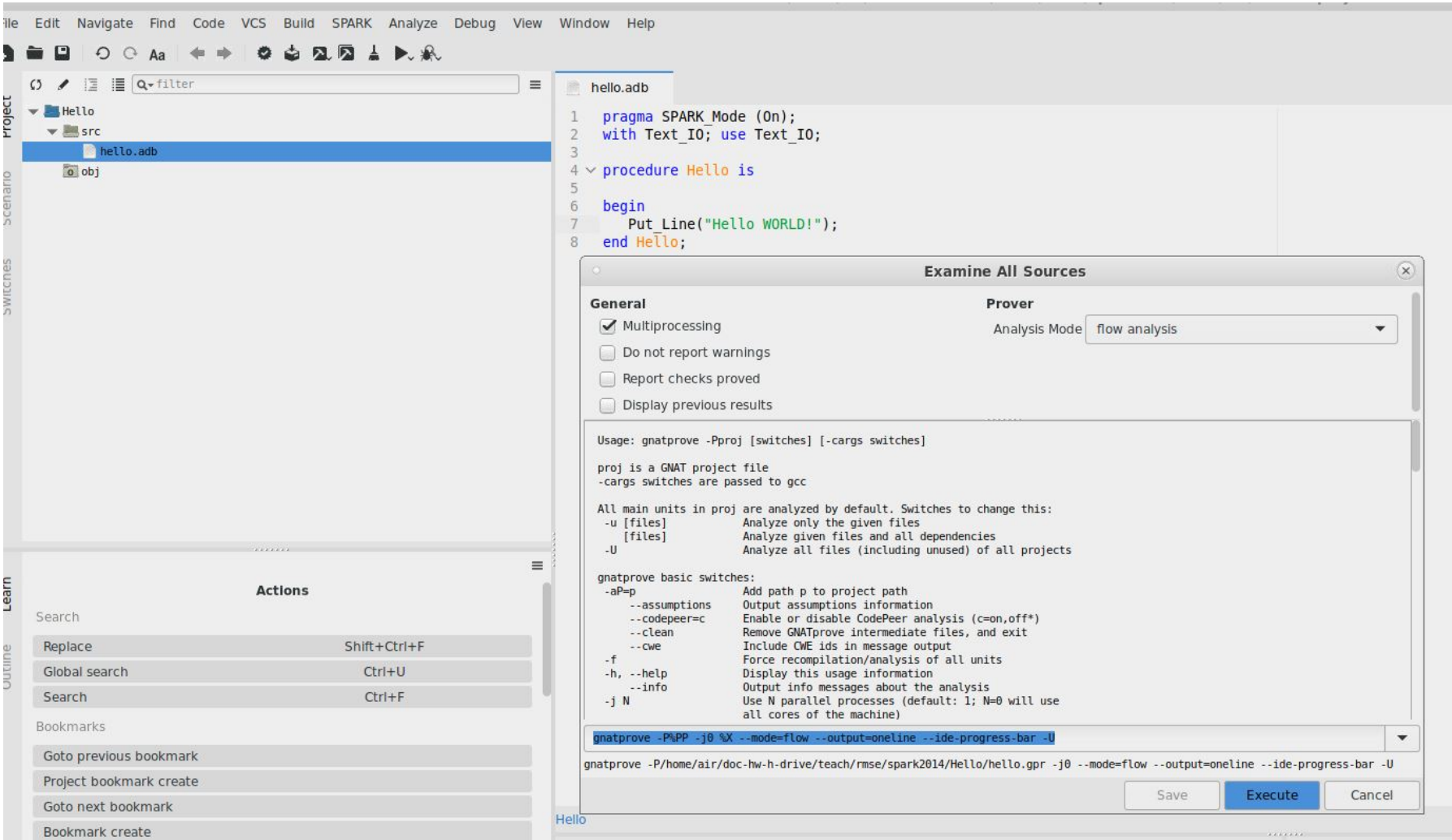
You can create your files within GNAT Studio or via your preferred editor and copy them to the **src** folder.

GNAT Studio – A Quick Tour



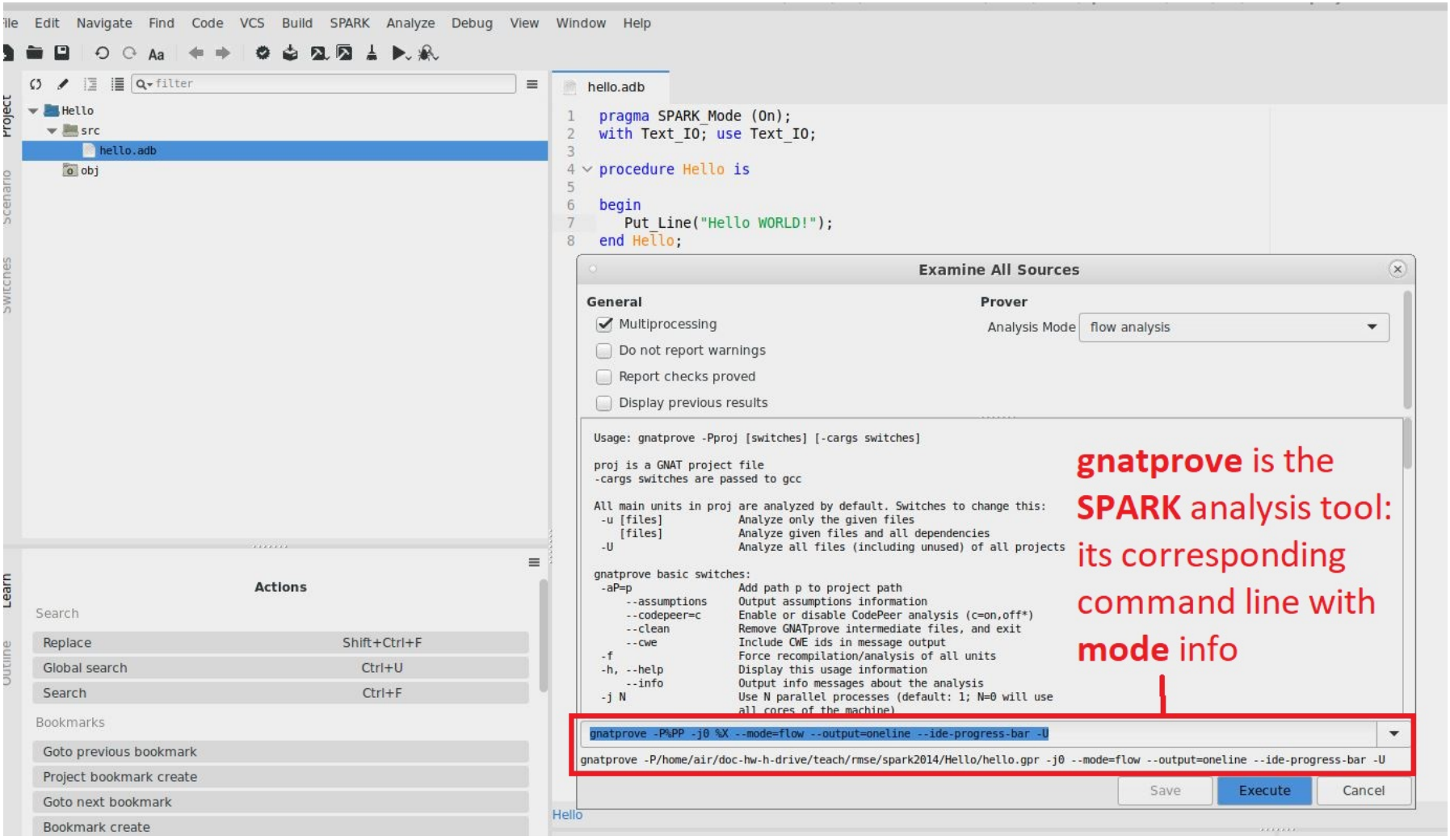
The **Examine All Sources** option from the **SPARK** menu performs a basic static analysis.

GNAT Studio – A Quick Tour



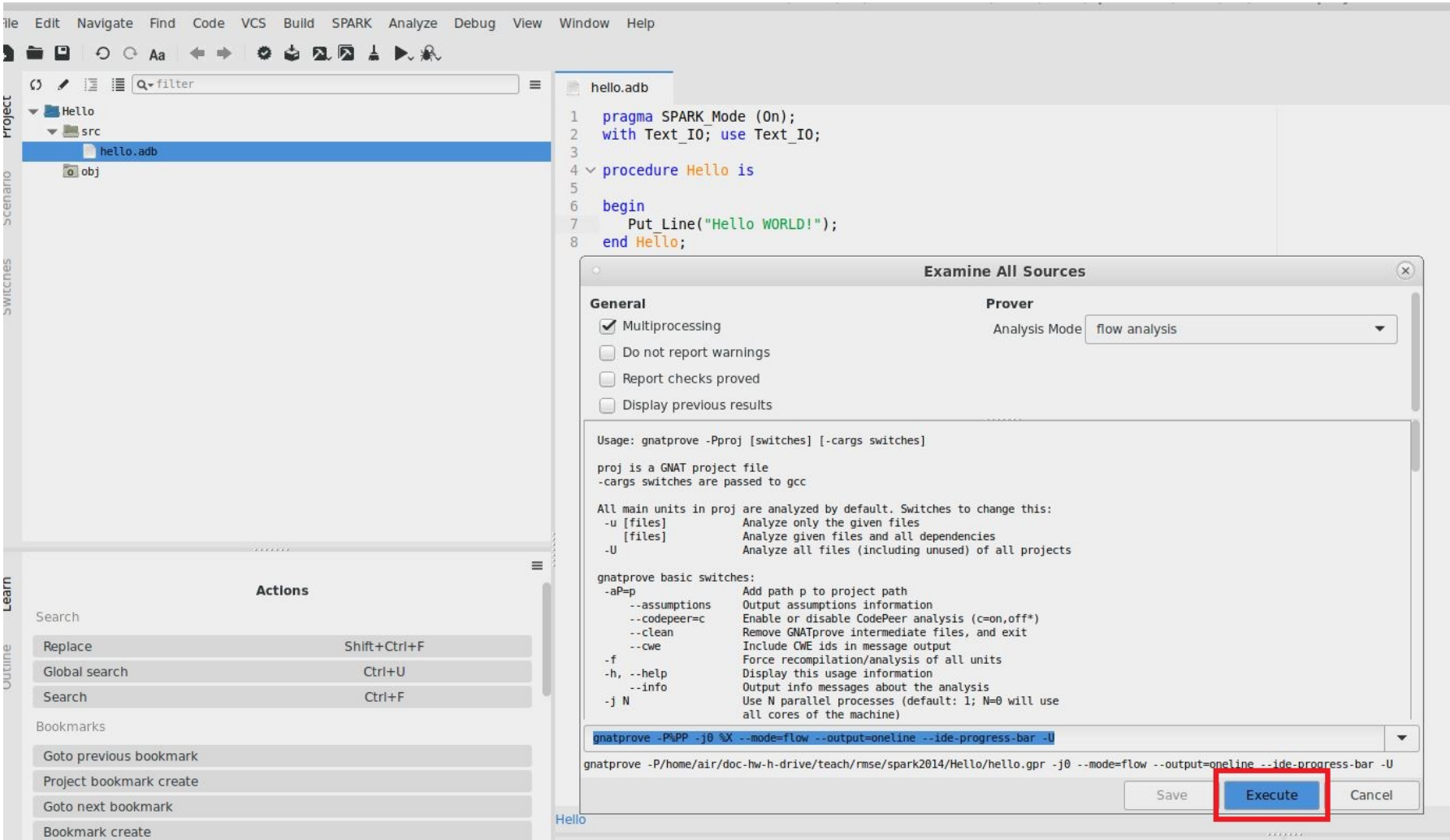
The **Examine All Sources** option from the **SPARK** menu performs a basic static analysis.

GNAT Studio – A Quick Tour



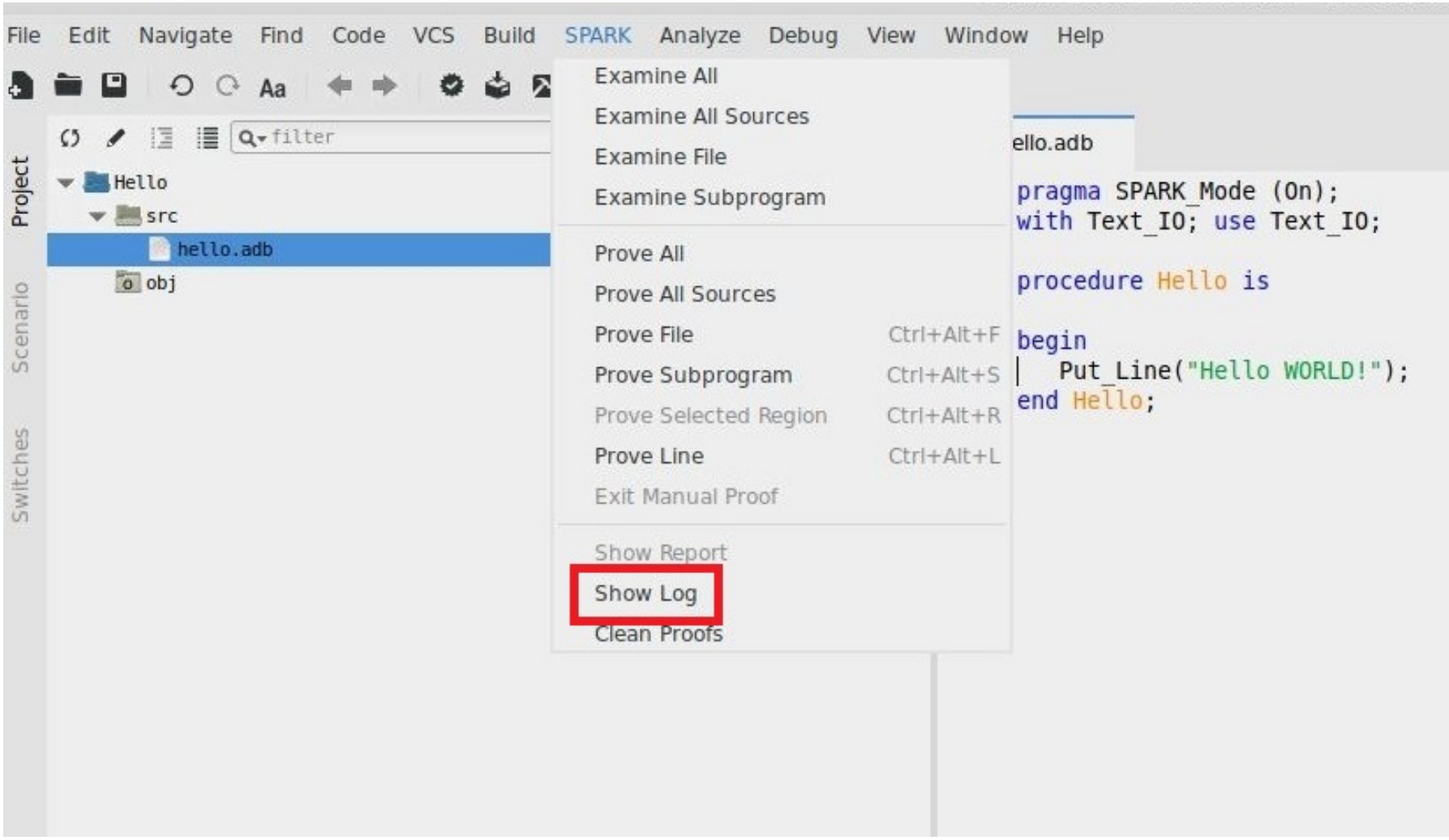
The **Examine All Sources** option from the **SPARK** menu performs a basic static analysis.

GNAT Studio – A Quick Tour



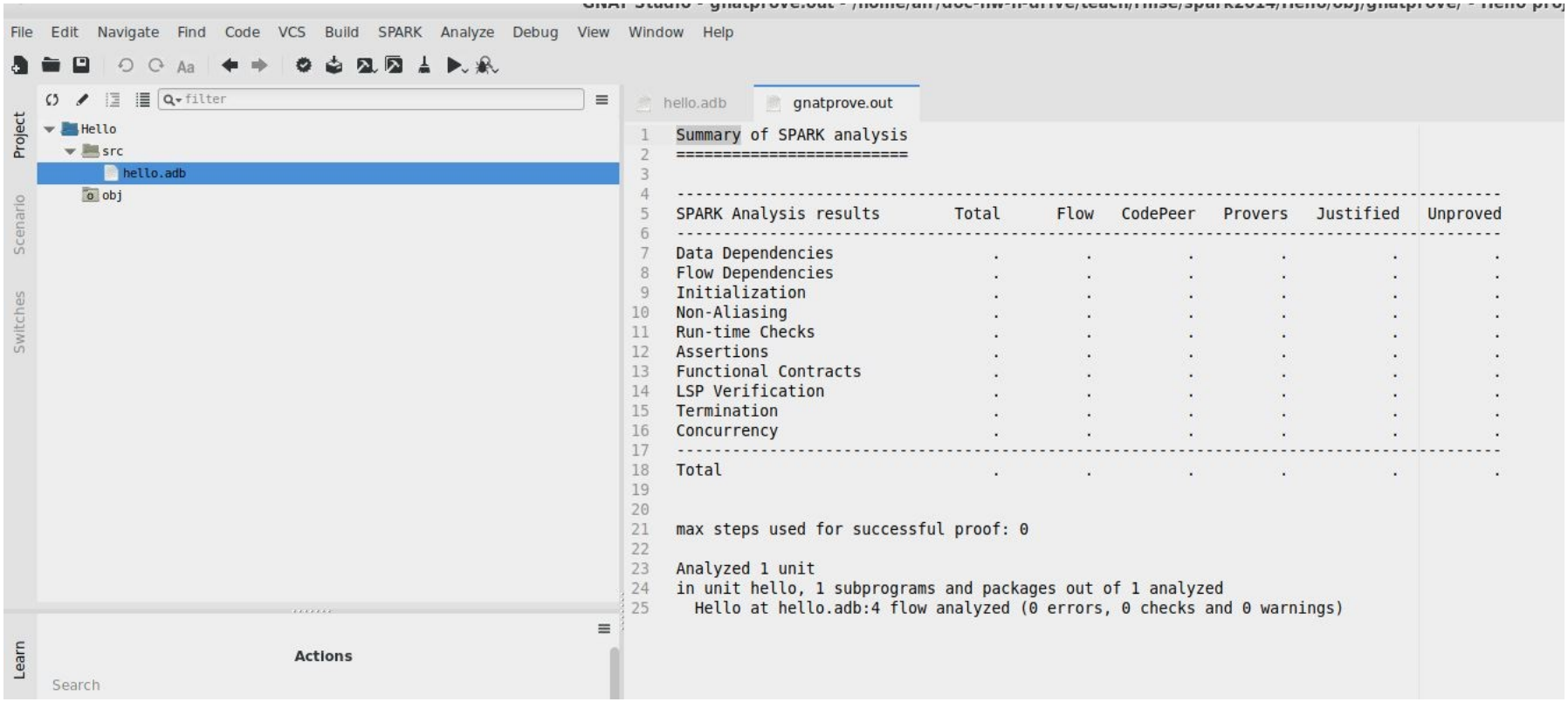
The **Examine All Sources** option from the **SPARK** menu performs a basic static analysis.

GNAT Studio – A Quick Tour



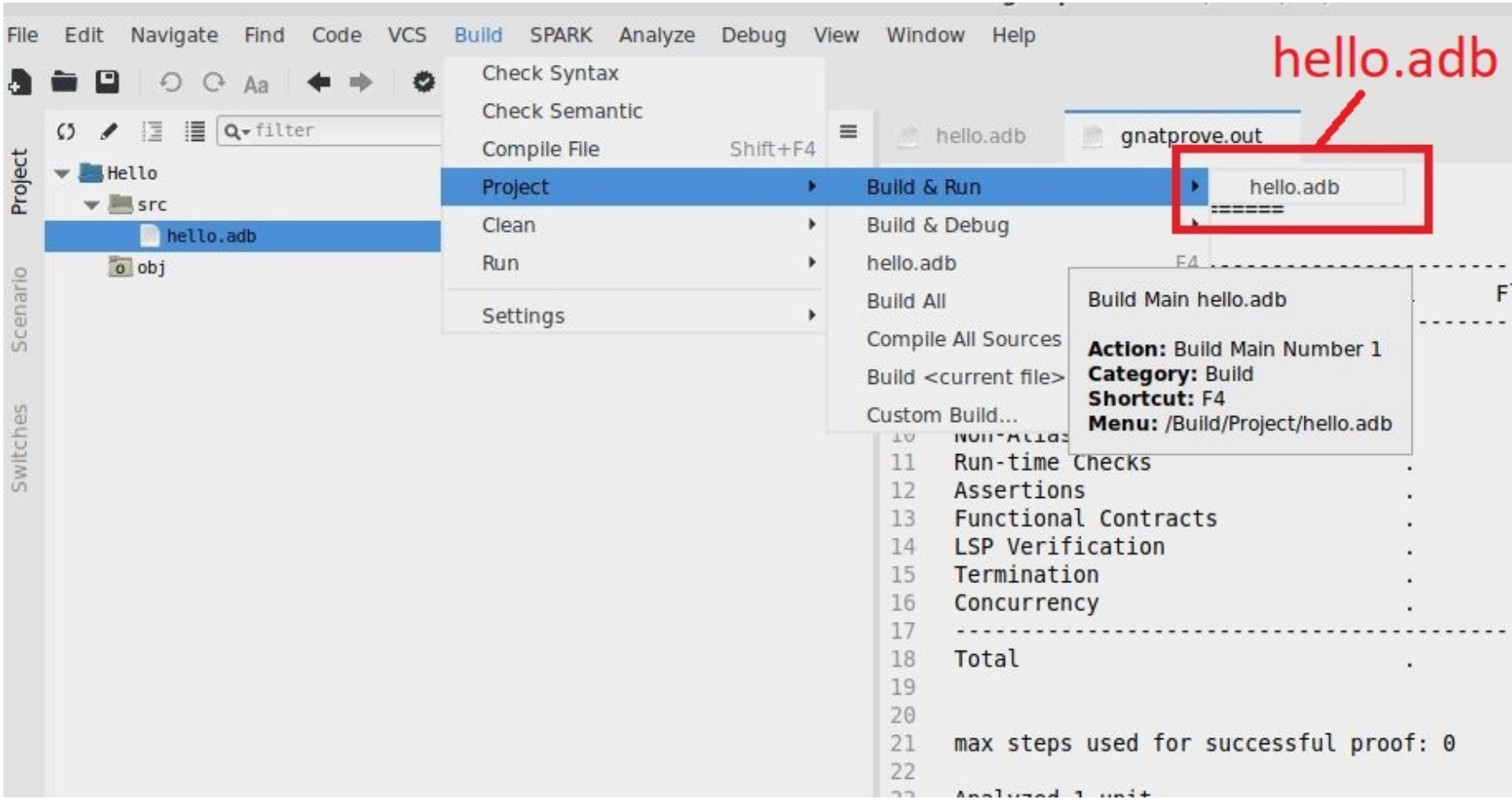
Select **Show Log** option from the **SPARK** menu to see a summary of the static analysis.

GNAT Studio – A Quick Tour



The **Summary of SPARK analysis** is shown above.

GNAT Studio – A Quick Tour



The **Project** ⇒ **Build & Run** option from the **Build** menu complies and runs your designated main file.

GNAT Studio – A Quick Tour

Hello WORLD!

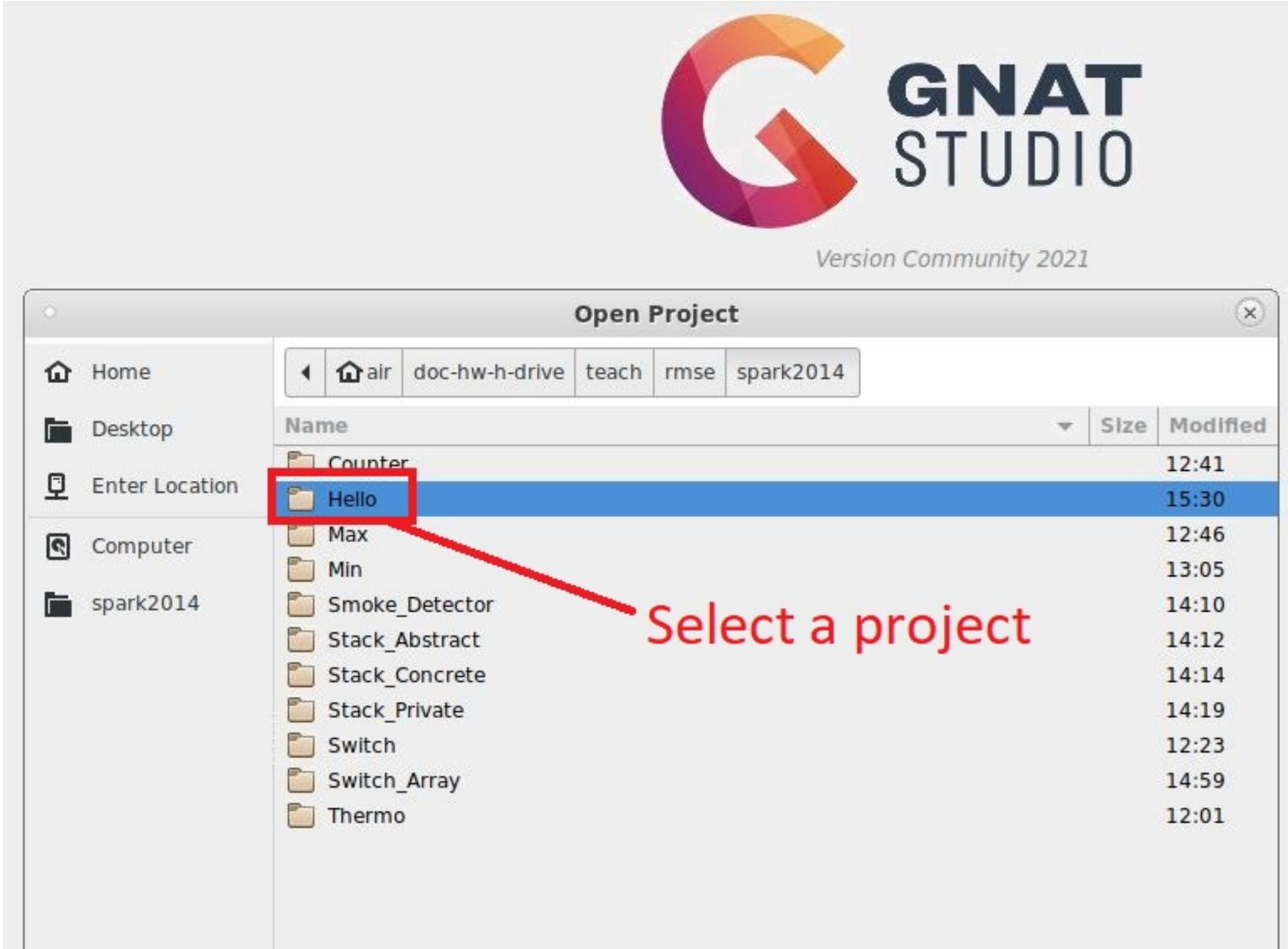
GNAT Studio – A Quick Tour

Openning an existing project.

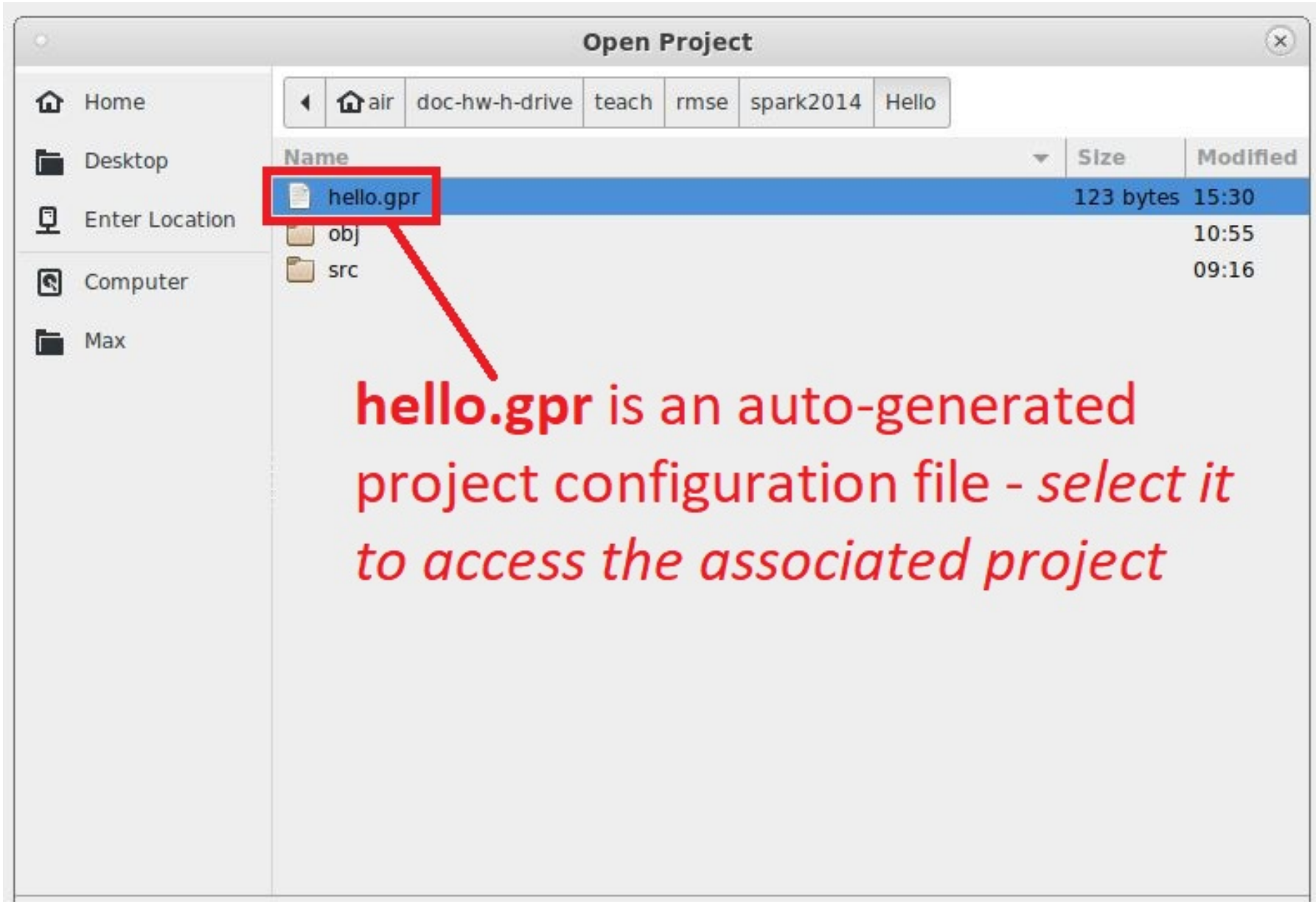
GNAT Studio – A Quick Tour

Select **Open project.**

GNAT Studio – A Quick Tour



GNAT Studio – A Quick Tour

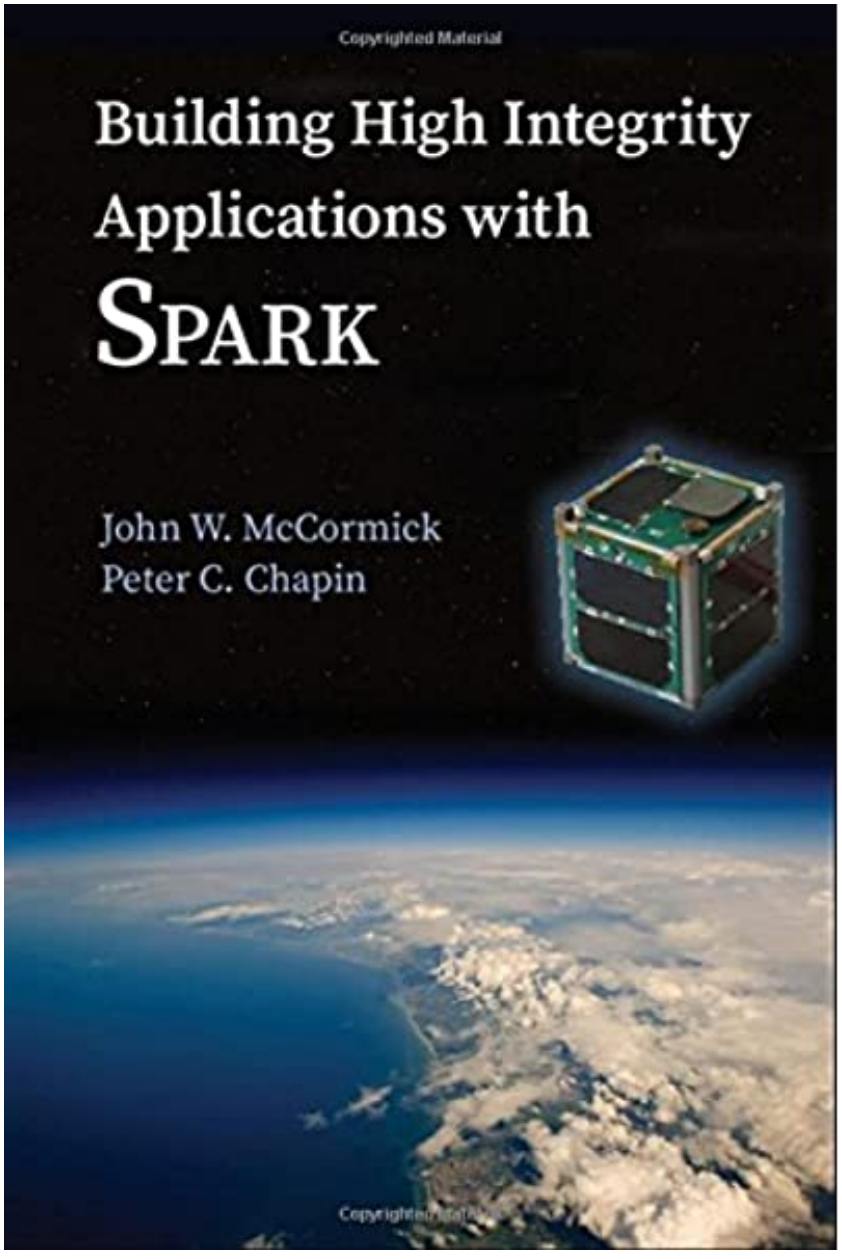


Additional Guidance

- A video showing how to create, analyse, compile and run the "Hello World" program using GNAT Studio on Windows is available via the **Week 1** module on the course's Canvas pages.
- An online tutorial for **GNAT Studio** is available via:

  `https://docs.adacore.com/live/wave/gps/html/gps_tutorial/`

Additional Guidance



Recommended reading:

- "Building High Integrity Applications with SPARK" Mccormick, J.W. and Chapin, P.C. Cambridge University Press, 2015.
- SPARK 2014 User's Guide: `https://docs.adacore.com/spark2014-docs/html/ug/`
- AdaCore: SPARK Pro: `https://www.adacore.com/sparkpro`

Summary

Learning outcomes:

- A brief history of Ada and its relationship with SPARK.
- How to access the SPARK toolkit and an Ada compiler.
- How to analyze, compile and execute SPARK code.