

RENESAS TOOL NEWS on October 16, 2004: RSO-SQMLint-041016D

MISRA C Rule Checker SQMLint Revised to V.1.01 Release 00

We have revised the SQMLint (a MISRA C rule checker) from V.1.00 Release 1A to V.1.01 Release 00.

1. Descriptions

1.1 Supported C Compiler Packages Increased

The SQMLint supports the following C compiler packages as well as those for the M32R and M16C family of MCUs:

All the C compiler packages for the SuperH RISC engine family whose version numbers are V.9.00 Release 00 and later.
(Windows version only)

1.2 Specifications Modified

- (1) When option "-ignore 1" selected, all the violations of MISRA C Rule 1 have hitherto been ignored. In this revision, the above point of view is modified as follows:
Only the messages concerning to the extended language specifications are ignored, and the syntax violations in C language are displayed on-screen.
- (2) When a function has a parameter of any enumeration type, passing an enumerator of this type as an argument to the function has hitherto been interpreted as a violation of Rule 77. In this revision, it is ignored by the rule.
- (3) Rule 18 requires that a suffix indicating a data type be added to constant value, if possible. So, if suffix U and/or L is not added to integer "1", it has been interpreted as the violation of this rule.
On the other hand, to make a negative constant, grammatically the unary - operator is operated on a positive constant. This means that "-1" is not a value but an expression, so "-1" has not been interpreted as a violation of Rule 18. In this revision, the expressions that are the result of the unary + operator and unary - operator of which operand is a constant value, and the parenthesized constant values, are considered to be constant values.

- (4) The case where a pointer to function is declared, and the pointer is used in function call has not been interpreted as a violation of Rule 22 in the SQMLint. In this revision, the above case is interpreted as a violation of Rule 22 as it says.
- (5) Rule 82 requires that the end of a function be only one. By the way, if "return" is placed in a local block, this means that there exists an implicit "return" at the end of a function, indicating the function has two ends. In this revision, the above case is interpreted as a violation of Rule 82.
- (6) Rule 85 prohibits placing a function identifier that is not followed by parentheses in a statement. However, this type of function identifier is generally used to express the initial value of a pointer to a function. So in this revision, the case where a function identifier is used for the above purpose is excluded from the interpretations of Rule 85.
- (7) The case where the name of a member of a structure and that of an auto variable are the same has hitherto been interpreted as a violation of Rule 21. In this revision, however, the above case is admitted by the rule because "MISRA C Technical Clarification" says the above case does not violate Rule 21.
- (8) When both a prototype declaration and a function definition exist, and a parameter in the prototype declaration has no name, the case has been interpreted as a violation of Rule 74.
In this revision, the above case is ignored by the rule because Rule 73 admits the parameters that have no names in a prototype declaration.

2. How to Revise the Current Product and Order the Revised One

2.1 Online Upgrade (without Charge)

Download the revised product from **HERE**.

2.2 First Ordering

When you place an order for the product, please provide the following information for your local Renesas Technology sales office or distributor:

Product Type : R0C00000SCW01R

Version No. : V.1.01

Release No. : Release 00

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.