Polyspace® quick reference card

The colors of Polyspace

Called function that contains grey code (and orange or green checks but no red)

Called function that contains green checks only

Called function that contains no check or checks that are filtered

Function names are colored in blue (syntax highlighting)

Unused global variable

Grey access : never executed

Global variable not shared

Shared and protected variable (green = no concurrent access problems)

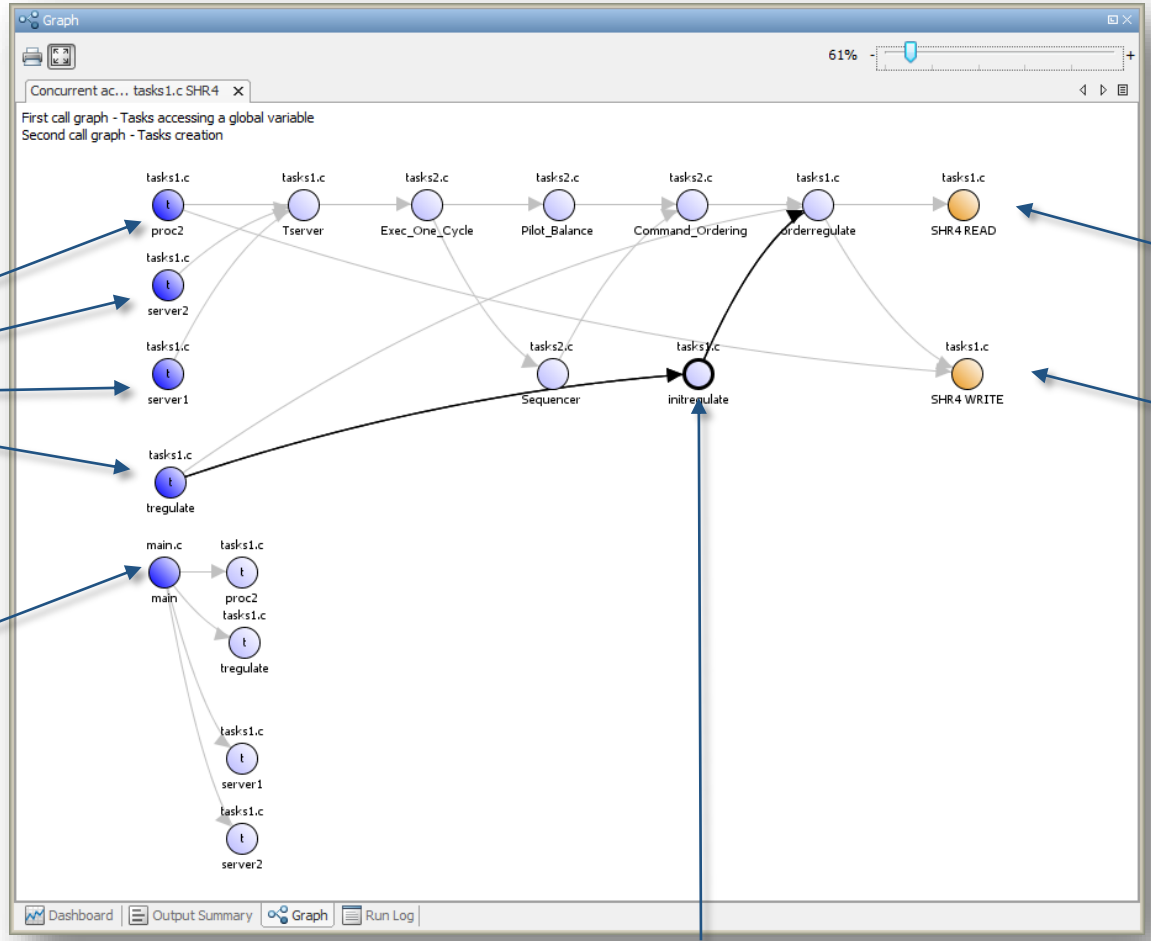
Shared and unprotected variable (orange = possible concurrent access problems)

Link on the whole line: it contains a macro (click to see the expanded code)

The current check is highlighted

Dashed underlined link: means that there is a tooltip

The access graph for global variables



The tasks that read or write the variable are on the left part of the graph

The main launches the tasks that will indirectly access the variable

The read and write accesses are represented by two orange or green spots.

Green if the variable is protected, orange if not.

Move the pointer over a node to highlight (bold line) the paths that directly lead to this item (the callers) and to the called functions (the callees) or operations (access)

Polyspace® quick reference card

C Acronyms

		Explanation
ZDV	**Z**ero **D**ivision	Division by zero
NIV	**N**on **I**nitialized **V**ariable	Variable used but not initialized (contains an unpredictable value)
NIVL	**N**on **I**nitialized **V**ariable **L**ocal	Local variable used but not initialized
COR	**Cor**rectness condition	Mix of other RTEs, mainly related to function pointers
OBAI	**O**ut of **b**ounds **A**rray **I**ndex	Index used to access an array element is too large or negative
IRV	**I**nitialized **R**eturn **V**alue	The function does not return an initialized variable
IDP	**I**llegally **D**ereferenced **P**ointer	Covers any issues dealing with pointer dereference : the pointer is null, the type of the pointed object is not the expected one…
SHF	**Shf**t	Left shift on negative variables / shift amount is too large
OVFL	**Ov**er**fl**ow	The assigned value is too large for the destination type
UNR	**Unr**eachable	Shows a statement that is never reached
NTL	**N**on **T**ermination of **L**oop	The loop never terminates or not as expected (before the end condition)
NTC	**N**on **T**ermination of **C**all	The call to this function never terminates (because of a RTE, an infinite loop…)
NIP	**N**on **I**nitialized **P**ointer	Pointer used but not initialized
STD_LIB	**Stand**ard **Lib**rary	Error when calling a function of the Ansi-C Standard Library
ASRT	**Ass**er**t**ion failure	The assertion condition is false

Keyboard shortcuts

Project Manager perspective

Open project	Ctrl O
New project	Ctrl N
Save project	Ctrl S

Run-time checks perspective

Save	Ctrl S
Cut	Ctrl X
Copy	Ctrl C
Paste	Ctrl V
Goto line	Ctrl L
Maximize current view	Ctrl M

Global

Search	Ctrl F

Code annotations

Introduced as comments (with `//` or `/* … */`)

Syntax for coding rules checker

`polyspace<`*`checker`*`:`*`rule_number[,rule2[,…]]`*
`[:class[:status]]``>` *`Comment`*

where '*`checker`*'**=** `MISRA-C, MISRA-CPP` **or** `JSF`
and '*`class`*'**=** `Unset, High, Medium, Low, Not a defect`
and '*`status`*'**=** `Undecided, Investigate,…`

Example 1:

```
/* polyspace<MISRA-C:11.3> Absolute address */
```

Example 2:

```
// polyspace<JSF:56,62:Medium> Robustness
```

Code annotations

For one block:

polyspace:begin<*checker*:*rule_number[,rule2[,…]]*[:*class[:status]]*> *Comment*

…

polyspace:end<*checker:rule_number[,rule2[,…]]*[:*class[:status]]*>

Example:

```
// polyspace:begin<JSF:33> Under development
…
// polyspace:end<JSF:33>
```

Code annotations

Syntax for checks

polyspace<RTE:*rte1[,rte2[,…]] [:class[:status]]*> *Comment*

where '*rte*' = NIV, OBAI, IDP…
and '*class*' = Unset, High, Medium, Low, Not a defect
and '*status*' = Undecided, Investigate,…

Example:

```
/* polyspace<RTE:NIV,IDP:Medium> Absolute address */
```

Code annotations

For one block

polyspace:begin<RTE:*rte1[,rte2[,…]] [:class[:status]]*>
Comment
…
polyspace:end<RTE:*rte1[,rte2[,…]] [:class[:status]]*>

Example:

```
/* polyspace:begin<RTE:NIV,IDP:Low:Investigate>
Absolute address */
…
/* polyspace:end<RTE:NIV,IDP:Low:Investigate> */
```