Polyspace® Products for Ada

Getting Started Guide

R2023a

MathWorks®

How to Contact MathWorks

Latest news:	`www.mathworks.com`
Sales and services:	`www.mathworks.com/sales_and_services`
User community:	`www.mathworks.com/matlabcentral`
Technical support:	`www.mathworks.com/support/contact_us`
Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Polyspace® Products for Ada Getting Started Guide

Trademarks

Patents

Revision History

March 2008	First printing	Revised for Version 5.1 (Release 2008a)
October 2008	Second printing	Revised for Version 5.2 (Release 2008b)
March 2009	Third printing	Revised for Version 5.3 (Release 2009a)
September 2009	Online only	Revised for Version 5.4 (Release 2009b)
March 2010	Online only	Revised for Version 5.5 (Release 2010a)
September 2010	Online only	Revised for Version 6.0 (Release 2010b)
April 2011	Fourth printing	Revised for Version 6.1 (Release 2011a)
September 2011	Online only	Revised for Version 6.2 (Release 2011b)
March 2012	Online only	Revised for Version 6.3 (Release 2012a)
September 2012	Online only	Revised for Version 6.4 (Release 2012b)
March 2013	Online only	Revised for Version 6.5 (Release 2013a)
September 2013	Online only	Revised for Version 6.6 (Release 2013b)
March 2014	Online Only	Revised for Version 6.7 (Release 2014a)
October 2014	Online Only	Revised for Version 6.8 (Release 2014b)
March 2015	Online Only	Revised for Version 6.9 (Release 2015a)
September 2015	Online Only	Revised for Version 6.10 (Release 2015b)
March 2016	Online Only	Revised for Version 6.11 (Release 2016a)
September 2016	Online Only	Revised for Version 6.12 (Release 2016b)
March 2017	Online Only	Revised for Version 6.13 (Release 2017a)
September 2017	Online Only	Revised for Version 6.14 (Release 2017b)
March 2018	Online Only	Revised for Version 6.15 (Release 2018a)
September 2018	Online Only	Revised for Version 6.16 (Release 2018b)
March 2019	Online Only	Revised for Version 6.17 (Release 2019a)
September 2019	Online Only	Revised for Version 6.18 (Release 2019b)
March 2020	Online Only	Revised for Version 6.19 (Release 2020a)
September 2020	Online Only	Revised for Version 6.20 (Release 2020b)
March 2021	Online Only	Revised for Version 6.21 (Release 2021a)
September 2021	Online Only	Revised for Version 6.22 (Release 2021b)
March 2022	Online Only	Revised for Version 6.23 (Release 2022a)
September 2022	Online Only	Revised for Version 6.24 (Release 2022b)
March 2023	Online Only	Revised for Version 6.25 (Release 2023a)

Contents

1 Introduction to Polyspace Products for Verifying Ada Code

2 Setting Up Polyspace Project

3 Run Verification

4 Review Verification Results

Introduction to Polyspace Products for Verifying Ada Code

Product Description

In this section...
"Polyspace Client for Ada" on page 1-2
"Polyspace Server for Ada" on page 1-2

Polyspace Client for Ada

Prove the absence of run-time errors in source code

Polyspace Client for Ada proves the absence of overflow, divide-by-zero, out-of-bounds array access, and certain other run-time errors in Ada83 and Ada95 source code. It produces results without requiring program execution, code instrumentation, or test cases. Polyspace Client for Ada uses abstract interpretation techniques based on formal methods to verify code. Analysis results are shown within the source code. Each code statement is color-coded to indicate whether it is free of run-time errors, proven to fail, unreachable, or unproven. Polyspace Client for Ada displays range information for variables and function return values and can prove which variables exceed specified range limits.

You can use Polyspace Client for Ada on your desktop to run and review code analyses before compilation and test.

Polyspace Server for Ada

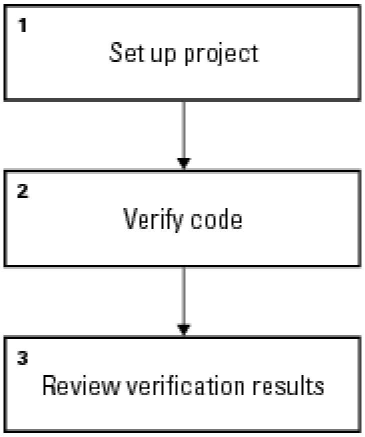
Perform code verification on computer clusters and publish metrics

Polyspace Server for Ada is a sound static analysis engine that proves the absence of overflow, divide-by-zero, out-of-bounds, array access, and certain other run-time errors in Ada83 and Ada95 code. It performs interprocedural analysis of all possible control and data flows, including multithreaded code, to identify each operation as always safe, always faulty, unreachable, or vulnerable. Polyspace Server for Ada identifies code segments that are free of run-time errors, proven to fail, unreachable, or unproven.

You can run Polyspace Server for Ada on a server-class machine and integrate it into build and continuous integration systems for automated verification using tools such as Jenkins®. The analysis results can be reviewed using the Polyspace Client for Ada or published to Polyspace Access™ for triage and resolution.

Basic Workflow

The basic workflow for using Polyspace software to verify Ada source code is:



In this workflow, you:

1. Set up a project file in the Polyspace user interface. See the tutorial “Set Up Polyspace Project” on page 2-2.
2. Verify code on a server or client.

   You can use the Polyspace user interface to start the verification or you can select files from a Microsoft® Windows® folder and send them to Polyspace software for verification. For verifications that run on a server, you can use the Polyspace Job Monitor to manage the verifications and download the results to a client. See the tutorial “Run Verification” on page 3-2
3. Review verification results in the Polyspace user interface. See the tutorial “Review Verification Results” on page 4-2.

Getting Help

In this section...

Product Help

To access the help that came with your installation, select **Help** > **Help** or clickⓘ in the Polyspace window.

Context Sensitive Help

In addition to the full documentation, you can also access contextual help for verification options and checks.

- In the **Configuration** window, hover your cursor over a verification option. In the tooltip, select **More Help** to get help on the option.
- In the **Check Details** window, select ⓘ to get help on the check.

MathWorks Online

For additional information and support, see:

www.mathworks.com/products/polyspace-ada

Related Products

In this section...

Polyspace Code Prover

For information about Polyspace products that verify C/C++ code, see the following:

`https://www.mathworks.com/products/polyspace-code-prover`

Polyspace Bug Finder

For information about Polyspace products that analyze C/C++ code to find possible defects, see the following:

`https://www.mathworks.com/products/polyspace-bug-finder`

Setting Up Polyspace Project

Set Up Polyspace Project

In this section...

Tutorial Overview

In this tutorial, you create a Polyspace project to verify Ada code.

What Is a Project?

A Polyspace project consists of:

- **Source** folders and their files.
- **Include** folders.
- One or more modules. You run verification on the source files in each module. Each module has the following folders:
  - **Source** — Contains files used for verification.
  - **Configuration** — Contains analysis options used for verification.
  - **Result** — Contains results of verification.

Prepare Project Folder

In the following procedures, `*Polyspace_Install*` is the Polyspace installation folder, for example, `C:\Program Files\Polyspace\`.

1. Create a folder `polyspace_project` in a particular location, for example `C:\`.
2. Open `polyspace_project` and create subfolders:
   - `sources`
   - `includes`
3. Copy `example.adb` and `example.ads` from *`Polyspace_Install`*`\polyspace\examples\ada\Demo_Ada_Single-File\sources` to `polyspace_project\sources`.
4. Copy the files from *`Polyspace_Install`*`\polyspace\examples\ada\Demo_Ada_Single-File\sources` to `polyspace_project\includes`.

Open Polyspace Verification Environment

- Windows: From the *`Polyspace_Install`*`\polyspace\bin` folder, double-click the `polyspace` executable.

- Linux® or Mac: Run the following command:

  ```
  /Polyspace_Install/polyspace/bin/polyspace
  ```

For project setup, you can use the following panes in the Polyspace user interface here.

Section	Use For
Project Browser	Specifying: • Source files • Include folders • Results folder
Configuration	Specifying analysis options
Output Summary	Monitoring the progress of a verification, and viewing status, log messages, and general verification statistics.

You can resize or hide sections.

Create Project

- "Create New Project" on page 2-3
- "Specify Source Files and Include Folders" on page 2-3
- "Next Steps" on page 2-4

Create New Project

1. Select **File** > **New Project**.
2. In the Project – Properties dialog box:

   - For **Project name**, enter `polyspace_project`.
   - Clear the **Use default location** check box. To specify where your `polyspace_project` folder is, click 📁.
   - Clear the boxes under **Project Configuration**.

     For more information on the option **Use template**, see "Create Project Using Template".

3. Click **Next**.

Specify Source Files and Include Folders

1. Use the **Browse** button to select the `sources` folder that you created.
2. Click **Add Source Folders**, then click **Next**.
3. Use the **Browse** button to select the `includes` folder that you created.
4. Click **Add Include Folders**, then click **Finish**.

   The analysis looks for include files relative to the folder paths that you specify. For instance, if your code contains the preprocessor directive `#include<../mylib.h>` and you include the folder:

```
C:\My_Project\MySourceFiles\Includes
```

the folder `C:\My_Project\MySourceFiles` must contain a file `mylib.h`.

You can see your project in the **Project Browser**.

Next Steps

1. “Run Verification” on page 3-2
2. “Review Verification Results” on page 4-2

3

Run Verification

Run Verification

In this section...

Tutorial Overview

In this tutorial, you run verification on your source code. Perform the steps outlined for remote verification if you want to perform verification on another machine. Otherwise, perform the steps outlined for local verification.

Before You Start the Tutorial

Before you start, you must:

- Complete "Set Up Polyspace Project" on page 2-2. You use the `polyspace_project` folder and the `polyspace_project.psprj` file in this tutorial.
- "Modify Polyspace Server Configuration" for remote verification.

Prepare for Verification

If `polyspace_project.psprj` is not already open in the **Project Browser**, then:

1. Select **File** > **Open**.
2. In the Open File dialog box, navigate to `polyspace_project`.
3. Select the project file `polyspace_project`.
4. Click **Open**.

Run Remote Verification

- "Start Verification" on page 3-2
- "Monitor Progress" on page 3-3
- "Stop Verification" on page 3-3

Start Verification

Before you start remote verification, you must perform a one-time setup. See "Modify Polyspace Server Configuration".

1. On the **Project Browser** pane, select **Module_1**.
2. On the **Configuration** pane, select **Machine Configuration**.

3. Select **Send to Polyspace Server**. By default, this action enables the **Add to results repository** option.
4. On the toolbar, click ▶.

   The following happens:

   a. On the local host computer, Polyspace Code Prover™ compiles your code.
   b. When the **Compile** phase is complete, you see the message `Verification queued on server` in the **Output Summary** tab.

If the verification fails, go to “Troubleshooting in Polyspace Products for Ada”.

Monitor Progress

To monitor the progress of a remote verification:

1. Select **Tools** > **Open Job Monitor**.
2. In the Polyspace Job Monitor, right-click your verification.
3. Select **View Log File**.

Stop Verification

To stop a remote verification:

1. Select **Tools** > **Open Job Monitor**.
2. In the Polyspace Job Monitor, right-click your verification.
3. Select **Remove From Queue**.

Run Local Verification

- “Start Verification” on page 3-3
- “Monitor Progress” on page 3-3
- “Stop Verification” on page 3-4

Start Verification

To start a verification on your local computer:

1. In the Polyspace user interface, in the **Project Browser**, select **Module_1**.
2. On the **Configuration** pane, select **Machine Configuration**. Clear **Send to Polyspace Server** if it is selected.
3. On the toolbar, click ▶.

If the verification fails, go to “Troubleshooting in Polyspace Products for Ada”.

Monitor Progress

To monitor the progress of a local verification, on the **Output Summary** pane, use the following tabs:

- **Output Summary**

- **Run Log**

  If this window is not visible by default, select **Window** > **Show/Hide View** > **Run Log**.

When the verification is complete, you see:

- Results on the **Results List** pane.
- Statistics, such as **Code covered by verification** and **Check distribution** on the **Dashboard** pane.

Stop Verification

To stop a local verification:

1. On the toolbar, click Stop.

   A warning dialog box opens.
2. Click **Yes**.

   The verification stops. If you restart the verification, it starts from the beginning.

Next steps

“Review Verification Results” on page 4-2

Review Verification Results

Review Verification Results

In this section...

Tutorial Overview

In this tutorial, you explore the results of verifying `example.c`. Before starting this tutorial, complete "Run Verification" on page 3-2.

Open Results

- "Remote Verification" on page 4-2
- "Local Verification" on page 4-2

Remote Verification

To open results from a remote verification:

1. Select **Access** > **Open Web Interface**.

   You might need to log in with your Polyspace Access credentials.
2. In your web browser, select a project in the **Project Explorer**.

   You can see an overview of your project statistics in the **Project Overview** dashboard.

Alternatively, select **Access** > **Open Results** to open the results hosted on Polyspace Access directly in the desktop interface. See also "Open or Export Results from Polyspace Access" (Polyspace Code Prover).

Local Verification

After verification, the results open automatically.

Review Results

Polyspace performs checks on each operation in your code. The software reports whether a check is green, red, orange or gray.

Check color	Indicates
Red	The code operation fails the check on every execution path.
Green	The code operation passes the check on every execution path.

Check color	Indicates
Orange	The code operation fails the check on some execution paths.
Gray	The code operation is unreachable from entry-point functions.

1. On the **Results List** pane, from the ☰▾ list, select **File**.

   The checks are grouped by file. Within each file, the checks are grouped by function.

2. Expand the following function names and select a check in the function. The corresponding line of code on the **Source** pane appears highlighted.

Function	Check	Source Code Appearance	Reason
`UNREACHABLE_CODE`	Gray **Unreachable code**	The `:` on line 182 is gray.	`x` is greater than 0. So the `if` statement branch cannot be reached.
`INFINITE_LOOP`	Red **Non-terminating loop**	The keyword `loop` on line 123 is red.	`x` cannot be less than 0. So the code cannot exit the loop.
`RECURSION`	Orange **Division by Zero**	The `/` sign on line 62 is orange.	The denominator can be zero.
`NON_INFINITE_LOOP`	Green **Overflow**	The `+` sign on line 112 is green.	`cur` does not overflow. The loop on line 110 terminates before `cur` can overflow.

3. To find further information about a check, do one of the following:

   - View the message on the **Result Details** pane.
   - Place your cursor on the check in the **Source** pane. View the tooltip.

4. Filter **Division by Zero** checks. To do this:

   a. Click ☑ on the **Check** column header.

   b. From the drop-down list, clear **All** and select **Division by Zero**.

   The **Results List** pane displays only the **Division by Zero** checks.

5. On the **Results List** pane, select the red **Division by Zero** check in the function `PROCEDURE_ZDV`. Enter the following review information.

Column	Action
Severity	`High`
Status	`Fix`
Comment	`x is always zero`

Generate Report

To generate a verification report:

1. If your verification results are not already open, open them.
2. Select **Reporting** > **Run Report**.
3. In the **Select Reports** section, select **Developer**.
4. For **Output folder**, select `C:\polyspace_project\Module_1\Result_1\Polyspace-Doc`.
5. For **Output format**, select `PDF` .
6. Click **Run Report**.

   The software creates the specified report and opens it.