

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

===== Be sure to read this note. =====

## C Compiler Package V.5.20 Release 02

for M32C/90, M32C/80, M16C/80 Series

### Release note

(Rev.4.0)

#### Renesas Solutions Corporation

Jun 1, 2005

#### Abstract

Welcome to C Compiler Package V.5.20 Release 02 for M32C/90, M32C/80, M16C/80 Series. This document contains supplementary descriptions to User's Manual. When you read certain items in the User's manual, please read this document as well.

Also, this document contains a License Agreement in the last. Please read it before using. By using the software, you are accepting and agreeing to such terms.

1.	Precautions on Product .....	3
1.1.	Precautions about Compiler .....	3
1.1.1.	On using if-else constructs .....	3
1.1.2.	On passing the address of an object qualified as const as an argument to a function.....	4
1.1.3.	On nesting inline functions .....	4
1.1.4.	Precautions about command option I .....	5
1.1.5.	Precautions about the search of an include file .....	5
1.1.6.	Precautions to be taken when using #pragma ASM/ENDASM and asm() .....	5
1.1.7.	Precautions about debugging of a program using _Bool type.....	5
1.1.8.	Precautions regarding the preprocessing directive #define.....	5
1.2.	Precaution of Assembler.....	6
1.2.1.	On using assemble option "-mode60" and "-mode60p" .....	6
1.3.	Precaution of MCU-Dependent Code.....	6
1.3.1.	M16Precautions regarding the M16C interrupt control register.....	6
1.3.2.	Precautions about access of SFR area.....	7
1.4.	Precautions about High-performance Embedded Workshop.....	7
1.4.1.	About the debugging function .....	7
1.4.2.	About the start of MAP viewer.....	7
1.5.	Precautions about TM.....	7
1.6.	Precautions about MS-Windows.....	7
1.6.1.	Precautions about environment of operation .....	7
1.6.2.	Suggestions Concerning File Names .....	8
1.6.3.	Precautions about virus check programs .....	8
1.6.4.	Precautions when upgrading .....	8
2.	Installing C Compiler Package .....	9
2.1.	Before installing C Compiler Package .....	9
2.2.	Precaution about installing this product.....	9
2.3.	C Compiler Package Installer.....	9
2.4.	Installation procedure.....	9
2.5.	Setting environment after installation.....	10

3.	Entering user registration .....	10
3.1.	User registration .....	10
4.	Contents of upgrade .....	10
4.1.	Function addition and revision of Compiler .....	10
4.1.1.	Change to error processing .....	10
4.2.	Problem correction of Compiler .....	10
4.3.	Functional revision of Assembler .....	11
4.3.1.	Revision of the function to set ID code and ROM protect code .....	11
4.4.	Problem correction of Assembler .....	11
5.	Software version list of C Compiler Package V.5.20 Release 02 .....	12
6.	Versions Useful for the Real-time Operating System for M32C/90, M32C/80, M16C/80 Series .....	12
7.	Conformance with MISRA C Rule in Standard Function Library .....	13
7.1.	Cause of Rule Violation .....	13
7.2.	Inspection No. running counter to the rule .....	13
7.3.	Evaluation Environment .....	13
8.	A Guide to Porting Projects Created with TM to High-performance Embedded Workshop V.4 .....	14
8.1.	Summary .....	14
8.2.	Porting Procedure .....	14
8.3.	Usage Notices .....	15
8.3.1.	TM-to-High-performance Embedded Workshop Portable and Non-Portable Information .....	15
8.3.2.	Cross Tools .....	15
8.3.3.	High-performance Embedded Workshop Versions .....	16
8.3.4.	Generated Project Workspace .....	16
8.3.5.	Load Module Converter .....	16
8.3.6.	Other Tools .....	17
8.3.7.	Linkage order .....	20
8.3.8.	Placing the Start Up program at the top of Linkage Order .....	20
9.	License Agreement .....	21

## 1. Precautions on Product

When using the compiler, please be sure to follow the precautions and suggestions described below.

### 1.1. Precautions about Compiler

#### 1.1.1. On using if-else constructs

- **Description**  
Regardless of whether the result of evaluation of the controlling expression in an if statement is TRUE or FALSE, incorrect code will be generated if statements for assigning a constant to a member of a bit field exist in the TRUE and FALSE statements.
- **Conditions**  
This problem occurs if the following conditions are all satisfied:
  - (1) An if-else construct exists.
  - (2) Regardless of whether the result of evaluation of the controlling expression in an if statement in (1) is TRUE or FALSE, constants are assigned to members of bit fields of a structure, where the bit fields have the same name as variables.
  - (3) The bit fields to which constants are assigned in (2) are 1 bit wide (those not assigned can be wider than 1).
  - (4) The bit fields in (2) are different in bit positions depending on whether each of them is put in the TRUE or FALSE statement.
  - (5) The constants in (2) are also different depending on whether each of them is assigned in the TRUE or FALSE statement.
- **Examples**

```

char    c;

struct S
{
    int    b0:1;        /* Conditions (3) and (4) */
    int    b1:1;        /* Conditions (3) and (4) */
    int    b2:1;
    int    b3:1;
    int    b4:1;
    int    b5:1;
    int    b6:1;
    int    b7:1;
    int    b8:8;
};

void    func( void )
{
    if ( c == 1 ){      /* Condition (1) */
        s.b0 = 0;      /* Conditions (2), (3), (4), and (5) */
    }
    else{              /* Condition (1) */
        s.b1 = 1;      /* Conditions (2), (3), (4), and (5) */
    }
}

```

- **Workaround**  
Place a dummy asm() function anywhere in the else block.

```

void    func( void )
{
    if ( c == 1 ){
        s.b0 = 0;
    }
    else{
        s.b1 = 1;
        asm();          /* Dummy asm() function placed */
    }
}

```

### 1.1.2. On passing the address of an object qualified as const as an argument to a function

- Description

Consider that a function takes the address of an object qualified as const as an argument. When such a function is called, an error message appears even if a correct type of argument is passed. However, code is properly generated in this case.

- Conditions

This problem occurs if the following conditions are all satisfied:

- (1) A parameter to the function to be called is a pointer pointing to an object qualified as const.
- (2) The argument corresponding to the parameter in (1) is either of the following:
  - The result of operation where an address operator is applied to an object qualified as const
  - The name of an array qualified as const
- (3) The object and the array in (2) above are of the same type as the object pointed to by the pointer in (1).

- Examples

```

const int  ten = 10;          /* Condition (3) */
const int  arr[2] = { 1, 2 }; /* Condition (3) */

void subr(const int *);      /* Conditions (1) and (3) */

void mainr(void)
{
    subr(&ten);              /* Condition (2) */
    subr(arr);               /* Condition (2) */
}

```

#### Examples of messages

[Warning(ccom):example.c,line 8] assignment from const pointer to non-const pointer

====> subr(&ten);

[Warning(ccom):example.c,line 9] assignment from const pointer to non-const pointer

====> subr(arr);

- Workaround

Operate a cast operator on the argument to which a warning message is sent.

```

void mainr(void)
{
    subr((const int *)&ten); /* Cast operator operated */
    subr((const int *)arr);  /* Cast operator operated */
}

```

### 1.1.3. On nesting inline functions

- Description

When an inline function that takes a parameter is nested, it may refer to an incorrect argument (a variable,

not an argument).

- **Conditions**

This problem occurs if the following conditions are both satisfied:

- (1) An inline function is nested in another.
- (2) Inline function A as a calling source and inline function B as the destination take the same parameter.

- **Example**

```

inline    B(int aaa, char ccc)                /* Condition (2) */
{
    .....
}

inline    A(int c, int aaa, char *ccc)        /* Condition (2) */
{
    int    i;
    char   c;

    B(i,c);                                  /* Condition (1) */
}

```

- **Workaround**

This problem can be circumvented any of the following ways:

- (1) Change the name of the parameter taken by the destination function (inline function B in the above example).
- (2) Don't nest any inline function.
- (3) Compile the program using the `-Ofoward_function_to_inline(-OFFTI)` option.

#### 1.1.4. Precautions about command option `I`

The number of directories that can be specified by the command option `"-I"` is 50 or less.

#### 1.1.5. Precautions about the search of an include file

If you give a file to include together with a drive name in the `#include` line, and attempt to compile the file from a directory different from the one in which the file to compile is present, instances may occur in which the file to include cannot be searched.

#### 1.1.6. Precautions to be taken when using `#pragma ASM/ENDASM` and `asm()`

- Regarding debug information when using `#pragma ASM` outside functions, if you write `#pragma ASM` anywhere outside functions, no C source line information will be output. For this reason, information regarding descriptions in `#pragma ASM` to `#pragma ENDASM`, such as error message lines when assembling or linking and line information when debugging, may not be output normally.
- C compilers generate code of arguments to be passed via registers and of register variables by analyzing their scopes. However, if manipulations of register values are described using inline assemble functions (such as `#pragma ASM` / `#pragma ENDASM` directives and `asm` function), C compilers cannot hold information on the scopes of the above-mentioned arguments and register variables. So, be sure to save and recover register contents on and from the stack when registers are loaded using inline assemble functions described above.

#### 1.1.7. Precautions about debugging of a program using `_Bool` type

When you debug the program which uses the `BOOL` type, please confirm whether the debugger is supporting the `BOOL` type.

In using the debugger which is not supporting the `BOOL` type, please use a debugging option `"-gbool_to_char (-gBTC)"` at the time of compile.

#### 1.1.8. Precautions regarding the preprocessing directive `#define`

To define a macro which will be made the same value as the macro `ULONG_MAX`, always be sure to add the prefix `UL`.

## 1.2. Precaution of Assembler

### 1.2.1. On using assemble option "-mode60" and "-mode60p"

The following mnemonics with which functions are different from M16C/60 series cannot be replaced even. And Warning message is not output to them.

Therefore, please confirm the function of each mnemonic when these mnemonics are described in the source program.

mnemonic
SMOVB
SMOVF
RMPA

## 1.3. Precaution of MCU-Dependent Code

### 1.3.1. M16Precautions regarding the M16C interrupt control register

When the "-O5" optimizing option is used, the compiler generates in some cases BTSTC or BTSTS bit manipulation instructions. In M16C, the BTSTC and BTSTS bit manipulation instructions are prohibited from rewriting the contents of the interrupt control registers. However, the compiler does not recognize the type of any register, so, should BTSTC or BTSTS instructions be generated for interrupt control registers, the assembled program will be different from the one you intend to develop. For detailed information about this, see below "Precautions for Interrupts" Described in Related Documents (Excerpts).

When using any of the products concerned, ensure that no incorrect code is generated.

- Example

When the -O5 optimizing option is used in the program shown below, a BTSTC instruction is generated at compilation, which prevents an interrupt request bit from being processed correctly, resulting in the assembled program performing improper operations.

```
#pragma ADDRESS TA0IC 006ch // M16C/80 MCU's Timer A0 interrupt control register
struct {
    char    ILVL : 3;
    char    IR   : 1; // An interrupt request bit
    char    dmy : 4;
} TA0IC;

void WaitUntillRisON(void)
{
    while( TA0IC.IR == 0 ) // Waits for TA0IC.IR to become 1
    {
        ;
    } // Returns 0 to TA0IC.IR when is become 1
}
```

- Workaround

- (1) Suppress the generation of the BTSTC and BTSTS instructions resulting from using an optimizing option by selecting the -ONA (or -Ono\_asmopt) option together with "-O5" optimizing option.
- (2) Add an asm function to disable optimization locally, as shown in the example below.

```

void WaitUntillRisON( void )
{
    while( TA0IC.IR == 0 )
    {
        asm();
    }
}

```

- Notes

Make sure that no BTSTC and BTSTS instructions are generated after these side-steppings.

### 1.3.2. Precautions about access of SFR area

You may need to use specific instructions when writing to or reading registers in the SFR area. Because the specific instruction is different for each model, see the User's Manual for the specific Machine. These instructions should be used in your program using the asm function.

## 1.4. Precautions about High-performance Embedded Workshop

### 1.4.1. About the debugging function

High-performance Embedded Workshop of C Compiler Package V.5.20 Release 02 provides neither the simulator debugging function nor the emulator debugging function. Please prepare external debugger package such as Emulator Debugger and Simulator debugger for M32C/90, M32C/80, M16C/80 Series when you debug programs.

### 1.4.2. About the start of MAP viewer

MAP viewer cannot be started from the menu of High-performance Embedded Workshop with M16C, R8C, and M32C. MAP viewer must start according to the following procedure.

- (1) The Customize dialogbox is displayed clicking menu [Setup]->[Customize...].
- (2) The [Menu] tab is clicked. The Add Tool dialogbox is displayed clicking the [Add...] button.

Name	MAPViewer (Any name can be specified.)
Command:	C:\¥Renesas¥NC308WA¥V520R02¥BIN¥MapView.exe (MapView.exe that exists in the compiler installation folder is specified.)
Arguments	\$(CONFIGDIR)¥\$(PROJECTNAME).x30
Initial directory	\$(CONFIGDIR)

The name specified for the tool menu according to procedure (2) is added. Please start MAP viewer by clicking this name.

## 1.5. Precautions about TM

- As for integrated development environment TM, use Version 3.00 or a later version. C Compiler Package in this version cannot be used in Version 2.01 or in an earlier version. So be careful.
- In an attempt to divert a project generated by TM V.2 for TM V.3, the -finfo option is not turned effective either in compiling or in assembling. Choose -finfo separately. For details, see the Release note of TM V.3.

## 1.6. Precautions about MS-Windows

### 1.6.1. Precautions about environment of operation

- (1) C Compiler Package operates under Windows 98, Windows NT 4.0 or later. It does not work under Windows 95 and Windows NT 3.5x or earlier.
- (2) If in Windows NT environment the command prompt size is set to other than "80 x 25," the command prompt size will change frequently as you start the compiler. Make sure the command prompt size is set to "80 x 25."

### 1.6.2. Suggestions Concerning File Names

The file names that can be specified are subject to the following restrictions:

- Directory and file names that contain kanji cannot be used.
- Only one period (.) can be used in a file name.
- Network path names cannot be used. Assign the path to a drive name.
- Keyboard shortcuts cannot be used.
- Directory and file names that contain a space character cannot be used.
- The "." symbol cannot be used as a means of specifying two or more directories.
- A file name in length of 128 characters or more including path specification cannot be used.

### 1.6.3. Precautions about virus check programs

If the virus check program is memory-resident in your computer, C Compiler Package may not start up normally. In such a case, remove the virus check program from memory before you start C Compiler Package.

### 1.6.4. Precautions when upgrading

To upgrade C Compiler Package, uninstall the currently installed C Compiler Package first before you install the new version.

- Procedure for uninstalling C Compiler Package  
To uninstall C Compiler Package, launch Add/Remove Programs in Control Panel and then execute Uninstall.

## 2. Installing C Compiler Package

### 2.1. Before installing C Compiler Package

Please confirm as follows before installing C Compiler Package in your computer.

- Please carefully read the "License Agreement" and "Release Note" included with your product before using C Compiler Package. If you've installed this product in your computer, it is assumed that you've agreed to the provisions stipulated in the License Agreement.
- In order that C Compiler Package operates comfortably, it requires at least 32Mbytes of memory and a hard disk having 20Mbytes or more of space.
- Use the dedicated installer to install C Compiler Package.
- You need to input a license ID in the middle of installation. Before you start installing C Compiler Package, check your license ID.

### 2.2. Precaution about installing this product

- When installing this product in a computer, the integrated development environment High-performance Embedded Workshop is also installed together with a compiler.
- When the following dialog is outputted during installation, please choose "*it is all yes*".

[Dialog message]  
 The following file is already on your computer.  
 C:¥WINDOWS¥TEMP¥....  
 Do you wish to overwrite this file?  
 :

### 2.3. C Compiler Package Installer

The installer is provided for each of the environments (supported host, supported OS and language) listed below. Check the product you've purchased to find the appropriate installer.

- Japanese environment

Supported host	Supported OS	Installer name	Directory on CD-ROM
IBM <sup>1</sup> PC/AT compatible	Microsoft Windows <sup>2</sup> 98 Microsoft Windows Me Microsoft Windows NT Microsoft Windows 2000 Microsoft Windows XP	SETUP.EXE	¥NC308WA¥W95J

- English environment

Supported host	Supported OS	Installer name	Directory on CD-ROM
IBM PC/AT compatible	Microsoft Windows 98 Microsoft Windows Me Microsoft Windows NT Microsoft Windows 2000 Microsoft Windows XP	SETUP.EXE	¥NC308WA¥W95E

### 2.4. Installation procedure

Please install C Compiler Package in the following procedure.

- (1) Go to the directory corresponding to your system, which can be found the name of the software you purchased, on the CD-ROM.
- (2) Start up the installer and follow the messages displayed on the screen as you install C Compiler Package.

<sup>1</sup> IBM and AT are registered trademarks of International Business Machines Corporation.

<sup>2</sup> Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the U.S. and other countries.

## 2.5. Setting environment after installation

After you finished installing C Compiler Package, set environment variables next.

The environment variables marked by "Auto" in the tables below do not need to be set because the Windows installer automatically rewrites AUTOEXEC.BAT.

Environment variable	Example of setting
BIN308	Auto (SET BIN308=C:\RENESAS\NC308WA\V520R02\BIN)
INC308	Auto (SET INC308=C:\RENESAS\NC308WA\V520R02\INC308)
LIB308	Auto (SET LIB308=C:\RENESAS\NC308WA\V520R02\LIB308)
TMP308	Auto (SET TMP308=C:\RENESAS\NC308WA\V520R02\TMP)
NCKIN	SET NCKIN=SJIS
NCKOUT	SET NCKOUT=SJIS
Command path	Auto (C:\RENESAS\NC308WA\V520R02\BIN is added)

## 3. Entering user registration

To be eligible for upgrade information, technical support, and other services, you must be registered as a user with Renesas Technology Corporation. Unless you are a registered user, the said services cannot be received.

Please register your name with Renesas Technology Corporation [within 30 days after purchase](#).

### 3.1. User registration

When you've installed C Compiler Package, the following file is created.

```
\renesas\nc308wa\v520r02\support\nc308wa\regist.txt
```

When you've installed the PC version of C Compiler Package, the following file is created.

Cut all contents of the regist.txt file and paste them into a file, then send it to the electronic mail address given below.

```
regist_tool@renesas.com
```

## 4. Contents of upgrade

### 4.1. Function addition and revision of Compiler

#### 4.1.1. Change to error processing

Warning of "mismatch prototype parameter type." was changed to the error.

### 4.2. Problem correction of Compiler

Improvements have been made to all of the following precaution that had been informed to you by tool news:

- On a division or remainder operation including a variable or constant of type long long
- On debug information
- On calling the same function in both cases where the controlling expression of an if statement is TRUE and FALSE
- On Typing three or more successive question marks
- On using options for the SQMLint (a MISRA C rule-checker)
- On reading members of bit fields in an array
- On using the near or far qualifier
- On a multiplication between variables of type long long
- On an inline function that returns a value of type \_Bool
- On using the `-Ofoward_function_to_inline (-OFFTI)` compile option
- On using the tool for enhancing the Scan All Dependencies function

- On calling a function declared to be typedef and returning a structure

### 4.3. Functional revision of Assembler

#### 4.3.1. Revision of the function to set ID code and ROM protect code

- ID code set by the directive command ".ID" is output to the absolute module file.
- The priority of the directive commands ( .ID, .PROTECT ) and lmc308's command options ( -ID, -protect1, -protect2, -protectx ) was changed. Priority is given to lmc308's command options.

### 4.4. Problem correction of Assembler

Improvements have been made to all of the following precaution that had been informed to you by tool news:

- On selection link option "-JOPT"
- On using assembler directive commands ".ID" and ".PROTECT"

## 5. Software version list of C Compiler Package V.5.20 Release 02

The following lists the software items and their versions include with C Compiler Package.

- nc308 V.1.05.06.000
- cpp308 V.4.05.04
- ccom308 V.5.02.11.000
- aopt308 V.1.02.00
- as308 V.4.01.00.000
- mac308 V.2.21.00.000
- pre30 V.1.10.12
- asp308 V.4.00.00
- ln308 V.4.02.00.000
- lb308 V.2.01.00
- lmc308 V.3.00.00
- xrf308 V.2.01.00
- abs308 V.1.02.00
- stk V.1.00.04
- utl308 V.1.00.10
- Stk Viewer V.1.00.01
- MapViewer V.3.00.00

## 6. Versions Useful for the Real-time Operating System for M32C/90, M32C/80, M16C/80 Series

The C Compiler Package compiler presented here supports the Real-time Operating System V.1.10 Release 1. When you are using Real-time Operating System in combination with the C Compiler Package, please be sure to use the above compiler version. User's manual supplementation

## 7. Conformance with MISRA C Rule in Standard Function Library

In C-Source code of standard function library C Compiler Package, it is found that 52 rules<sup>3</sup> are against the MISRA C Rule NOTE, but these violations do not constitute a drawback to any operation.

### 7.1. Cause of Rule Violation

In C-Source code of standard function library C Compiler Package, the major causes for rule violation are as follows:

- C-Compiler specifications (near/far modifier, asm () function and #pragma)
- Declaration of function based on ANSI Standard
- The evaluation sequence in the conditional statement is not described explicitly, using a parenthesis.
- Implicit type conversion

### 7.2. Inspection No. running counter to the rule

The following are Inspection Nos. that run counter to the Rule:

1	12	13	14	18	21	22	28	34	35
36	37	38	39	43	44	45	46	48	49
50	54	55	56	57	58	59	60	61	62
65	69	70	71	72	76	77	82	83	85
99	101	103	104	105	110	111	115	118	119
121	124								

### 7.3. Evaluation Environment

Compiler	C Compiler Package V.5.20 Release 1 for M32C/90, M32C/80, M16C/80 Series
Compile Option	-O -c -as308 "-DOPTI=0" -gnone -finfo -fNII -misra_all -r \$*.csv
MISRA C Checker	MISRA C Rule Checker V.1.00 Release 1A

<sup>3</sup> These results were produced after inspection using MISRA C Rule Checker for M32C/90, M32C/80, M16C/80 Series.

## 8. A Guide to Porting Projects Created with TM to High-performance Embedded Workshop V.4

This document explains how to port projects created with TM V.2.xx or V.3.xx into High-performance Embedded Workshop V.4.

### 8.1. Summary

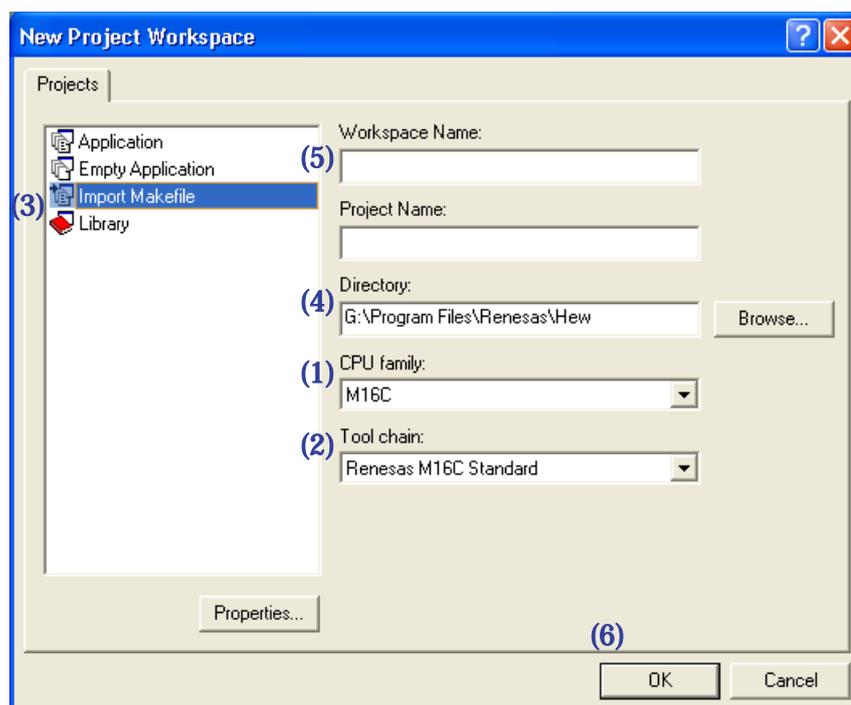
To port projects created using TM V.2.xx or V.3.xx into High-performance Embedded Workshop V.4, the Import Makefile function of High-performance Embedded Workshop is used. This function can create projects from such items of information as source files and build options described in the specified makefile files.

In TM, project files are created in the makefile format executable in GNU make format. When project files created with TM are selected as makefile files using High-performance Embedded Workshop Import Makefile function, they are converted to files that can run in High-performance Embedded Workshop. In addition to TM project files, the Import Makefile function can also convert files in the makefile formats for hmake, nmake, and gmake to High-performance Embedded Workshop projects.

### 8.2. Porting Procedure

To port projects created using TM into High-performance Embedded Workshop, perform the following steps:

- (1) Open the File menu and select the New Workspace command.
- (2) The New Project Workspace dialog box opens.

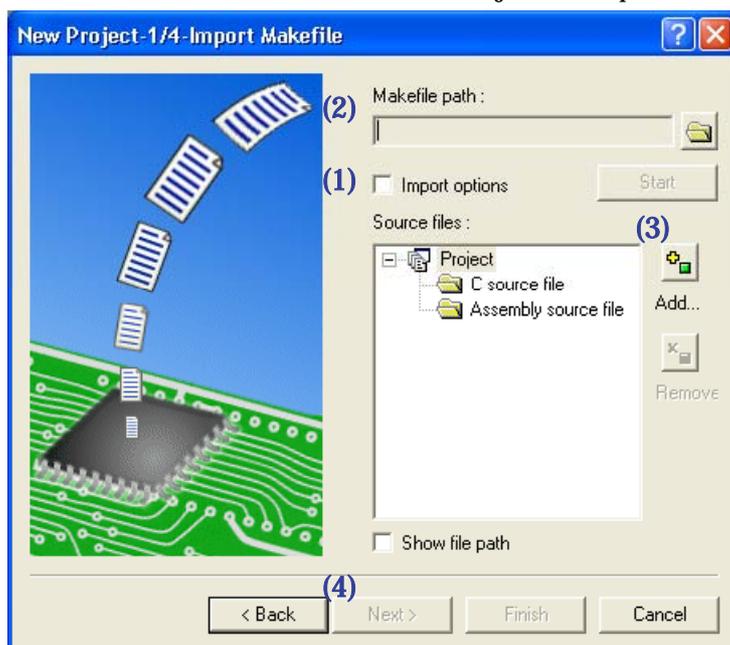


- (1) Select the type of CPU used in the TM project from the Type of CPU drop-down list.
- (2) Select the tool chain (cross tool) used for the TM project from the Toolchain drop-down list. The names of tool chains and corresponding cross tools are shown in Table 1.

Tool Chain	Cross Tool
Renesas M16C Standard	NC30WA
Renesas R8C Standard	NC8C
Renesas M32C Standard	NC308WA
Renesas M32R Standard	CC32R

- (3) Select Import Makefile from the Project list.
- (4) Type the directory path in the Directory text box.
- (5) Type the workspace name in the Workspace Name text box. The same name will be automatically entered as the project name in the Project Name text box.

- (6) Click OK.
- (3) You should now be able to see the New Project-1/4-Import Makefile wizard.



- (1) Select the Import options check box; this will enable information on build options (compiling and assembling options etc.) to be used to create High-performance Embedded Workshop projects. If you clear the Import options check box, the above information is neglected and not used in High-performance Embedded Workshop.
- (2) Type the name of the TM project file (with extension .tmk) in the Makefile path text box. As soon the name is input, the specified file is analyzed, and upon analysis completion, the analyzed source files are displayed in a tree structure in the Source files box. Click the Start button to analyze the specified file again.
- (3) If there are any errors in the analysis results (tree structure in the Source files box), rectify the tree structure with the Add and Remove buttons.
- (4) Click Next.
- (4) Follow the instructions according to the Wizard as it continues in the procedure.

### 8.3. Usage Notices

#### 8.3.1. TM-to-High-performance Embedded Workshop Portable and Non-Portable Information

When you port a project created using TM into High-performance Embedded Workshop, not all the components of the project can be ported.

Portable information is as follows:

- (1) Paths of assembler source files
- (2) Paths of C-language source files
- (3) Assembling options
- (4) C-compiling options
- (5) Linking options (except linkage order)

Non-Portable Information:

- (1) Linkage order
- (2) Tool configurations, dependencies, and options other than Assembler, C Compiler, Linker

To transfer these items, edit the High-performance Embedded Workshop project as described in Section 3.4 and further after processing the Import Makefile.

#### 8.3.2. Cross Tools

Import Makefile cannot enable all cross tool versions for use in High-performance Embedded Workshop projects regardless of whether they are used with TM or not; only the following cross tools versions are valid for High-

performance Embedded Workshop projects:

NC30WA	V.5.20 Release1, V.5.30 Release1, V.5.30 Release 02
NC8C	V.5.30 Release1
NC308WA	V.5.20 Release1

### 8.3.3. High-performance Embedded Workshop Versions

When TM projects are ported into High-performance Embedded Workshop information portable to High-performance Embedded Workshop varies according to the High-performance Embedded Workshop version. The information that can be ported from each cross tool to various High-performance Embedded Workshop versions are shown in Table 2.

		High-performance Embedded Workshop				
		V.3.01.02	V.3.01.04	V.3.01.05	V.3.01.06	V.4.00
NC30WA	V.5.20 Release 1	B	B	B	B	A
	V.5.30 Release 1	B	B	B	B	A
	V.5.30 Release 02	--	--	--	--	A
NC8C	V.5.30 Release 1	B	B	B	B	B
NC308WA	V.5.20 Release 1	B	B	B	B	B

A: All the items of information listed in Section 3.1 are portable.

B: Only the paths of assembler and C-language source files are portable.

### 8.3.4. Generated Project Workspace

Because the project workspace created for a TM project ported to the High-performance Embedded Workshop environment is simply the contents of the makefile itself, its configuration (object output directory) will be different than that of a newly generated project workspace in High-performance Embedded Workshop.

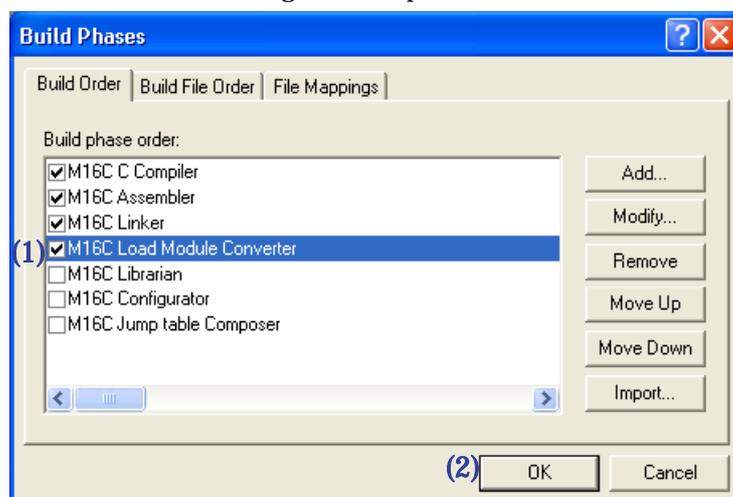
To validate the configuration, modify the output directory file names for the compiler, assembler and linker as follows:

Output Directory (compiler, assembler)	\$(CONFIGDIR)
Output Directory (linker)	\$(CONFIGDIR)\\$(PROJECTNAME).x30

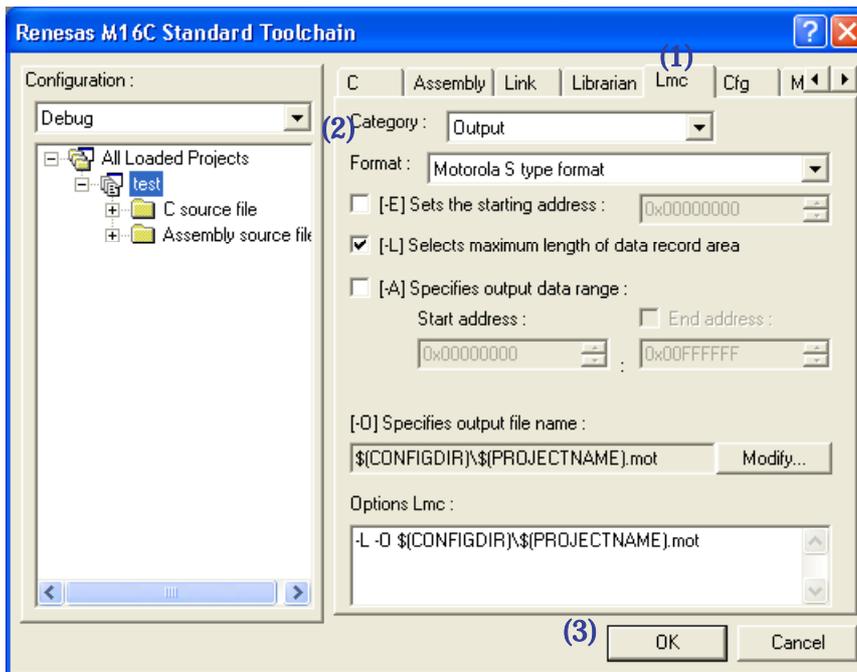
### 8.3.5. Load Module Converter

Import Makefile cannot port the information contained in any load module converter (for example, information on options, command executions, or dependencies) into the High-performance Embedded Workshop project. If using a load module converter to create projects in TM, change the settings of the load module converter as follows after completing the Makefile processing:

- (1) Open the Build menu and select the Build Phases command.
- (2) The Build Phases dialog box will open.



- (1) Select the Mxxx Load Module Converter check box from the Order of Build Phases list.
- (2) Click OK.
- (3) Open the Build menu and select Renesas Mxxx Standard Toolchain.
- (4) The Renesas Mxxx Standard Toolchain dialog box appears.



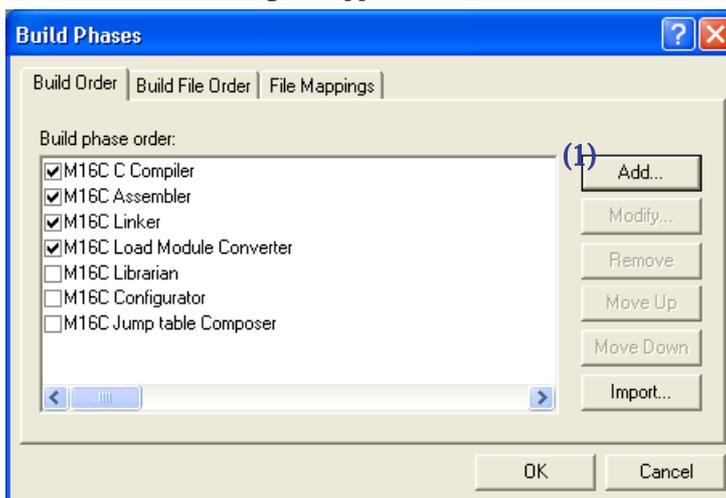
- (1) Click the Lmc tab.
- (2) Select the Category type from the Category drop-down list.
- (3) Click OK.

### 8.3.6. Other Tools

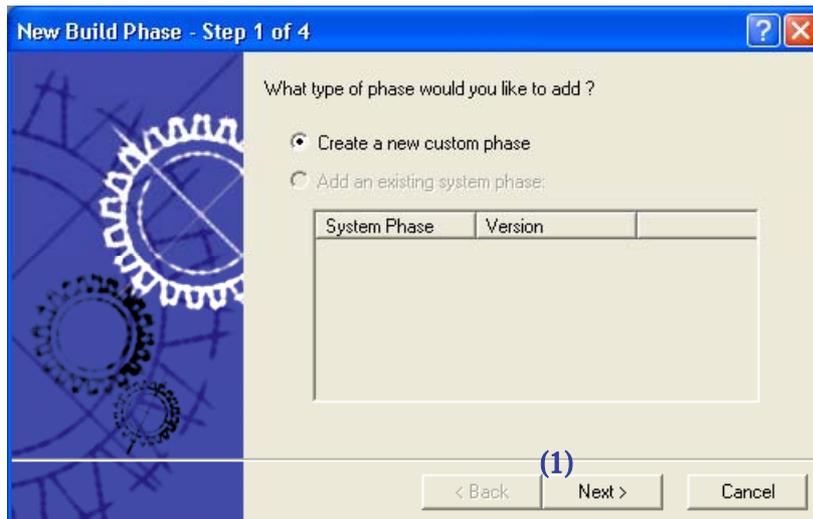
Import Makefile cannot port any information (options, command executions, dependencies) contained in tools other than the assembler, C compiler, and linker. If any tools other than the assembler, C compiler, linker, and load module converter are used to create projects in TM, custom build phases must be created in High-performance Embedded Workshop. Custom build phases are specifically for operating other tools before, after, or during standard builds (in the assembler, C compiler, and linker).

For more details, see Section 3.2 “Creating Custom Build Phases” in the High-performance Embedded Workshop 3 User’s Manual. The following is provided as an example of how to register the cross-reference generation tool xrf30 with High-performance Embedded Workshop.

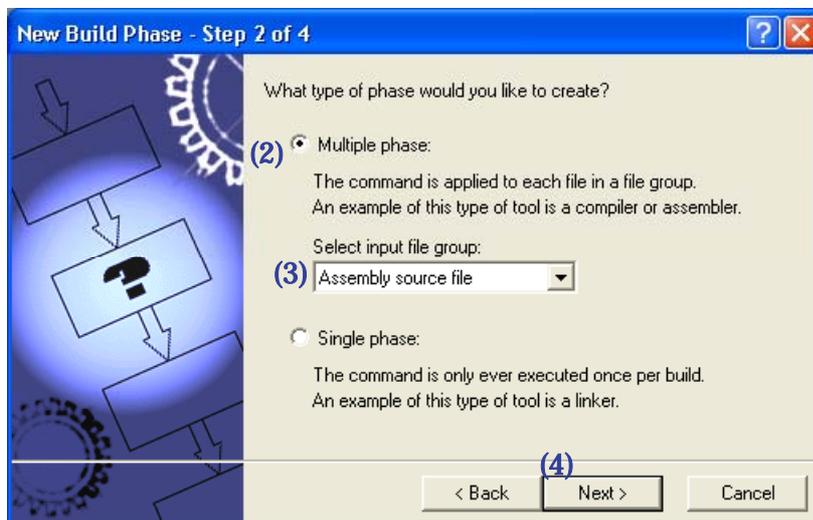
- (1) Open the Build menu and select the Build Phases command.
- (2) The Build Phases dialog box appears; click Add.



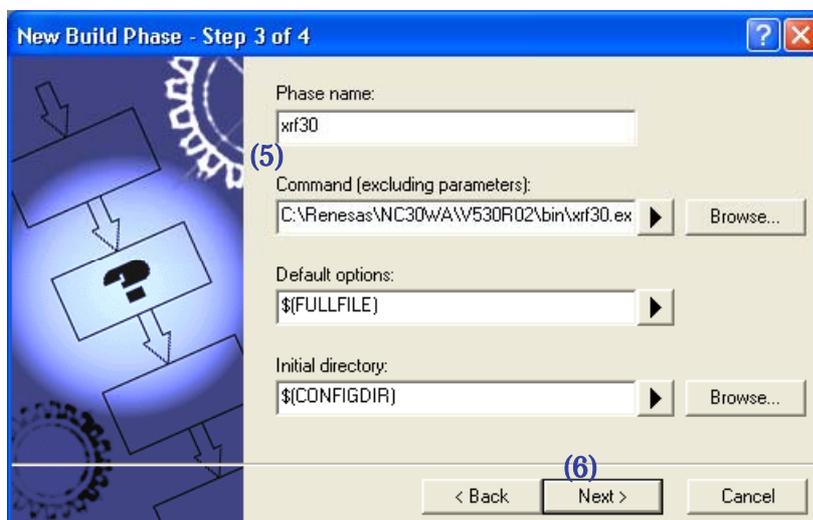
- (3) The New Build Phase- Step 1/4 wizard opens. Follow the instructions to register the tool as follows:



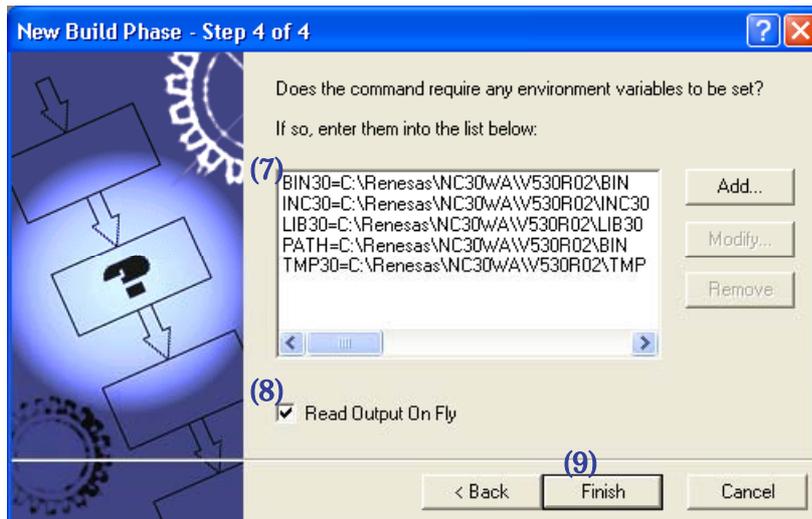
- (1) Click Next (the Create a New Custom Phase check box is selected by default); the New Build Phase-2/4 Step wizard opens.



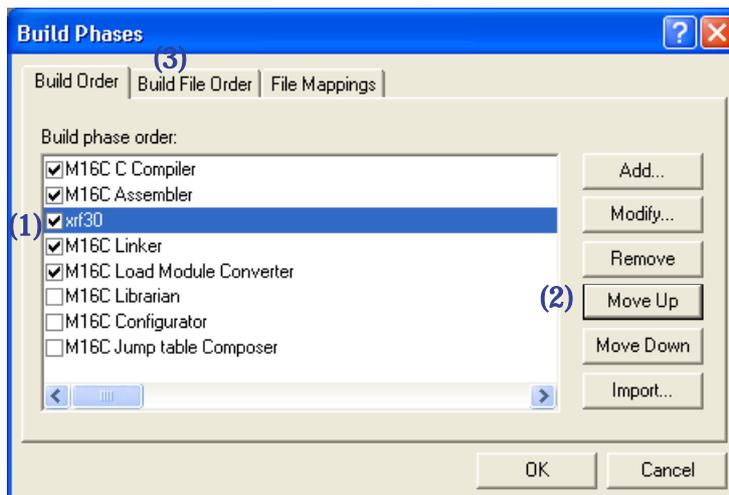
- (2) In this wizard, select the Multiple Phase check box.  
 (3) Select Assembly Source file from the Select input file group.  
 (4) Click Next; the New Build Phase- Step 3/4 wizard opens.



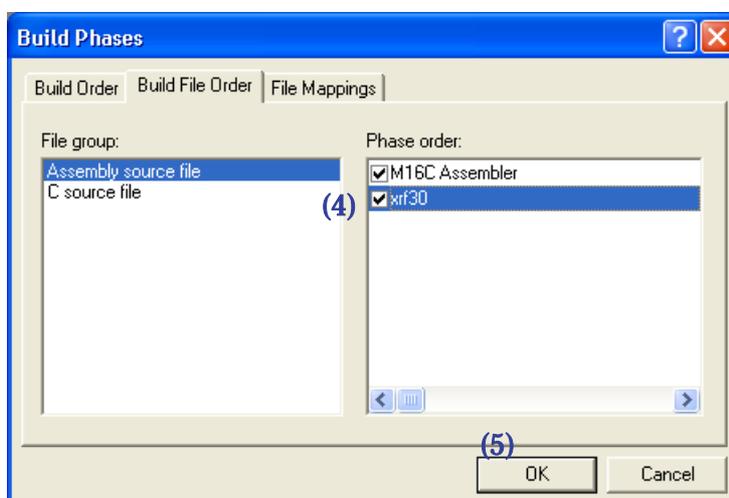
- (5) Type xrf30 and its fullpath name in the Phase Name and the Command text box.  
 (6) Click Next; the New Build Phase- Step 4/4 wizard opens.



- (7) In this wizard, enter the necessary environment variables in the list.
  - (8) Select “Read Output On Fly” check box.
  - (9) Click Finish.
- (4) You return to the Build Phases dialog box at this point, where you can see that xrf30 has been registered as a build phase at the end of the Order of Build phase order.



- (1) Select xrf30 from the Order of Build phase order.
- (2) Click Move Up to move xrf30 next to the assembler name.
- (3) Click the Build File Order tab.



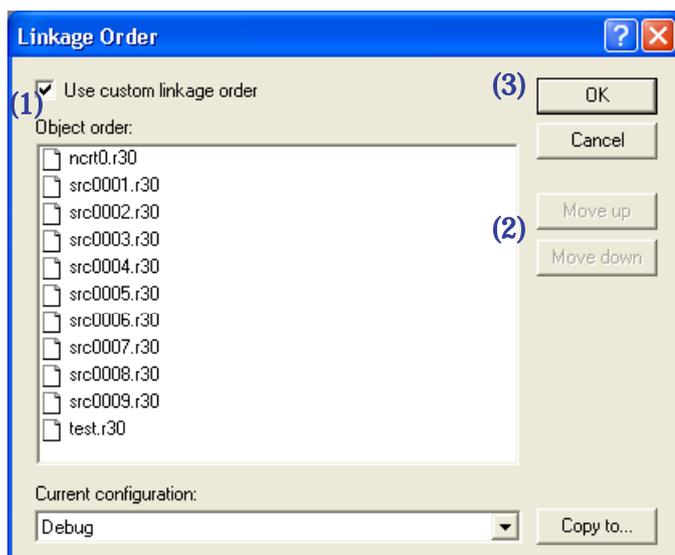
- (4) Select the xrf30 check box in the Order of Phase order.
- (5) Click OK.

- (5) Open the Options menu and select the xrf30 command.
- (6) The xrf30 Options dialog box appears; select options as necessary. This setting executes xrf30 for all assembler source files after assemble is completed at a build (before linking files).

### 8.3.7. Linkage order

Import Makefile cannot port the linking order information to High-performance Embedded Workshop . High-performance Embedded Workshop arranges the linking order alphabetically. To change this order, go through the following steps:

- (1) Open the Build menu and select the Linkage Order command.
- (2) The Linkage Order dialog box opens.



- (1) Select “Use custom linkage order” check box.
- (2) Select a file from the Object order list, and click Move up or Move down to move the file. Repeat this step for all files that need to be rearranged.
- (3) Click OK.

### 8.3.8. Placing the Start Up program at the top of Linkage Order

As the Import Makefile cannot port linking order information to High-performance Embedded Workshop , and links are order alphabetically, the start up program may not be placed at the top of the linking order. To place it at the top, follow the steps described previously in “Linkage Order.

## 9. License Agreement

### IMPORTANT

Licensed Copy Number : 1 Copies

DO NOT OPEN THIS PACKAGE NOR INSTALL THE SOFTWARE UNTIL YOU HAVE READ AND AGREED TO THE TERMS AND CONDITIONS SPECIFIED IN THE SOFTWARE USER LICENSE AGREEMENT BELOW

Permission to use this software is subject to your agreeing to the terms and conditions specified in the agreement entitled "Software User License Agreement" below. Please read carefully the Software User License Agreement before you open the package. If you do not agree to the terms and conditions specified in this agreement, do not open the package and return the product to your distributor, with the program disk packaging seal unbroken for a full refund.

### Software User License Agreement

The customer (hereafter referred to as "Licensee") and Renesas Technology Corp. (hereafter referred to as "Licensor") do hereby agree to the terms and conditions as specified in this Software User License Agreement (hereafter referred to as "Agreement") concerning the enclosed this software and its explanatory manuals.

#### ARTICLE 1. Definition

- (1) "Licensed Software" shall mean the main unit of the C compiler, Assembler, Simulator and related executable programs, the "Library" as defined in Section 1.3 below and documentation such as explanatory manuals for the Licensed Software and all other materials provided to Licensee in connection with use of the Licensed Software.
- (2) "Designated System" shall mean the computer system to be installed Licensed Software. In case of Network environment system, "Designated System" shall mean the computer system to be executed Licensed Software.
- (3) "Licensed Copy Number" shall mean the number which is printed at right upper part in this sheet. And it shall mean the number of "Designated System" which you can install Licensed Software and can make the copy of the Master Disk of Licensed Software.
- (4) "Library" shall mean a group of function programs that enable operations including, but not limited to input/output and character recognition and that are offered in the forms of a re-locatable program.
- (5) "Developed Software" shall mean software created by Licensee incorporating all or part of a Library in accordance with the licenses granted to Licensee under the terms and conditions of this Agreement.

#### ARTICLE 2. License Grant

Licensor grants Licensee a non-exclusive, non-assignable, non-transferable license:

- (1) to use the Licensed Software solely on the Designated System for Licensee's business purpose of creating and copying Developed Software to be incorporated in the Licensee's products installed with Licensor's Renesas microcomputer (hereafter referred to as "Licensee's Products");
- (2) to make copies of Licensed Copy Number by installer program of Licensed Software on Designated System for the purposes of using the Licensed Software in accordance with subparagraph (1) above of this Section 2 to develop Developed Software in accordance with subparagraphs (3) below of this Section 2;

- (3) to merge the Library with one or more other programs to develop Developed Software, provided any portion of the Library so merged continues to be subject to the terms and conditions of this Agreement; and
- (4) to copy the Developed Software in connection with the manufacture and sale of the Licensee's Products incorporating Developed Software, subject to provisions of Article 6 hereunder.

### ARTICLE 3. Restrictions

- (1) Licensee shall not transfer the rights granted herein or allow any third party to use the Licensed Software.
- (2) Licensee shall not remove the copyright notice contained in the Licensed Software.
- (3) Licensee shall not sublicense, rent, lease, or assign the Licensed Software.
- (4) Licensee shall not reverse assemble, reverse compile or otherwise reverse engineer the Licensed Software.

### ARTICLE 4. Title to Licensed Software

- (1) All right, title and interest in and to the Licensed Software are shall remain with Licensor. The copyrights in the Licensed Software owned by Licensor are protected by Japanese copyright laws, other applicable intellectual property laws and international treaty provisions.
- (2) Licensor retains all rights not expressly granted.
- (3) The copyrights in the Developed Software as a derivative work of Library shall be owned by Licensee, provided, however, that copyrights in any part of Library contained in such Developed Software shall remain the property of Licensor.

### ARTICLE 5. Confidentiality

Licensee acknowledges that the Licensed Software contains valuable proprietary information and trade secrets of Licensor. Licensee shall preserve the Licensed Software in confidence, and shall not provide or disclose the contents of all or any part of the Licensed Software to any third party without written authorization from Licensor.

Licensee will take appropriate action, by instruction, agreement or otherwise, with any persons permitted access to the Licensed Software so as to enable them to hold the Licensed Software in confidence and otherwise to satisfy their obligations under this Agreement.

This Agreement shall not restrict disclosure or use of any information which:

- can be demonstrated by Licensee to be in the public domain when received, or thereafter enters the public domain through no fault of Licensee;
- is known to Licensee at the time of the disclosure by Licensor;
- is received from a third party without any restrictions on disclosure and use; or
- is independently developed by Licensee through persons who have not had access to the Licensed Software.

### ARTICLE 6. Term and Termination

- (1) This Agreement shall become effective upon Licensee's breaking the packaging seal of the Licensed Software and shall remain in force until terminated by any of the provisions in (2) and (3) below.
- (2) Licensee may terminate this Agreement at any time on thirty (30) day prior written notice to Licensor.
- (3) In the event that Licensee shall be in default under this Agreement, Licensor may terminate this Agreement and all licenses granted hereunder by giving written notice of termination to Licensee, effective upon its sending.

### ARTICLE 7. Post-Termination Procedures

- (1) In the event of any termination of this Agreement, Licensee shall cease all use of the Licensed Software and shall remove any and all copies of the Licensed Software from the Designated System, shall remove any Libraries contained in the Developed Software from such Developed Software (except for Libraries

- incorporated in the Licensee's Products already sold), and shall destroy all such copies of Licensed Software.
- (2) Licensee shall complete all the obligations described in Section 7.1 above and shall submit written proof to that effect to Licensor within thirty (30) days from the termination of this Agreement.

## ARTICLE 8. Proprietary Rights Indemnification

- (1) Licensor shall defend or, at its option, settle any claim or proceedings brought against Licensee to the extent that it is based on an assertion that the Licensed Software, used within the scope of this Agreement, constitute direct infringement of any copyright, patent or trademark of a third party and shall indemnify Licensee against all costs, damages and expenses finally awarded against Licensee which result from any such claim, provided that Licensee notifies Licensor promptly in writing of such claim or proceedings and gives Licensor full and complete authority, information and assistance to defend such claim or proceedings, and further provided that Licensee gives Licensor sole control of the defense of any such claim or proceedings and all negotiations for its compromise or settlement.
- (2) Should the Licensed Software become, or, in Licensor's opinion, be likely to become the subject of a claim of infringement, Licensor may procure for the Licensee the right to continue using the Licensed Software or replace or modify it to make it non infringing. If neither of the foregoing alternatives is reasonably available to Licensor, then Licensor's sole and exclusive remedy shall be to terminate this Agreement, and to return to Licensee refund of License fees and other expenses paid to Licensor.
- (3) Licensor's obligations under this Article 8 shall not apply to any claims of infringement based upon (i) use of other than the latest upgrades of the Licensed Software made available to Licensee by Licensor according to Section 9.2 hereunder if such infringement would have been avoided by the use of such upgrades of the Licensed Software, (ii) combination, operation or use of the Licensed Software with any programs, data or products not furnished by Licensor if such infringement would not have occurred without such combination, operation or use, (iii) use of the Licensed Software after receiving written notice from Licensor that the Licensed Software infringes any patent, copyright or trademark of a third party, or (iv) claimant expresses its intent to license or otherwise resolve its claim only with the Licensee of any apparatus incorporating the Licensee's Products. LICENSOR'S OBLIGATION UNDER THIS ARTICLE 8 CONSTITUTE LICENSEE'S SOLE AND EXCLUSIVE REMEDY FOR A CLAIM OR PROCEEDINGS FOR AND INTELLECTUAL PROPERTY INFRINGEMENT OF A THIRD PARTY.

## ARTICLE 9. Limited Warranty, Indemnification and Limitation of Liability

- (1) Licensor warrants that it has the right to grant to Licensee a license to use the Licensed Software and that Licensor has the right to enter into this Agreement. EXCEPT AS SPECIFICALLY PROVIDED HEREIN, LICENSOR DISCLAIMS ALL WARRANTIES, REPRESENTATION, PROMISE OR GUARANTEE, EITHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND INCLUDING WITHOUT LIMITATION, DAMAGES RESULTING FROM LOSS OF USE, LOSS OF DATA, LOSS OF PROFITS OR LOSS OF BUSINESS. THE LICENSED SOFTWARE IS PROVIDED "AS IS", WITH ALL FAULTS. LICENSEE AGREES THAT LICENSOR'S LIABILITY HEREUNDER FOR DAMAGES, REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE CHARGES PAID BY LICENSEE FOR THE LICENSED SOFTWARE. Notwithstanding the foregoing, in the event Licensee find what it believes to be an error or failure of the Licensed Software, Licensor shall make reasonable efforts to correct any such error or failure by taking measures which Licensor deems appropriate provided that Licensee has returned the Customer Registration Card to Licensor. However, Licensor does not guarantee that any measures described above will correct all of the errors or failures.
- (2) The warranty coverage under this Agreement does not apply to any error or failure of the Developed Software or any error or failure caused by a modification of the Licensed Software by Licensee or by any negligent act or omission of Licensee.

- (3) In the event Licensor provides Licensee with any upgrades or error corrections of the Licensed Software, Licensee may upgrade the Licensed Software under the terms and conditions determined by Licensor. Licensor may request Licensee to submit the Master Disk for the Licensed Software at the time an upgrade is conducted.

#### ARTICLE 10. Export Control

- (1) Licensee hereto represents and warrants that it shall not use any products, software and/or technology provided by Licensor, or any other products, software and/or technology manufactured or developed by using them (collectively hereinafter called, "Products"), for the purposes of disturbing international peace and security, including (i) the design, development, production, stockpiling or use of weapons of mass destruction such as nuclear, chemical or biological weapons or missiles, (ii) the other military activities, or (iii) any use supporting these activities.
- (2) Licensee also represents and warrants that it shall not sell, export, dispose of, license, rent, transfer, disclose or otherwise provide the Products to any third party, whether directly or indirectly, with knowledge or reason to know that the third party or any other party will engage in the activities described above.
- (3) Furthermore, Licensee represents and warrants that it shall not directly or indirectly, export, re-export, transship or otherwise transfer the Products in violation of any applicable export control laws or regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or their transactions.

#### ARTICLE 11. General

- (1) This Agreement constitutes the entire agreement between the parties concerning Licensee's use of the Licensed Software and no waiver, alteration, modification or cancellation shall be binding unless made in writing and signed by the parties.
- (2) If any provision of this Agreement shall be held by a court of competent jurisdiction to be illegal, invalid or unenforceable, the remaining provisions shall remain in full force and effect.
- (3) This Agreement is made under and shall be construed in accordance with the Laws of Japan. Licensee agrees that any disputes related to this Agreement shall be settled by the jurisdiction of the court determined by Licensor. LICENSEE ACKNOWLEDGES THAT LICENSEE HAS READ THIS AGREEMENT AND AGREES TO ALL TERMS AND CONDITIONS STATED HEREIN.

Renesas Technology Corp.