New Features in Polyspace Products

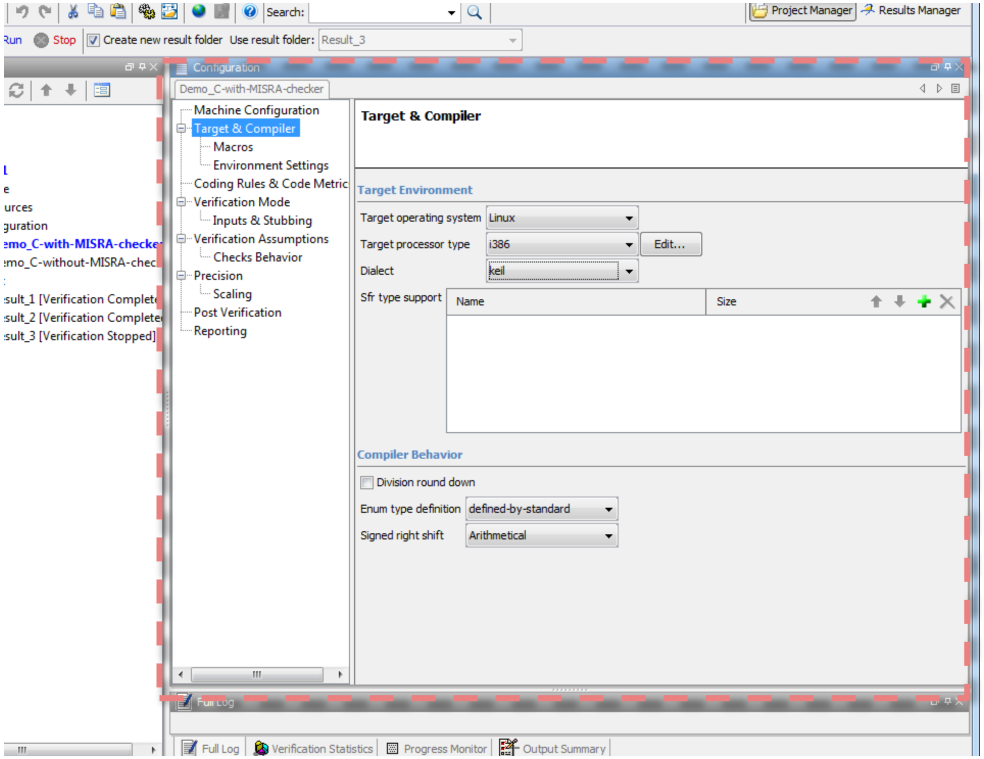
September 2012

R2012b

Simplified Configuration Option Setting

Reorganized configuration options

- Simplified setup process
- Categories for compilation settings, coding rules, results accuracy options, and other settings
- Options organized in line with the relevant step of the setup process
- Fewer popup dialog boxes
- New options



☑ Create new result folder Use result folder: Result_3

Target & Compiler

Target Environment

Target operating system: Linux

Target processor type: i386

Dialect: keil

Sfr type support

Compiler Behavior

□ Division round down

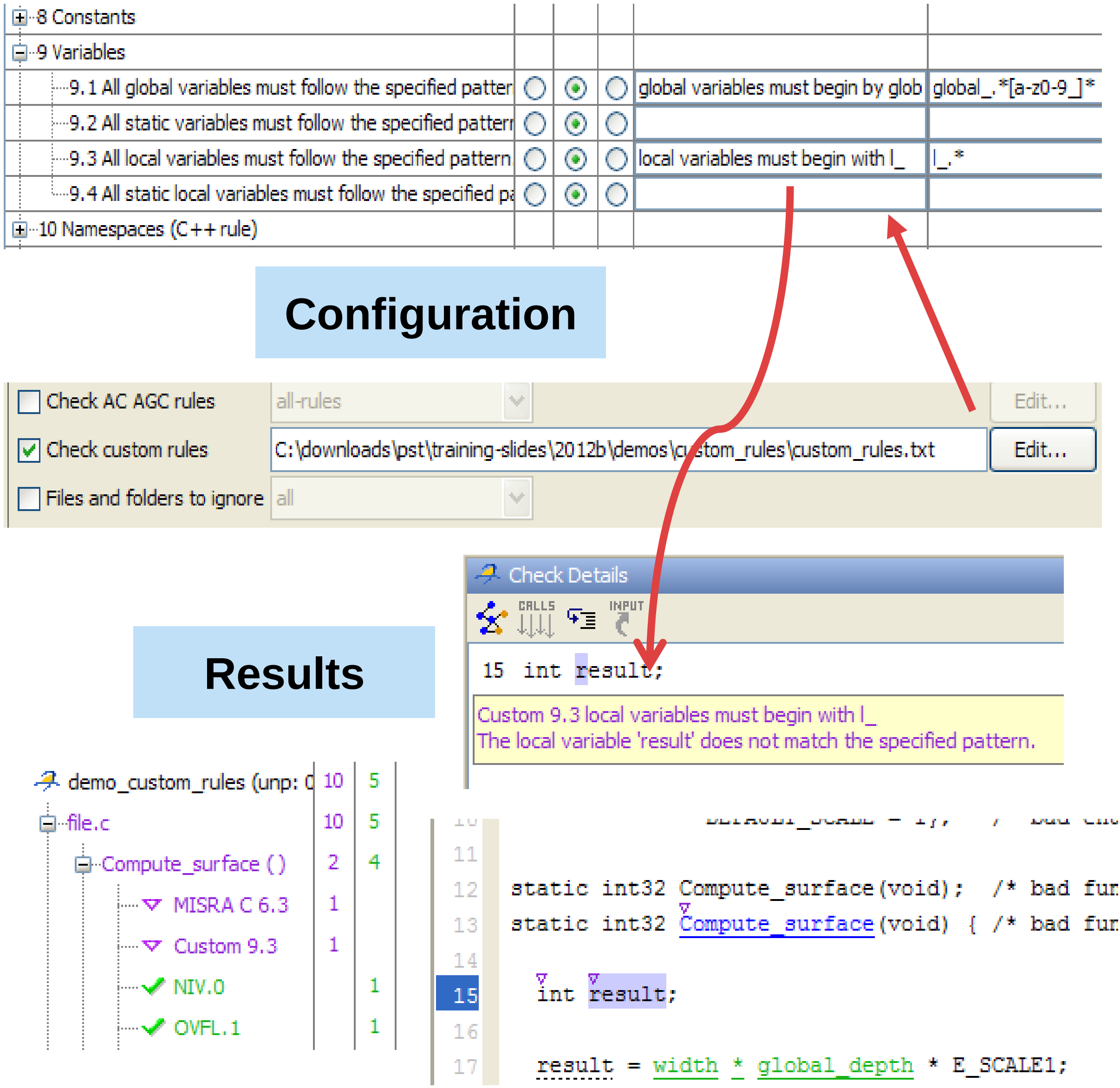
Enum type definition: defined-by-standard

Signed right shift: Arithmetical

MISRA Checker Improvements

Custom rule checking capability

- Create naming convention rules for variables, functions, and other elements
- Use new command-line option or custom rules GUI to create rules
- Define rules:
  - Pattern in the form of regular expression
  - Insert comments



⊞ 8 Constants					
⊟ 9 Variables					
9.1 All global variables must follow the specified patter	○	●	○	global variables must begin by glob	global_.*[a-z0-9_]*
9.2 All static variables must follow the specified patterr	○	●	○		
9.3 All local variables must follow the specified pattern	○	●	○	local variables must begin with l_	l_.*
9.4 All static local variables must follow the specified pa	○	●	○		
⊞ 10 Namespaces (C++ rule)					

Configuration

□ Check AC AGC rules	all-rules	Edit...
☑ Check custom rules	C:\downloads\pst\training-slides\2012b\demos\custom_rules\custom_rules.txt	Edit...
□ Files and folders to ignore	all	

Check Details

```
15  int result;
```

Custom 9.3 local variables must begin with l_
The local variable 'result' does not match the specified pattern.

Results

demo_custom_rules (unp: 0	10	5
⊟ file.c	10	5
⊟ Compute_surface ()	2	4
▽ MISRA C 6.3	1	
▽ Custom 9.3	1	
✓ NIV.0		1
✓ OVFL.1		1

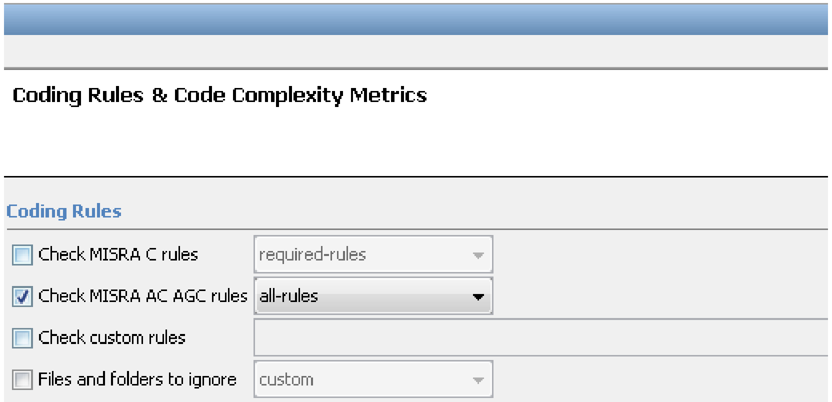
```
static int32 Compute_surface(void);   /* bad fun
static int32 Compute_surface(void) { /* bad fun

  int result;

  result = width * global_depth * E_SCALE1;
```

MISRA Checker Improvements

Enhancements for MISRA AC AGC

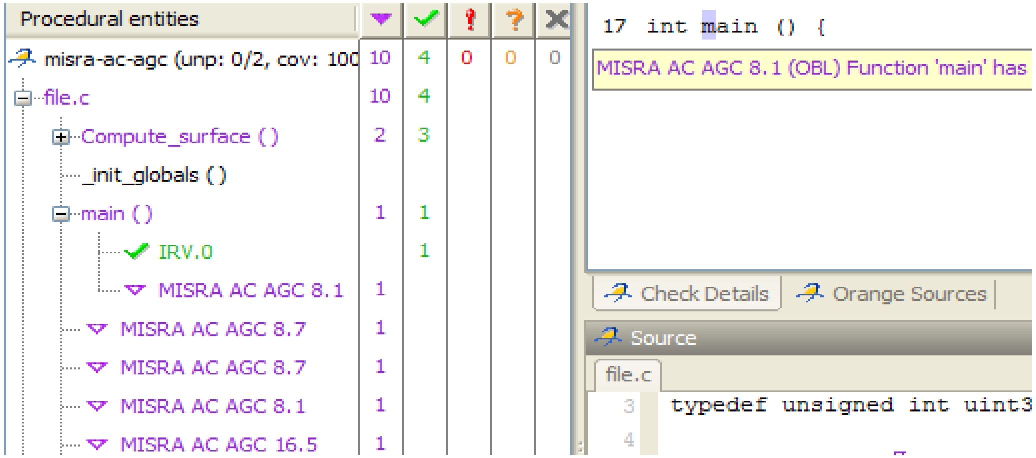


Coding Rules & Code Complexity Metrics

Coding Rules

- □ Check MISRA C rules: required-rules
- ☑ Check MISRA AC AGC rules: all-rules
- □ Check custom rules
- □ Files and folders to ignore: custom

- Check MISRA AC AGC rules:
  - With dedicated checker
  - With new option: `-misra-ac-agc`
- Use predefined subsets:
  - MISRA-C and MISRA C++ checkers: *required-rules*
  - MISRA-AC-AGC: *OBL-rules* *OBL-REC-rules*



Procedural entities	▽	✓	!	?	✗
misra-ac-agc (unp: 0/2, cov: 100	10	4	0	0	0
file.c	10	4			
Compute_surface ( )	2	3			
_init_globals ( )					
main ( )	1	1			
✓ IRV.0		1			
▽ MISRA AC AGC 8.1	1				
▽ MISRA AC AGC 8.7	1				
▽ MISRA AC AGC 8.7	1				
▽ MISRA AC AGC 8.1	1				
▽ MISRA AC AGC 16.5	1				

```
int main () {
```

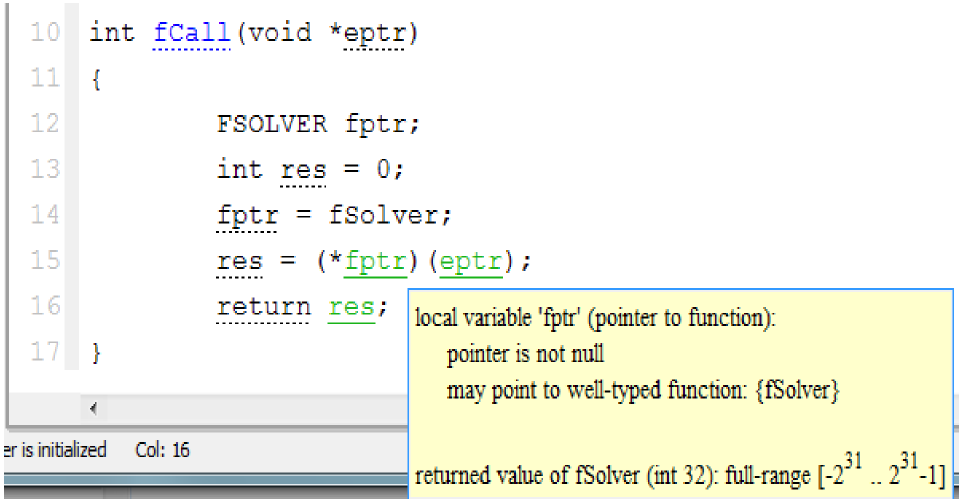
MISRA AC AGC 8.1 (OBL) Function 'main' has

```
typedef unsigned int uint3
```

New Option `-permissive-function-pointer`

Use pointer parameters to enable permissive function pointer calls

- When this option is enabled, the function pointer call is considered to be well-typed.

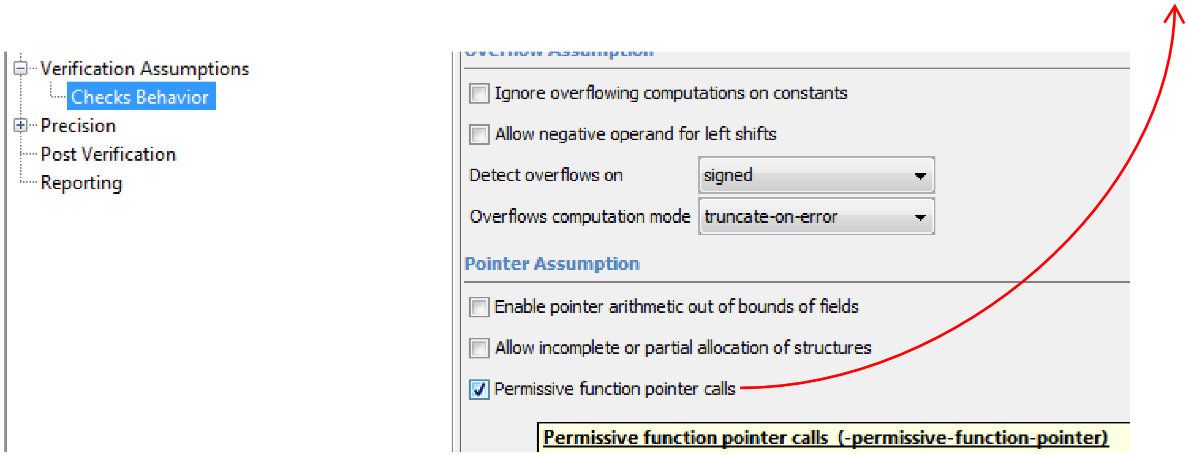


```
int fCall(void *eptr)
{
        FSOLVER fptr;
        int res = 0;
        fptr = fSolver;
        res = (*fptr)(eptr);
        return res;
}
```

local variable 'fptr' (pointer to function):
pointer is not null
may point to well-typed function: {fSolver}

returned value of fSolver (int 32): full-range [-2^{31} .. $2^{31}-1$]

er is initialized Col: 16



Overflow Assumption

□ Ignore overflowing computations on constants

□ Allow negative operand for left shifts

Detect overflows on: signed

Overflows computation mode: truncate-on-error

Pointer Assumption

□ Enable pointer arithmetic out of bounds of fields

□ Allow incomplete or partial allocation of structures

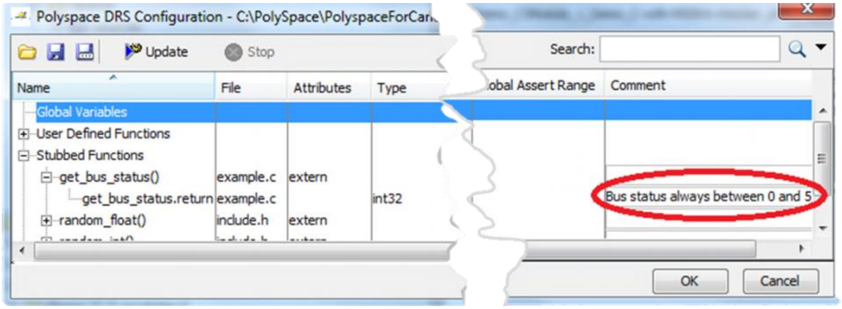
☑ Permissive function pointer calls

Permissive function pointer calls (-permissive-function-pointer)

Data Range Specification Improvements

- Data Range Specification (DRS) can be generated in unit-by-unit mode.

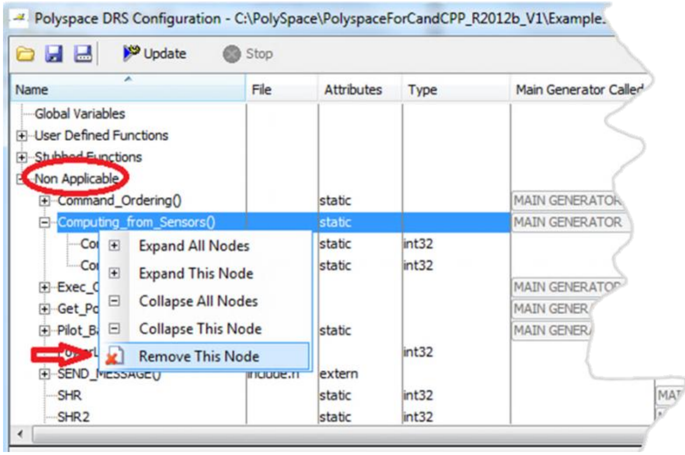
- Comments can be added to a DRS Configuration.



Polyspace DRS Configuration - C:\PolySpace\PolyspaceForCan

Name	File	Attributes	Type	obal Assert Range	Comment
Global Variables					
User Defined Functions					
Stubbed Functions					
get_bus_status()	example.c	extern			
get_bus_status.return	example.c		int32		Bus status always between 0 and 5
random_float()	include.h	extern			

- Non-Applicable DRS entries can be removed (by a right-click on the table).



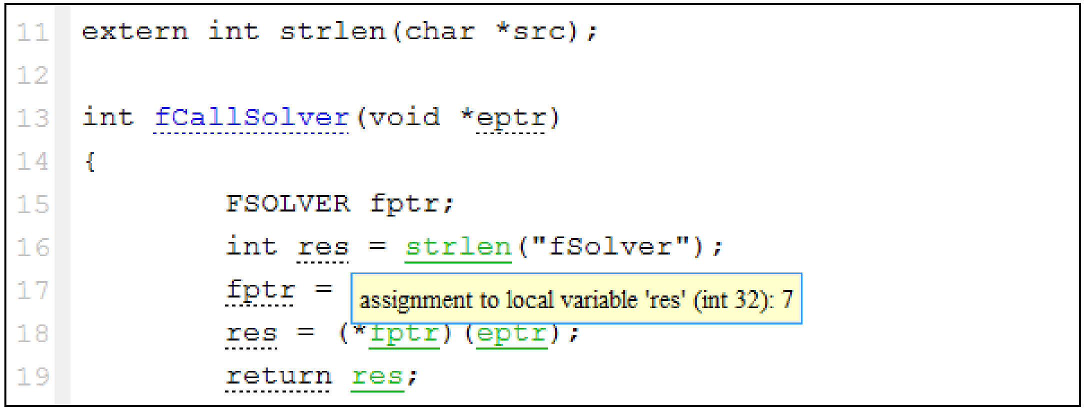
Polyspace DRS Configuration - C:\PolySpace\PolyspaceForCandCPP_R2012b_V1\Example.

Name	File	Attributes	Type	Main Generator Called
Global Variables				
User Defined Functions				
Stubbed Functions				
Non Applicable				
Command_Ordering()		static		MAIN GENERATOR
Computing_from_Sensors()		static		MAIN GENERATOR
Co		static	int32	
Co		static	int32	
Exec_C				MAIN GENERATOR
Get_Po				MAIN GENER
Pilot_B		static		MAIN GENER
			int32	
SEND_MESSAGE()	include.h	extern		
SHR		static	int32	MAT
SHR2		static	int32	

Support User-Defined Standard Functions*

Automatic stubber adapts standard stubs to user version

- Eases compilation when using non-standard prototypes on standard functions
- Replaces previous implementation requiring `-D __polyspace_no_<function name>`



```
extern int strlen(char *src);

int fCallSolver(void *eptr)
{
        FSOLVER fptr;
        int res = strlen("fSolver");
        fptr =
        res = (*fptr)(eptr);
        return res;
```

assignment to local variable 'res' (int 32): 7

* C code only

Easier Project Configuration and Use

Resolve relative paths automatically in the project file

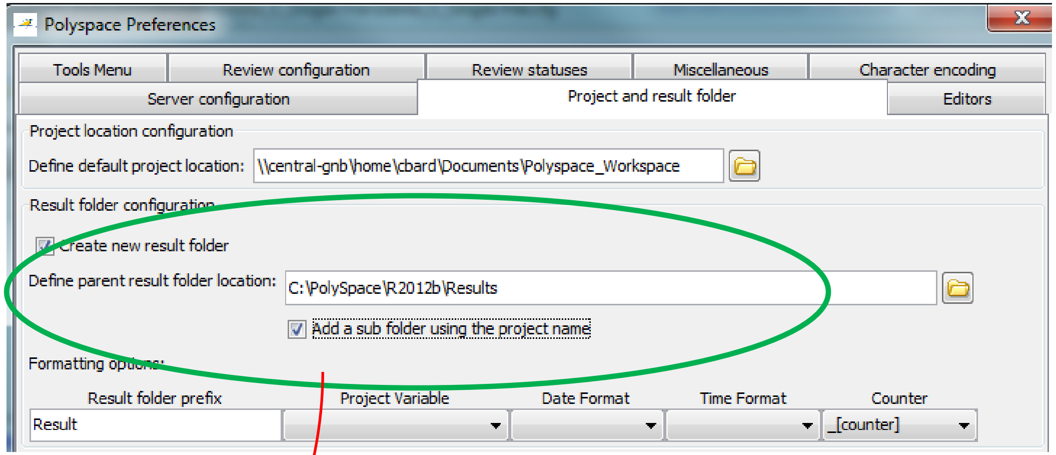
- Relative paths are found when a project file is opened, based on both:
  - The project file location
  - Absolute paths stored in the project

- A fallback mechanism exists:
  - Use absolute paths if relative paths do not exist

Define Folder Location for Parent Results

Add a root directory for all result folders

- All result folders can be grouped in a root folder.
- Results can be grouped hierachically.



Polyspace Preferences

Project location configuration

Define default project location: \\central-gnb\home\cbard\Documents\Polyspace_Workspace

Result folder configuration

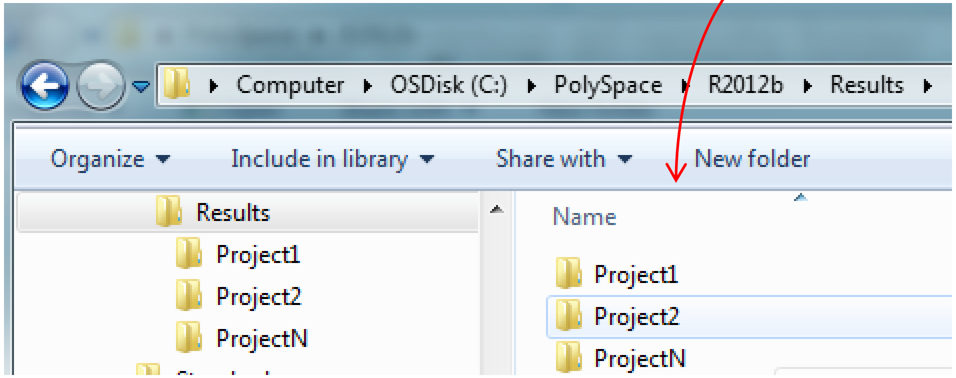
☑ Create new result folder

Define parent result folder location: C:\PolySpace\R2012b\Results

☑ Add a sub folder using the project name

Formatting options:

Result folder prefix	Project Variable	Date Format	Time Format	Counter
Result				_[counter]



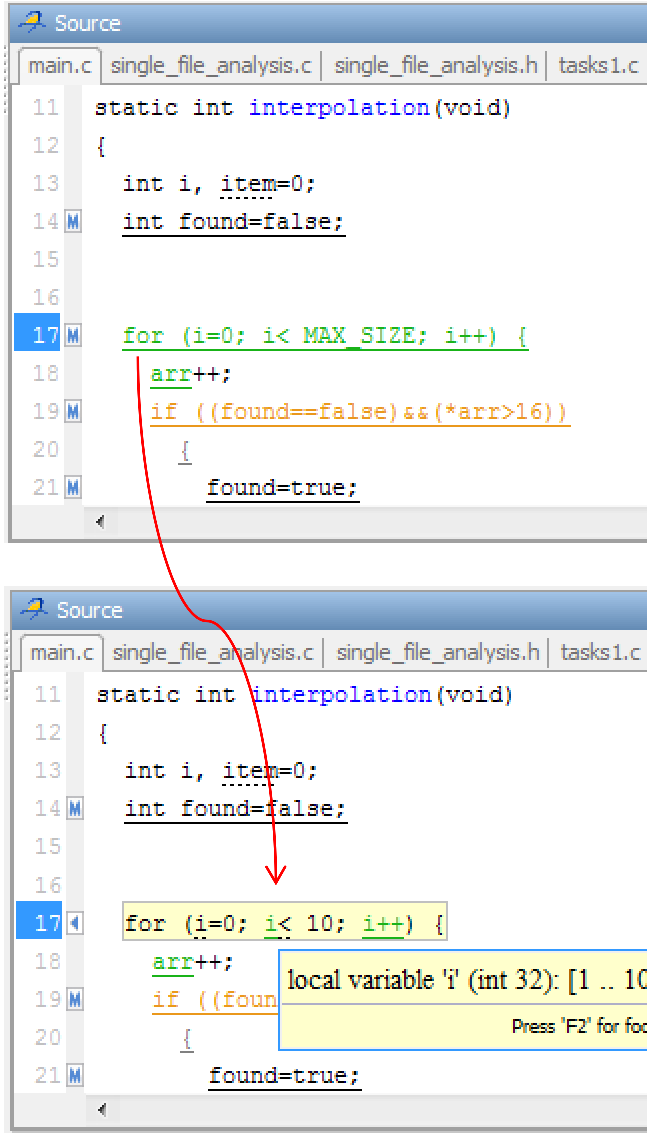
Computer ▸ OSDisk (C:) ▸ PolySpace ▸ R2012b ▸ Results ▸

Macro Expansion in Source Code View*

Single source code window to review all Polyspace checks

- Macros now expand directly in the source code view.
- These macros are identified with an M icon.
- Macro code can be read directly in the context of the surrounding code.

* C and C++ code only



```
static int interpolation(void)
{
  int i, item=0;
  int found=false;


  for (i=0; i< MAX_SIZE; i++) {
    arr++;
    if ((found==false)&&(*arr>16))
      {
        found=true;
```

```
static int interpolation(void)
{
  int i, item=0;
  int found=false;


  for (i=0; i< 10; i++) {
    arr++;
    if ((foun
      {
        found=true;
```

local variable 'i' (int 32): [1 .. 10

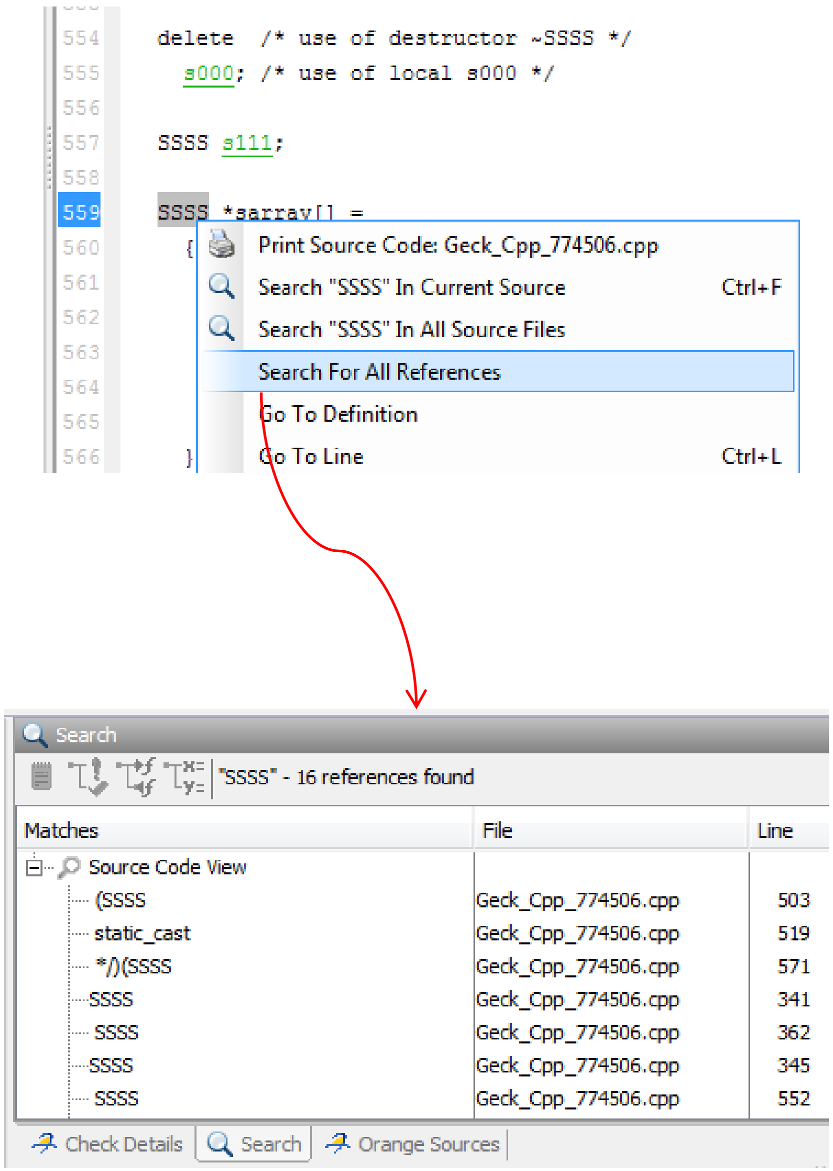
Press 'F2' for foc

Navigation Improvements*

Enhanced search features

- Search references for:
  - Class
  - Type
  - Local variable
  - Global variable
  - Function
- Click a reference in list to navigate directly to the source code

* C and C++ code only



```
delete  /* use of destructor ~SSSS */
  s000; /* use of local s000 */

SSSS s111;

SSSS *sarray[] =
  {

  }
```

Print Source Code: Geck_Cpp_774506.cpp
Search "SSSS" In Current Source Ctrl+F
Search "SSSS" In All Source Files
Search For All References
Go To Definition
Go To Line Ctrl+L
Search

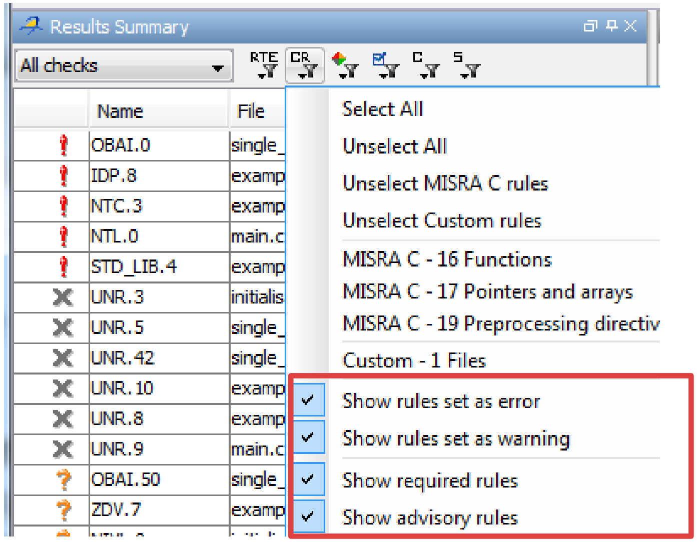
"SSSS" - 16 references found

Matches	File	Line
Source Code View		
(SSSS	Geck_Cpp_774506.cpp	503
static_cast	Geck_Cpp_774506.cpp	519
*/)(SSSS	Geck_Cpp_774506.cpp	571
SSSS	Geck_Cpp_774506.cpp	341
SSSS	Geck_Cpp_774506.cpp	362
SSSS	Geck_Cpp_774506.cpp	345
SSSS	Geck_Cpp_774506.cpp	552

Check Details
Search
Orange Sources

Review Enhancements

- **Improved filtering capabilities**
  - Filters are now available in the Results Summary view.
  - New coding rule filters for MISRA distinguish rules set as error or warning as well as required or advisory rules.

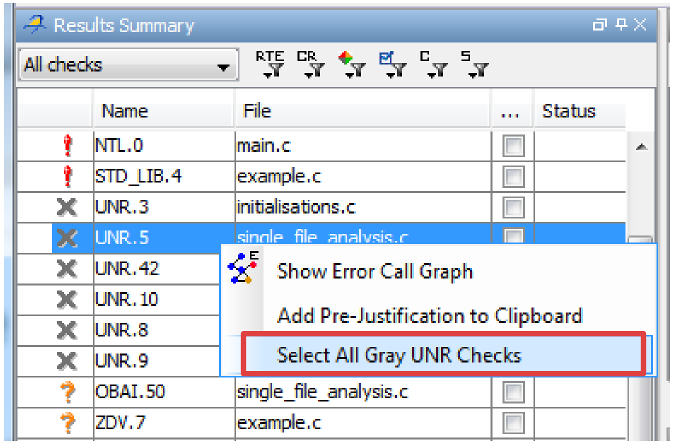


Results Summary

All checks

	Name	File
!	OBAI.0	single_
!	IDP.8	examp
!	NTC.3	examp
!	NTL.0	main.c
!	STD_LIB.4	examp
✕	UNR.3	initialis
✕	UNR.5	single_
✕	UNR.42	single_
✕	UNR.10	examp
✕	UNR.8	examp
✕	UNR.9	main.c
?	OBAI.50	single_
?	ZDV.7	examp

- **Easier to review checks:**
  - Ctrl-click to select multiple checks to justify in one click
  - Right-click check to select all checks of the same category



Results Summary

All checks

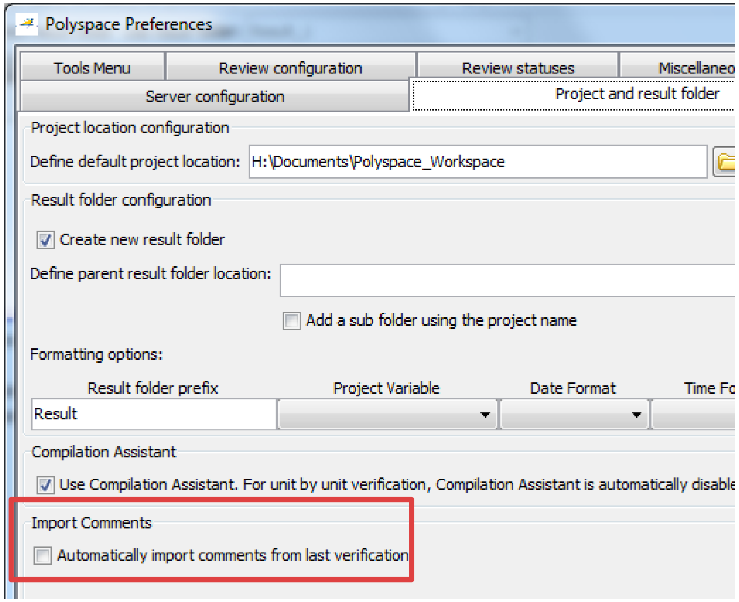
	Name	File	...	Status
!	NTL.0	main.c	□	
!	STD_LIB.4	example.c	□	
✕	UNR.3	initialisations.c	□	
✕	UNR.5	single_file_analysis.c	□	
✕	UNR.42			
✕	UNR.10			
✕	UNR.8			
✕	UNR.9			
?	OBAI.50	single_file_analysis.c	□	
?	ZDV.7	example.c	□	

Show Error Call Graph
Add Pre-Justification to Clipboard
Select All Gray UNR Checks

Review Enhancements

MathWorks R2012b

- **Easier to import comments:**
  - Import comments from previous verification



Polyspace Preferences

Project location configuration

Define default project location: H:\Documents\Polyspace_Workspace

Result folder configuration

☑ Create new result folder

Define parent result folder location:

□ Add a sub folder using the project name

Formatting options:

Result folder prefix	Project Variable	Date Format	Time Fo
Result			

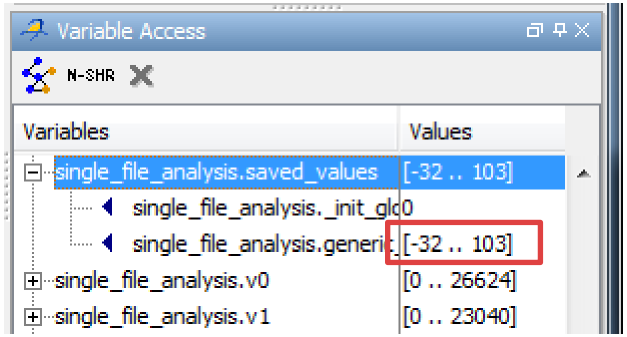
Compilation Assistant

☑ Use Compilation Assistant. For unit by unit verification, Compilation Assistant is automatically disable

Import Comments

□ Automatically import comments from last verification

- **Better understanding of results:**
  - Variables and values are provided on each read and write access.
  - Users can more easily find which write access on a variable is the source of an issue.



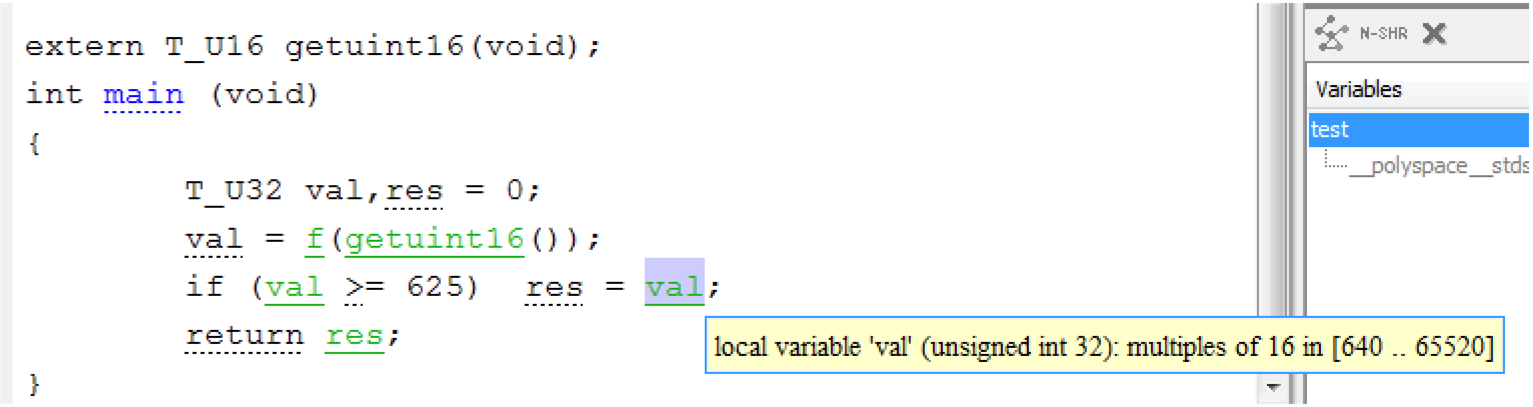
Variable Access

Variables	Values
single_file_analysis.saved_values	[-32 .. 103]
single_file_analysis._init_glc	0
single_file_analysis.generic	[-32 .. 103]
single_file_analysis.v0	[0 .. 26624]
single_file_analysis.v1	[0 .. 23040]

More Comprehensive Tooltip Data

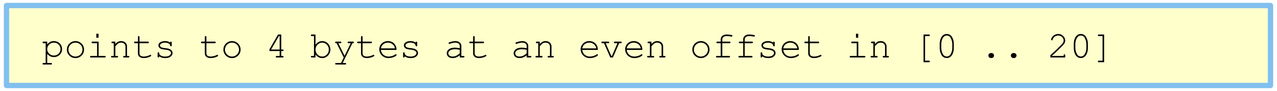
Shows hidden constraints on variables

- Information such as odd/even, multiples, etc.
- Additional details for pointers



```
extern T_U16 getuint16(void);
int main (void)
{
        T_U32 val,res = 0;
        val = f(getuint16());
        if (val >= 625)  res = val;
        return res;
}
```

local variable 'val' (unsigned int 32): multiples of 16 in [640 .. 65520]



points to 4 bytes at an even offset in [0 .. 20]

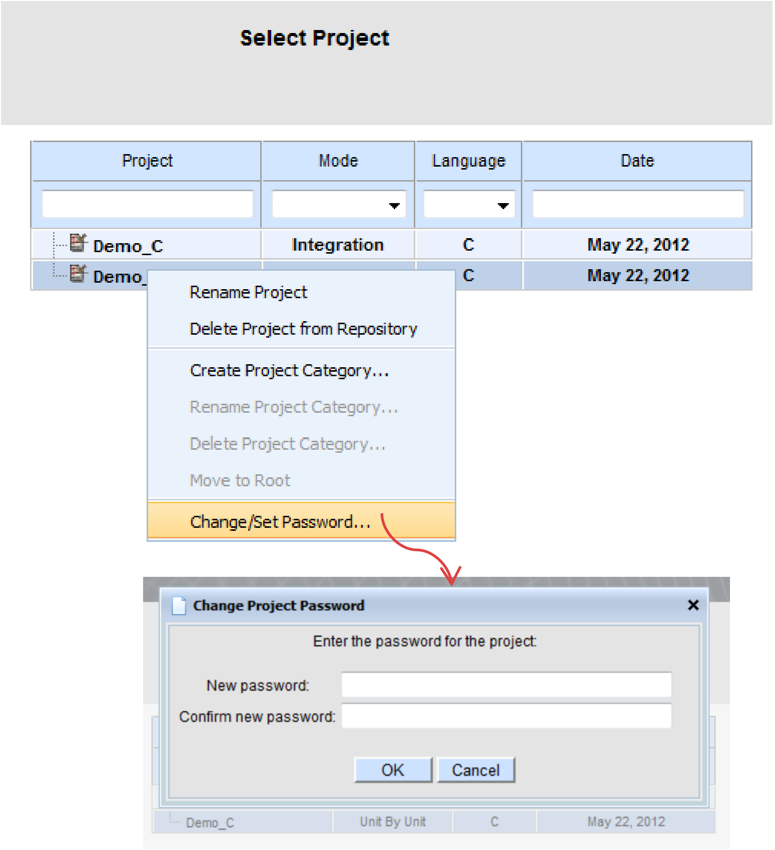
Additional Precision-Related Improvements

- Increase precision on global variables
  - When using multitasking
- Increase precision on math functions
  - Example: `sqrt ()` and `fmod()`
- Increase precision in nonsymbolic situations
  - Pointers and initialization
- Increase precision on multilinear expressions
  - When divisions are involved
- Increase precision on references for C++
- Increase precision on loop condition with `'!='` operator

Polyspace Metrics

Password protection to restrict and control access

- Set or change password directly from web page
- Require a password to access metrics, download results, and perform admin operations (rename, delete, etc.)



Select Project

Project	Mode	Language	Date
Demo_C	Integration	C	May 22, 2012
Demo_		C	May 22, 2012

Rename Project
Delete Project from Repository
Create Project Category...
Rename Project Category...
Delete Project Category...
Move to Root
Change/Set Password...

Change Project Password

Enter the password for the project:

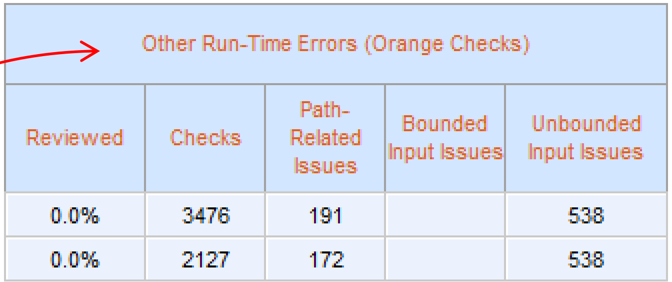
New password:

Confirm new password:

Demo_C	Unit By Unit	C	May 22, 2012

Level 0 bugs

- Run-time checks such as path-related issues, bounded input issues, unbounded input issues

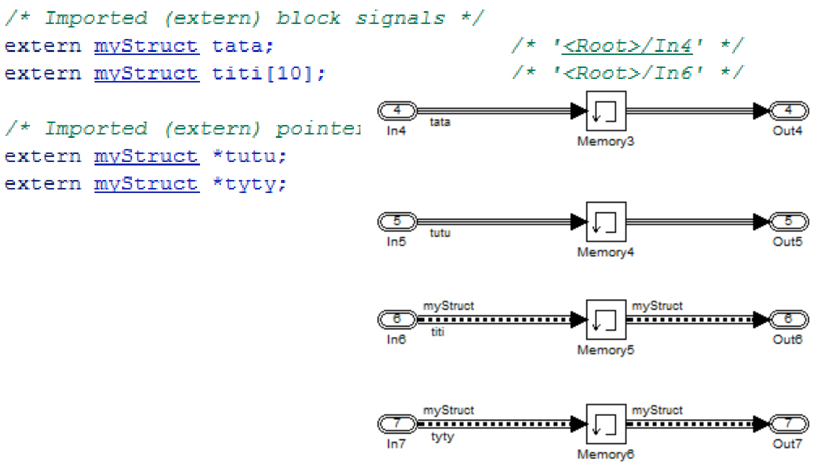


Other Run-Time Errors (Orange Checks)				
Reviewed	Checks	Path-Related Issues	Bounded Input Issues	Unbounded Input Issues
0.0%	3476	191		538
0.0%	2127	172		538

Polyspace Model Link Enhancements

Support nested structures on DRS for Polyspace Model Link

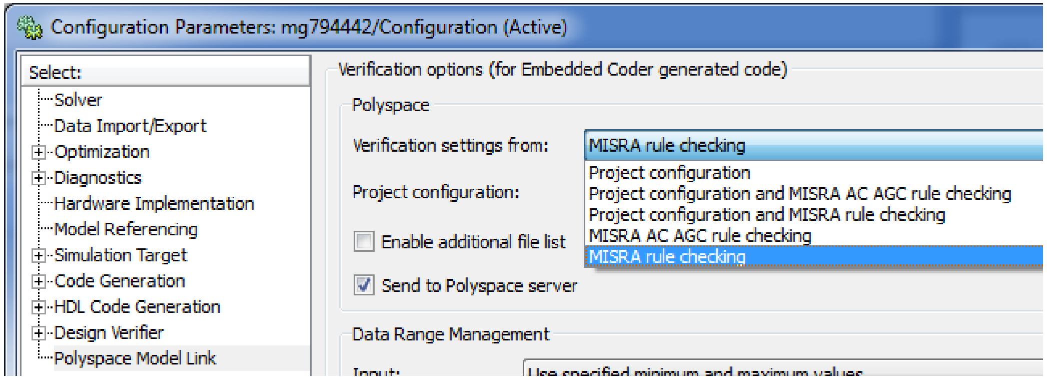
- Ranges on nested structures are automatically generated.
- DRS results in higher-precision verification.



```
/* Imported (extern) block signals */
extern myStruct tata;                /* '<Root>/In4' */
extern myStruct titi[10];            /* '<Root>/In6' */

/* Imported (extern) pointe
extern myStruct *tutu;
extern myStruct *tyty;
```

Select MISRA modes from the Embedded Coder configuration parameters panel



Configuration Parameters: mg794442/Configuration (Active)

Verification options (for Embedded Coder generated code)

Polyspace

Verification settings from: MISRA rule checking

- Project configuration
- Project configuration and MISRA AC AGC rule checking
- Project configuration and MISRA rule checking
- MISRA AC AGC rule checking
- MISRA rule checking

Project configuration:

☐ Enable additional file list

☑ Send to Polyspace server

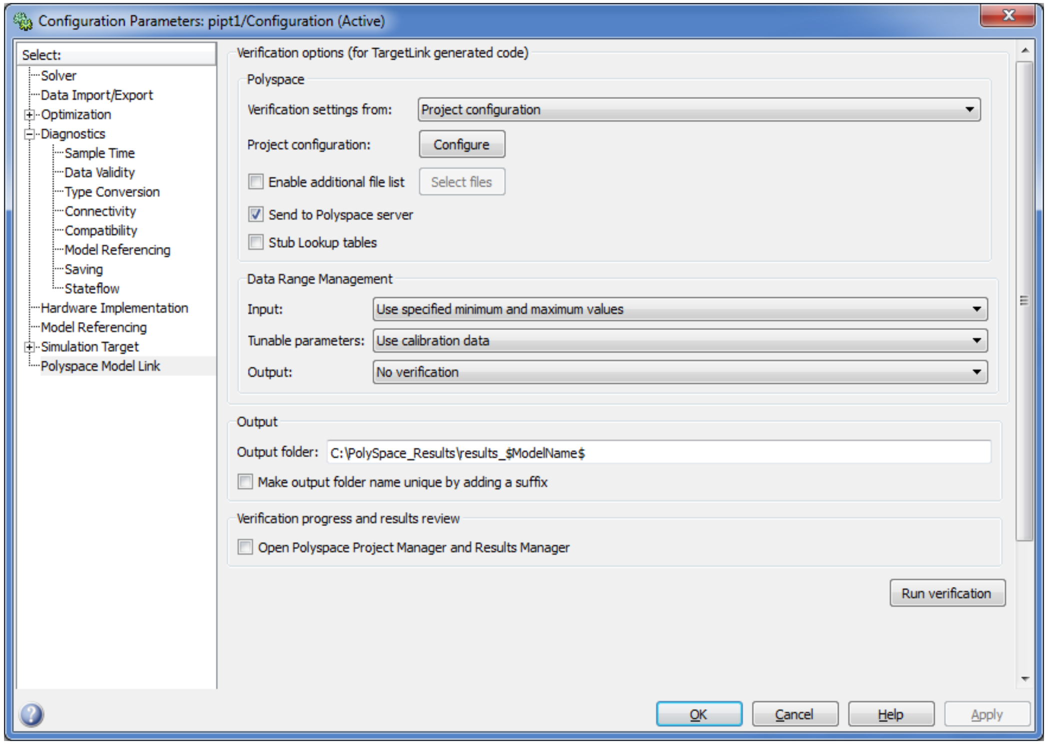
Data Range Management

Input: Use specified minimum and maximum values

Polyspace Model Link Enhancements

Enhanced support for TargetLink 3.1, 3.2, and 3.3

- Improve main generation and structure and data-range management
- Improve verification results



Verification options (for TargetLink generated code)

Polyspace

Verification settings from: Project configuration

Project configuration: Configure

□ Enable additional file list

☑ Send to Polyspace server

□ Stub Lookup tables

Data Range Management

Input: Use specified minimum and maximum values

Tunable parameters: Use calibration data

Output: No verification

Output

Output folder: C:\PolySpace_Results\results_$ModelName$

□ Make output folder name unique by adding a suffix

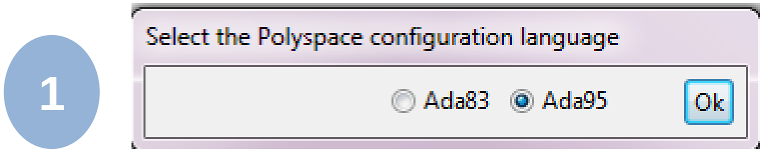
Verification progress and results review

□ Open Polyspace Project Manager and Results Manager

GNATbench Eclipse* Plug-In for Polyspace Client for Ada

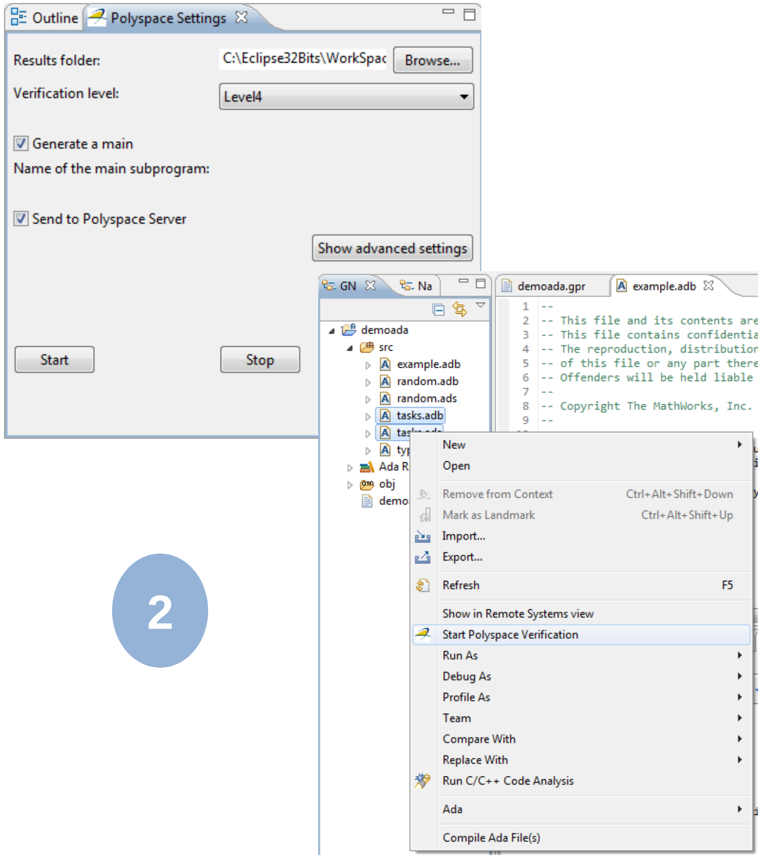
1. Select language

2. Launch a complete verification or a package analysis



Select the Polyspace configuration language

○ Ada83 ● Ada95 Ok



Results folder: C:\Eclipse32Bits\WorkSpac Browse...

Verification level: Level4

☑ Generate a main

Name of the main subprogram:

☑ Send to Polyspace Server

Show advanced settings

Start Stop

```
--
-- This file and its contents are
-- This file contains confidentia
-- The reproduction, distribution
-- of this file or any part there
-- Offenders will be held liable
--
-- Copyright The MathWorks, Inc.
--
```

* GNATbench plug-in 2.5.1 for Eclipse requires Eclipse 32-bit, version 3.5.x or 3.6.x for Windows