[Notes]

R20TS0157EJ0100
Rev.1.00
Mar. 16, 2017

CS+ Integrated Development Environment

Outline

When using the CS+ integrated development environment, note the following point.

1. Acquisition of information on symbols of static variables and static functions inside a file when using CC-RH and CC-RL

1. Acquisition of Information on Symbols of Static Variables and Static Functions inside a File When Using CC-RH and CC-RL

1.1 Applicable Products

(1) If you are using the C compiler package for the RL78 family (CC-RL)

The version of the CS+ for CC common program is from V3.00.00 to V5.00.00.

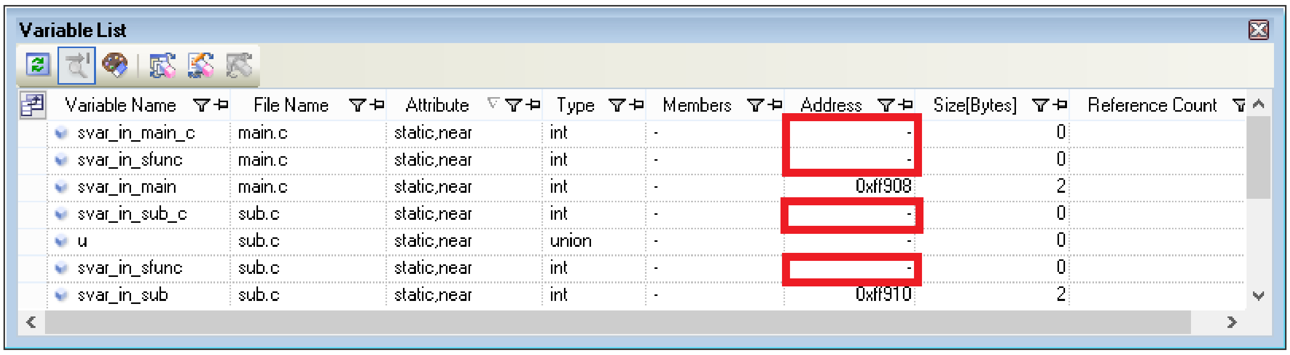
(2) If you are using the C compiler package for the RH850 family (CC-RH)

The version of the CS+ for CC common program is from V3.00.00 to V5.00.00, or the version of the CubeSuite+ common program is V2.01.00 or later.

1.2 Details

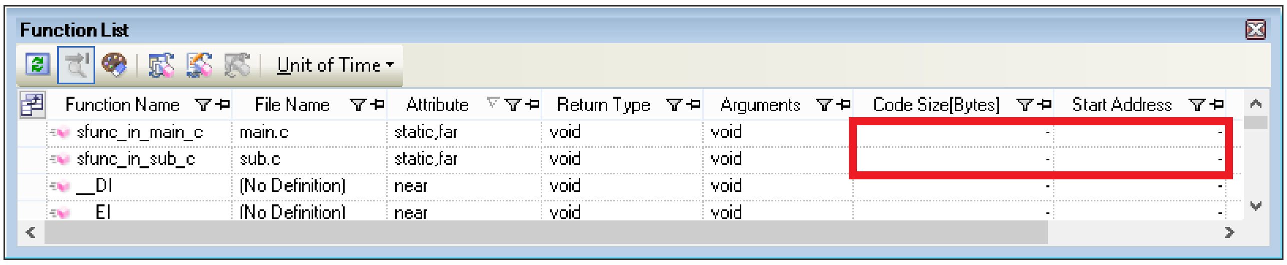
Information on symbols associated with static variables and static functions inside a file cannot be acquired and either of the following phenomena may occur:

(1) In the Variable List panel, the addresses of static variables inside a file are not displayed and a hyphen (-) is displayed instead.



Variable Name	File Name	Attribute	Type	Members	Address	Size[Bytes]	Reference Count
svar_in_main_c	main.c	static,near	int	-	-	0	
svar_in_sfunc	main.c	static,near	int	-	-	0	
svar_in_main	main.c	static,near	int	-	0xff908	2	
svar_in_sub_c	sub.c	static,near	int	-	-	0	
u	sub.c	static,near	union	-	-	0	
svar_in_sfunc	sub.c	static,near	int	-	-	0	
svar_in_sub	sub.c	static,near	int	-	0xff910	2	

(2) In the Function List panel, the code sizes and start addresses of static functions inside a file are not displayed and a hyphen (-) is displayed instead.

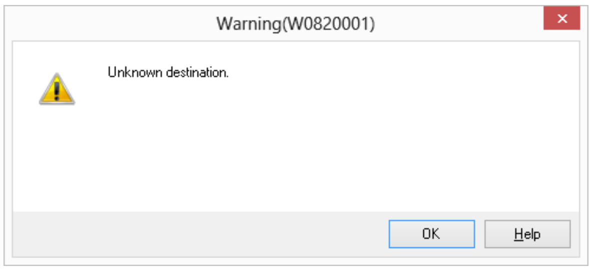


Function Name	File Name	Attribute	Return Type	Arguments	Code Size[Bytes]	Start Address
sfunc_in_main_c	main.c	static,far	void	void	-	-
sfunc_in_sub_c	sub.c	static,far	void	void	-	-
__DI	(No Definition)	near	void	void	-	-
EI	(No Definition)	near	void	void	-	-

(3) In the Call Graph panel or Class/Member panel, when you execute either of the following context menus in the selected static variable or static function inside a file, the "W0820001 (Unknown destination.)" warning message appears.

- [Jump to Disassemble]
- [Jump to Memory]

Remark: The context menu is displayed by right-clicking the mouse.



Warning(W0820001)

Unknown destination.

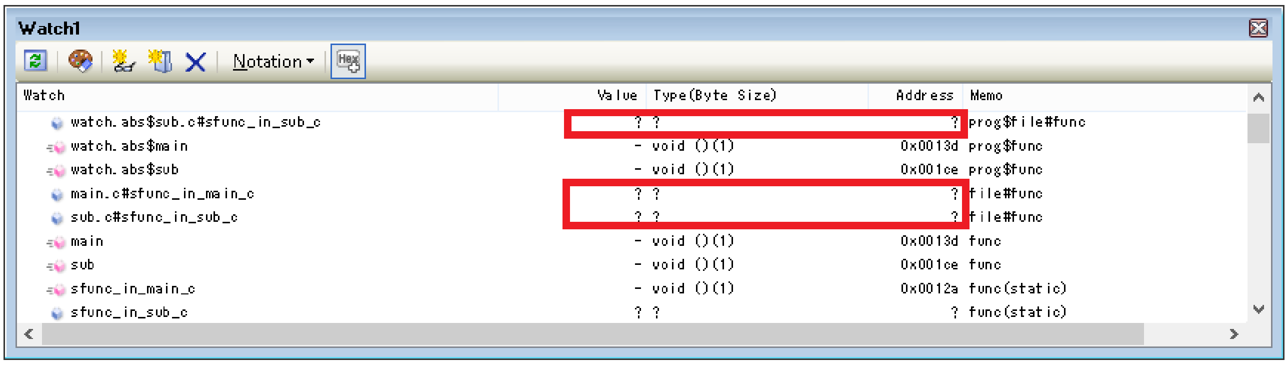
(4) When you specify the following settings in the [Download File Settings] tab on the Debug Tool, the program does not run to the location of the specified symbol, and instead breaks at the location immediately after CPU reset:

[CPU Reset after download]: Yes (default)

[Execute to the specified symbol after CPU Reset]: Yes (default)

[Specified symbol]: Specify a static function inside a file.

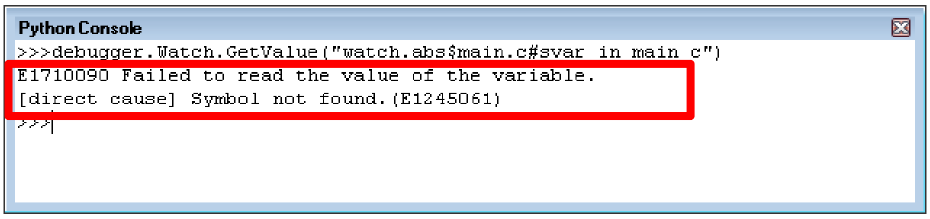
(5) In the Watch panel, the values, types, and addresses of static variables(Note) or static functions(Note) inside a file are not displayed and a question mark (?) is displayed instead.



Watch	Value	Type(Byte Size)	Address	Memo
watch.abs$sub.c#sfunc_in_sub_c	?	?	?	prog$file#func
watch.abs$main	-	void ()(1)	0x0013d	prog$func
watch.abs$sub	-	void ()(1)	0x001ce	prog$func
main.c#sfunc_in_main_c	?	?	?	file#func
sub.c#sfunc_in_sub_c	?	?	?	file#func
main	-	void ()(1)	0x0013d	func
sub	-	void ()(1)	0x001ce	func
sfunc_in_main_c	-	void ()(1)	0x0012a	func(static)
sfunc_in_sub_c	?	?	?	func(static)

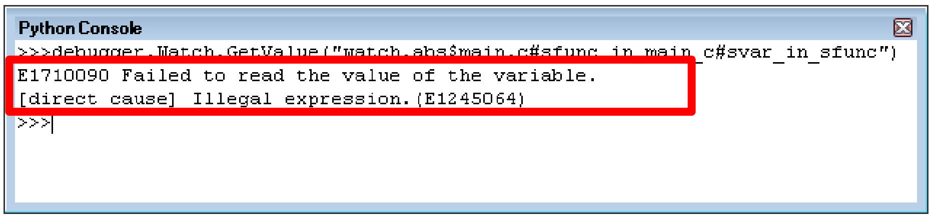
(6) When static variables(Note) inside a file are referenced (debugger.Watch.GetValue) or set (debugger.Watch.SetValue) in the Python console, any of the following errors occurs:

(a) E1710090 Failed to read the value of the variable.
[direct cause] Symbol not found. (E1245061)



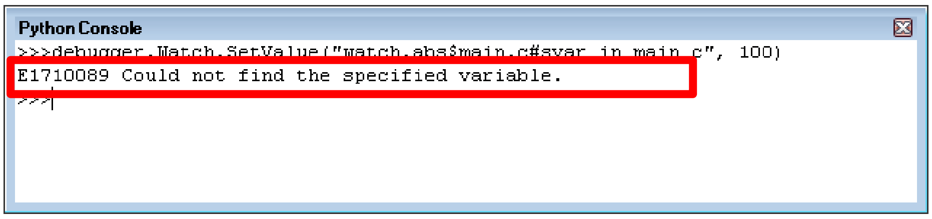
```
>>>debugger.Watch.GetValue("watch.abs$main.c#svar_in_main_c")
E1710090 Failed to read the value of the variable.
[direct cause] Symbol not found.(E1245061)
>>>
```

(b) E1710090 Failed to read the value of the variable.
[direct cause] Illegal expression. (E1245064)



```
>>>debugger.Watch.GetValue("watch.abs$main.c#sfunc_in_main_c#svar_in_sfunc")
E1710090 Failed to read the value of the variable.
[direct cause] Illegal expression.(E1245064)
>>>
```

(c) E1710089 Could not find the specified variable.



```
>>>debugger.Watch.SetValue("watch.abs$main.c#svar_in_main_c", 100)
E1710089 Could not find the specified variable.
>>>
```

Note: This is only applied when the source file name is used with a specified scope.

<Example>

- prog$file#func
- file#func
- prog$file#func#var
- prog$file#var
- file#func#var
- file#var

prog: Load module name
file: Source file name
func: Function name
var: Variable name

For details about scope specification, see the following user's manual at the link on the web page below:
https://www.renesas.com/search/keyword-search.html#genre=document&q=r20ut3939
CS+ V5.00.00 Integrated Development Environment User's Manual: RL78 Debug Tool
A. WINDOW REFERENCE
Watch panel

1.3 Condition

This error may occur when a variable or function is declared with a static qualifier in a source file.

1.4 Workaround

(1) If you are using the C compiler package for the RL78 family (CC-RL)

Perform (a) and (b), and then if the problem is not resolved, perform also (c).

Note that when the version of CC-RL is earlier than V1.03, you only need to perform (c).

(a) Do not use the -vfinfo option.

(b) Do not use the __callt keyword or #pragma callt directive in the applicable source file.

(c) Define or allocate an empty function in __near type as follows:

CC-RL V1.02 or earlier: Define an empty function at the beginning of the source file.

CC-RL V1.03 or later: Allocate an empty function at the beginning of the .text section.

Example of an empty function in __near type:

```
__near void dummy() {}
```

(2) If you are using the C compiler package for the RH850 family (CC-RH)

Define or allocate an empty function as follows:

CC-RH V1.03 or earlier: Define an empty function at the beginning of the source file.

CC-RH V1.04 or later: Allocate an empty function at the beginning of the .text section.

Example of an empty function:

```
void dummy() {}
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in CS+ for CC V6.00.00.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 16, 2017	-	First edition issued

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan

Renesas Electronics Corporation

■Inquiry
https://www.renesas.com/contact/