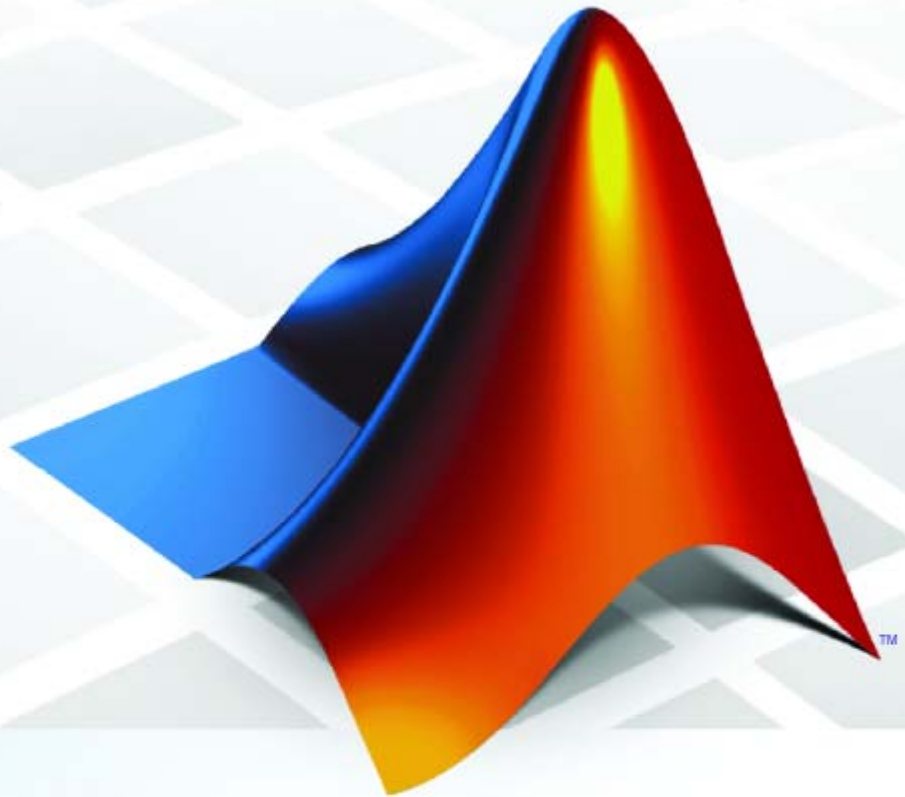


PolySpace™ Client/Server for Ada 5

Getting Started Guide



How to Contact The MathWorks



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



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

For contact information about worldwide offices, see the MathWorks Web site.

PolySpace™ Client/Server for Ada Getting Started Guide

© COPYRIGHT 2007–2008 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

March 2008 First printing New for Version 5.1 (Release 2008a)

General Requirements

1

Computer Configuration	1-2
Timing Information	1-3
Installation Guide	1-4
Structure of This Document	1-5

PolySpace™ Client — Setting Up and Launching an Analysis of a Single Ada File

2

Overview	2-2
Analysis Prerequisites	2-3
Setting Up a PolySpace™ Client Analysis	2-4
PolySpace™ Client: Running the Analysis	2-10
Parsing Errors During Preliminary PolySpace™ Analysis	
Stages	2-12
Progression of the analysis	2-16
End of the Analysis	2-17

PolySpace™ Viewer — Exploration of Results

3

Overview	3-2
Modes of Operation	3-3
Download Results into the Viewer	3-5
Analyzing PolySpace™ Results in “Expert” Mode	
(example.adb)	3-7
Procedural Entities View (RTE View)	3-8
Colors in the Source Code View	3-14
More Examples of Run-Time Errors	3-15
Advanced Results Exploration	3-17
Miscellaneous	3-20
Methodological Assistant	3-21
Assistant Dashboard	3-22
Choose a Methodological Assistant	3-26
Report Generation	3-28

Launch PolySpace™ Remotely

4

Overview	4-2
Steps of Launching	4-3
Management of PolySpace™ Analysis in Remote: the PolySpace™ Spooler	4-5
Batch Commands	4-9
Launching an Analysis in Batch	4-9
Managing an Analysis in Batch	4-9

Share Analyses Between Accounts **4-12**
 Analysis-key.text File **4-12**
 Magic Key or Shared Analysis Between Projects **4-13**

Summary

5

General Requirements

Computer Configuration (p. 1-2)	Describes how to access hardware requirements
Timing Information (p. 1-3)	Describes how long it takes to perform this tutorial
Installation Guide (p. 1-4)	Describes the PolySpace™ products available on the installation CD
Structure of This Document (p. 1-5)	Describes the exercises included in this document

Computer Configuration

Please refer to the PolySpace™ installation manual for the minimum hardware requirements to follow this tutorial on a Microsoft® Windows® PC.

Timing Information

The installation of PolySpace™ products takes around 5 minutes (see the complete installation guide is available from the PolySpace installation CD-ROM in \Docs\Install\PolySpace_Installation_Guide.pdf).

The first step of this tutorial takes about 15 minutes.

The second step of this tutorial takes about 15 minutes.

Installation Guide

Note If the PolySpace™ products are already installed on your computer, please go directly to Chapter 2, “PolySpace™ Client — Setting Up and Launching an Analysis of a Single Ada File”.

The PolySpace products are delivered on a CD-ROM. There are 4 modules:

- *PolySpace™ Client™ for Ada* for analyzing single files. Note that this module is available with the icon “*PolySpace Launcher*”.
- *PolySpace™ Server™ for Ada* for multi-file or composite analysis. Note that this module is available with the icon “*PolySpace Launcher*”.
- *PolySpaceViewer* is the graphical user interface to explore the results computed by PolySpace Server or PolySpace Client.
- *PolySpace Spooler* is the graphical interface to manage analysis sent in remote.

Please refer to PolySpace Installation Guide for installing the PolySpace products.

Structure of This Document

Once the installation is done, you can launch PolySpace™ by using the following icons that were placed on your desktop PC:



This Getting Started will focus on the following three exercises using PolySpace Client, the Viewer and the launching of an analysis remotely:

- In Chapter 2, “PolySpace™ Client — Setting Up and Launching an Analysis of a Single Ada File” we will analyze a simple package “example” by using PolySpace Client.
- In Chapter 3, “PolySpace™ Viewer — Exploration of Results” we will review analysis results by using PolySpace Viewer.
- Chapter 4, “Launch PolySpace™ Remotely” instead of performing a PolySpace Client or Server analysis locally, analysis will be sent remotely to a server.

PolySpace™ Client — Setting Up and Launching an Analysis of a Single Ada File

Overview (p. 2-2)

Provides an overview of the exercise performed in this chapter

Analysis Prerequisites (p. 2-3)

Describes requirements for performing the analysis

Setting Up a PolySpace™ Client Analysis (p. 2-4)

Describes how to set up the analysis

PolySpace™ Client: Running the Analysis (p. 2-10)

Describes how to run the analysis

Overview

This chapter describes a basic file analysis. It focuses on the analysis of the “example” package, which is included in the PolySpace™ installation directory and located at:

PolySpaceInstallDir\Examples\Demo_Ada\sources\example.adb.

The PolySpace analysis process is composed of three main phases:

First, PolySpace software checks the syntax and semantic of the analyzed file(s). However, as PolySpace software is not associated to a particular compiler, **benefits** of this phase are triple for the analyzed source code: **Ada Standard compliance**, **portability** and **maintainability**.

Then, PolySpace software seeks the main procedure. If none is found, PolySpace Client will generate one automatically. This function will call all the functions which are declared in the specification of the package.

Finally, PolySpace software proceeds with the code analysis phase, during which run time errors are detected and highlighted in the code.

Analysis Prerequisites

Any analysis requires the following:

- PolySpace™ products and related license file and dongle correctly installed
- Source code files (in this case, `example.adb`) and all other specifications that it may directly or indirectly require

Setting Up a PolySpace™ Client Analysis

- 1 Double-click on the PolySpace Launcher icon (release number could not be same):

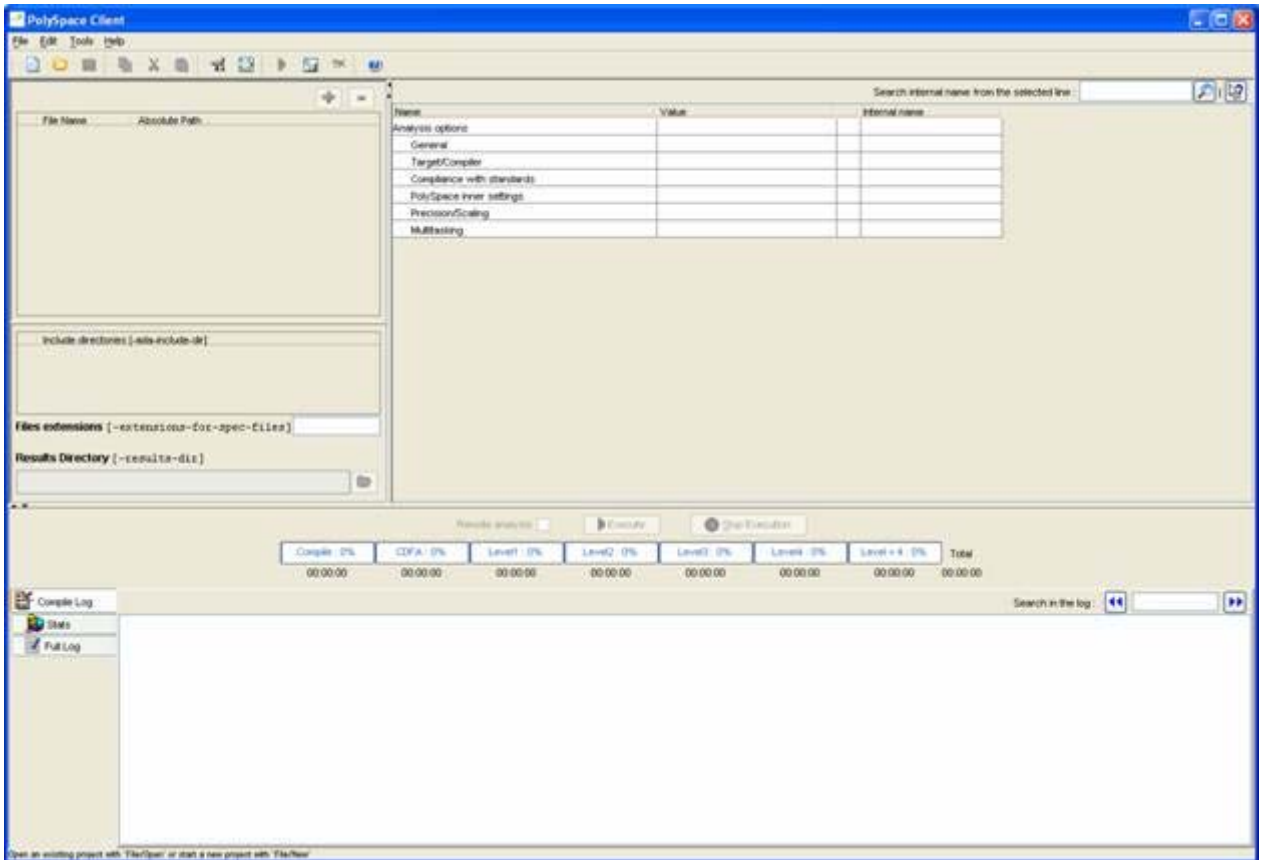


A dialog box window appears proposing to launch one of the following categories of analysis mixing the type of product and the language:



- 2 The language to select depends on available installed PolySpace™ products.

The Graphical Interface of PolySpace analysis Launcher is displayed as below after having chosen Client Launcher and Ada95:



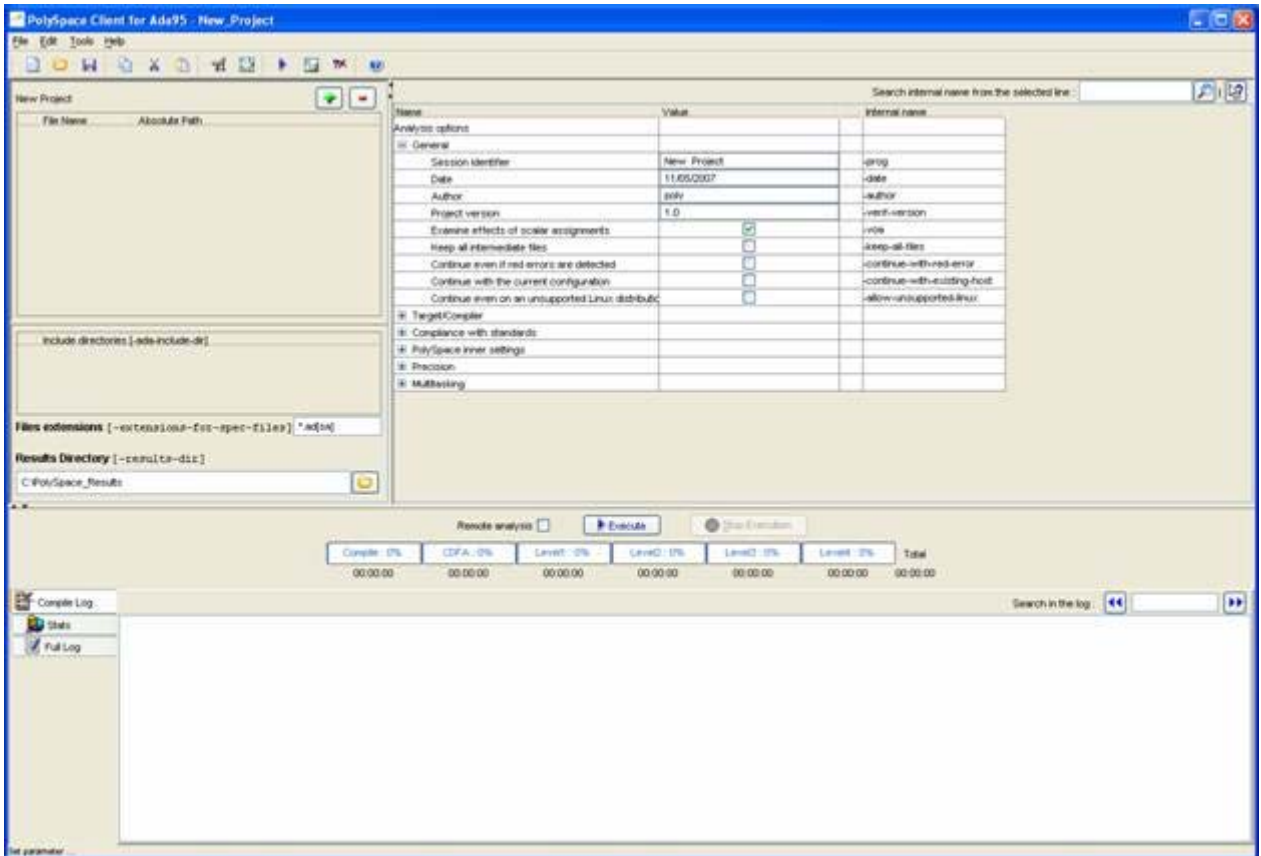
3 Click on File/New Project to start an analysis:




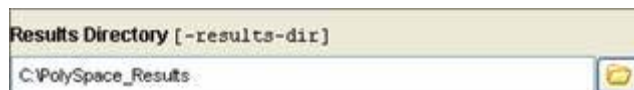
- 4** If required, Select Ada95 as the language and click on “OK”.

The PolySpace Client for Ada95 New Project window opens (see figure below). It contains four sections:


- At the very top, the title bar, which contains usual icons and menus;
- Top left is the list of files to analyze, along with include and results directories;
- Top right is the set of options associated with the analysis that will be processed;
- Finally the bottom area allows following the execution and progress of the analysis.



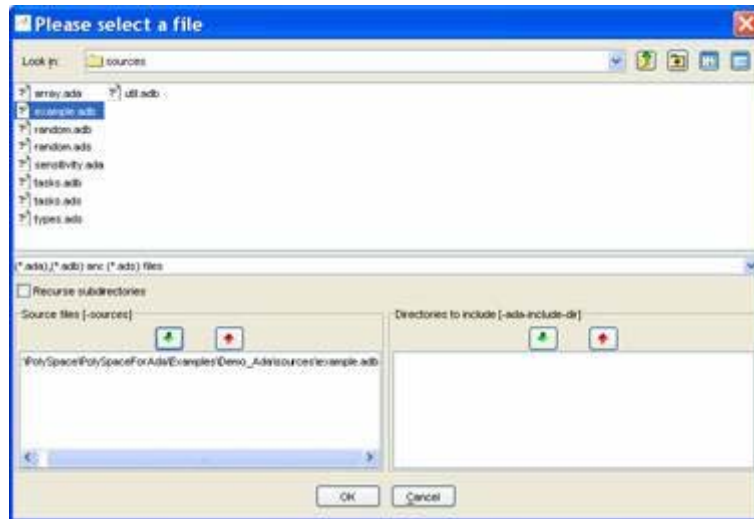
- 5 Start by updating the result directory name by clicking on the browse button .





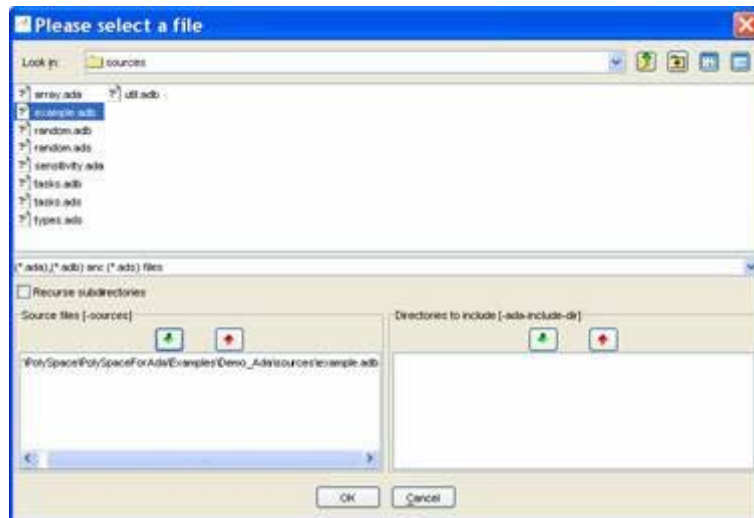
This directory is the one where PolySpace Client will store the results of the analysis. By default, PolySpace will store results in C:\PolySpace_Results. This is the directory that we will choose for the analysis.


- 6 Now, Click on the  button (right of the “New Project” label).

It opens the “Please select a file” window, from which you can select one or several files to analyze.




- 7 In the “Look in” section, click on , and select *PolySpaceInstallDir\Examples\Demo_Ada\sources*. A list of files appears in the box (*PolySpaceInstallDir* corresponds to *C:\PolySpace\PolySpaceForAda* in the figure above).
- 8 Select *example.adb* and click on  in the “Source files [-sources]” section (bottom left) of the window. The file is now listed among the source files to be analyzed.



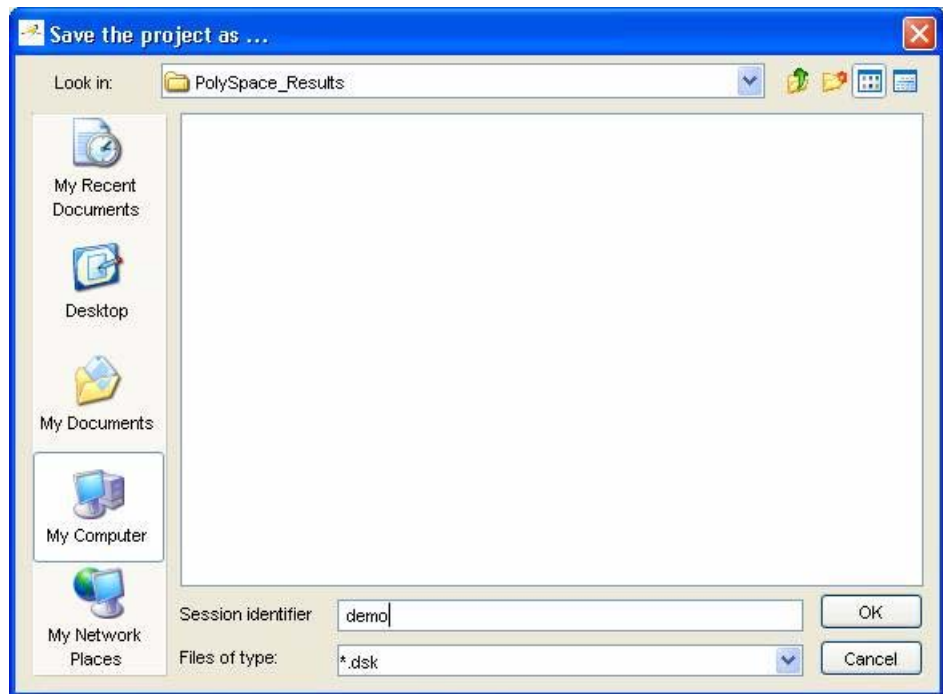
- 9 Click on  to go back to the “PolySpace Client for Ada95 - New_Project” window.

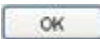

Note it is also possible to drag a directory or source files and drop them directly in the “File Name/Absolute Path” part (top left of PolySpace Client) without using the “Please select a file” window.

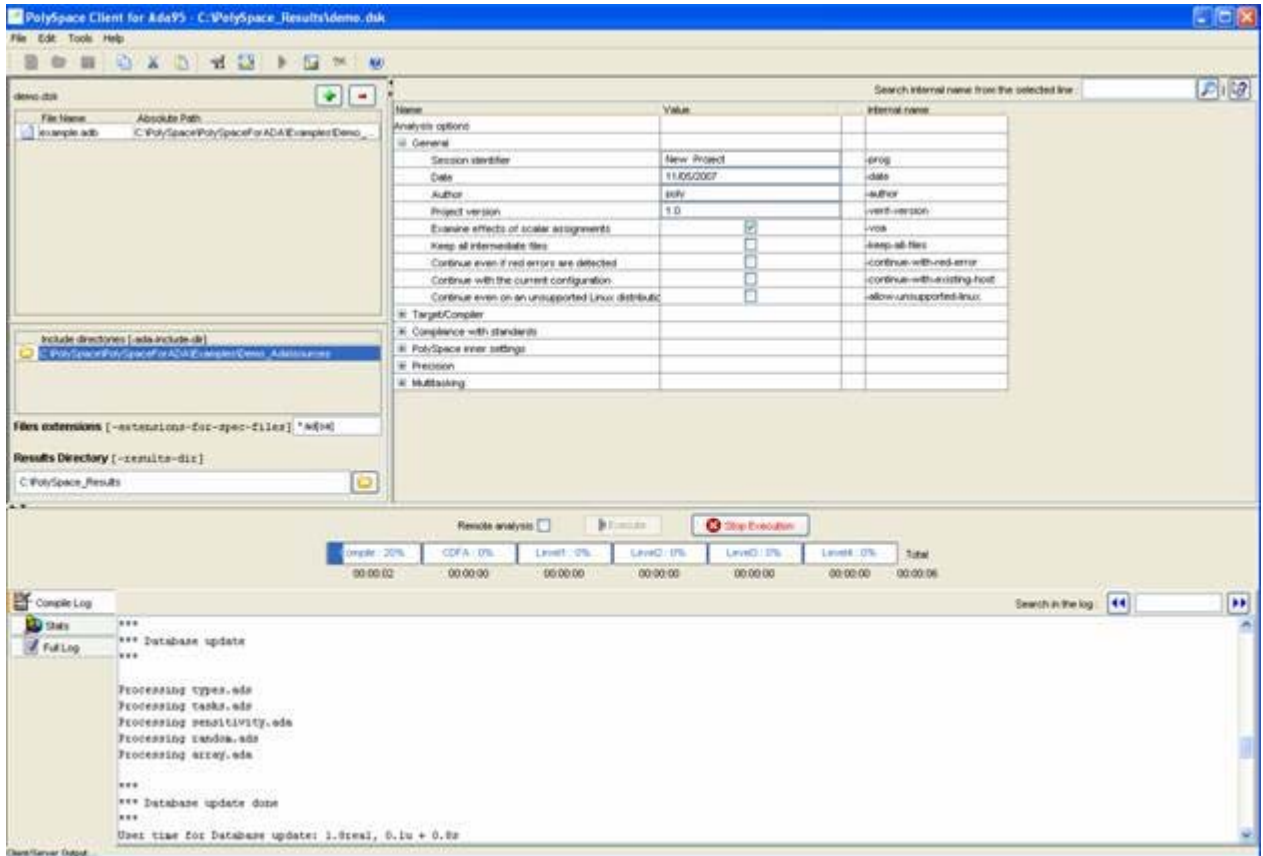
PolySpace™ Client: Running the Analysis


- 1 Click on  to start the analysis. Alternatively, you can click on the button in the title bar to run PolySpace™ Client with the current setting.


The window titled “Save the project as” opens. You can decide where to store the configuration information related to the analysis. Here, create a file called demo and save it under PolySpace result directory. The full name of that file will be demo.dsk.



- 2 Click on  to go back to the “PolySpace Client for Ada95 - New_Project” window and click again on  to proceed.



A progress report is displayed in the bottom part of the graphical interface, indicating that the analysis is being performed. The  button is also grayed out.

Note You may press the Stop Execution button -  to interrupt the analysis but it is not part of the current tutorial.

Parsing Errors During Preliminary PolySpace™ Analysis Stages

After some checks, PolySpace software will show an error message:




Let's try and understand why we get this error message.

First possible cause for the error message: Hardware recommendation

If this happens, please verify whether your computer fits the minimal hardware configuration requirements described in the general requirements. Moreover, a message like the following one is displayed in the bottom part of the graphical interface:



- 1 Type “host” in the “Search in the log:” box and click on  to search if the error corresponds to a hardware recommendation problem.

If the error message corresponds to the one shown above and in order to continue analysis, you can either:

- upgrade your computer to meet the minimal requirements, or
- use the `-continue-with-existing-host` option which overrides the initial check for minimal hardware configuration. To do so, please follow the following steps:

- 2 To set up the `-continue-with-existing-host` option, please type “continue” in the Search internal name from the selected line (top right box).

Search internal name from the selected line : 

- 3 Click .

It will show all options containing “continue” in the set of options part below:

Name	Value	Internal name
Analysis options		
[-] General		
Session Identifier	New Project	-prog
Date	18/05/2006	-date
Author	berd	-author
Project version	1.0	-vert-version
Examine effects of scalar assignments	<input type="checkbox"/>	-vob
Keep all intermediate files	<input type="checkbox"/>	-keep-all-files
Continue even if red errors are detected	<input type="checkbox"/>	-continue-with-red-error
Continue with the current configuration	<input type="checkbox"/>	-continue-with-existing-host
Continue even on an unsupported Linux distribution	<input type="checkbox"/>	-allow-unsupported-linux
[+] Target/Compiler		
[+] Compliance with standards		
[+] PolySpace inner settings		
[+] Precision/Scaling		
[+] Multitasking		

- 4 Check the box



in the “Value” column that is associated to the “`-continue-with-existing-host`” line as shown below.

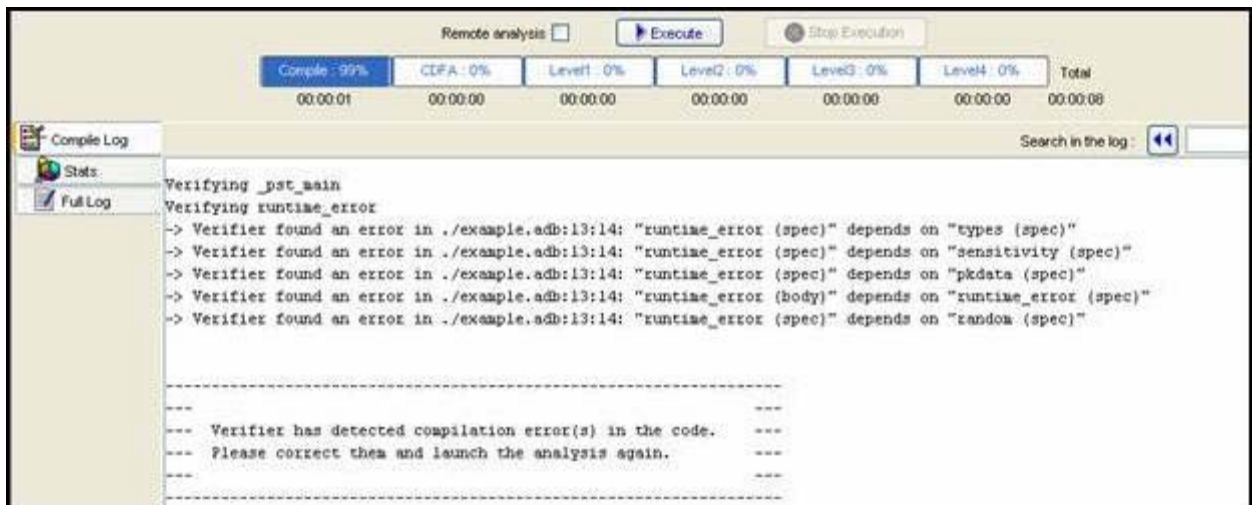
It is also recommended to select the `-continue-with-red-error` option. Indeed, example.adb contains — on purpose — code with some definite

errors, later called red errors. This option allows you to continue the analysis even if red errors are detected in previous passes.

Continue even if red errors are detected	<input checked="" type="checkbox"/>	-continue-with-red-error
Continue with the current configuration	<input checked="" type="checkbox"/>	-continue-with-existing-host

Second possible cause for the error message: Information about Header files

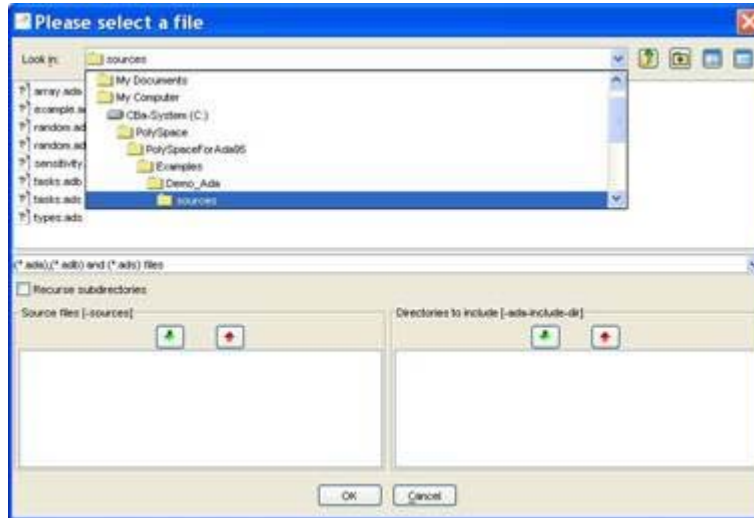
Another cause of error may be that PolySpace Desktop misses some package specifications.




In the tutorial, as shown above, some specification are missing: “types”, “sensitivity”, “pkdata”, “runtime_error” and “random”. To fix these compilation errors, you need to indicate where to find these specifications. As PolySpace is not associated with one particular compiler, it is mandatory to indicate where library files are stored.

In our example.adb file analysis, the related specifications are located in the same directory as the .adb file:
PolySpaceInstallDir\Examples\Demo_Ada\sources.

- 1 Open the “Please select a file” window by using button (right of the “demo.dsk” label in the top right of the interface):



- 2 Select “*PolySpaceInstallDir*\Examples\Demo_Ada\sources”, where the specifications are located.
- 3 Click on  in the “Directories to include [-ada-include-dir]” section, then click **OK** to close the window.

Note All specifications are in this folder only. It is also possible to drag a directory and drop it directly in the “include directories [-ada-include-dir]” part (top left of PolySpace Client) without using the “Please select a file” window.

At the end, a last compilation error remains:

```
Command : rte-fel -feT0 -feS -fe95 -feVF -fea -quiet -fed3 -febX -feh -fec -fet -feo
Verifying _got_main
Verifying runtime_error :
-> Verifier found an error in ./example.adb:13:14: "runtime_error (body)" depends on "runtime_error (spec)"
-> Verifier found an error in ./example.adb:13:14: "runtime_error (spec)" depends on "sensitivity (spec)"
-> Verifier found an error in ./example.adb:13:14: "sensitivity (spec)" depends on "pkutil (spec)"

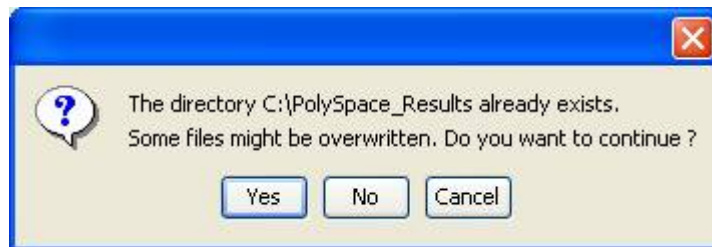
-----
--- Verifier has detected compilation error(s) in the code. ---
--- Please correct them and launch the analysis again. ---
```

It means that “pkutil” specifications are missing even the -ada-include-dirdirectory added just before. Searching for “pkutil” specifications, in the sources directory, we can see that it is defined in the util.adb file. Changing the following option “Files extension” by “-extensions-for-specs-file*.ad[sab]” allows you to indicate that specifications can be found in *.ada, *.ads and *.adb file extensions.

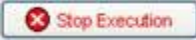
Progression of the analysis

1 Click on  to restart the analysis.

If you previously clicked **Execute**, some results may have already been written in the C:\PolySpace_Results directory. Therefore, a window opens to check whether you want to overwrite in this directory or not:







2 If this happens, click **Yes**.

Note Closing the PolySpace Desktop window will not stop the PolySpace analysis. If you wish to stop it, click  (a window of confirmation follows the click). If the window is closed without stopping the analysis, it continues in background. Opening again PolySpace Desktop with the same project automatically updates the analysis with its current status.

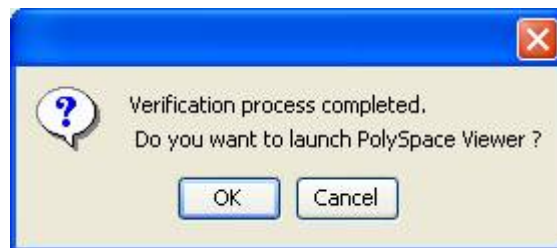
The progress bar allows to follow the progress of the analysis:




- 3 To obtain a progress report, click on  for the compilation phase, or  for the full analysis in the low level window.
- 4 Click  to get other pieces of information about current analysis (list of options, stubbed functions, functions used during main construction, checks found after each phase, etc.).
- 5 Click the  icon to refresh the summary.

End of the Analysis

When the analysis ends, PolySpace software proposes to review the results:



If you Click **OK**, go to the next section of the tutorial to view the results.

If you click **Cancel**, and no other analyses are running, you can access the results via the  icon in the title bar.

PolySpace™ Viewer — Exploration of Results

Overview (p. 3-2)	Provides an overview of the exercise performed in this chapter
Modes of Operation (p. 3-3)	Describes the two ways you can review analysis results
Download Results into the Viewer (p. 3-5)	Describes how to open results into the viewer
Analyzing PolySpace™ Results in “Expert” Mode (example.adb) (p. 3-7)	Describes how to use Expert mode
Methodological Assistant (p. 3-21)	Describes how to use the methodological assistant
Report Generation (p. 3-28)	Describes how to generate reports containing analysis results

Overview

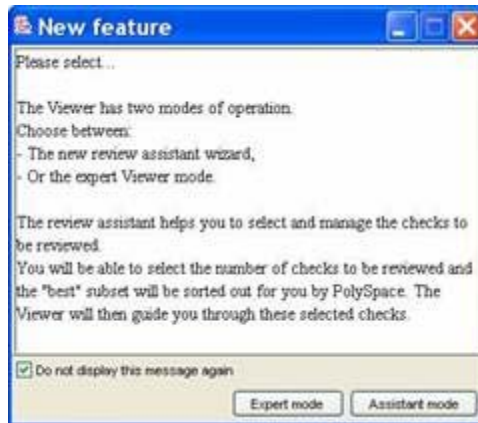
This step illustrates how to explore analysis results that were generated by either PolySpace™ Client or PolySpace Server. We review the results of the analysis of `example.adb` performed in Chapter 2, “PolySpace™ Client — Setting Up and Launching an Analysis of a Single Ada File” using the following icon:



If you clicked **OK** at the end of the previous analysis (see previous section), PolySpace Viewer automatically opens results. Please go directly to “Analyzing PolySpace™ Results in “Expert” Mode (`example.adb`)” on page 3-7.

Modes of Operation

The first time The PolySpace™ Viewer is opened, a sub-window will appear after the splash screen of the viewer. It is aimed to warn user about different modes of operation. User has to choose between launching the Viewer in an “expert” mode or in an “assistant” mode.



The mode will define the reviewing process of checks highlighted during an analysis:

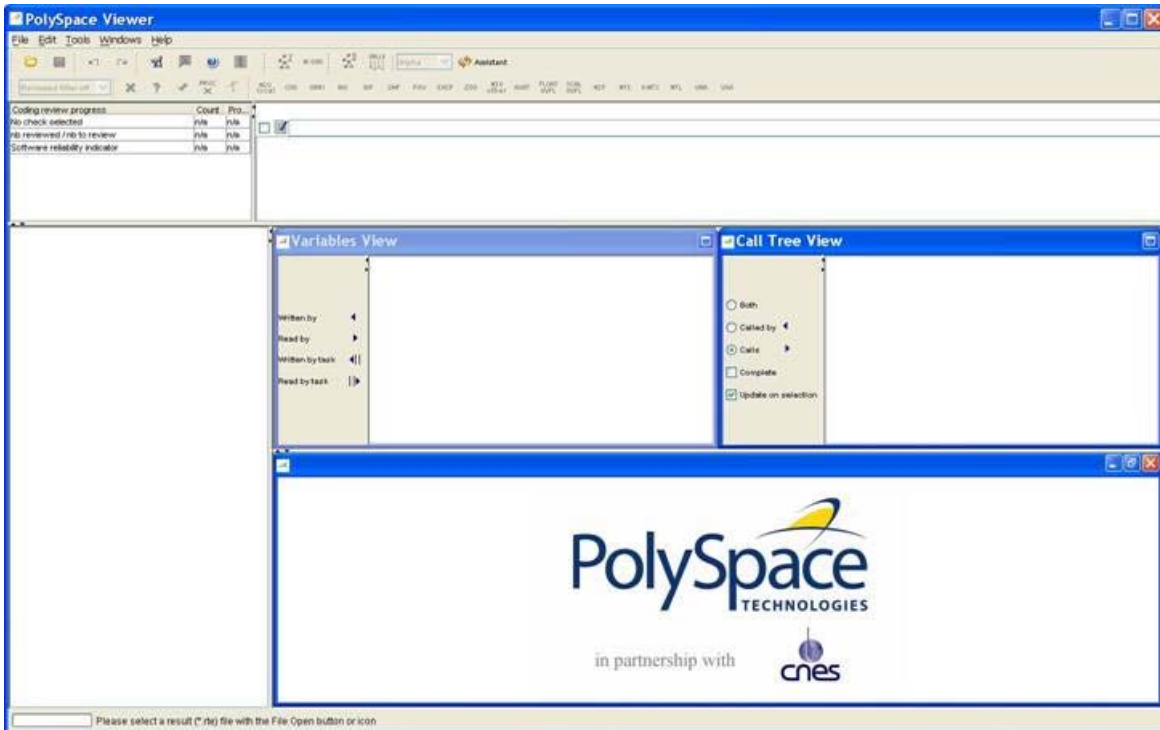
- **Expert mode** — The Viewer is opened in a mode where all checks can be seen. The number, the order and the categories of checks can be reviewed can be chosen by the user himself (See next section).
- **Assistant mode** — the reviewing rules for an Ada analysis results follows a methodology selected by PolySpace. It concerns the “best” subset of checks sorted out for user. The PolySpace Viewer will then guide user through these selected checks. First selected checks concern the Automatic Methodology.

For the need of this tutorial, please untick “Do not display this message again” and then click on “Expert mode”.

Note Even if the user has chosen one mode it is easy in one click to change the mode inside the PolySpace Viewer.

Download Results into the Viewer

After having clicked on “Expert mode” the PolySpace™ Viewer window looks like the figure below:

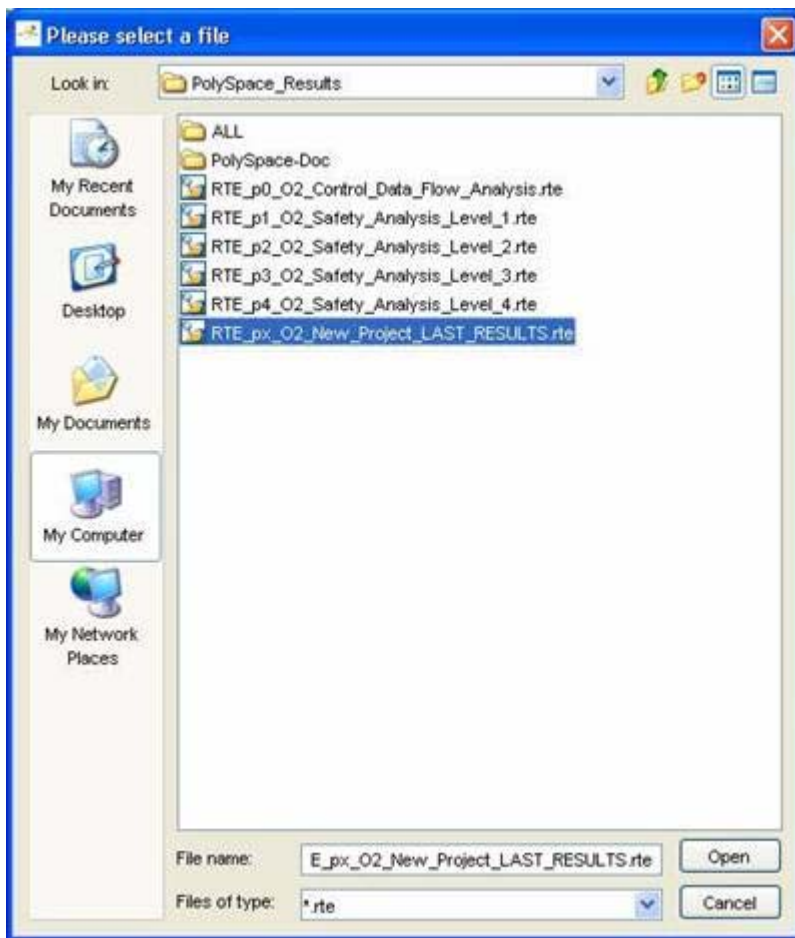


- 1 Click **File > Open** to load result files.

Note If you did not perform the analysis, you can still review the results by opening the following file:

```
PolySpaceInstallDir\Examples\Demo_Ada\
RTE_px_02_Demo_Ada_LAST_RESULTS.rte
```

- 2 Select the following file located in C:\PolySpace_Results.

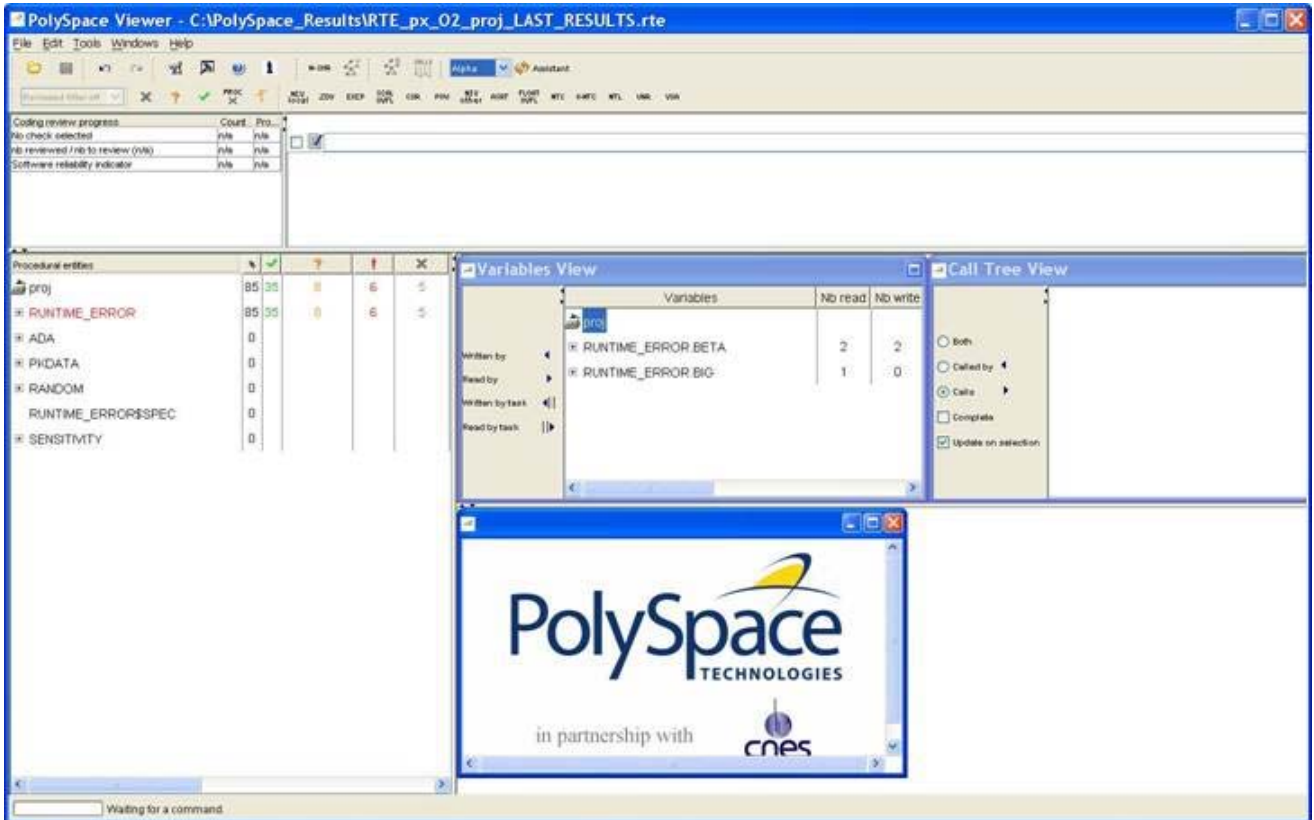


3 Click **Open** to proceed with further steps

Note The RTE_px_02_Demo_Ada_LAST_RESULTS.rte file is a sort of “link” on the best analysis in terms of precision. This analysis is represented by the RTE_p4_02_Safety_Analysis_Level14.rte file. Lower level files represent lower precision analysis.

Analyzing PolySpace™ Results in “Expert” Mode (example.adb)

After loading the results, and PolySpace™ Viewer window looks like below:

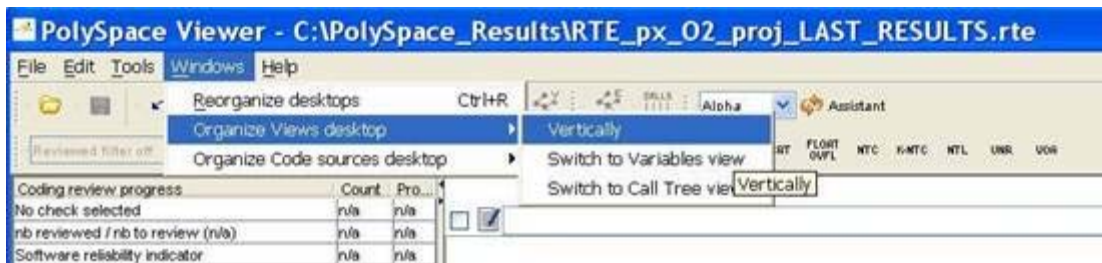


- On the left is the Procedural entities view (or RTE view). It displays the list of packages which have been analyzed or used during the analysis (specifications).
- In the bottom right area is the source code view with colored instructions. Each operation checked is displayed using meaningful color scheme and related diagnostic:

- **Red** — Errors which occur at every execution.
- **Orange** — Warning - an error may occurs sometimes.
- **Grey** — Shows unreachable code.
- **Green** — Error condition that will never occur.
- The two windows just below the tool bar concern details of a currently reviewed check (when the check has been selected):




- The top right area is used for displaying both control and data flow results. You can switch from one view to the other by using the “Windows” menu:



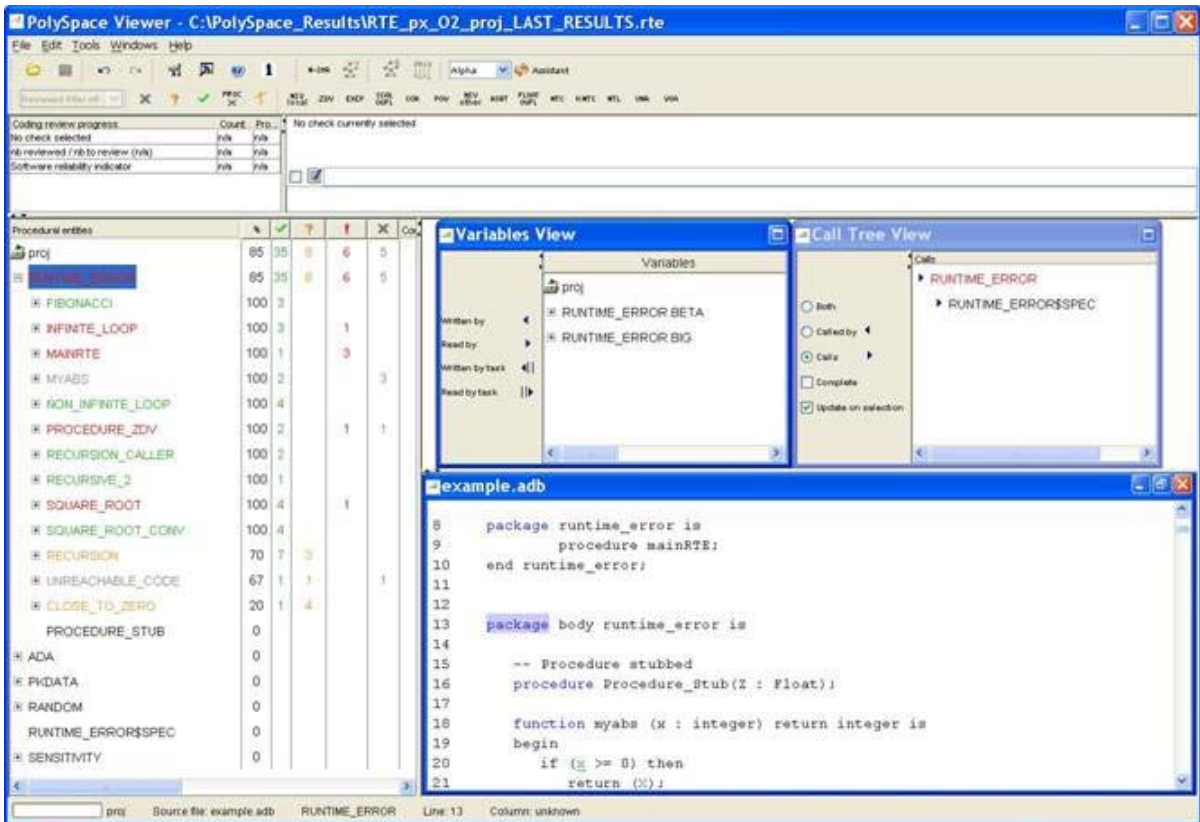
Procedural Entities View (RTE View)






Each package and underlying functions in the RTE view is colored according to the most critical error found:






- In black color — The packages specification has been used to perform analysis
- In red color — The package is red; one or more *definite* run-time errors have been found in it.

Click once on the  left of “RUNTIME_ERROR” to find out more about this package.

“RUNTIME_ERROR” is expanded and the list of functions defined within “RUNTIME_ERROR” is displayed. The functions in red or grey have code sections that need to be inspected (PROCEDURE_ZDV, SQUARE_ROOT, etc.) first because they are definite diagnosis of PolySpace (either runtime errors or dead code).




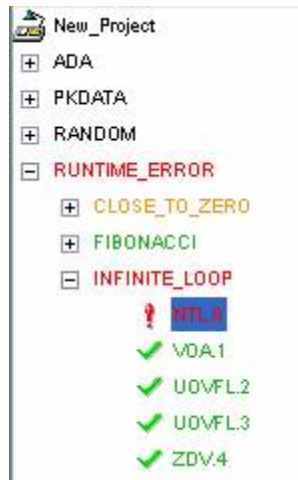
The columns (, , , , and ) provide information about run-time errors found in each function. The following table describes each of these columns.

Column	Indicates
	Reliability of the code (level of proof).
	Number of definite run-time errors or reds.
	Number of warnings or oranges (that may hide run-time errors that do not occur systematically).
	Number of safe operations or greens.
	Number of unreachable instructions or grey code sections.

Lets have a look at some errors found by PolySpace in the analyzed package.

First Example of Runtime Error Found by PolySpace™ Software: Memory Corruption

- 1 Click on  to expand “INFINITE_LOOP “ to find out more about the red error. It displays a list of red, green, and orange symbols, featuring the complete list of code areas that PolySpace checked within the “INIFINITE_LOOP” function inside package RUNTIME_ERROR.



- 2 Click on the red “NTL.0” item - which stands for **Non-Termination of Loop** -, to precisely locate this error in the source code. The bottom right section is updated showing the location of the “NTL.0” item.

3 PolySpace™ Viewer — Exploration of Results

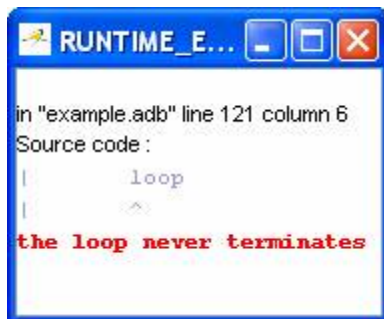
The screenshot displays the PolySpace Viewer interface. At the top, the title bar reads "PolySpace Viewer - C:\PolySpace_Results\RTE_px_O2_proj_LAST_RESULTS.rte". Below the menu bar, a coding review progress table is visible:

Coding review progress	Count	Pro...
nb NTL reviewed / nb NTL to review (Red)	0/1	0
nb reviewed / nb to review (Red)	0/6	0
Software reliability indicator	3554	64

Below the table, a message states "the loop never terminates" with a red "loop" annotation above it. The "Procedural entities" pane on the left lists various entities with their counts and error indicators. The "Variables View" and "Call Tree View" panes are also visible. The main source code editor shows the following code:

```
114     end Non_Infinite_Loop;
115
116     -- Infinite loop
117     procedure Infinite_loop is
118       x : integer := 1;
119       y : integer := 2;
120     begin
121       loop
122         exit when x < 0;
123         if (Random.random > 0) then
124           x := (x + myabs(x))/myabs(x); -- x is always positive
125         end if;
126       end loop;
127       -- this code section is unreachable
128       Recursion(v):
```

3 Click on red **loop** in the source code at line 121. An error message is opened:



Indeed, the condition to exit the loop is that “x” becomes a negative value.

```

121           loop
122            exit when x < 0;
123            if (Random.random > 0) then
124                x := (x + myabs(x))/myabs(x); -- x is always positive
125            end if;
126           end loop;

```

But according to the formula, “x” will always be a positive value. So, the loop can not terminate.

Second Example of Runtime Error Found by PolySpace™ Software: Unreachable Code

Select “UNREACHABLE_CODE” in the RTE View. You can see that the division “z := x / y” is unreachable (gray color on the check) because of the non satisfied Boolean condition: “x” is never negative when evaluating “x<0”. PolySpace has detected some dead code.

```
169      -- Here we demonstrate PolySpace Verifier's ability to
170      -- identify unreachable sections of code due to the
171      -- value constraints placed on the variables.
172      procedure Unreachable_Code is
173          x : integer := Random.random;
174          y : integer := Random.random;
175          Z : Integer;
176      begin
177          if (x > y) then
178              x := x - Y;
179              if (x < 0) then
180                  Z := x / Y;
181              end if;
182          end if;
183      end Unreachable_Code;
184
```

Colors in the Source Code View

Each operation checked is also displayed using meaningful color scheme and related diagnostic in the source code view as links:


- **Red** — A link to the error message associated to the error which occurs at every execution.
- **Orange** — A link to an unproven message - an error may occur sometimes.
- **Grey** — A link to a check shown as unreachable code. The error message is in grey.
- **Green** — A link to a VOA (Value on Assignment) or an error condition that will never occur.
- **Black** — Represents some comments, source code that does not contain any operation to be checked by PolySpace in terms of run time errors and optimized operations, e.g. `x := 0;`
- **Blue** — Text highlighting the keyword “procedure” and “function”

- **Blue Underlined** — A link to a global variable in the “Global variable View”. In the next figure (see next paragraph), **Beta** is an example of this kind of global variable.

More Examples of Run-Time Errors

Unlike most other testing techniques, PolySpace provides the benefit of finding the exact location of run-time errors in the source code. Below are some examples that you can review with PolySpace Viewer.

In an First Example of the Second Set: Arithmetic Error

Click  to expand “SQUARE_ROOT” function. You can see the source code view in the bottom right.

You can also display the call tree for that function by using the “Windows” menu (see previous paragraph).

“SQUARE_ROOT” is called by MAINRTE function. It is displayed in the “*Call tree view*” window (right of the top right section).

“SQUARE_ROOT” calls “RANDOM.random“ (automatically stubbed function), “SQUARE_ROOT_CONV” (from RUNTIME_ERROR package) and “SQRT” (from the standard library).

```
151  -- The table provided below the example shows the domain of
152  -- values for the expressions in the example.
153  procedure Square_Root_conv (alpha : in float; y : out long_float) is
154  begin
155      y := (1.5 + cos (long_float(alpha)))/5.0;
156  end Square_Root_conv;
157
158  Beta : Long_Float;
159  procedure Square_Root is
160      Alpha : Float := Random.random;
161      Gamma : long_float;
162  begin
163      Square_Root_conv (Alpha, Beta);
164      Beta := Beta - 0.75;
165      Gamma := sqrt(Beta); -- always sqrt(negative number)
166  end Square_Root;
167
```

The green sections into the source code view are error-free but the red (**sqrt**) is an issue that needs to be fixed. Indeed, when the local float variable gamma is computed in the line “gamma=sqrt(beta - 0.75);”, the operation will cause a run-time error, as the parameter passed to “sqrt” is always negative.

Note Using -voa option at launching time, PolySpace can help more suitably by giving information of range on scalar assignment.

Second Example of the Second Set: Non-Infinite Loop

Select “NON_INFINITE_LOOP” in RTE View. The function is fully green: it means that the locale variable x never overflows, even if the exit condition of loop deals with y that is smaller than x. PolySpace confirms that the function always terminates.




```

104  procedure Non_Infinite_Loop (X : out Integer) is
105      cur : Integer :=0;
106  begin
107      X := 0;
108      loop
109          exit when x > big;
110          cur := cur + 2;
111          x := cur / 2;
112      end loop;
113      X := Cur / 100;
114  end Non_Infinite_Loop;
115

```

Advanced Results Exploration

You can filter the information provided by PolySpace to focus on the type of errors you wish to investigate.


There are pre-defined composite filters ( ,  and  that you can choose depending on your development process. These filters are accessible through a combo list:



To illustrate the use of these filters, we will focus on the Square Root function that we have examined in the previous section.

Gamma Mode

Gamma mode provides all the “red” and “grey” code sections. It is mainly used during the earliest development stages to focus quickly on critical bugs.

To select Gamma mode, click the  button.

The software reduces the information checks related to “SQUARE_ROOT”.





This list of acronyms - for type of operations checked - shows what PolySpace automatically analyzed for you.

Beta Mode

Beta mode highlights checks that could cause a processor halt, memory corruptions or overflows. Beta mode is the default mode.

To select Beta mode:

- 1 Click .
- 2 Select “RECURSION” in the “Procedural entities” view.
- 3 Click  to get the list of the checks.



Alpha Mode

Alpha Mode provides a comprehensive list of operations checked by PolySpace.



To switch to Alpha mode, click



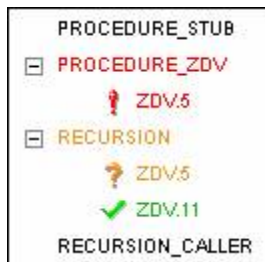
You may also want to use filters to focus on particular categories of errors. Those filters are located at the top of the PolySpace Viewer window:




Note When the mouse pointer moves on the filter, a tool tips gives its definition.

- Click  (top of the window) to suppress all checks, then click .

You will get list of checks containing only ZDV (Zero DiVision) reds, oranges, or greens:



- Click  (top of the window) to suppress green code sections. You will get a reduced list of checks reds, oranges and grays:



Miscellaneous

The  icon gives access to the PolySpace Manual. All views have a pop-up menu (right click on mouse).

Close the PolySpace Viewer window by clicking on the upper right



symbol (PolySpace Viewer can also be closed using **File > Close**).


Methodological Assistant

After a first navigation into the PolySpace™ Viewer, some simple questions remain:

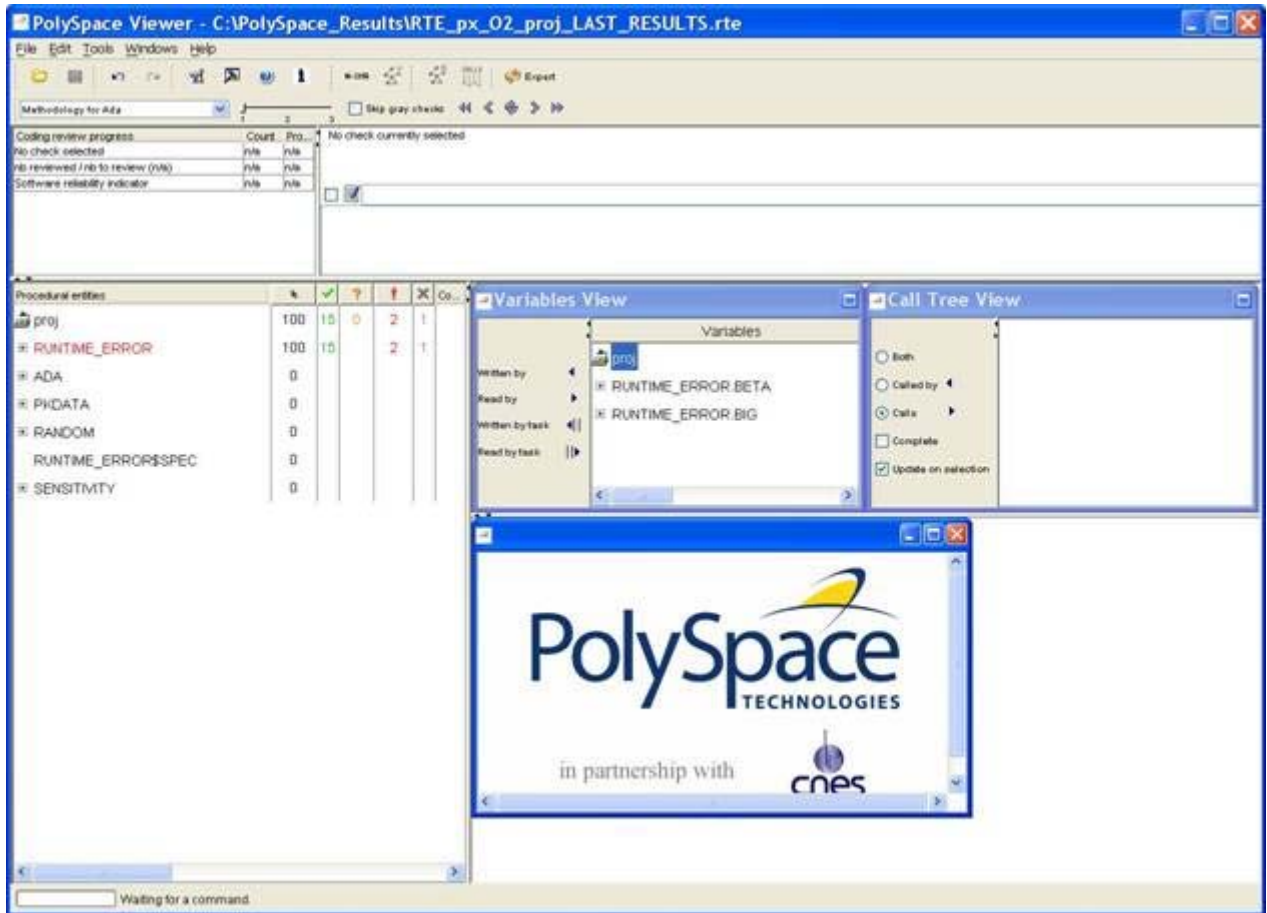
- Do all checks need to be reviewed?
- What are the checks to review?
- How many?
- What is the best order?

The Methodological assistant is here to answer to all these questions: It helps to select and manage the checks to be reviewed. It selects a “best” subset and sorts out them. The Assistant mode in the PolySpace Viewer will then guide through these selected checks.

To open the assistant:

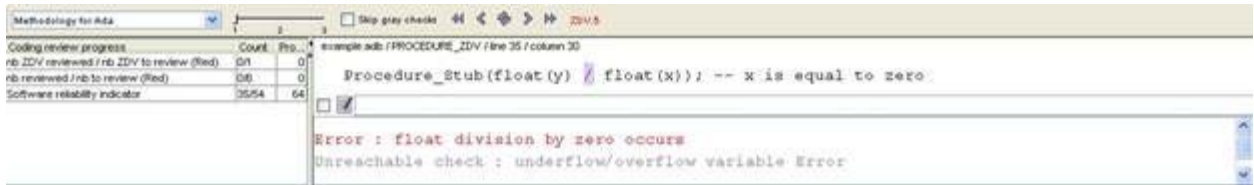
- 1** If the PolySpace Viewer is still open, close it by clicking on the upper right  symbol.
- 2** Open the PolySpace Viewer again, then load the same results.
- 3** Choose “Assistant” mode.

After having loaded the results in “Assistant” mode, PolySpace Viewer window looks like below:




Assistant Dashboard

The second line of buttons on the toolbar and the two views just below are the navigation centre based on the methodological method used in the assistant mode:



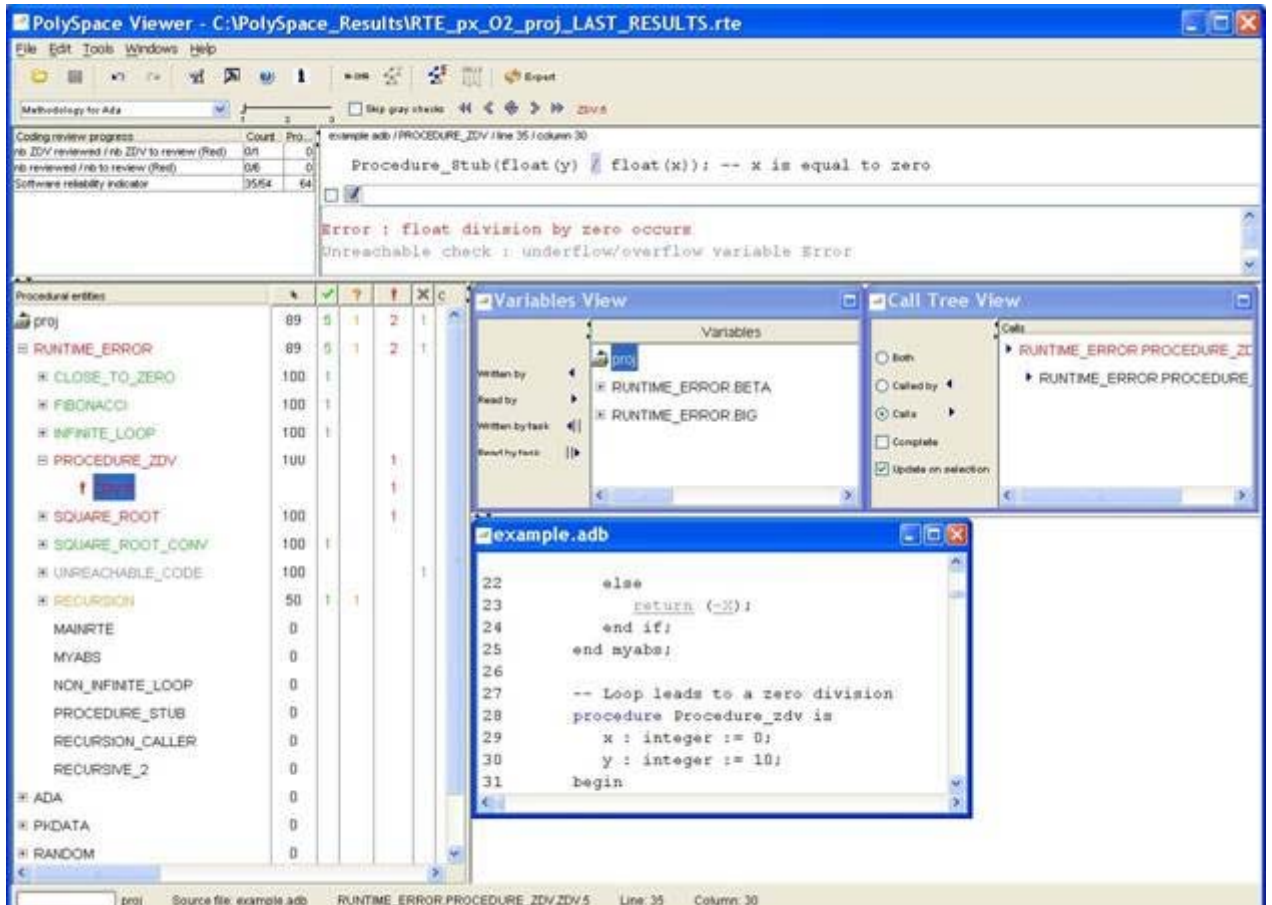
Some other changes can be seen in the viewer:

- Now, in the “Procedural Entities” view the list of files analyzed *is sorted by the methodological assistant* used.
- In the bottom right area is the source code view with colored instructions. Each operation will be checked and sorted by the methodological method using meaningful color scheme and related diagnostic and in the following order:
 - **Red** — Assistant browses all errors which occur at every execution.
 - **Gray** — Assistant browses each block of unreachable code depending if radio button “Skip gray checks” has been ticked or not.
 - **Orange** — Assistant chooses and reviews the “best” unproven operations -errors that may occur sometimes.

- 1 Click  to navigate to next check.

The PolySpace Viewer has been refreshed with the first check selected by the Methodology of review:

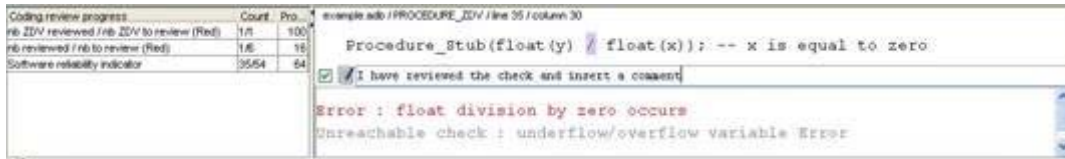
3 PolySpace™ Viewer — Exploration of Results



The Methodological dashboard gives details and allows reviewing the check. On the selected check, it is possible to mark the fact that it has been reviewed.

- 2 Select the radio button box.
- 3 Enter a comment in the associated edit box on the right.

After, it looks like:



The left part of the dashboard has been updated, and displays some statistics in three lines:

- The first line gives the number and percentage of remaining checks to review of the current category. In the previous example, it concerns red IDP checks.
- The second line gives values in the color category (red, grey and unproven).
- The Last line gives in permanence the Software reliability indicator.

Other buttons in the Methodological dash board allow navigating to previous check, coming back to current one



and going to next

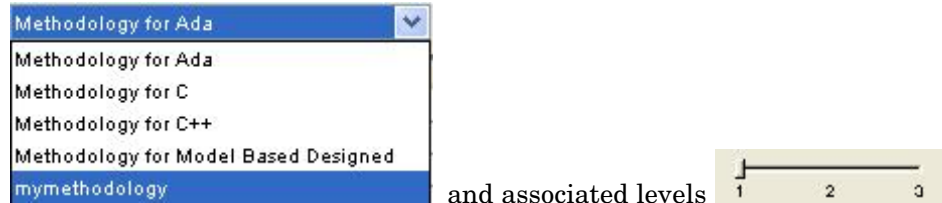


/ previous



category of reviewed checks selected by the Methodology.

Choose a Methodological Assistant



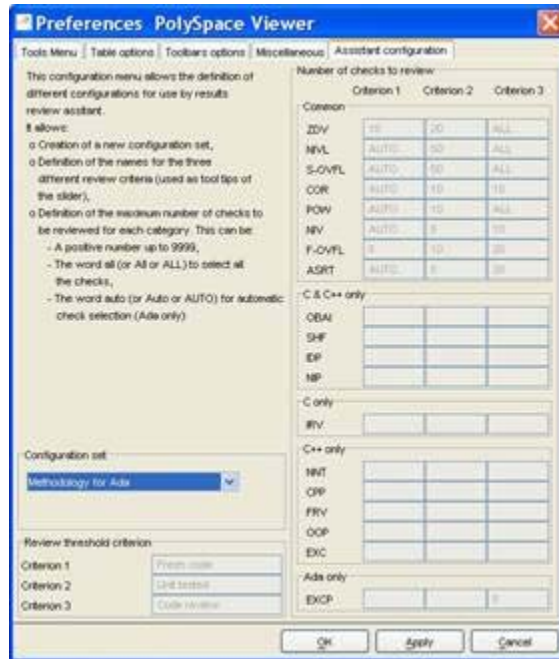
have been pre-selected by PolySpace.

The methodology allows selecting the categories of checks to review, the number for each category and their order depending of a statistical algorithm.

The level (or criterion) defines the number of checks to review by category. Explicit name have been associated to each criterion like “Fresh code”, “Unit test” and “Code review”

It is possible to refine a self-created one or define its own Methodology.

- 1 Select **Edit > Preferences** in the PolySpace Viewer.
- 2 Select the Assistant Configuration tab.



3 Create a new configuration set

Define the categories of check to review for each criterion, how many in each one.

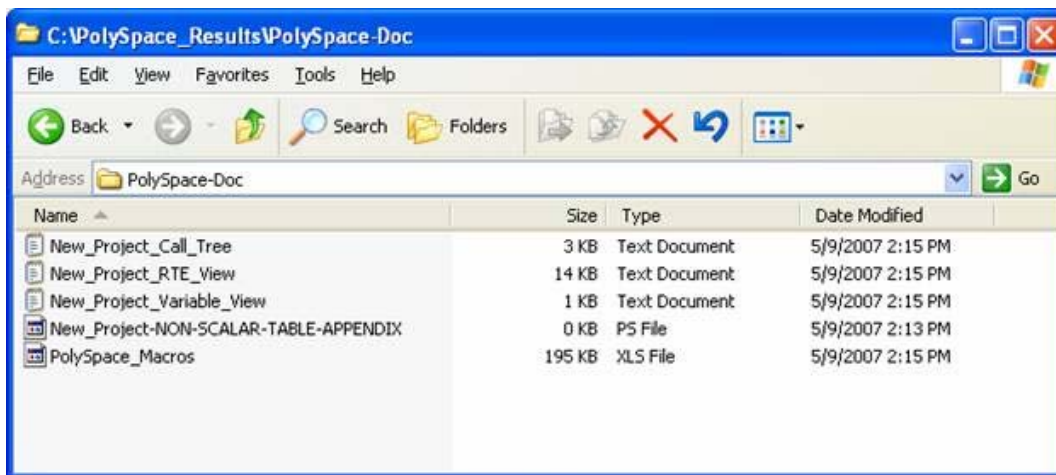
Note You can not change an existing configuration except by duplication and refinement.

Report Generation

When PolySpace performs an analysis, it generates textual files that can be used to generate Microsoft® Excel® reports. These files are located in the results directory (See C:\PolySpace_Results\PolySpace-Doc or *PolySpaceInstallDir*\Examples\Demo_C\PolySpace-Doc).

All views (except source code) are printable and can be exported to textual or Excel format (protected by license).

The C:\PolySpace_Results\PolySpace-Doc directory should contain the following files:



To generate a report:

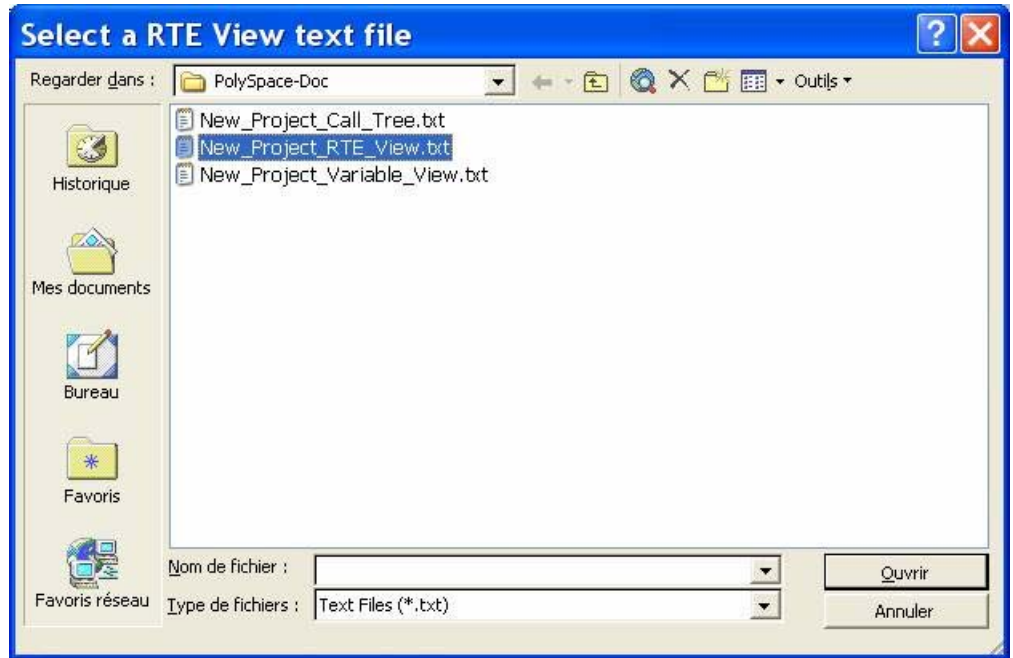
- 1 Open the file called `PolySpace_Macros.xls`, enable macros when asked and then the following window opens:

	A	B	C	D	E	F	G	H
1								
2	Copyright © PolySpace Technologies, 1999-2006							
3								
4	<div style="border: 1px solid black; padding: 10px;"> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Apply filters? <input type="checkbox"/></p> <p><input checked="" type="radio"/> No filters</p> <p><input type="radio"/> Beta filters</p> </div> <div style="width: 45%;"> <p>Generate checks by file? <input type="checkbox"/></p> <p><input checked="" type="radio"/> yes</p> <p><input type="radio"/> no</p> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="border: 1px solid gray; padding: 2px 5px;">Help</div> <div style="text-align: center;"> <p>Use this button to create the complete synthesis in one file. Select the RTE export view and a file in which to save results. If the other views are in the same directory as the RTE view then they will automatically be incorporated into the same file.</p> </div> <div style="border: 1px solid gray; padding: 2px 5px;">Help</div> </div> <div style="text-align: center; margin-top: 10px;"> <div style="border: 1px solid gray; padding: 5px; display: inline-block; color: red; font-weight: bold;">Generate PolySpace Results Synthesis</div> </div> </div>							
9								
10								
11								
12								
13								
14								
15								
16								
17	Reports can be generated from all PolySpace txt file format results. These are generated							
18	by the PolySpace Verifier during an analysis, the export option in the PolySpace Viewer,							
19	or from the command line using the "gen-excel-files" command.							
20								
21	Individual PolySpace text result files can be processed using the below macros:							
22								
23	<u>The macros are:</u>							
24	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid gray; padding: 2px 5px; background-color: #e0e0e0;">RTE</div> <div>Apply to RTE views exported from PolySpace Viewer</div> </div>							
25	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid gray; padding: 2px 5px; background-color: #e0e0e0;">Call Tree</div> <div>Apply to Call Tree views exported from PolySpace Viewer</div> </div>							
26	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid gray; padding: 2px 5px; background-color: #e0e0e0;">Variables</div> <div>Apply to Variable views exported from PolySpace Viewer</div> </div>							
27								
28								
29	Version 3.4.1D				RTE = Run Time Error			
30								

2 Click on

Generate PolySpace Results Synthesis

A file browser opens.



3 Select the file called `New_Project_RTE_View.txt`.

After a few seconds, an Excel file is generated. It contains several spreadsheets related to the application analyzed.



For example, in “Checks Synthesis” all statistics about checks and colors are reported in a summary table.

	A	B	C	D	E	F	G
1	RTE Statistics						
2	Check category	Check detail	R	O	Gy	Gr	% proved
3	OBAI	Out of Bounds Array Index	0	0	0	0	0,00%
4	NIVL	Uninitialized Local Variable	0	0	1	28	100,00%
5	IDP	Illegal Dereference of Pointer	1	1	0	7	88,89%
6	NIP	Uninitialized Pointer	0	0	0	12	100,00%
7	NIV	Uninitialized Variable	0	0	0	8	100,00%
8	IRV	Initialized Value Returned	0	0	0	15	100,00%
9	COR	Other Correctness Conditions	0	0	0	2	100,00%
10	ASRT	User Assertion Failure	0	0	0	0	0,00%
11	POW	Power Must Be Positive	0	0	0	0	0,00%
12	ZDV	Division by Zero	0	1	0	4	80,00%
13	SHF	Shift Amount Within Bounds	0	0	0	0	0,00%
14	OVFL	Overflow	0	3	2	8	76,92%
15	UNFL	Underflow	0	1	2	9	91,67%
16	UOVFL	Underflow or Overflow	0	3	0	4	57,14%
17	EXCP	Arithmetic Exceptions	0	0	0	0	0,00%
18	NTC	Non Termination of Call	3	0	0	0	100,00%
19	k-NTC	Known Non Termination of Call	0	0	0	0	0,00%
20	NTL	Non Termination of Loop	0	0	0	0	0,00%
21	UNR	Unreachable Code	0	0	0	0	0,00%
22	UNP	Uncalled Procedure	0	0	0	0	0,00%
23	IPT	Inspection Point	0	0	0	0	0,00%
24	OTH	other checks	0	0	0	0	0,00%
25	Total :		4	9	5	97	92,17%

This ends ways of results review.

Launch PolySpace™ Remotely

Overview (p. 4-2)	Provides an overview of the exercise performed in this chapter
Steps of Launching (p. 4-3)	Describes how to perform an analysis using PolySpace™ Server™ for Ada
Management of PolySpace™ Analysis in Remote: the PolySpace™ Spooler (p. 4-5)	Describes how to manage remote analyses
Batch Commands (p. 4-9)	Describes how to perform and manage analyses in Batch
Share Analyses Between Accounts (p. 4-12)	Describes how to share the results of a Server analysis between multiple Client users

Overview

This section describes the basic steps to launch an analysis in remote.

To do so you need:


- A Queue Manager server (QM) installed.
- Your desktop PC configured with a PolySpace™ Client.
- A networked machine configured with a PolySpace Server.

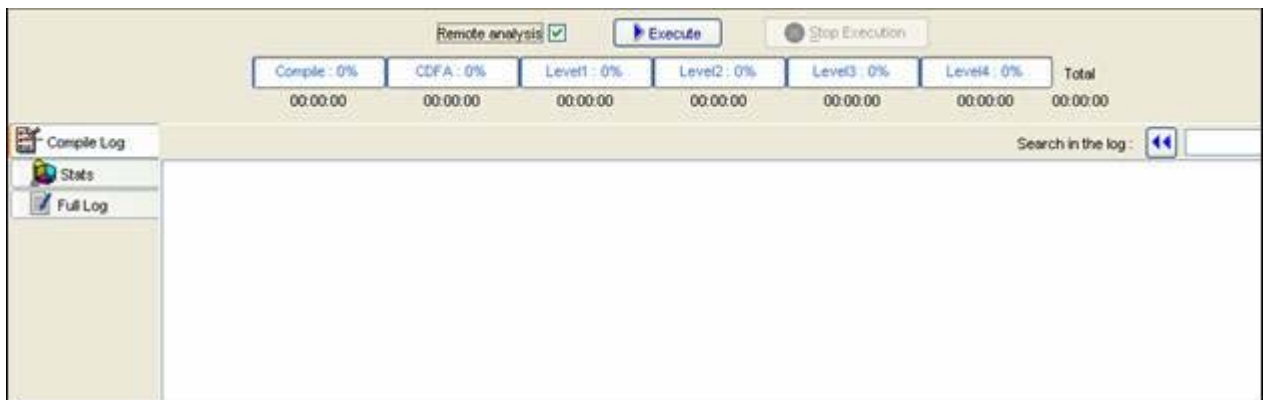
Please see the PolySpace Installation Guide (available on the PolySpace CD-ROM in \Docs\Install or the PolySpace Install Guide Manual) to install and configure a Client and a Server.

Note Launching an analysis remotely requires a PolySpace Server product and associated license.

Steps of Launching

To launch PolySpace™ remotely:

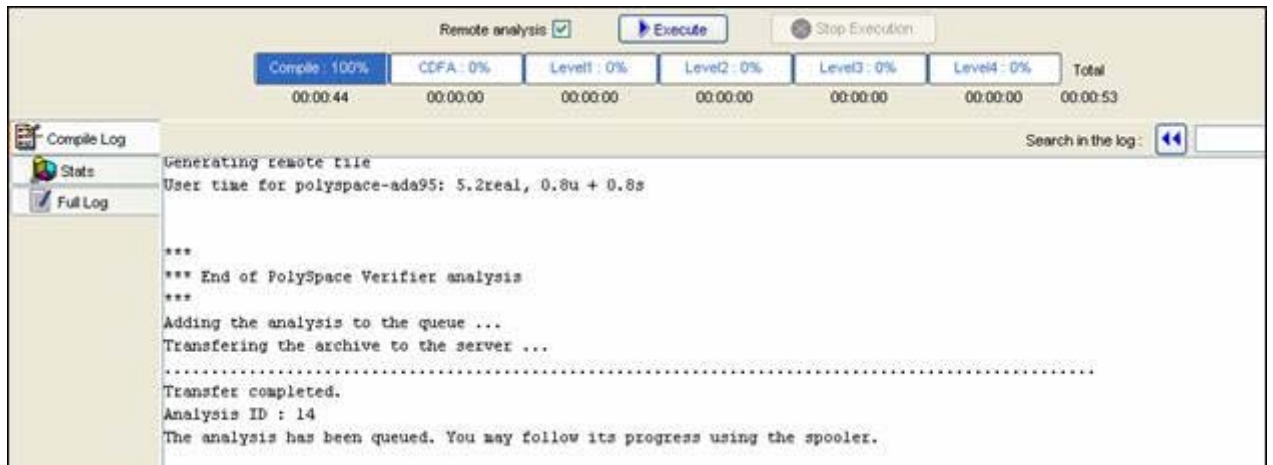
- 1 Set up an analysis as described in Chapter 2, “PolySpace™ Client — Setting Up and Launching an Analysis of a Single Ada File”, but do not launch it.
- 2 Select the **Remote analysis** radio button (see next figure).
- 3 Click  to launch the analysis.



The analysis starts and the compilation phase is performed on the desktop PC. At the end of the “Compilation phase” the analysis is sent to the Queue Manager server.

- 4 Click on the “Full Log” tab. You will have a message like this:

4 Launch PolySpace™ Remotely



The analysis has been queued with an ID number, and you can follow its progression using the PolySpace Spooler.

Note If you do not tick the “Remote analysis” radio button, the analysis continues locally.

Management of PolySpace™ Analysis in Remote: the PolySpace™ Spooler

You can check the analysis processes in the queue using the PolySpace™ Spooler.

To manage an analysis in the queue:

- 1 Open the PolySpace Spooler by either:
 - Clicking on the short cut on your desktop PC



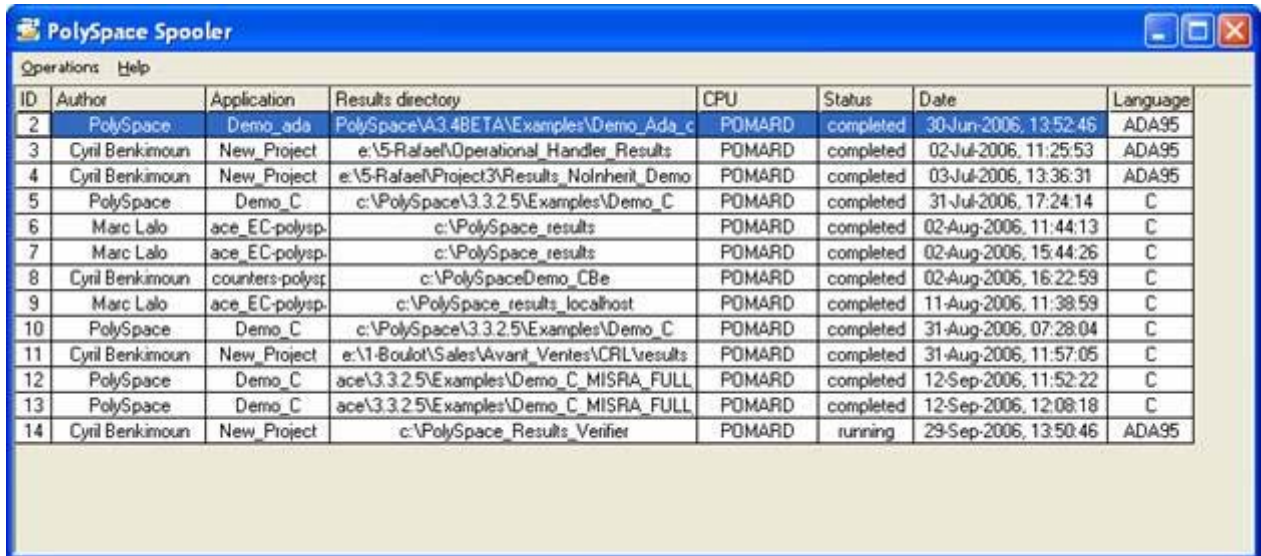
- Clicking on the icon



in the menu tab of the launcher.

The PolySpace Spooler appears.

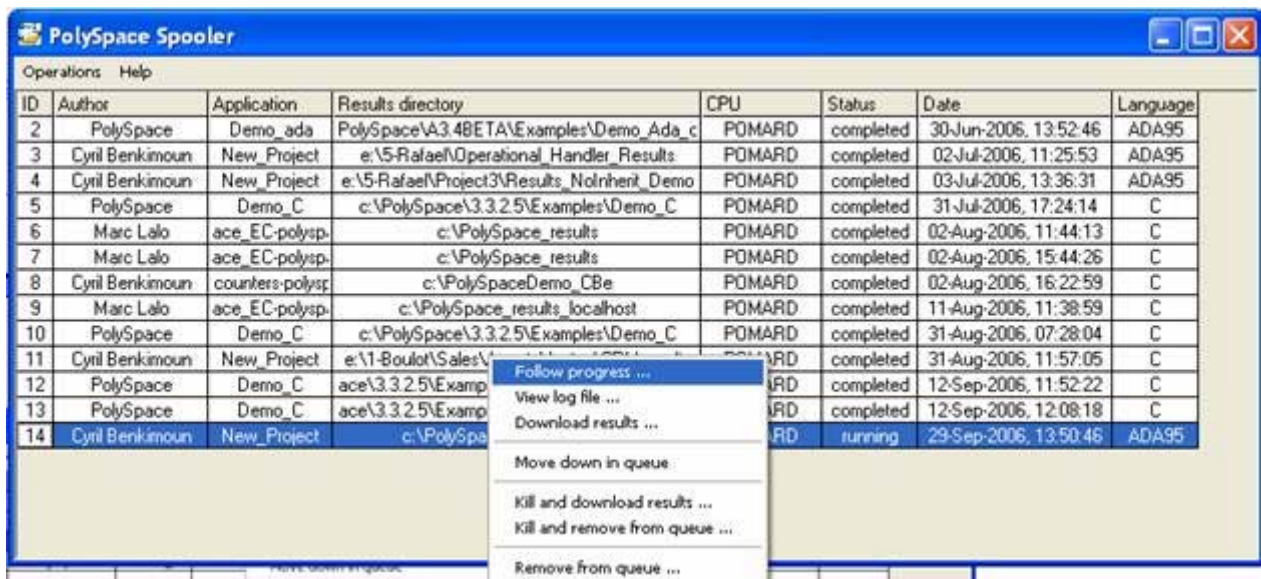
4 Launch PolySpace™ Remotely



The screenshot shows the PolySpace Spooler application window. The title bar reads "PolySpace Spooler". Below the title bar is a menu bar with "Operations" and "Help". The main area contains a table with the following columns: ID, Author, Application, Results directory, CPU, Status, Date, and Language. The table lists 14 jobs, with the last job (ID 14) currently in a "running" state.

ID	Author	Application	Results directory	CPU	Status	Date	Language
2	PolySpace	Demo_ada	PolySpace\A3.4BETA\Examples\Demo_Ada_c	POMARD	completed	30-Jun-2006, 13:52:46	ADA95
3	Cyril Benkimoun	New_Project	e:\5-Rafael\Operational_Handler_Results	POMARD	completed	02-Jul-2006, 11:25:53	ADA95
4	Cyril Benkimoun	New_Project	e:\5-Rafael\Project3\Results_NoInherit_Demo	POMARD	completed	03-Jul-2006, 13:36:31	ADA95
5	PolySpace	Demo_C	c:\PolySpace\3.3.2.5\Examples\Demo_C	POMARD	completed	31-Jul-2006, 17:24:14	C
6	Marc Lalo	ace_EC-polysp	c:\PolySpace_results	POMARD	completed	02-Aug-2006, 11:44:13	C
7	Marc Lalo	ace_EC-polysp	c:\PolySpace_results	POMARD	completed	02-Aug-2006, 15:44:26	C
8	Cyril Benkimoun	counters-polysp	c:\PolySpace\Demo_CBe	POMARD	completed	02-Aug-2006, 16:22:59	C
9	Marc Lalo	ace_EC-polysp	c:\PolySpace_results_localhost	POMARD	completed	11-Aug-2006, 11:38:59	C
10	PolySpace	Demo_C	c:\PolySpace\3.3.2.5\Examples\Demo_C	POMARD	completed	31-Aug-2006, 07:28:04	C
11	Cyril Benkimoun	New_Project	e:\1-Boulot\Sales\Avant_Ventes\CRL\results	POMARD	completed	31-Aug-2006, 11:57:05	C
12	PolySpace	Demo_C	ace\3.3.2.5\Examples\Demo_C_MISRA_FULL	POMARD	completed	12-Sep-2006, 11:52:22	C
13	PolySpace	Demo_C	ace\3.3.2.5\Examples\Demo_C_MISRA_FULL	POMARD	completed	12-Sep-2006, 12:08:18	C
14	Cyril Benkimoun	New_Project	c:\PolySpace_Results_Verifier	POMARD	running	29-Sep-2006, 13:50:46	ADA95

2 Right-click on an analysis to manage it in the queue:



The screenshot shows the same PolySpace Spooler application window as above, but with a context menu open over the last job (ID 14). The menu options are:

- Follow progress ...
- View log file ...
- Download results ...
- Move down in queue
- Kill and download results ...
- Kill and remove from queue ...
- Remove from queue ...

3 Select one of the following options:

- **Follow progress** — This action lists the associated log file in a Launcher window. If the analysis is running, you can follow the update of the log file and associated progress bar in real time on the Launcher window.
- **View log file** — This action lists the associated log file in a “Command prompt” window, in which you can the last 100 updated lines of the log file in real time. This option is only available when the analysis is running.
- **Download results** — This action downloads the results of an analysis onto the client. If the analysis is still running, available results are downloaded on the client, without disturbing the analysis. The option is not possible for a “queued” analysis.
- **Move down in queue** — This action reduces the priority of a “queued” analysis.
- **Kill and download results** — This action stops the analysis definitively and the results are downloaded. The status of the analysis changes from “running” to “aborted”. The analysis remains on the queue.
- **Kill and remove from queue** — This action stops the analysis definitively, and the analysis is removed from the queue.

Note The results will be lost.

- **Remove from queue** — This action removes a “queued”, “aborted” ora “completed” analysis.

Note The results will be lost.

You can also manage the queue from an administrator point of view using the **Operations** menu:

- Select **Operations > Purge queue**, to purge the entire queue or purge only completed and aborted analysis (see next figure).



Note The queue manager password is required.

- Select **Operations > Change root password**, to change the administrator password of the queue manager or the default one.

Note By default the password is “administrator”.

Batch Commands

In this section...
“Launching an Analysis in Batch” on page 4-9
“Managing an Analysis in Batch” on page 4-9

Launching an Analysis in Batch

A set of commands allow you to launch an analysis in batch.

All these commands begin with the following prefixes:

- **Server analysis** —
PolySpaceInstallDir/Verifier/bin/polyspace-remote-ada95
- **Client analysis** — *polyspace-remote-desktop-ada95*

These commands are equivalent to commands with a prefix
PolySpaceInstallDir/bin/polyspace-.

For example, *polyspace-remote-desktop-ada95 -server*
[<hostname>:[<port>] | auto] allows you to send an Ada desktop analysis remotely.

Note If your PolySpace server is running on Microsoft® Windows®, the batch commands are located in the */wbin/* directory. For example, *PolySpaceInstallDir/Verifier/wbin/polyspace-remote-ada95.exe*

Managing an Analysis in Batch

In batch, a set of commands allow the management of analysis in the queue.

On UNIX®, all these command begin with the prefix
PolySpaceCommonDir/RemoteLauncher/bin/psqueue-.

On Windows, these commands begin with the prefix
PolySpaceCommonDir/RemoteLauncher/wbin/psqueue-:

- `psqueue-download <id> <results dir>` — download an identified analysis into a results directory.
 - `[-f]` force download (without interactivity)
 - `-admin -p <password>` allows administrator to download results.
 - `[-server <name>[:port]]` selects a specific Queue Manager.
 - `[-v|version]` gives release number.
- `psqueue-kill <id>` — kill an identified analysis.
- `psqueue-purge all|ended` — remove all or finished analyses in the queue.
- `psqueue-dump` — gives the list of all analyses in the queue associated to default Queue Manager.
- `psqueue-move-down <id>` — move down an identified analysis in the Queue.
- `psqueue-remove <id>` — remove an identified analysis in the queue.
- `psqueue-get-qm-server` — give the name of the default Queue Manager.
- `psqueue-progress <id>`: give progression of the currently identified and running analysis.
 - `[-open-launcher]` display the log in the graphical user interface of launcher.
 - `[-full]` give full log file.
 - `psqueue-set-password <password> <new password>` — change administrator password.
- `psqueue-check-config` — check the configuration of Queue Manager.
 - `[-check-licenses]` check for licenses only.
- `psqueue-upgrade` — Allow to upgrade a client side (cf. PolySpace Install Guide in the `PolySpaceCommonDir/Docs` directory).
 - `[-list-versions]` give the list of available release to upgrade.
 - `[-install-version <version number> [-install-dir <directory>]] [-silent]` allow to install an upgrade in a given directory and in silent.

Note *PolySpaceCommonDir/bin/psqueue-<command>* -h gives information about all available options for each command.

Share Analyses Between Accounts

In this section...
“Analysis-key.text File” on page 4-12
“Magic Key or Shared Analysis Between Projects” on page 4-13

Analysis-key.text File

From a security point of view, all analysis spooled on a same Queue Manager are owned by the user who sent the analysis from a specific account. Each analysis has a unique cryptic key.

The public part of the key is stored in a file `analysis-keys.txt` associated to a user account. This file is located in:

- **UNIX®** — `/home/username/.PolySpace`
- **Windows®** — `C:\Documents and Settings\username\Application Data\PolySpace`

The format of the ASCII file is the following (spaces are tabulation):

```
<id of launching> <server name of IP address> <public key>
```

where *<public key>* is a value in the range [0..F]

Example:

```
1 m120 27CB36A9D656F0C3F84F959304ACF81BF229827C58BE1A15C8123786
2 m120 2860F820320CDD8317C51E4455E3D1A48DCE576F5C66BEEF391A9962
8 m120 2D51FF34D7B319121D221272585C7E79501FBCC8973CF287F6C12FCA
```

When we make an attempt of management (download, kill and remove, etc.) on a particular analysis, the Queue Manager will examine this file and find the associated public key to authenticate the analysis on the server.

If the key does not exist, an error message appears: “key for analysis <ID> not found”. So sharing an analysis with another user account necessitates the public key.

Sharing an analysis is quite simple, ask to the owner of the analysis the line in `analysis-key.txt` which containing the associated `<ID>` and put it the line in your own file. After, it will be able to download the analysis.

Magic Key or Shared Analysis Between Projects

A magic key allows sharing analyses without taking into account the `<ID>`. It allows same key for all analysis launched by a user account. The format is the following:


```
0 <Server id> <your hexadecimal value>
```

All analyses spooled will have this key instead of random one. In the same way, if this kind of key is available in an `analysis-key.txt` file of another user, it allows to authorize any operation on any analyses pushed with this key.

Note It only works for all analysis launched after having put the magic key in the file. If the analysis has been launched before, the allowed key associated to the ID will be used for the authentication.

Summary

After having followed each step of this tutorial, you are now able to launch an analysis using PolySpace™ Client and explore some results with PolySpace Viewer. All these commands can be performed locally on your desktop PC or in a client/server architecture.

You will find more information on advanced options available with our tools in the PolySpace Ada documentation available on the CD-ROM (in `\Docs\Manual\`) or by clicking on  in PolySpace tools.