

To all our customers

---

## **Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.**

---

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.  
Customer Support Dept.  
April 1, 2003

# STK Viewer V.1.00

User's Manual

User's Manual

Rev.1.00  
Mar. 01, 2005

Renesas Technology  
[www.renesas.com](http://www.renesas.com)

- Microsoft, MS-DOS, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. HP-UX is a registered trademark of Hewlett-Packard Company.
- Sun, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries, and are used under license
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds.
- Turbolinux and its logo are trademarks of Turbolinux, Inc.
- IBM and AT are registered trademarks of International Business Machines Corporation.
- HP9000 is a product name of Hewlett-Packard Company.
- SPARC and SPARCstation are registered trademarks of SPARC International, Inc.
- Intel and Pentium are registered trademarks of Intel Corporation.
- Adobe and Acrobat are registered trademarks of Adobe Systems Incorporated.
- Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation in the U.S. and other countries.

All other brand and product names are trademarks, registered trademarks or service marks of their respective holders.

#### **Keep safety first in your circuit designs!**

- Renesas Technology Corporation and Renesas Solutions Corporation put the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

#### **Notes regarding these materials**

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation, Renesas Solutions Corporation or a third party.
- Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation and Renesas Solutions Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation, Renesas Solutions Corporation or an authorized Renesas Technology product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corporation and Renesas Solutions Corporation by various means, including the Renesas home page (<http://www.renesas.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation, Renesas Solutions Corporation or an authorized Renesas Technology product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation and Renesas Solutions Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation or Renesas Solutions Corporation for further details on these materials or the products contained therein.

For inquiries about the contents of this document or product, fill in the text file the installer generates in the following directory and email to your local distributor.

\\SUPPORT\Product-name\SUPPORT.TXT

Renesas Tools Homepage <http://www.renesas.com/en/tools>

# Contents

Contents .....	3
Tables .....	4
Figures .....	4
1. Outline of the STK Viewer .....	5
1.1 Outline of Processing Performed by the STK Viewer .....	5
1.2 Main Window .....	8
1.3 The Select Functions Dialog Box .....	11
1.4 The About Dialog Box .....	11
1.5 The Exit Dialog Box .....	12
2. Limitations .....	13
2.1 About the Mouse Cursor .....	13
2.2 About the stk Execution File .....	13
3. Starting the STK Viewer .....	14
4. Using the STK Viewer .....	15
4.1 Calculating the Stack Size .....	15
4.2 Stack Size of a Function with Unknown Stack Size .....	15
4.3 Reusing the Set Information .....	16
4.4 Showing Version Information .....	16
5. Error Messages of the STK Viewer .....	17
5.1 Error Messages .....	17
Appendix A. ....	19
A.1 Stack Calculation Utility .....	19
A.1.1 Input Format .....	19
A.1.2 Option Reference .....	20
A.1.3 Output Format .....	24

## Tables

Table 1.1 Menu Commands .....	9
Table 1.2 Toolbar.....	9
Table 5.1 Error Messages of the STK Viewer.....	17
Table 5.2 Error Messages of the STK Utility .....	18

## Figures

Figure 1.1 Flow of Stack Calculation Processing .....	5
Figure 1.2 Main Window .....	8
Figure 1.3 Select Functions Dialog Box .....	11
Figure 1.4 About Dialog Box.....	11
Figure 1.5 Exit Dialog Box .....	12

# 1. Outline of the STK Viewer

## 1.1 Outline of Processing Performed by the STK Viewer

The STK Viewer finds the stack size needed for program operation and the relationship of function calls by processing the inspector information which is added to the object file when compiled after specifying the compiler driver's startup option `-finfo`. Also, by specifying information, if any, that could not be fully analyzed with only the inspector information via GUI, the STK Viewer recalculates the stack size and the relationship of function calls.

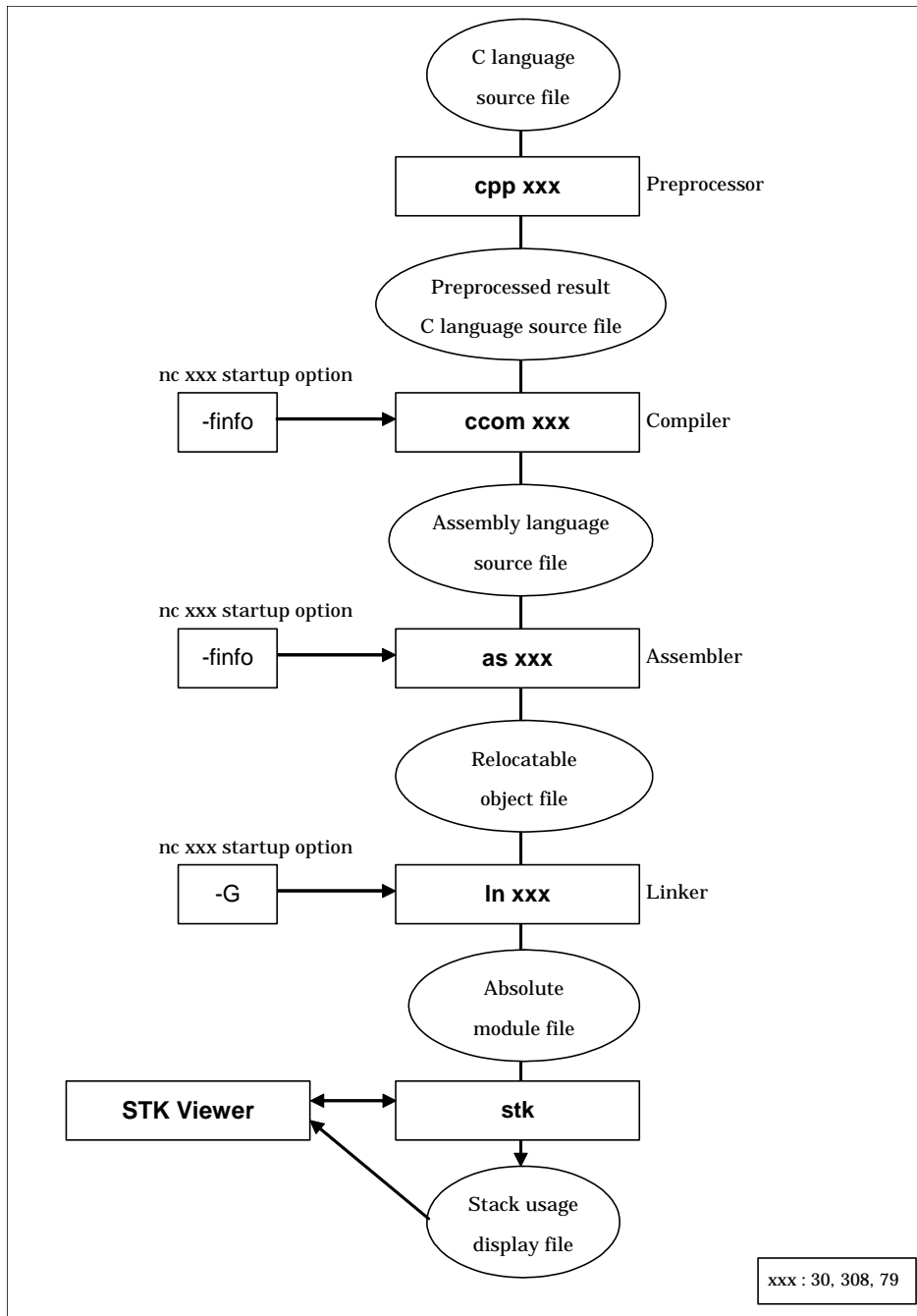


Figure 1.1 Flow of Stack Calculation Processing

The STK Viewer starts a stack size calculation utility (stk). The utility **stk** analyzes the inspector information added to the absolute module file and calculates the stack size needed for each function and the relationship of function calls. This information is output to a stack usage display file (extension “stk”).

The STK Viewer analyzes the stack usage display file and displays the result on its window.

### **About assembler files**

For the assembler files to be analyzed by the STK Viewer, the following information is required.

#### **(1) .insf**

##### **Description**

Indicates the beginning of a function.

##### **Format**

.insf assembler function name, storage class, 0  
assembler function name: Indicates a function name.  
storage class: S (static function) or G (global function)  
0: Fixed to 0

##### **Example statement**

.insf \_func,S,0

#### **(2) .einsf**

##### **Description**

Indicates the end of a function.

##### **Format**

.einsf

#### **(3) .call**

##### **Description**

Indicates a function call.

##### **Format**

.call assembler function name, storage class  
assembler function name  
storage class: S (static function)/G (global function)

##### **Example statement**

.call \_func,S

#### (4) .stk

##### Description

Indicates stack operation.

##### Format

```
.stk push/pop size
      push/pop size: Specify in bytes
```

##### Example statement

```
.stk  8      (For pushing 8 bytes)
.stk -8      (For popping 8 bytes)
```

#### (5) Example statement

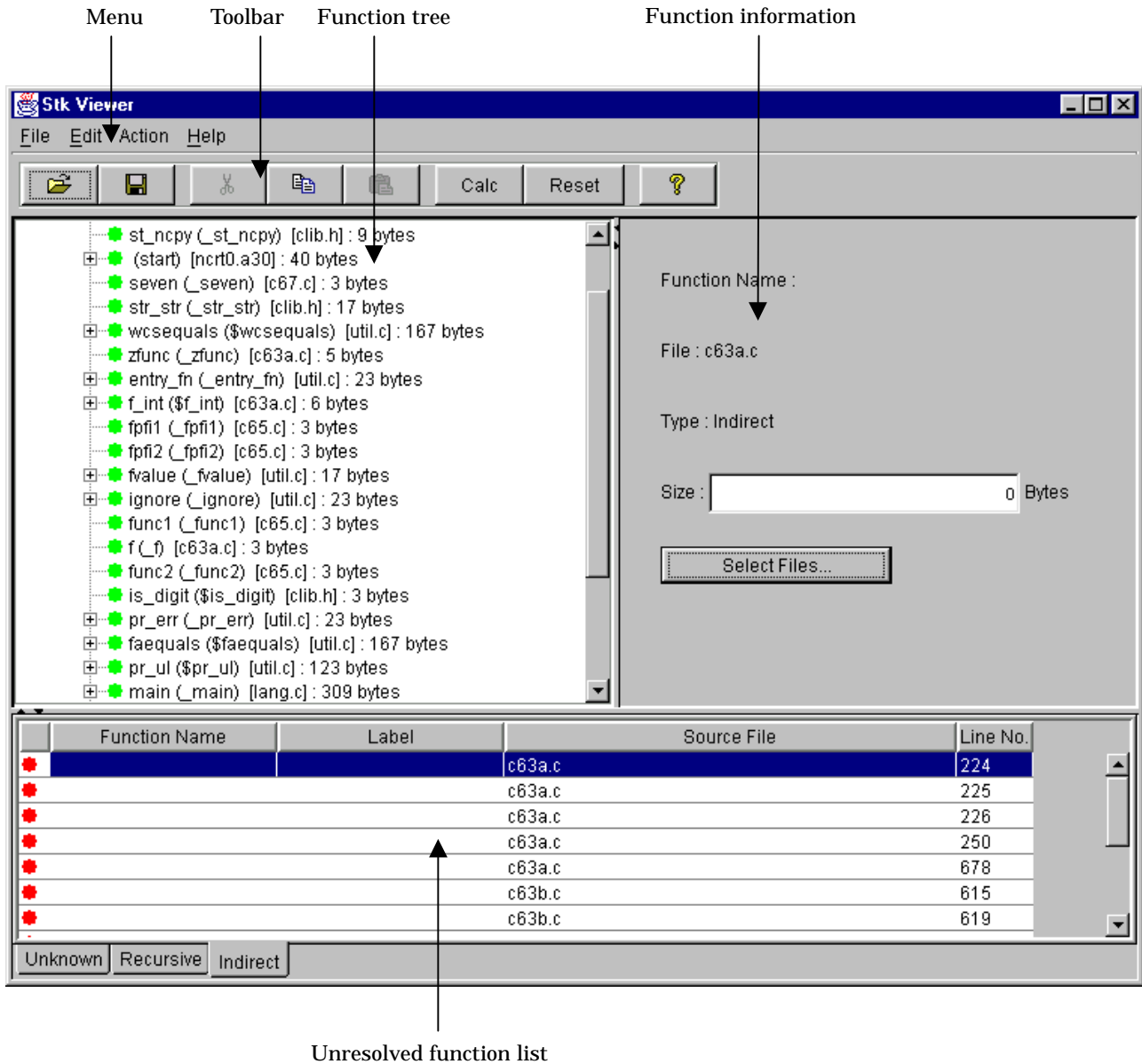
```
      .glb          __xxx
__xxx:
      .insf  __xxx,G,3      <- Specify the position at which the function starts
      pushm  R1,R3,A0,A1
      .stk   8              <- Use 8 bytes of stack for pushing
      mov.w  #0,A0
      mov.w  #0,A1
      .
      .
      .glb    _yyy
      .call  _yyy,G        <- Specify the position at which the function is
called
      jsr.a  _yyy
      .
      .
      popm   R1,R3,A0,A1
      .stk   -8            <- Free 8 bytes of stack by popping
      rts
      .einsf                <- Specify the position at which the function ends
      .end
      .
```

#### Executing stk singly

The utility **stk** can be executed singly from the command line to display the stack usage or the amount of stack used. (Refer to Appendix A.)

## 1.2 Main Window

Displays the stack usage after calculating it. The functions for which the stack size could not be determined by **stk** are displayed for each category in the unresolved function list in the lower part of the window.



**Figure 1.2 Main Window**

(Note: This is an example window when run in Windows)

**Table 1.1 Menu Commands**

Command name	Description
File	
Open...	Loads absolute module file or saved stack size.
Save...	Saves the specification of stack size.
Exit	Closes the STK Viewer.
Edit	
Cut	Copies a selected string to the clipboard and cuts it from the body text.
Copy	Copies a selected string to the clipboard.
Paste	Pastes a string from the clipboard into a position.
Action	
Calc	Recalculates the stack usage from the specified stack usage and the relationship of function calls.
Reset	Reacquires stack information. Clears the information that has been set.
Help	
About...	Shows version information. Opens the About Dialog Box.

**Table 1.2 Toolbar**

Command name	Description
Open	Loads absolute module file or saved stack size.
Save	Saves the specification of stack size.
Cut	Copies a selected string to the clipboard and cuts it from the body text.
Copy	Copies a selected string to the clipboard.
Paste	Pastes a string from the clipboard into a position.
Calc	Recalculates the stack usage from the specified stack usage and the relationship of function calls.
Reset	Reacquires stack information. Clears the information that has been set.
About	Shows version information. Opens the About Dialog Box.

### **(1) Function tree**

Shows the relationship of function calls.

The functions for which the stack size could not be calculated are marked with red circles. The red circles change to green when the problem is resolved by a use-specified stack size.

[Display format]

function name (assembler symbol name)[definition file name]: stack usage bytes

### **(2) Function information**

Shows information on a function selected from the function tree or unresolved function list.

For unknown functions or recursively called functions, the stack size to be used by such function can be specified. For indirectly called functions, click the [Select Files...] button to open the Select Functions Dialog Box, from which you can specify the called function.

### **(3) Unresolved function list (unknown functions and recursive or indirect calls)**

The functions for which the stack usage could not be calculated are listed here.

The functions for which the stack usage has been set have their markings change from red to green circles.

### 1.3 The Select Functions Dialog Box

The utility **stk** cannot find the stack size when the program contains indirect calls. In such a case, use this dialog box to specify any function that may probably be called and calculate the stack size newly again.

The functions in this dialog box are displayed in order of the defined file name, function name, and assembler symbol name.

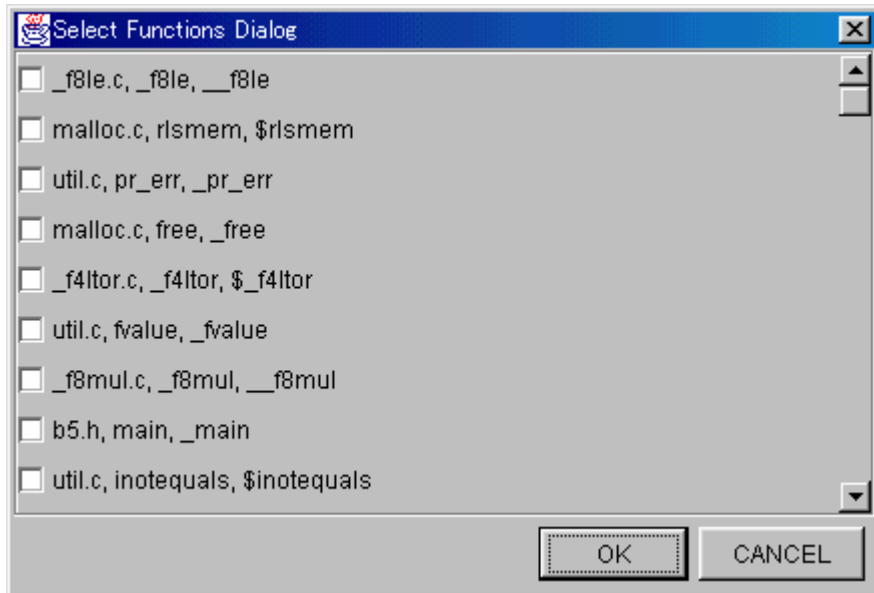


Figure 1.3 Select Functions Dialog Box

### 1.4 The About Dialog Box

This dialog box is used to show the version numbers of the STK Viewer and **stk**.

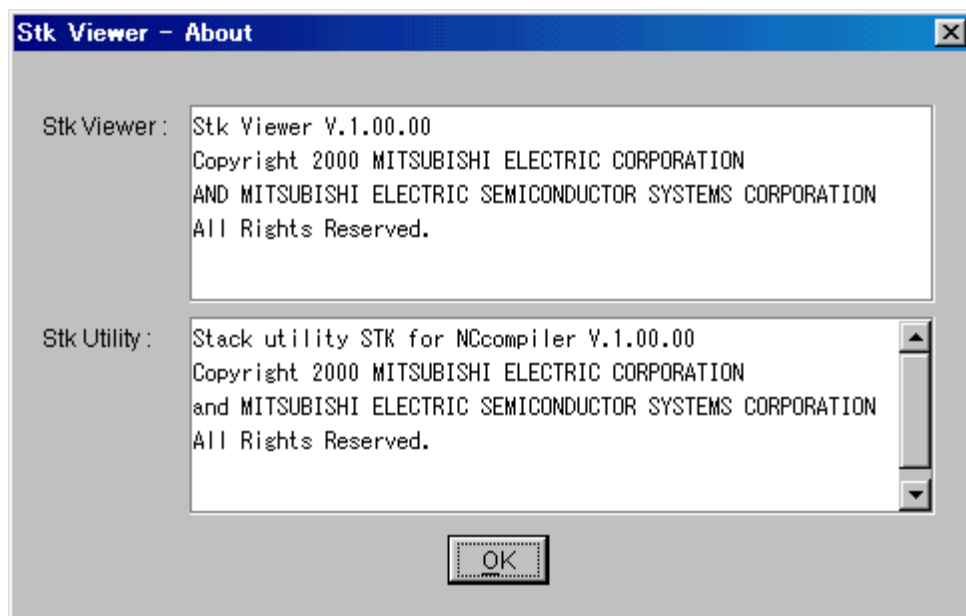
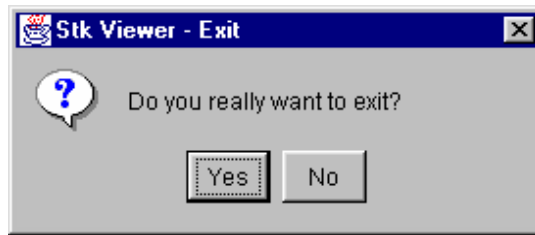


Figure 1.4 About Dialog Box

## 1.5 The Exit Dialog Box

This dialog box is provided for confirming whether or not to exit the STK Viewer.



**Figure 1.5 Exit Dialog Box**

## 2. Limitations

### 2.1 About the Mouse Cursor

If the mouse on your computer is used for an animation or a 3D cursor in Windows, or the mouse is set to leave its locus as a lag, the STK Viewer may not be displayed correctly. In such a case, please change the current mouse settings and use the mouse with standard settings.

If the problem still exists even after changing to standard settings, change the system properties performance setting by lowering the hardware accelerator in its detail settings of graphic. This will help to solve the problem.

### 2.2 About the `stk` Execution File

For the STK Viewer to be able to start `stk`, one of the following conditions must be met. If neither of these conditions are met, the STK Viewer will run properly.

- The STK Viewer and `stk` exist in the same directory
- The path to `stk` is specified

### 3. Starting the STK Viewer

#### When running in Windows

##### Starting from the TM

Click the [Stk Viewer Start] button on the project bar of TM V.3.00 or later. When a project is open in the TM, **stk** by default uses the project's absolute module file to calculate the stack size.

##### Starting from the Start menu

#### When running in UNIX

##### Starting from the command line

Enter the command shown below from the command line.

```
[path in which java installed]java -jar [path in which STK Viewer installed]StkViewer.jar
```

## 4. Using the STK Viewer

### 4.1 Calculating the Stack Size

- (i) Choose [File] -> [Open] from the menu or click the [Open] button on the toolbar to open a file selection dialog box. From this dialog box, specify the absolute module file you want to load.
- (ii) Click the [Open] button, and **stk** starts calculating the stack size after closing the dialog box. The result is shown in the function tree.

If the STK Viewer is started from the TM, **stk** by default selects the absolute module file opened by the TM as it calculates the stack size.

### 4.2 Stack Size of a Function with Unknown Stack Size

#### (1) Assembler functions

- (i) Choose an assembler function from the function tree or click the [Unknown] tab on the unresolved function list to choose the function whose stack size is unknown.
- (ii) When information on the selected function is shown in function information column of the window, enter the desired stack size.
- (iii) Choose [Action] -> [Calc] from the menu or click the [Calc] button on the toolbar, and **stk** recalculates the stack size for the function and updates the display accordingly.

#### (2) Indirectly called functions

- (i) Choose an indirectly called function from the function tree or click the [Indirect] tab on the unresolved function list to choose the function whose stack size is unknown.
- (ii) When information on the selected function is shown in function information column of the window, click the [Select Files...] button to open the Select Functions Dialog Box. From this dialog box, choose the called function by turning its check box on.
- (iii) Choose [Action] -> [Calc] from the menu or click the [Calc] button on the toolbar, and **stk** recalculates the stack size for the function and updates the display accordingly.

#### (3) Recursively called functions

- (i) Choose a recursively called function from the function tree or click the [Recursive] tab on the unresolved function list to choose the function whose stack size is unknown.
- (ii) When information on the selected function is shown in function information column of the window, enter the desired stack size.

(iii) Choose [Action] -> [Calc] from the menu or click the [Calc] button on the toolbar, and **stk** recalculates the stack size for the function and updates the display accordingly.

#### **(4) Clearing a specified stack size**

Choose [Action] -> [Reset] from the menu or click the [Reset] button on the toolbar. After clearing all of the specified stack size, **stk** recalculates the stack size for the functions and updates the display accordingly.

### **4.3 Reusing the Set Information**

#### **(1) Saving the set information**

- (i) Choose [File] -> [Save...] from the menu or click the [Save] button on the toolbar.
- (ii) A file selection dialog box is displayed. From this dialog box, specify the path and file name you want and click the [Save] button.

#### **(2) Loading the saved information**

- (i) Choose [File] -> [Open] from the menu or click the [Open] button on the toolbar.
- (ii) A file selection dialog box is displayed. From this dialog box, specify the path and file name you want and click the [Open] button. Based on the saved information, **stk** recalculates the stack size for the function and shows the result on the window. The contents hitherto displayed on the window are erased.

### **4.4 Showing Version Information**

Choose [Help] -> [About] from the menu or click the [About] button on the toolbar. An About Dialog Box is displayed, showing the version information on the STK Viewer and **stk**.

## 5. Error Messages of the STK Viewer

### 5.1 Error Messages

Table 5.1 lists the error messages output by the STK Viewer along with the contents of errors and the actions to be taken.

**Table 5.1 Error Messages of the STK Viewer**

Error message	Content of error and action
Error No.0 Stk Viewer: Internal Error Contact to Mitsubishi.	An internal error occurred. -> Please contact Renesas.
Error No.1 Stk Viewer: 'file-name' is not found. Confirm 'file-name'.	The file "file-name" could not be found. -> Check the file name.
Error No.2 Stk Viewer: STK Utility failed.	This is an error of the STK utility. -> Refer to Table 5.2, "Error Messages of the STK Utility."
Error No.3 Stk Viewer: File is not specified.	No file is specified. -> Specify the file name you want.
Error No.4 Stk Viewer: Invalid Stack Size. Stack size is a decimal number.	The specified stack size is incorrect. -> Specify any stack size in decimal.
Error No.5 Stk Viewer: Invalid STK File. Stk Viewer has gotten an invalid Stk file.	An STK file in incorrect format has been loaded. -> Check whether the STK file is correct.
Error No.6 Stk Viewer: Out of memory.	Memory is insufficient. There are too many functions that are called in the source file. Processing is stopped.

**Table 5.2 Error Messages of the STK Utility**

Error message	Content of error and action
Error : multiply input file.	Multiple input file names exist. -> Make sure there is only one input file name.
Error : target file not found.	The absolute module file does not exist. -> Specify another absolute module file that does exist.
Error : ignore option 'option-name'.	The option “option-name” is incorrect. -> Specify another option name that is correct.
Error : not enough memory.	Memory is insufficient. -> Close other applications or take other measures to secure the necessary memory capacity.
Error : cannot open 'file-name'.	Failed to open the file “file-name.” -> Check to see if the file exists.
Error : cannot close 'file-name'.	Failed to close the file “file-name.” -> Check the disk system of your computer.
Error : different format 'file-name'.	The absolute module file is in incorrect format. -> Check the absolute module file and, if necessary, recreate it.
Error : not found start function data.	The function data for which to start calculations does not exist. -> Please contact Renesas.
Warning : not found function 'function-name'.	Information for the function specified with -func= cannot be found. -> 1. Check to see if the specified function name is correct. 2. Check to see if the .x30 and .x79 files have been created after specifying -finfo.
System error [xxxxxx].	This is an internal error of STK. -> Please contact Renesas.

## Appendix A.

### A.1 Stack Calculation Utility

#### A.1.1 Input Format

- When specifying an option from the command line  
%STK [startup option] <absolute module file name>
- When specifying an option in the command file  
%STK @<command file name>

[: Represents the prompt.

<>: Represents an essential item.

[: Represents an optional item to be written when necessary.

When writing multiple startup options, separate each entry with a space.

#### Files needed to start STK

**Absolute module file** (essential item): This file is generated by specifying the -finfo option to add inspector information as a compiler/assembler startup option.

**Command file:** Use this file to specify startup options and the absolute module file name.

## A.1.2 Option Reference

### a. **-recursive**

Sets a recursively called function for which the stack size could not be calculated.

#### **Description**

When this option is specified, **stk** sets a recursively called function for which the stack size could not be calculated.

#### **Format**

-recursive = function name, notation in assembler, source file name, module information, storage class, line number, stack size

C function name \* Leave blank for assembler functions

notation in assembler

source file name: Full path to the file from which the specified function is called (File attributes cannot be omitted.) Essential for static functions.

module information: Module file name (File attributes cannot be specified)

storage class: G (external function)/ S (static function)/ I (inline function).  
The default storage class is "G."

line number: The line number from which the specified function is called

stack size: Specify in assembler function and recursively called function

#### **Input example**

-recursive = func,\_func,sample.c,sample,G,10,20

**b. -indirect**

Sets an indirectly called function for which the stack size could not be calculated.

**Description**

When this option is specified, **stk** sets an indirectly called function for which the stack size could not be calculated.

**Format**

-indirect = function name, notation in assembler, source file name, module information, storage class, line number

C function name: The called function name

\* Leave blank for assembler functions

notation in assembler: The name of the called function in assembler

source file name: Full path to the file from which the specified function is called (File attributes cannot be omitted.) Essential for static functions.

module information: The module file name that calls the specified function (File attributes cannot be specified)

storage class: The storage class of the called function

G (external function)/ S (static function)/ I (inline function).

The default storage class is "G."

line number: Line number from which the specified function is called

**Input example**

-indirect= func,\_func,sample.c,sample,G,10

### **c. -unknown**

Sets a function whose information is unknown

#### **Description**

When this option is specified, **stk** sets the function whose function information is unknown.

#### **Format**

-unknown = function name, notation in assembler, source file name, module information, storage class, line number, stack size

C function name           \* Leave blank for assembler functions

notation in assembler

source file name: Full path to the file from which the specified function is called (File attributes cannot be omitted.) Essential for static functions.

module information: The module file name (File attributes cannot be specified)

storage class: G (external function)/ S (static function)/ I (inline function).

The default storage class is "G."

line number: Line number from which the specified function is called

stack size: Specify in assembler function and recursively called function

#### **Input example**

-unknown= func,\_func,sample.c,sample,G,10,30

**d. -func**

Specifies the C function name for which to calculate the stack usage

**Description**

Only a C function name can be specified.

The function attribute is global. (Static not accepted)

**Format**

-func= function name

**Input example**

-func= main

**e. -V**

Shows version

**Description**

When this option is specified, **stk** shows its version information.

**Format**

-V

### A.1.3 Output Format

The calculation result is shown for only the target function for which **stk** made the calculation.

#### **Format**

function type C function name (notation in assembler): stack size

#### **function type**

Outputs the type of function. This information is not output for ordinary functions. There are following types of functions.

**\*INTERRUPT\***

Denotes an interrupt function.

**\*HANDLER\***

Denotes an MR handler such as an ALM handler, INT handler, or CYC handler.

**\*TASK\***

Denotes a task.

#### **C function name**

Outputs a C function name. This information is not output for assembler functions.

#### **notation in assembler**

Outputs the function name in assembler. No information is output for indirect calls.

#### **stack size**

Outputs the maximum stack size of the target function. However, if that function is followed by a recursively called function, etc, this is the maximum stack size within a calculable range.

#### **Output example**

```
D:¥ Temp>stk c:¥mtool¥smp79¥smp79.x79
      _sget(__sget):7 bytes
      (start):72 bytes,static
      init_dev($init_dev):5 bytes
      _pput($_pput):7 bytes
      calloc($calloc):45 bytes
      realloc(_realloc):81 bytes
*INTERRUPT* timer_a0int(_timer_a0int):193 bytes

D:¥ Temp>
```

---

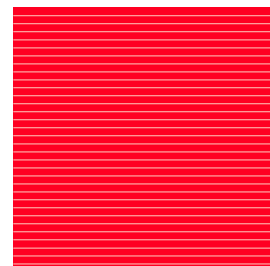
STK Viewer V.1.00 User's Manual

Publication Date: Mar. 01, 2005      Rev.1.00

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

Edited by: Microcomputer Tool Development Department  
Renesas Solutions Corp.

STK Viewer V.1.00  
User's Manual



**Renesas Technology Corp.**

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan