

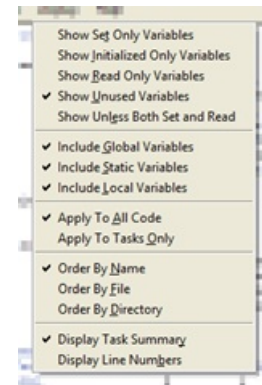
## Unused Variables

The Unused Variables report might be considered a collection of five reports. You can choose whether you want to review variables that are set only, initialized only, read only, never unused, or not both set and read. Each of the five alternatives locates a different type of potential problem in your code. For example, consider the following source code:

```
int globalA, globalB;

int func2(int paramX, int paramY) {
    paramY = globalB;
    globalA = paramY;
    return globalB;
}

int func1(void) {
    int localW = 1;
    int localX, localY, localZ;
    return func2(localX, localY);
}
```



The reported results will depend greatly on which alternative you've selected. Unused Variables locates variables that are neither read nor set anywhere in the source code. The report lists each variable, along with information about where it is declared. These might simply be spurious declarations, but further review may show more serious problems, such as inconsistent naming, in the software being checked.

Variables That Are Never Used				
Settings:				
Usage Type:	never used			
Global Variables:	displayed			
Static Variables:	displayed			
Local Variables:	displayed			
Code Range:	all			
Summary				
Variables	Total	Set Only	Read Only	Unused
Global	2	1	1	0
Static	0	0	0	0
Local	6	2	2	1
Variable				
				File (Line)
localZ				unused_vars.c (12)

Initialized Only Variables locates variables that are assigned an initial value, but then not subsequently used in the source code. The potential causes of these are similar to the unused variables, except that the initialization make a spurious declaration less likely.

Variables That Are Initialized Only				
Settings:				
Usage Type:	initialized only			
Global Variables:	displayed			
Static Variables:	displayed			
Local Variables:	displayed			
Code Range:	all			
Summary				
Variables	Total	Set Only	Read Only	Unused
Global	2	1	1	0
Static	0	0	0	0
Local	6	2	2	1
Variable				
				File (Line)

Set Only Variables could indicate a failure to use important data. Note that you can filter by the variables' scope. Here, global variables are omitted from the report, which you might do if you are checking only a portion of your full system.

Variables That Are Set Only (including Initialized Only)

Settings:

```
Usage Type:      set only
Global Variables: omitted
Static Variables: displayed
Local Variables: displayed
Code Range:     all
```

Summary

Variables	Total	Set Only	Read Only	Unused
Static	0	0	0	0
Local	6	2	2	1

Variable

Variable	File (Line)
localW	unused_vars.c (11)
paramX	unused_vars.c (4)

By configuring the report to show variables which are read only, you're able to find problems such as uninitialized variables, which can lead to non-deterministic results when they're read.

Variables Which Are Read Only

Settings:

```
Usage Type:      read only
Global Variables: displayed
Static Variables: displayed
Local Variables: displayed
```

Summary

Variables	Total	Set Only	Read Only	Unused
Global	2	1	1	0
Static	0	0	0	0
Local	6	2	2	1

Variable

Variable	File (Line)
globalA	unused_vars.c (2)
localW	unused_vars.c (12)
paramX	unused_vars.c (4)

The Unless Both Set and Read alternative generates the most complete results. It effectively combines all variables reported by the other alternatives, listing every variable where the usage is suspect.

Note that in this run, the code range is set to Tasks Only. You can [define tasks](#) and then use those definitions to focus the analysis on variable usage within a specific task or a set of tasks. Because only one task exists in this tiny example, selecting Tasks Only does not have an impact here. Other reports focus on variable flow and data sharing between tasks.

Variables That Are Either Set Only (including Initialized Only), Read Only, or Never Used

Settings:

```
Usage Type:      non-complete (not both set and read)
Global Variables: omitted
Static Variables: displayed
Local Variables: displayed
Code Range:     tasks only
```

Task Definitions

Tasks are from Auto Task Generation: Any root functions

Name	Members	Graph	Root
autotask 1 - func1	2	[+]	func1

Summary

Variables	Total	Set Only	Read Only	Unused
Static	0	0	0	0

Local	6	2	2	1
Variable	File (Line)			
localW	unused_vars.c (11)			
localX	unused_vars.c (12)			
localY	unused_vars.c (12)			
localZ	unused_vars.c (12)			
paramX	unused_vars.c (4)			