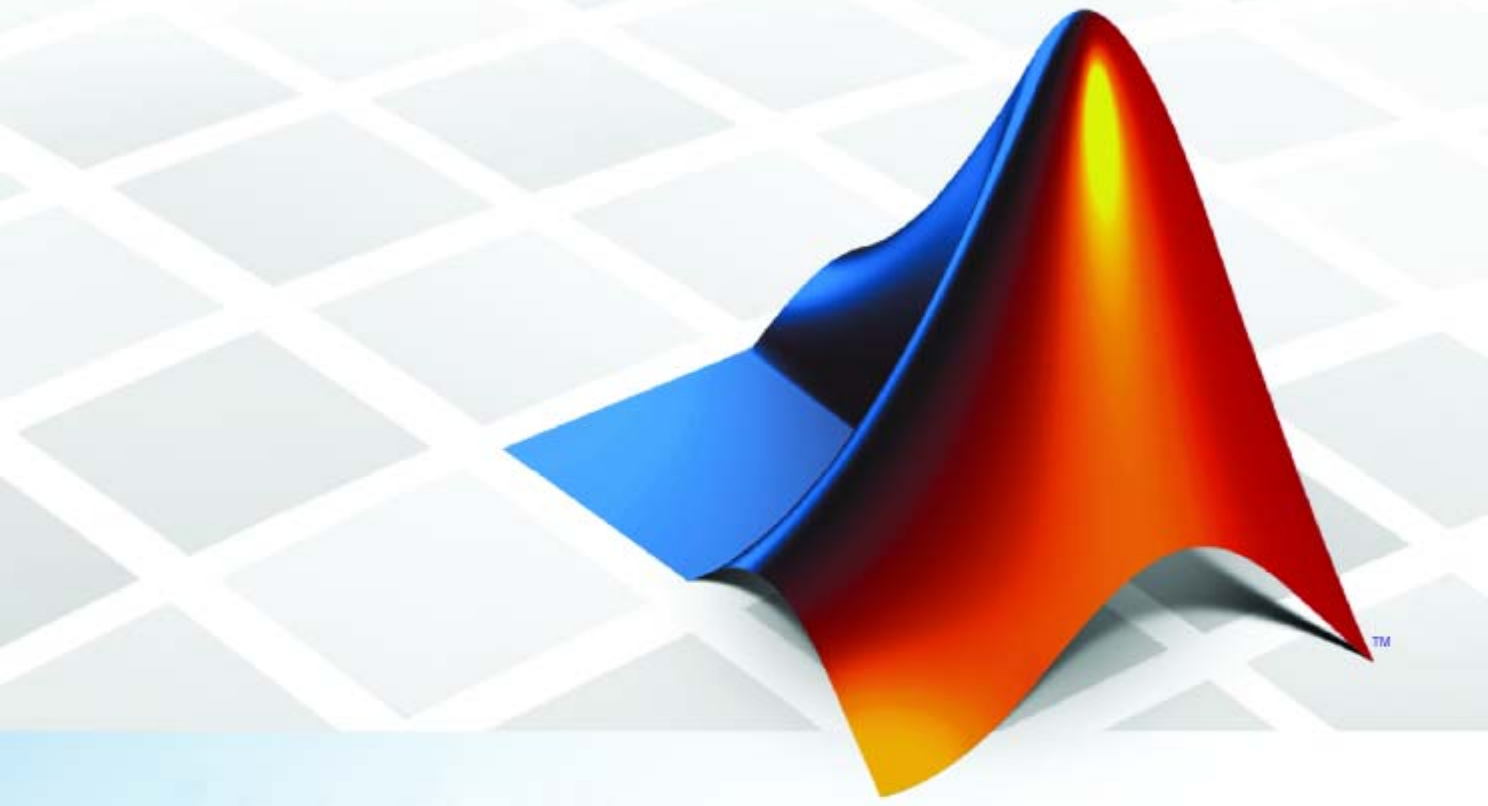
PolySpace® UML Link RH 5

User's Guide



The MathWorks™

Accelerating the pace of engineering and science

How to Contact The MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support

`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	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® UML Link™ RH User's Guide

© COPYRIGHT 1999–2009 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from 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 2009 Online Only Revised for Version 5.3 (Release 2009a)

Contents

PolySpace UML Link RH User's Guide

1

Getting Started **1-2**
Overview of PolySpace UML Link RH Software **1-2**
Opening the Rhapsody Model **1-2**
Starting a Verification **1-3**
Navigating from the PolySpace Results to the Rhapsody Model **1-7**

PolySpace Panel **1-10**
Overview of the PolySpace Panel Interface **1-10**
Start Button **1-11**
Stop Button **1-14**
Compilation Log Button **1-14**
Configure Button **1-14**
Manage Analyses Button **1-15**
View Results Button **1-16**
Browse for Results Button **1-16**
Help Button **1-16**

Installing the Integration into an Existing Model **1-17**
Overview **1-17**
Merging the PolySpace Code with Rhapsody **1-17**

Other Topics **1-19**
Verification Information **1-19**
Default Template CFG Options **1-19**
Back to Model Viewer Link **1-20**

1

PolySpace UML Link RH User's Guide

- "Getting Started" on page 1-2
- "PolySpace Panel" on page 1-10
- "Installing the Integration into an Existing Model" on page 1-17
- "Other Topics" on page 1-19

Getting Started

In this section...

"Overview of PolySpace UML Link RH Software" on page 1-2

"Opening the Rhapsody Model" on page 1-2

"Starting a Verification" on page 1-3

"Navigating from the PolySpace Results to the Rhapsody Model" on page 1-7

Overview of PolySpace UML Link RH Software

While using Collaborative Model-Driven Development, run-time errors can be caused either by design issues in the model itself or faulty hand written code. These reliability flaws can sometimes be found using code reviews and intensive testing - but these techniques are time-consuming and costly. PolySpace® UML Link™ RH software saves you both time and money by performing an exhaustive verification of the code and automatically flagging flaws directly in the original Rhapsody® model, enabling developers to fix these issues quickly and early during the design process.

This getting started guide takes you through the steps required to analyze a model.

Note The PolySpace plug-in has already been integrated into the example model. Before other models can be analyzed the plug-in may need manually installing into the Rhapsody project directory. During the getting started the following conventions will be used: "`<PolySpaceInstallCommon>`" will refer to the installation location of the PolySpace common folder.

Opening the Rhapsody Model

To open the model and display the PolySpace panel:

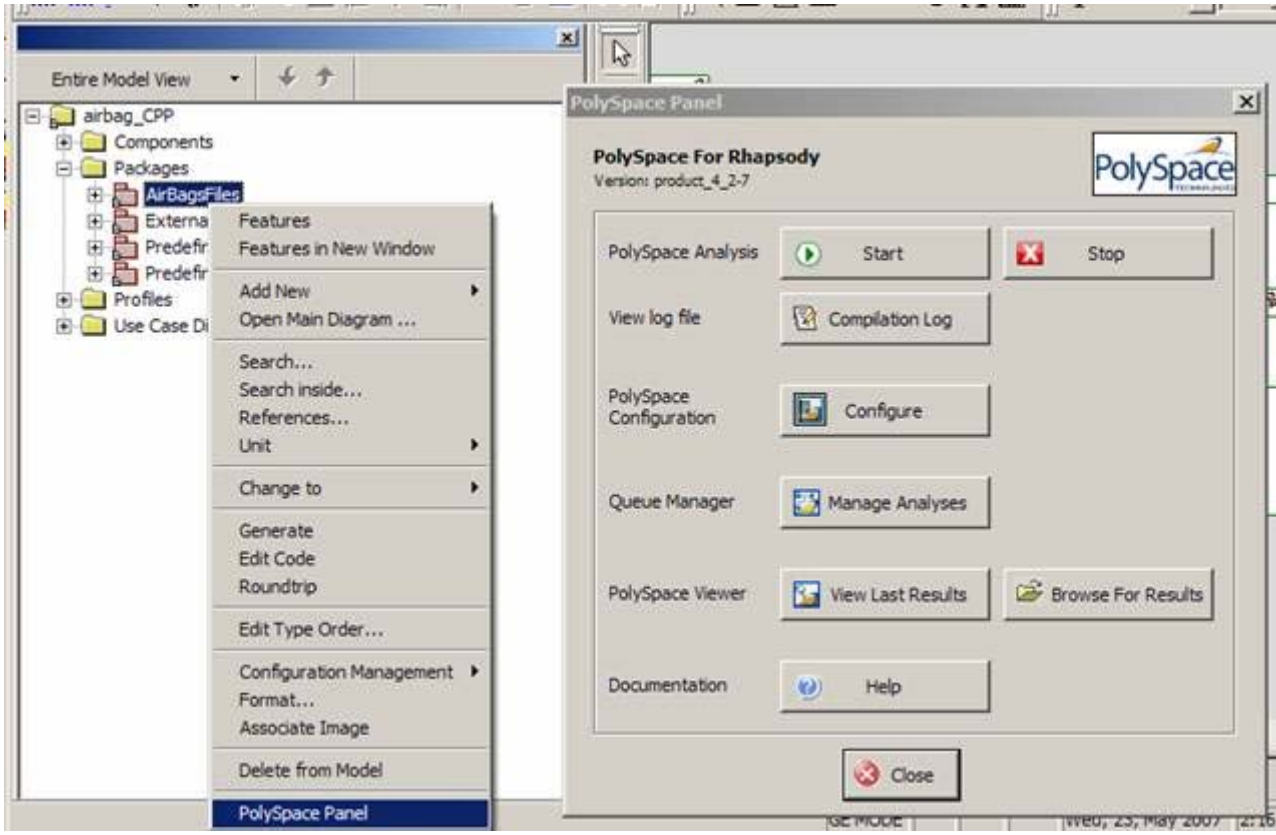
1 Open the Rhapsody model you want to verify.

For the example used in this guide, open the `airbag_CPP.rpy` model in `<PolySpaceInstallCommon>/PolySpaceUMLLink/example`.

2 Select the files/packages you want to verify in the model view.

For example, select **Packages** list in the **airbag_CPP** Model View.

3 Right-click **AirBagFiles**, then select **PolySpace Panel** from the menu.



The PolySpace Panel opens. The PolySpace Panel is the interface to the PolySpace UML Link RH within Rhapsody.

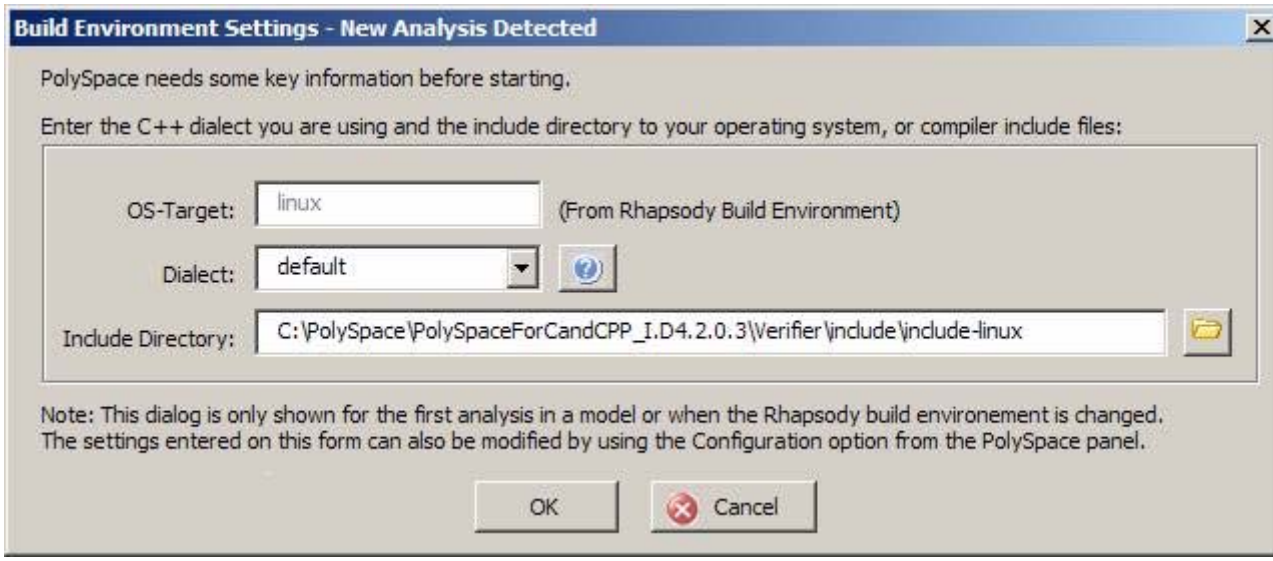
Starting a Verification

To start a verification:

1 Click on the **Start** button in the PolySpace Panel.

If you are starting the first verification of a model, or if the Rhapsody configuration environment has changed since the last verification, the **Build Environment Settings** dialog is displayed.

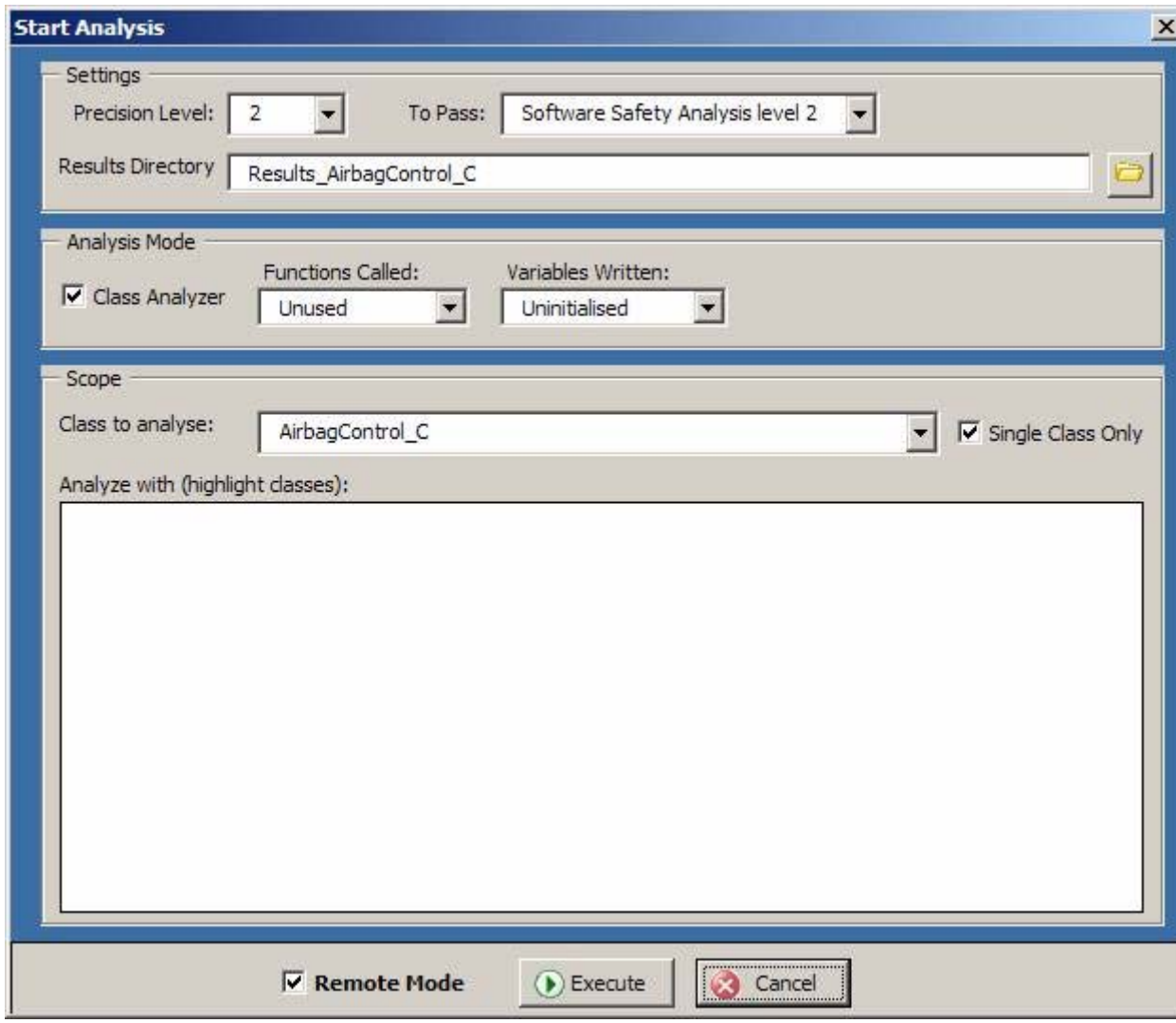
The operating system target (`-OS-target`) is set automatically from the models environment. When the Linux® environment is detected the dialect will be set to default (`-dialect`) and include directory will be configured to use the Linux header files supplied with PolySpace software.



Note Make sure that the generated code for the model is up to date before starting a verification.

2 Click **OK**.

The Start Analysis panel opens.



3 Select the type of analysis you want to perform in the **Analysis Mode** section.

4 Select the file, class, or package you want to analyze in the **Scope** section.

The results directory is automatically set according to the name of the selected package, but it can be overwritten once you select the package to analyze.

5 In the **Settings** section, specify the verification precision, the number of passes to be performed, and the location of the results directory.

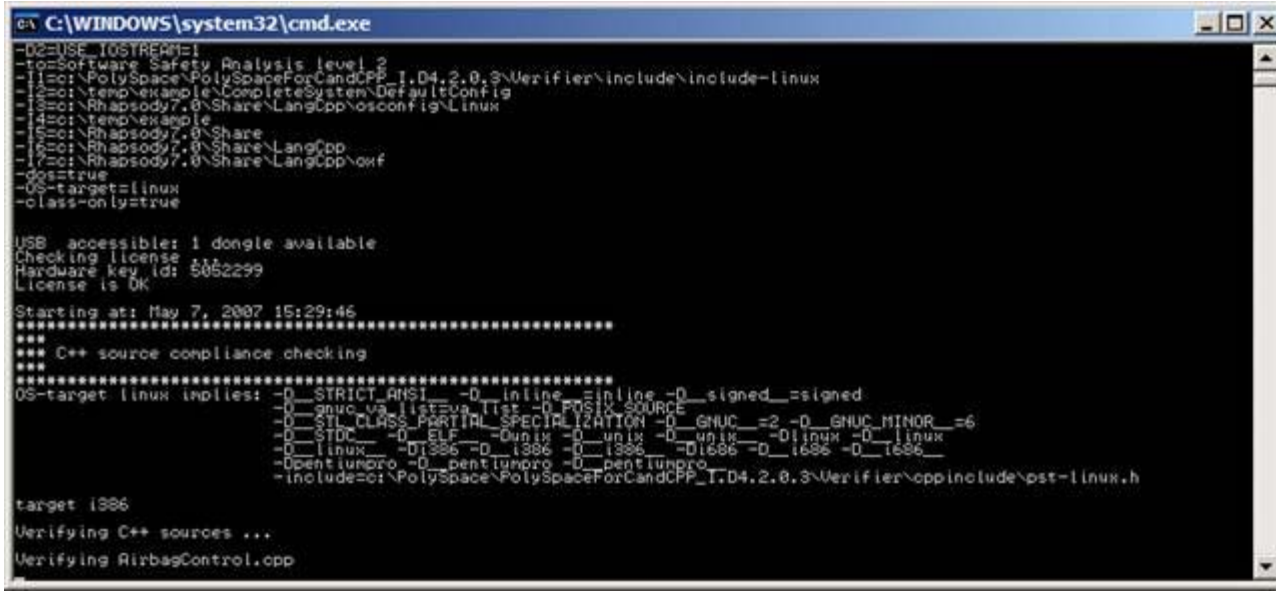
Note The results directory is set automatically when the package to analyze or the analysis mode is changed.

---

6 Select **Remote Mode** to send your verification to the PolySpace Server, or deselect the option to perform the verification locally.

7 Click the **Execute** button.

A command window is displayed, showing the phases of the verification performed locally:



Note The settings (size of window, number of lines of history, font, etc.) for the command window can be changed by right clicking on the window title and selecting properties. It follows standard settings of the "Command Windows" associated with Windows® OS.

You can follow the analysis on the server (if you selected **Remote Mode**), by clicking the **Manage Analyses** button in the PolySpace panel to display the PolySpace Queue Manager interface (or Spooler).

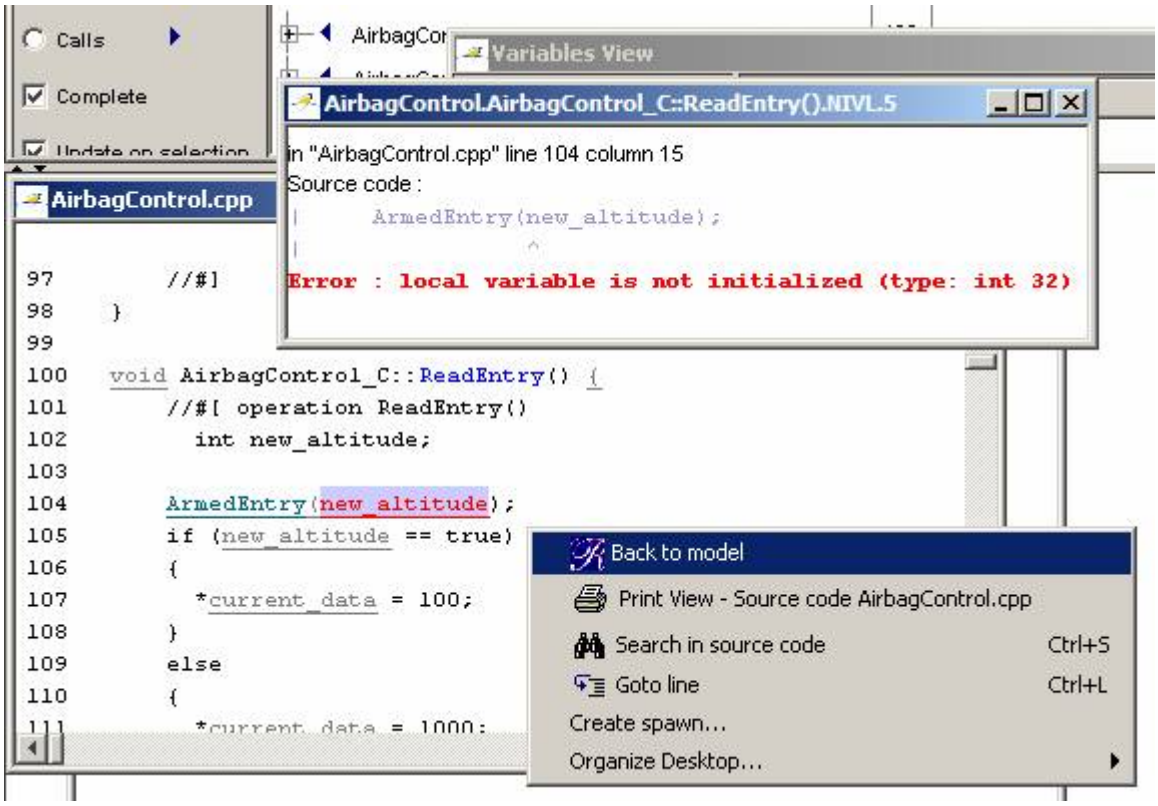
8 When the analysis has completed, download the results and when prompted open with the PolySpace Viewer.

Navigating from the PolySpace Results to the Rhapsody Model

To navigate from the PolySpace results to the model:

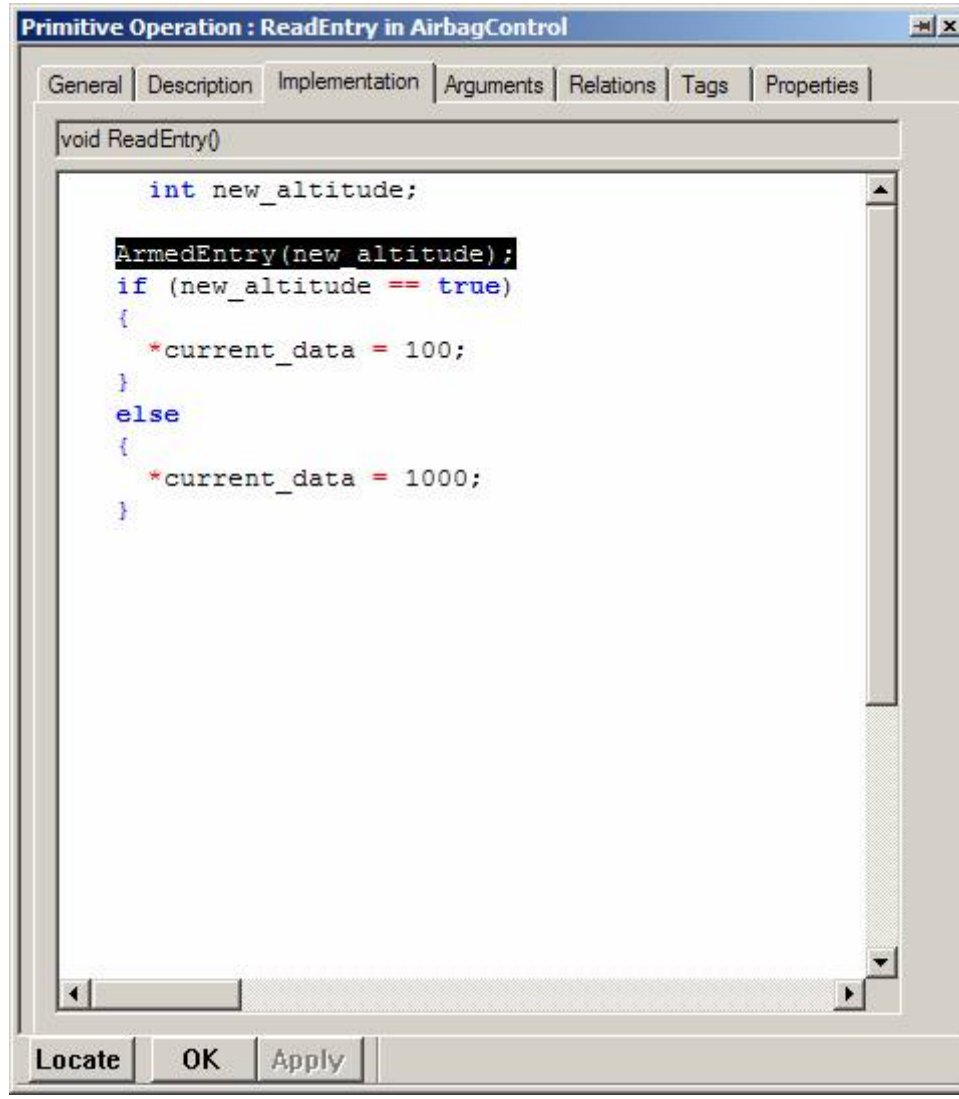
1 Navigate to an error displayed in the PolySpace Viewer. In this example, a non initialized variable detected in the model at line 104 of `Airbag Control_C`.

2 Right click the error, then select **Back to Model** from the pop-up menu.



Note For the "Back To Model" feature to work Rhapsody must be running with the model open.

The software locates the code within the Rhapsody model. Depending on the Rhapsody configuration, this will either be shown in a popup dialog (such as the one below) or in the code view:


Primitive Operation : ReadEntry in AirbagControl
General
Description
Implementation
Arguments
Relations
Tags
Properties
void ReadEntry()
int new_altitude;
ArmedEntry(new_altitude);
if (new_altitude == true)
{
*current_data = 100;
}
else
{
*current_data = 1000;
}
Locate
OK
Apply

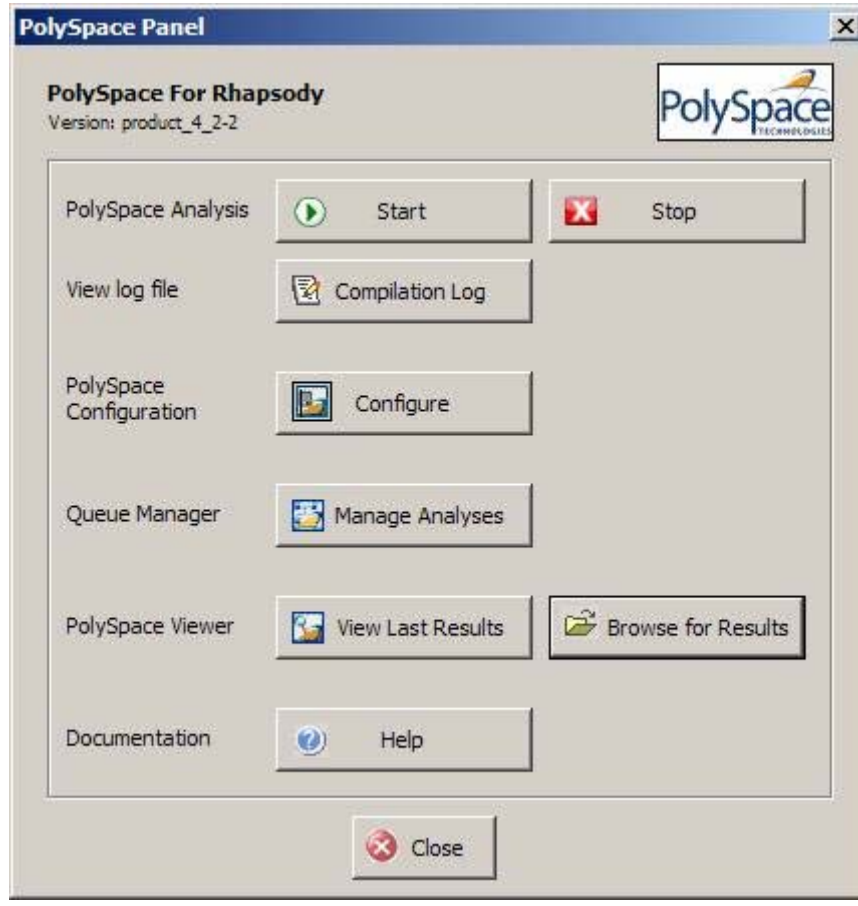
PolySpace Panel

In this section...

"Overview of the PolySpace Panel Interface" on page 1-10
"Start Button" on page 1-11
"Stop Button" on page 1-14
"Compilation Log Button" on page 1-14
"Configure Button" on page 1-14
"Manage Analyses Button" on page 1-15
"View Results Button" on page 1-16
"Browse for Results Button" on page 1-16
"Help Button" on page 1-16

Overview of the PolySpace Panel Interface

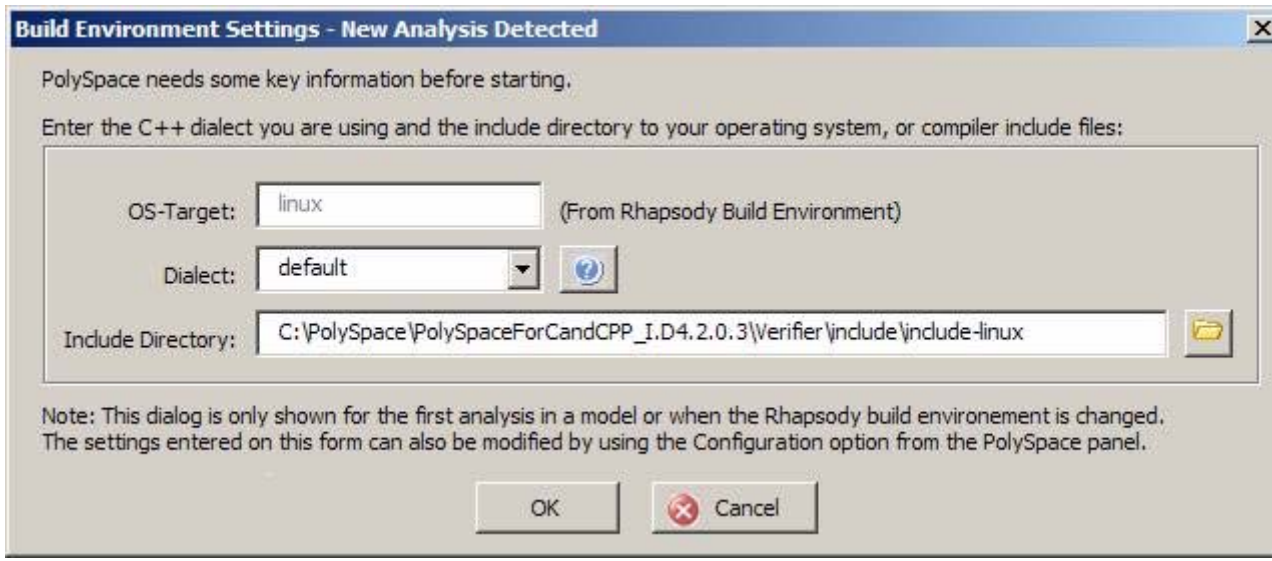
The PolySpace panel is the main interface between PolySpace and Rhapsody. You open the panel by right-clicking on a package in the Rhapsody **Entire Model View** and selecting **PolySpace Analysis**.



Start Button

The **Start** button is used to start a verification.

Either for the first verification of the model, or if the Rhapsody configuration environment has changed since the last verification the **Build Environment Settings** dialog is displayed:



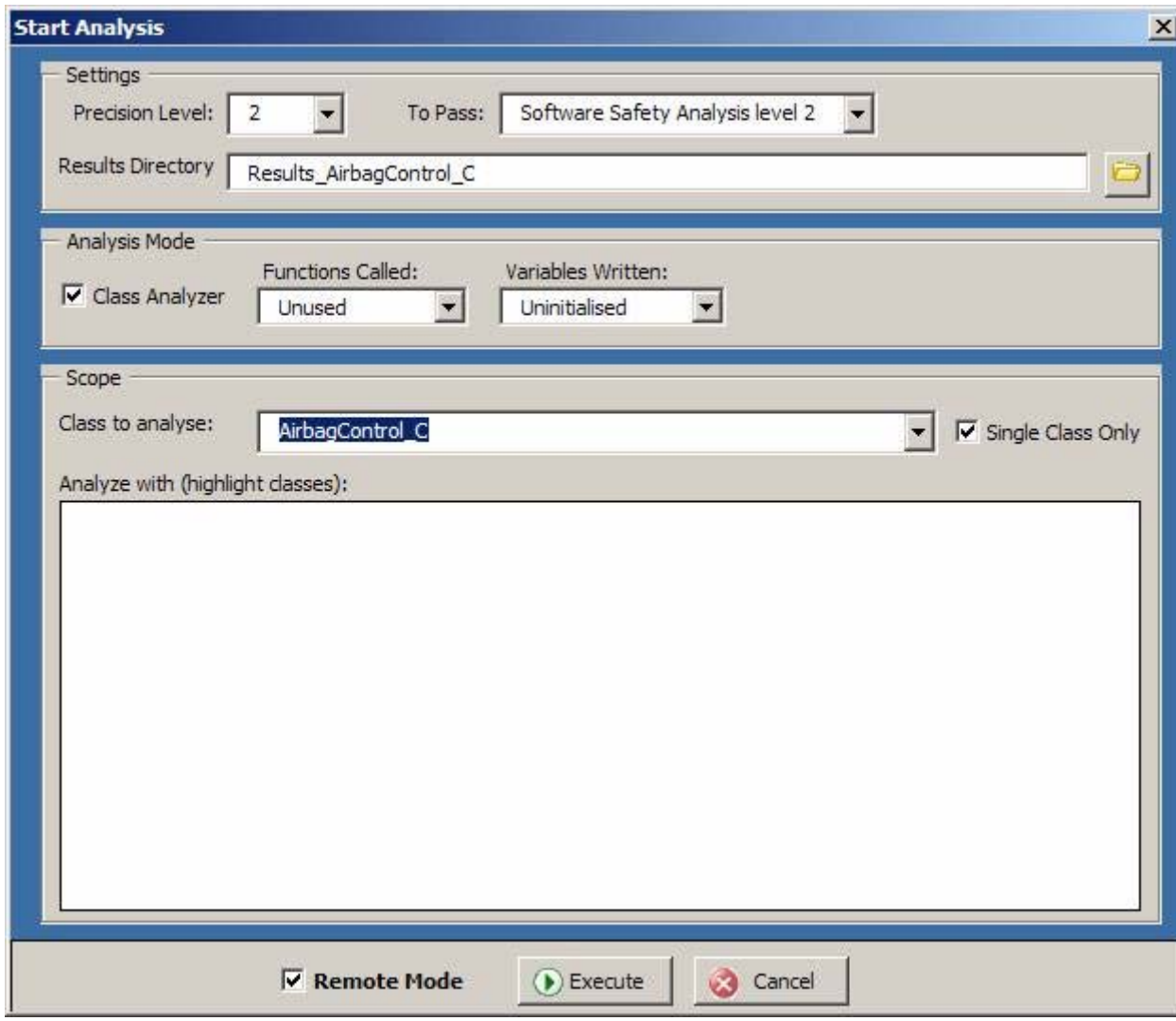
The Operating System Target (`-OS-target`) is detected automatically from the active Rhapsody build environment.

Select the appropriate C, C++, or Ada dialect and the location of the include files for the compiler. If more than one include directory is required this can be added later using the "Configure" option on the PolySpace Panel.

Note If the visual OS-Target is detected and a PolySpace supported version of the Microsoft® Visual C++® compiler is installed the "Dialect" (`-dialect`) and "Include Directory" fields will be automatically completed. This also applies if Linux is detected as the -OS-Target, the dialect and include directory will be configured to process the header files from the PolySpace® Client™ for C/C++ product directory.

Start Analysis Dialog

When you select OK in the Build Environment Settings dialog, the Start Analysis dialog is displayed:



From the Start Analysis dialog:

- Select the class to analyze from the scope section. The results directory is automatically set according to the name of the selected class, but can be overwritten once the class to analyze has been selected.

- Select **Execute** to start the verification. If **Remote Mode** is selected the verification will be sent to the PolySpace® Server™ for C/C++ machine at the end of the compilation phase.
- The **Settings** section allows setting of the analysis precision, the number of passes of the analysis to perform and the results directory.
- The **Analysis Mode** section allows configuration of the type of analysis to perform. The options are either to use the "Class Analyzer" to analyze individual classes, or, without in which case a valid "main" function needs to be present in the code. To analyze multiple classes at the same time deselect the "Single Class Only" option, and highlight the classes to analyze in the list. The control and shift keys can be used to control the selection of classes from the list.

Stop Button

The client based phase (compilation) of the verification can be stopped by clicking on the **Stop** button.

For a verification running on a PolySpace Server, use the "Queue Manager" to control the jobs.

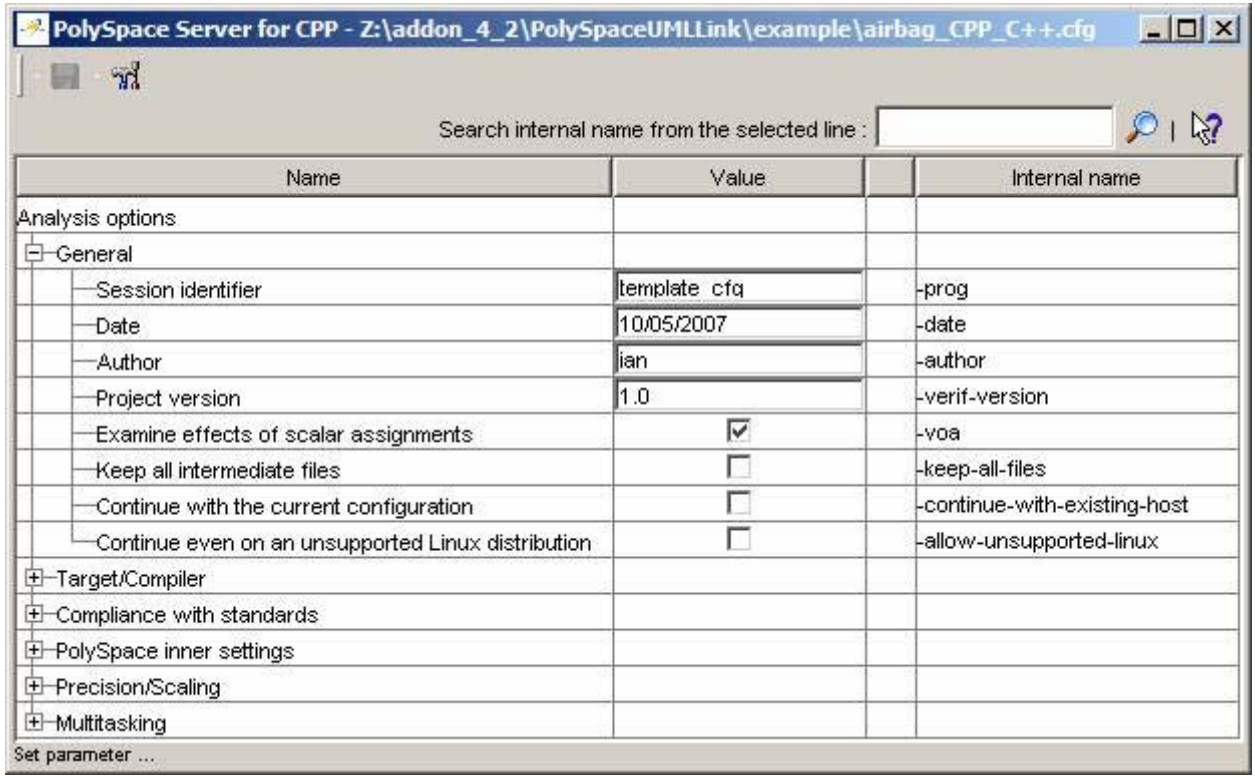
Compilation Log Button

The latest compilation log can be viewed at any time by clicking on the "**Compilation Log**" button.

Configure Button

The **Configure** button displays a scaled-down version of the PolySpace Launcher. You can use this interface to set up advanced PolySpace options, and to enter extra source code compilation parameters, if necessary.

Click the disk button in the top left corner to save the configuration.



Note PolySpace extracts the include directories and compilation flags from the current build environment. In many cases, no further configuration beyond that requested by the **Build Environment** dialog should be required for a standard verification.

Manage Analyses Button

To download the results from a PolySpace Server or to follow the progress of a verification running remotely, click on the **Manage Analyses** button.

View Results Button

To view the results from the last completed verification, click on the **View Results** button. If no results are available (they are still on the server), the user will be prompted to start the PolySpace Queue Manager. The results can be downloaded and opened from the Queue Manager.

Browse for Results Button

You can browse and open all of the results for a given model by clicking on the **Browse for Results** button.

Help Button

The **Help** button opens the PDF help document.

Installing the Integration into an Existing Model

In this section...

"Overview" on page 1-17

"Merging the PolySpace Code with Rhapsody" on page 1-17

Overview

This section details the configuration required to use PolySpace UML Link RH link in a Rhapsody project.

The integration is written using the Visual Basic® extension provided by Rhapsody. To install the integration into new Rhapsody projects it uses the copyVBA feature contained in the Rhapsody `.ini` file (Windows directory `c:\rhapsody.ini`).

However, for existing projects, the PolySpace Visual Basic module needs copying from `<PolySpaceCommonInstall>\PolySpaceUMLLink\bin\polyspace.vba` to the project directory and then renaming to `<project_name>.vba`, replacing the existing `vba` file.

Merging the PolySpace Code with Rhapsody

For sites already using the Rhapsody Visual Basic feature, you can merge PolySpace code with the existing code.

To perform the merge:

1. Export each form and module for the existing code using the Rhapsody Visual Basic editor.
2. Close your model, copy the `PolySpace.vba` file into the model directory and rename it to `<project_name>.vba`.
3. Reopen the model, start the Visual Basic Editor and import all of the module/form code that was exported in the first step.

4 Optional: The master `polyspace.vba` file can be updated if necessary with the contents of `<project_name>.vba` for use in new projects.

Other Topics

In this section...
"Verification Information" on page 1-19
"Default Template CFG Options" on page 1-19
"Back to Model Viewer Link" on page 1-20

Verification Information

The first time an analysis is performed, a template PolySpace configuration file is copied from `<PolySpaceInstallCommon>/PolySpaceForUML/etc/template_<Language>.cfg` to the project directory and is renamed `<model>_<language>.cfg` where `<model>` is the name of your model and `<language>` is the name of the language the model targets (C, C++, or Ada).

The template `.cfg` file can be updated either by editing it with the PolySpace Launcher, by double-clicking the file in a Windows Explorer window, or by manually copying a `.cfg` file from a Rhapsody model directory over the template file. This means that a configuration can be shared among project members and used with other projects.

Default Template CFG Options

The following options are set by default in the `template_C++.cfg` file:

```
-lang=CPP
-prog=template_cfg
-results-dir=r->results
-allow-undef-variables=true
-respect-types-in-globals=true
-respect-types-in-fields=true
-dos=true
-target=i386
-D=[OM_NO_FRAMEWORK_MEMORY_MANAGER]
-to=7
-OS-target=no-predefined-OS
```

The following options are set by default in the `template_Ada.cfg` file:

```
-lang=ADA
-prog=template_cfg
-results-dir=r->results
-allow-undef-variables=true
-respect-types-in-globals=true
-respect-types-in-fields=true
-dos=true
-target=i386
-D=[OM_NO_FRAMEWORK_MEMORY_MANAGER]
-to=7
-OS-target=no-predefined-OS
```

Back to Model Viewer Link

The **Back to Model** command in the Viewer (right click on a check) is currently limited to source code lines containing a PolySpace check that additionally do not contain a macro.

- A warning "`Unable to go back to the UML model from this location`" will be given when trying to use the command on a line containing a macro.
- A warning "`No element found in model`" will be given for locations for which Rhapsody does not support the **Back to Model** feature, or for lines of code not in the model.