User's Manual

External Flash Definition Editor（EFE）

User's Manual

There is addition on page 34 in this document.

Renesas Electronics
www.renesas.com

Rev.1.01 Mar 2012

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.

Introduction

The External Flash Definition Editor (hereafter referred to as the EFE) is the utility software to create the definition files that are needed to use the "external flash download function."

When you've created an external flash memory definition file with the EFE, register it in the emulator software that supports this function. That way, it becomes possible to use the external flash download function.

Abbreviations

Following abbreviations and abbreviated words are used in this manual.

Abbreviation/abbreviated word	Description
EFE	External Flash Definition Editor (this product)
RFD file	Renesas Flash Definition file (flash memory device definition files)
USD file	User System Definition file (user system definition files)
Emulator	Emulator systems made by Renesas
External flash memory	Flash memory devices connecting to a microprocessor's external bus
External flash download function	Function to download data into external flash memory
Write program	Program for writing to external memory
Standard program	Standard write program preinstalled in this product
JEDEC method	JEDEC standard command-compliant flash program method
CUI method	Intel/Sharp CUI command flash program method
HEW	High-performance Embedded Workshop, or the integrated development environment from Renesas

Manual Notation

Following notations are used in this manual.

Notation	Meaning
[item name]	Strings enclosed in [ ] represent the EFE GUI item names.
'parameter'	Strings enclosed in single quotes ('') represent the parameters selected in the EFE GUI.
"string"	Strings enclosed in double quotes ("") represent the values set for each EFE GUI item.

Contents

1. Summary

The flash memory devices connected externally to the external buses of a microprocessor are referred to as "external flash memory." Also, the function of an emulator system to download into or rewrite data in external flash memory is referred to as the "external flash download function."

The External Flash Definition Editor (hereafter referred to as the EFE) is the utility software to create the definition files that are needed to use the external flash download function.

When you've created an external flash memory definition file with the EFE, register it in the emulator software that supports this function. That way, it becomes possible to use the external flash download function.

1.1 Basic Composition

The EFE consists of the "editor body" and "standard program."

The "editor body" generates the following definition files in GUI-based software:

(1) Flash memory device definition file (RFD file)
Defines the specification of a flash memory device.

(2) User system definition file (USD file)
Defines how external flash memory will be used in a user system environment and includes a write program. It is this file that is registered in the emulator software.

The "standard program," preinstalled in this product, is a program for writing to flash memory.

The standard program is implemented by using the standard flash memory commands generally accepted for flash devices, and can therefore be used for most flash memory.

The standard program supports the following functions:

(1) JEDEC method
(2) CUI method
(3) Fast programming mode (write buffer programming and successive programming by UnlockBypass)
(4) Parallel connection of flash memory devices
(5) Program to flash after the lock bit is cleared

1.2 Position of the Product

The EFE is a common utility software independent of any specific emulator system.

Figure 1.1 schematically shows the position of the product in user systems.

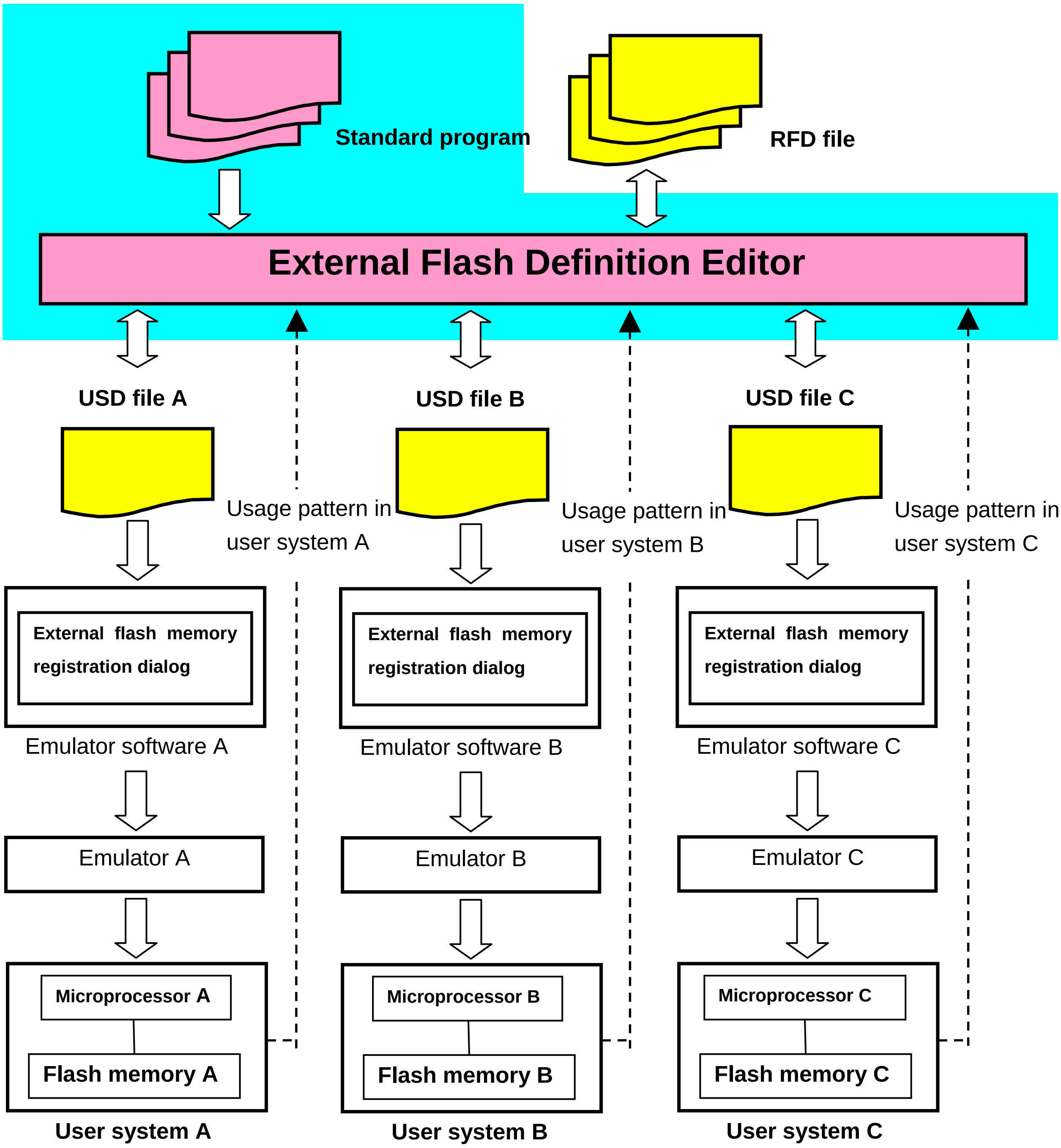


Figure 1.1 Position of the Product

1.3 Setup Procedure

1. Download the External Flash Definition Editor from the EFE product site.
2. Extract the downloaded file in any folder you want.
3. Launch EFE.exe in the extracted folder.

1.4 Folder Structure

Figure 1.2 shows the folder structure of this product.

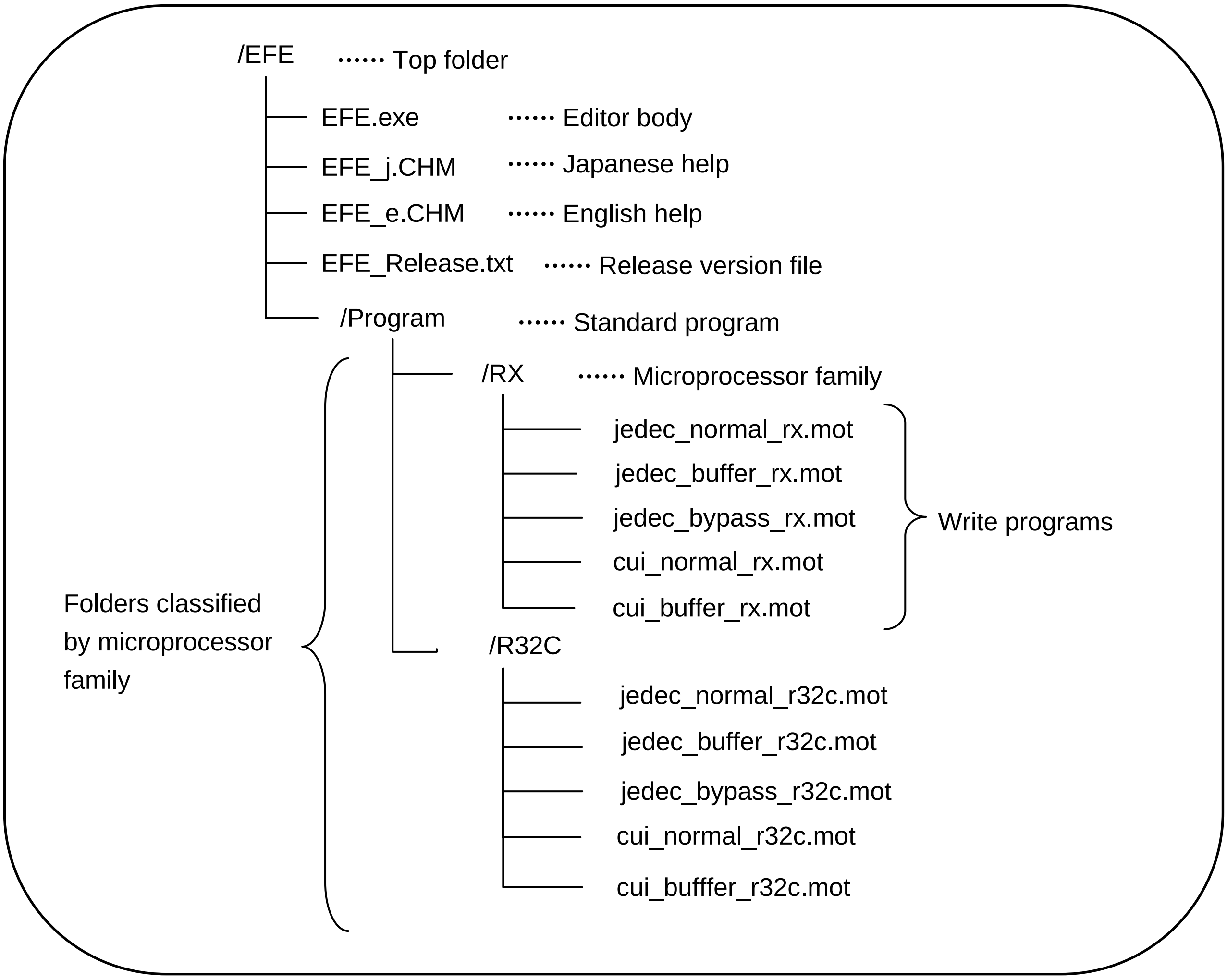


Figure 1.2 Folder Structure

1.5 Definition File Creation Flow

There are following two types of definition files that can be created with the EFE:

- ■ RFD File (Renesas Flash Definition file)
- ■ USD File (User System Definition file)

The RFD file is the one that defines the specification of the flash memory device used.
These files serve as input files for a USD file when it is created.

The USD file is the one that defines the manner of how flash memory will be used in a user system and the additional processing to be performed when programming to flash memory. The program used for writing to flash memory is included in this file.
By registering this file in the emulator software, it becomes possible to use the "external flash download function."

Figure 1.3 shows a flow of operations according to which definition files are generated.

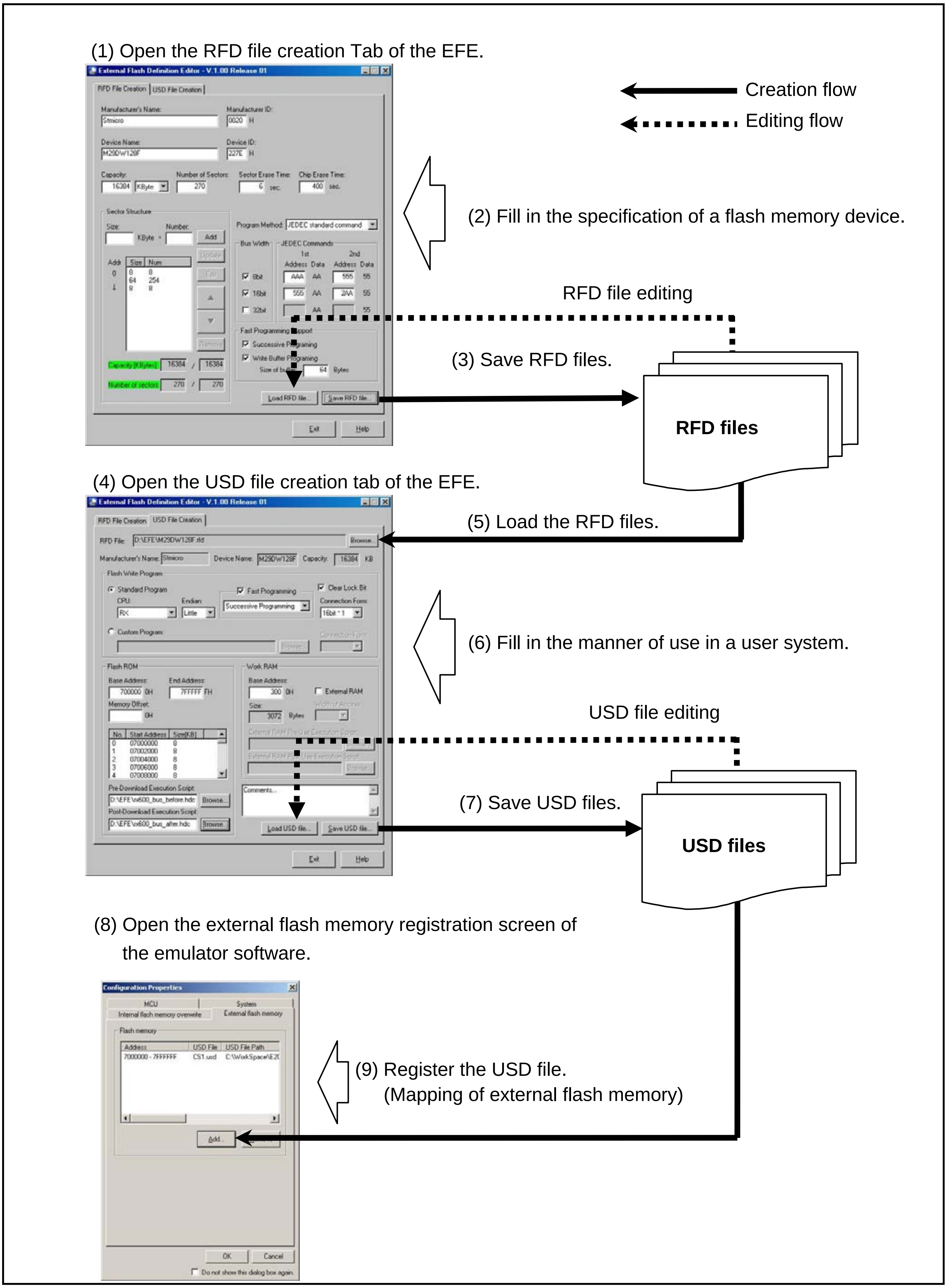


Figure 1.3 Definition File Creation Flow

2. RFD File Creation Tab

On the RFD file creation tab, fill in the specification of a flash memory device to generate an RFD file. Before proceeding, please prepare a data sheet for the flash memory device you use.

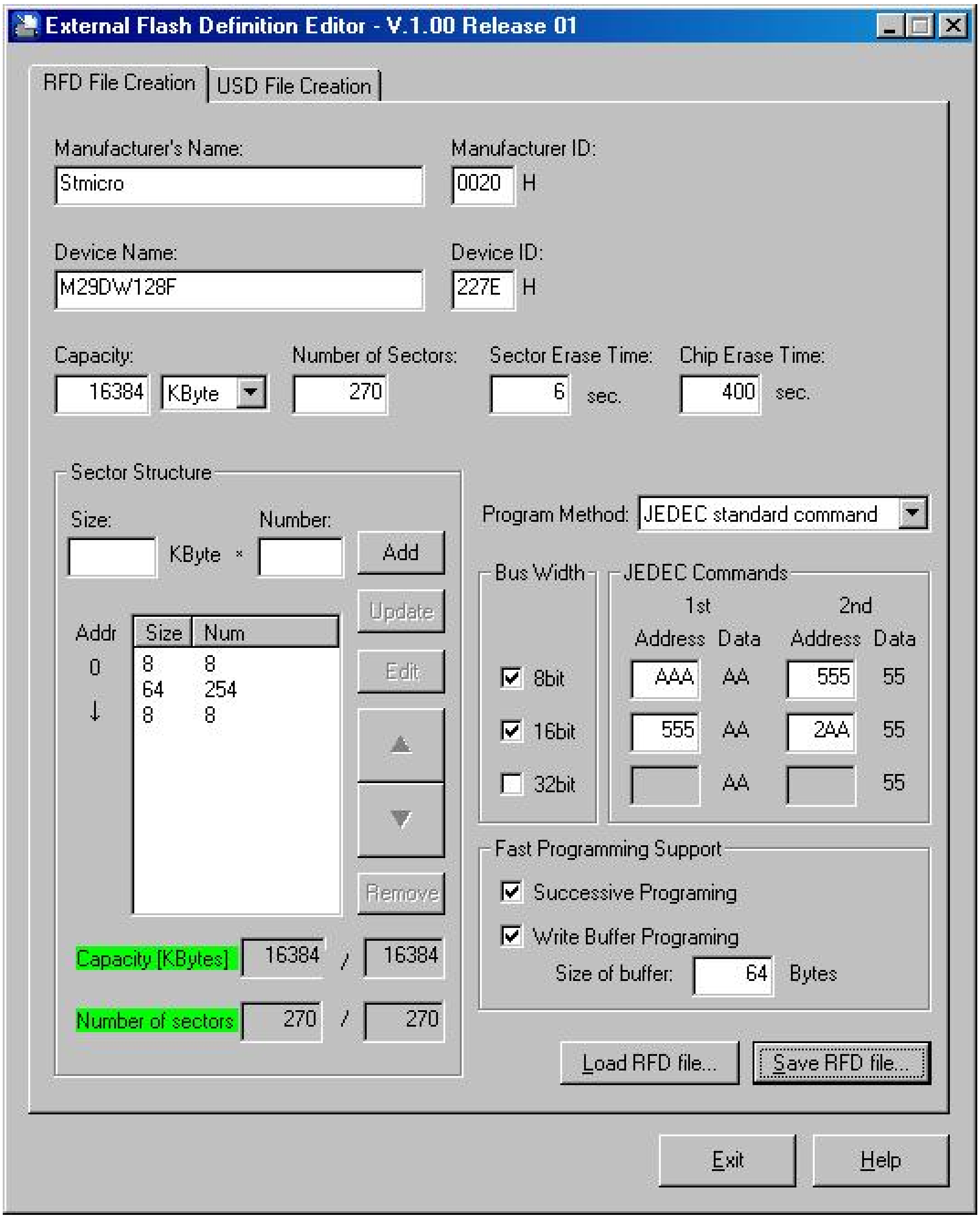


Figure 2.1 RFD File Creation Tab

2.1 Manufacturer's Name

Fill in the manufacturer's name of your flash memory device.

The manufacturer's name you enter here is displayed on screen when RFD files are loaded on the USD file creation tab.

2.2 Manufacturer ID

Fill in the manufacturer ID of your flash memory device.

The manufacturer ID, in the data sheet of each device manufacturer, may be stated as "maker ID," "maker code," "manufacturer ID," or "manufacturer code," for example.

Table 2.1 Examples of Manufacturer ID

Device manufacturer	Manufacturer ID	
	×8-bit device	×16-bit device
Spansion (AMD)	01h	0001h
Spansion (Fujitsu)	04h	0004h
Numonyx (STMicro electronics)	20h	0020h
Numonyx (Intel)	89h	0089h
Samsung	ECh	00ECh
Macronix	C2h	00C2h
Silicon Storage Technology	BFh	00BFh
Excel Semiconductor	4Ah	004Ah
Sharp	B0h	00B0h

Note: The above examples are shown for reference only.
Please be sure to consult the data sheet of your device manufacturer.

2.3 Device Name

Fill in the product type name of your flash memory device.

The product type name you enter here is displayed on screen when RFD files are loaded on the USD file creation tab.

2.4 Device ID

Fill in the device ID of your flash memory device.

The device ID, in the data sheet of each device manufacturer, may be stated as "device ID," "device code," "signature ID," or "signature code," for example.

Depending on the type of flash memory device concerned, the device ID may consist of multiple words.
In such a case, enter only the first word of a device ID.
For example, if the ID of your flash memory device is stated as follows in its data sheet, then enter "227E."

Table 2.2 Example of Entering a Device ID (1)

DEVICE CODE: 227Eh + 2220h + 2200h

Also, for flash memory devices whose data bus width is switchable, the device ID may be stated separately by bus width. In that case, enter the value given for the larger bus width than the other.
For example, if the ID of your flash memory device is stated as follows in its data sheet, then enter "1234."

Table 2.3 Example of Entering a Device ID (2)

	8-bit mode		16-bit mode	
	DQ15-DQ8	DQ7-DQ0	DQ15-DQ8	DQ7-DQ0
DEVICE CODE	Hi-Z	34h	12h	34h

2.5 Capacity

Enter the memory capacity in Kbyte or Mbyte units.
For memory capacities, the following applies:

1 Kbyte = 1,024 bytes
1 Mbyte = 1,024 Kbytes

For example, if the capacity of your flash memory device is stated as

128 Mbits (16 Mb × 8 or 8 Mb × 16)

then enter "16" [Mbyte] or "16384" [Kbyte].

2.6 Number of Sectors

Enter the aggregate number of sectors in your flash memory device.

For even flash memory devices whose sector sizes are irregular, enter simply the number of sectors, irrespective of the difference in sizes of individual sectors.

2.7 Sector Erase Time

Enter the maximum sector erase time in seconds. Fractional digits after the decimal point can also be entered.

However do not enter a value more than 32 seconds.

2.8 Chip Erase Time

Enter the maximum erase time of the entire chip in seconds.

2.9 Sector Structure

Figure 2.2 shows the composition of the sector structure group of the editor screen.

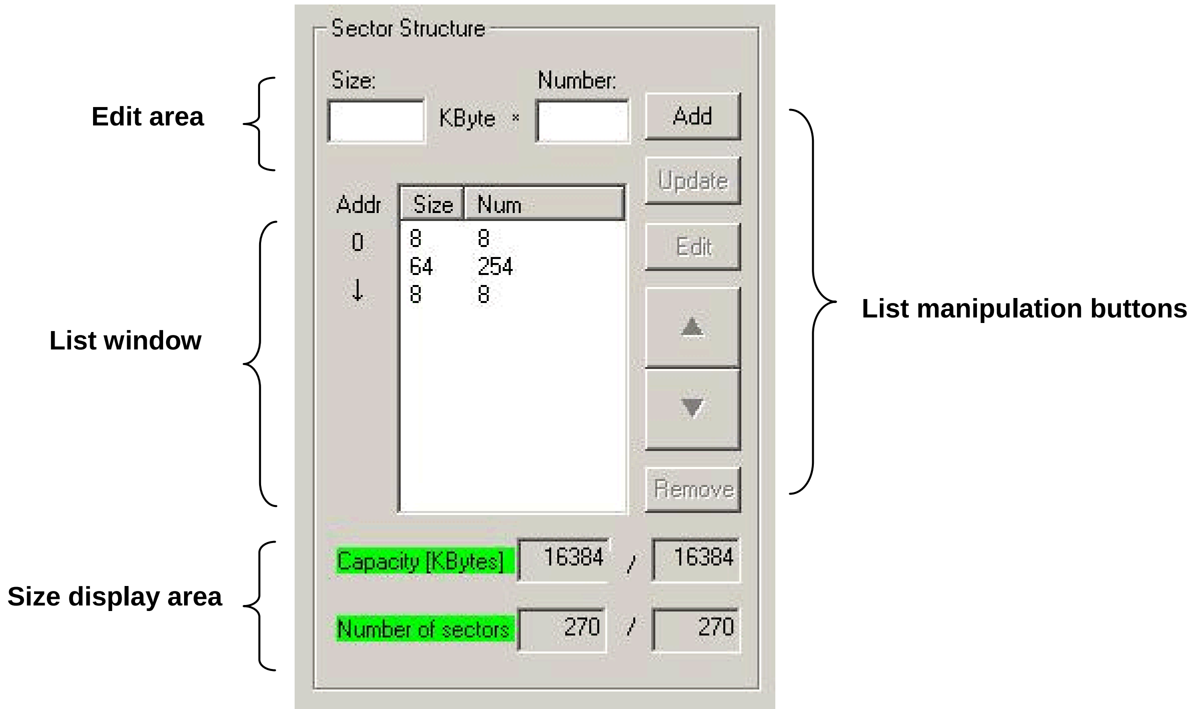


Figure 2.2 Sector Structure Group

2.9.1 Edit Area

Enter the sector structure of your flash memory device in the form "size of one sector × number of consecutive sectors."

When you click the [Add] or the [Update] button, the contents you've entered are reflected in the list window.

Until you finish defining the entire sector structure of your flash memory device in the list window, add list lines from here.

2.9.2 List Manipulation Buttons

These buttons permit you to manipulate the list.

[Add] button	Adds the content of the edit area to the list window. If any line in the list window is selected, the content is inserted into a place above the selected line. If no lines are selected, the content is inserted into the bottom line of the list.
[Update] button	Replaces the content of a selected line with that of the edit area.
[Edit] button	Shows the content of a selected line of the list window in the edit area.
[▲] button	Moves a selected line one line up in the list to change places.
[▼] button	Moves a selected line one line down in the list to change places.
[Remove] button	Removes a selected line from the list window.

2.9.3 List Window

Shows the sector structure you've entered in list form.

Each line in the list is selected when you click on it, allowing you to perform any operation on it using the list manipulation button.

Be sure the list is aligned in order of ascending memory addresses, from top to bottom of the list window.

For flash memory devices that have a sector structure like the one in Table 2.4, for example, set up the list as in Figure 2.3.

Table 2.4 Example of a Sector Structure

No	Size [KByte]	Address Range (×8)		No	Size [KWord]	Address Range (×16)
0	64	1F_0000h–1F_FFFFh		0	32	0F_8000h–0F_FFFFh
1	64	1E_0000h–1E_FFFFh		1	32	0F_0000h–0F_8FFFh
2	64	1D_0000h–1D_FFFFh		2	32	0E_8000h–0F_0FFFh
⋮	⋮	⋮	64KB×31	⋮	⋮	⋮
28	64	03_0000h–03_FFFFh		28	32	01_8000h–01_FFFFh
29	64	02_0000h–02_FFFFh		29	32	01_0000h–01_7FFFh
30	64	01_0000h–01_FFFFh		30	32	00_8000h–00_FFFFh
31	32	00_8000h–00_FFFFh	32KB×1	31	16	00_4000h–00_7FFFh
32	8	00_6000h–00_7FFFh	8KB×2	32	4	00_3000h–00_3FFFh
33	8	00_4000h–00_5FFFh		33	4	00_2000h–00_2FFFh
34	16	00_0000h–00_3FFFh	16KB×1	34	8	00_0000h–00_1FFFh

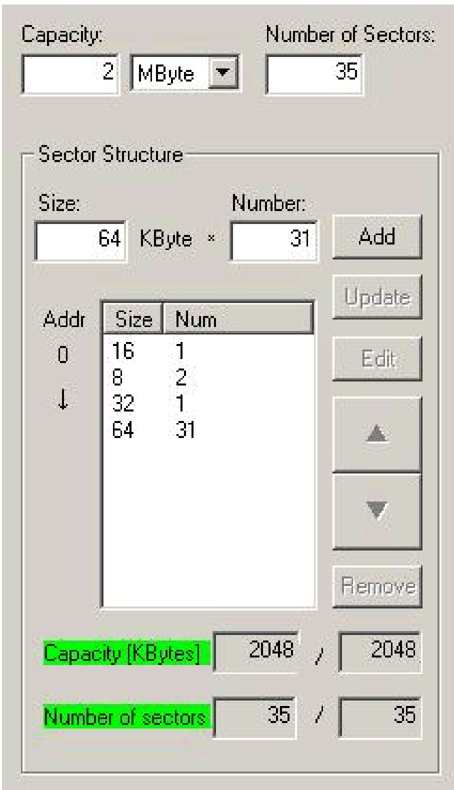


Figure 2.3 Example of Setting Up the List Window

2.9.4 Size Display Area

After calculating the entire memory capacity of the list and the number of sectors in the list window, it shows the result in the form shown below.

Capacity [Kbytes]: Total memory capacity of list lines / aggregate memory capacity
Number of sectors: Total number of sectors of list lines / aggregate number of sectors

The aggregate memory capacity and the aggregate number of sectors here show the values you've set in Section 2.5, "Capacity," and Section 2.6, "Number of Sectors," respectively.

The total memory capacity of list lines and the total number of sectors of list lines are automatically recalculated and redisplayed each time the list is updated.
When the left and right values of either item match, the color of the item name changes to green, showing an agreement with the expected value.
Conversely, if the left-side value of either item exceeds its right-side value, the color of the item name changes to red, showing that the input value does not agree.

When the entire sector structure is entered correctly, Capacity [Kbytes] and Number of Sectors both are displayed in green.

2.10 Program Method

For the program method, select either 'JEDEC standard command method' or 'CUI command method'.

2.11 Supported Bus Width

Put a check mark in the appropriate checkbox for the data bus width supported by your flash memory device.
If your flash memory device is a type that permits the bus width to be switched by pin control, put a check mark in all of the supported bus width checkboxes.

For example, if the data sheet of your flash memory device stipulates its bus width as follows

128 Mbit (16 Mb × 8 or 8 Mb × 16)

then check the bus width checkboxes as shown below.

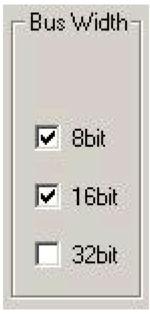


Figure 2.4 Example of Setting Up the Supported Bus Width

2.12 JEDEC Commands

Set the command address pattern used in the JEDEC method.

When you put a check mark in the supported bus width checkbox, the default command address pattern is displayed.

For almost all flash memory devices, these default settings do not need to be altered without causing any problem. However, products from some device manufacturers require different settings.

In that case, alter the values of the default command address pattern according to the contents stipulated in the product data sheet.

2.13 Fast Programming Support

If your flash memory device supports the fast programming mode, put a check mart in the appropriate checkbox.

For the JEDEC method, you can check the [Successive Programming] and [Write Buffer Programming] checkboxes. If both of these methods are supported, put a check mark in both checkboxes.

For the CUI method, you can check only the [Write Buffer Programming] checkbox.

If you've checked [Write Buffer Programming], enter the write buffer size in byte units which is internally available in your flash memory device.

Figure 2.5 shows, when the JEDEC method selected, an example of how to specify for a flash memory device that supports both [Successive Programming] and [Write Buffer Programming].

Figure 2.5 Example of Specifying Fast Programming Support

■ Successive Programming (JEDEC method)

Table 2.5 Successive Programming (JEDEC method)

Command	mode	1st		2nd		3rd	
		addr	data	addr	data	addr	data
Unlock Bypass Entry	16bit	555h	AAh	2AAh	55h	555h	20h
	8bit	AAAh	AAh	555h	55h	AAAh	20h
Unlock Bypass Program	-	X	A0h	PA	PD	-	-
Unlock Bypass Reset	-	X	90h	X	00h	-	-

* PA: Program Address

* PD: Program Data

■ Write Buffer Programming (JEDEC method)

Table 2.6 Write Buffer Programming (JEDEC method)

Command	mode	1st		2nd		3rd		4th		5th		6th	
		addr	data	addr	data	addr	data	addr	data	addr	data	addr	data
Write to Buffer and Program	16bit	555h	AAh	2AAh	55h	BA	25h	BA	N	PA	PD	BA	29h
	8bit	AAAh	AAh	555h	55h	BA	25h	BA	N	PA	PD	BA	29h

* BA: Block Address (sector address)
* N: Number of Words to be programmed - 1
* PA: Program Address
* PD: Program Data

■ Write Buffer Programming (CUI method)

Table 2.7 Write Buffer Programming (CUI method)

Command	1st		2nd		3rd		4th	
	addr	data	addr	data	addr	data	addr	data
Write to Buffer and Program	BA	E8h	BA	N	PA	PD	X	D0h

* BA: Block Address (sector address)
* N: Number of Words to be programmed - 1
* PA: Program Address
* PD: Program Data

2.14 Load RFD File

To edit the RFD file you've created or create a new RFD file based on the RFD file you've created, specify the source RFD file here that you want to edit or from which to create.

2.15 Save RFD File

This button saves the RFD file.

If the input content of the file is incomplete or erroneous, an error message is output and the save operation is canceled. In that case, correct the content according to the instructions of the error message.

3. USD File Creation Tab

On the USD file creation tab, you create a USD file that defines the manner of how your flash memory will be used.

By registering a USD file in the emulator software, you'll have your flash memory device recognized as external flash memory.

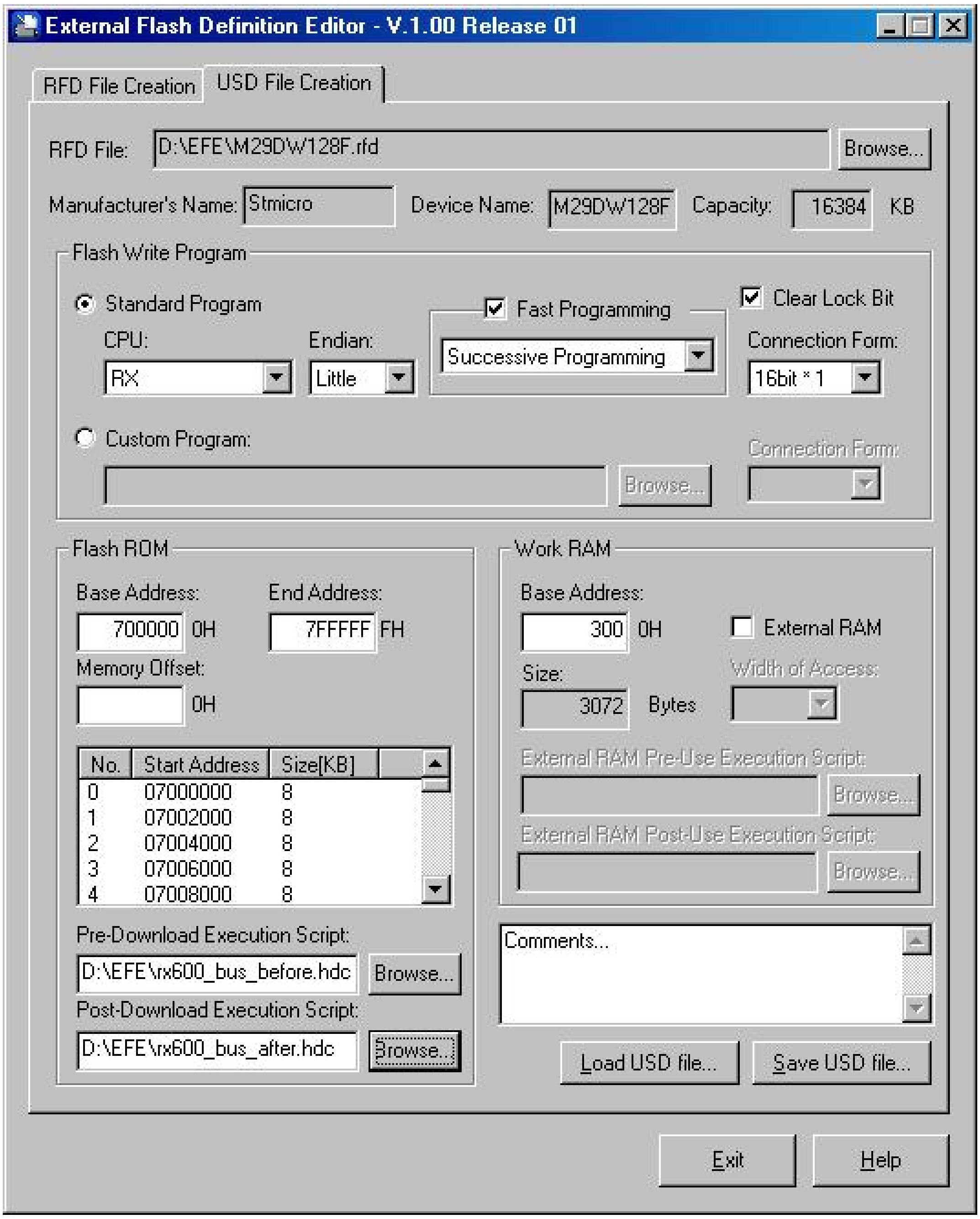


Figure 3.1 USD File Creation Tab

3.1 Referring to the RFD File

Specify the RFD file to be referred to.

To create a new USD file, you first need to specify the appropriate RFD file here.

Of the loaded information, three items of information—[Manufacturer Name], [Device Name], and [Capacity]—are displayed in the area below this box.

3.2 Flash Write Program

This group of the dialog is used to set a program for writing to flash memory.

3.2.1 Selecting a Write Program

Select the type of write program you want to use for program to flash memory.

■ Standard Program

Selecting this radio button uses the standard program preinstalled in this product.

The standard program is written using the standard flash memory commands supported by every device manufacturer, and can therefore be used for the majority of flash memory devices.

For the options available when [Standard Program] is selected, see Sections 3.2.2 through 3.2.6.

■ Custom Program

For flash memory devices for which the standard program is not usable, select this radio button.

In this case, you specify a write program you've prepared yourself.

For the options available when [Custom Program] is selected, see Sections 3.2.7 through 3.2.8

3.2.2 CPU

Choose the CPU type of your microprocessor from the pulldown menu.

3.2.3 Endian

For bi-endian microprocessors, specify the appropriate endian that suits the working environment.

For endian-fixed microprocessors, Endian is automatically displayed.

3.2.4 Fast Programming

To execute programs to flash memory in fast programming mode, check this checkbox.

You can select this checkbox when your flash memory device supports one or both of the following programming modes.

If you cannot program normally in fast programming mode, deselect this checkbox.

■ Write Buffer Programming

This programming mode is provided for flash memory devices that have write buffer memory.

Data is transferred one word in one bus cycle.

■ Successive Programming

This mode can be specified for only the JEDEC method.

With part of program commands omitted, data is transferred one word in two bus cycles.

3.2.5 Clear Lock Bit

If you want to program to flash memory after clearing the lock (protect) bits set for its sectors, check this checkbox. Note that the editor does not perform a relock process after completion of the program operation.

■ JEDEC method

The commands listed in Table 3.1 are used to clear the lock bits.

Table 3.1 Lock Set/Clear Commands when JEDEC Method is selected

	1st		2nd		3rd		4th	
	addr	data	addr	data	addr	data	addr	data
Set Lock bit	555h	00AAh	2AAh	0055h	555h	0048h	BA	XX01h
Clear Lock bit	555h	00AAh	2AAh	0055h	555h	0048h	BA	XX00h

* BA: Block Address (sector address)

* These locks are effective only during the 16-bit mode. Therefore, these locks are effective only when the flash memory is connected in the [16-bit × 1] or [16-bit × 2] form.

* The command names may not always be expressed as "Set Lock bit" and "Clear Lock bit" depending on device manufacturers.

■ CUI method

The commands listed in Table 3.2 are used to clear the lock bits.

Table 3.2 Lock Set/Clear Commands when CUI Method is selected

	1st		2nd	
	addr	data	addr	data
Block Protect	XXh	60h	BA	01h
Block Unprotect	XXh	60h	XXh	D0h

* BA: Block Address (sector address)

* When Block Unprotect is executed, the lock of the entire chip is unlocked.

* The command names may not always be expressed as "Block Protect" and "Block Unprotect" depending on device manufacturers.

3.2.6 Connection Form (Standard Program)

Specify the form of connection between the microprocessor and flash memory.

From the pulldown menu list, select the connection form that suits your working environment.

The list shows only the usable connection forms as determined from the flash memory used and the currently selected CPU.

The meaning of each displayed form of connection is as follows:

(1) 8bit × 1: One flash memory device in data bus width of 8 bits is connected. Therefore, the data bus is 8 bits wide.

(2) 8bit × 2: Two flash memory devices in data bus width of 8 bits are connected in parallel. Therefore, the data bus is 16 bits wide.

(3) 8bit × 4: Four flash memory devices in data bus width of 8 bits are connected in parallel. Therefore, the data bus is 32 bits wide.

(4) 16bit × 1: One flash memory device in data bus width of 16 bits is connected. Therefore, the data bus is 16 bits wide.

(5) 16bit × 2: Two flash memory devices in data bus width of 16 bits are connected in parallel. Therefore, the data bus is 32 bits wide.

(6) 32bit × 1: One flash memory device in data bus width of 32 bits is connected. Therefore, the data bus is 32 bits wide.

Figure 3.2 shows the relationships between the connection forms and aggregate memory capacities in respective cases where 1 MB (data 8 bits wide), 1 MB (data 16 bits wide), and 1 MB (data 32 bits wide) flash memory devices are connected.

(1) 8bit×1

D7-D0
0_0000h
0_0001h
· · · ·
F_FFFEh
F_FFFFh

Total 1MB

(2) 8bit×2

D15-D8	D7-D0
00_0001h	00_0000h
00_0003h	00_0002h
· · · ·	· · · ·
1F_FFFDh	1F_FFFCh
1F_FFFFh	1F_FFFEh

Total 2MB

(3) 8bit×4

D31-D24	D23-D16	D15-D8	D7-D0
00_0003h	000002h	00_0001h	00_0000h
00_0007h	000006h	00_0005h	00_0004h
· · · ·	· · · ·	· · · ·	· · · ·
3F_FFFBh	3F_FFFAh	3F_FFF9h	3F_FFF8h
3F_FFFFh	3F_FFFEh	3F_FFFDh	3F_FFFCh

Total 4MB

(4) 16bit×1

D15-D8	D7-D0
0_0001h	0_0000h
0_0003h	0_0002h
· ·	· ·
F_FFFDh	F_FFFCh
F_FFFFh	F_FFFEh

Total 1MB

(5) 16bit×2

D31-D24	D23-D16	D15-D8	D7-D0
00_0003h	00_0002h	00_0001h	00_0000h
00_0007h	00_0006h	00_0005h	00_0004h
· ·	· ·	· ·	· ·
1F_FFFBh	1F_FFFAh	1F_FFF9h	1F_FFF8h
1F_FFFFh	1F_FFFEh	1F_FFFDh	1F_FFFCh

Total 2MB

(6) 32bit×1

D31-D24	D23-D16	D15-D8	D7-D0
0_0003h	0_0002h	0_0001h	0_0000h
·	·	·	·
F_FFFFh	F_FFFEh	F_FFFDh	F_FFFCh

Total 1MB

Figure 3.2 Relationships between the Connection Forms and Aggregate Memory Capacities

3.2.7 Specifying a Custom Program Path

Specify a file path to the custom program.
The custom program must meet the following requirements:

- File format: Motorola S format
- Code size: 8,192 bytes or less
- Endian: For bi-endian CPUs, it is necessary that the program be compiled using little-endian format, irrespective of its run-time CPU endian.
  (Endian conversions executed in the emulator software)
- Write buffer programming
  Write to buffer 256 bytes or less.

When you create a custom program, see the separately supplied sample program for reference.

3.2.8 Connection Form (Custom Program)

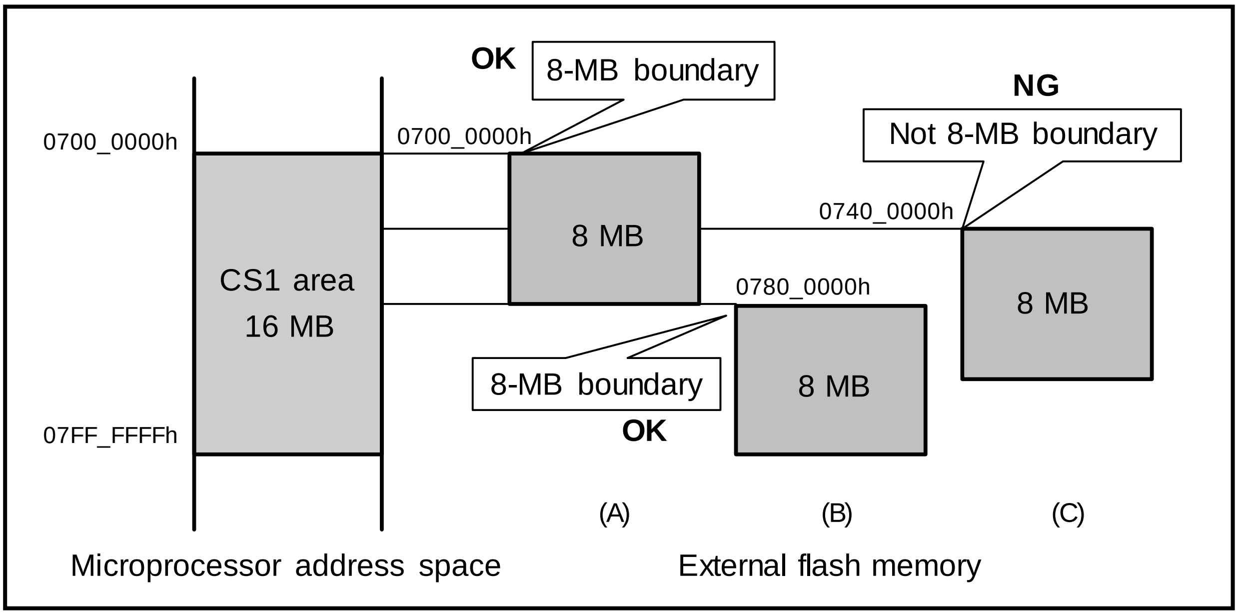
Specify the form of connection between the microprocessor and flash memory.
From the pulldown menu list, select the connection form that suits your usage conditions.
The meaning of each displayed form of connection is the same as in Section 3.2.6.

3.3 Flash ROM

Specify the address mapping of flash memory and the command script automatically issued before and after a download.

3.3.1 Base Address

Set the mapping start address of the flash memory.
Specify the address boundary that matches the flash memory size.
However, if you specify [Memory Offset] here there is no need to specify it.



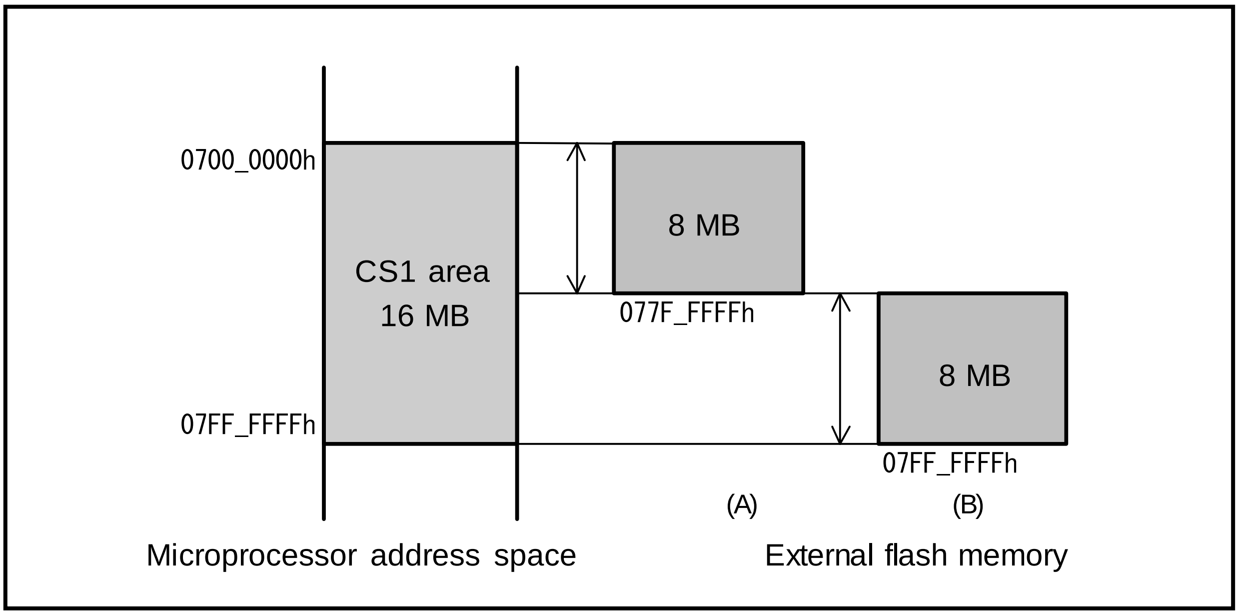
*Example for the case where 8-MB flash memory devices are mapped to the CS1 area by the RX610

Figure 3.3 Example of Setting the Base Address

3.3.2 End Address

Set the mapping end address of the flash memory.

Normally, set (base address + flash memory size – 1).



* Example for the case where 8-MB flash memory devices are mapped to the CS1 area by the RX610

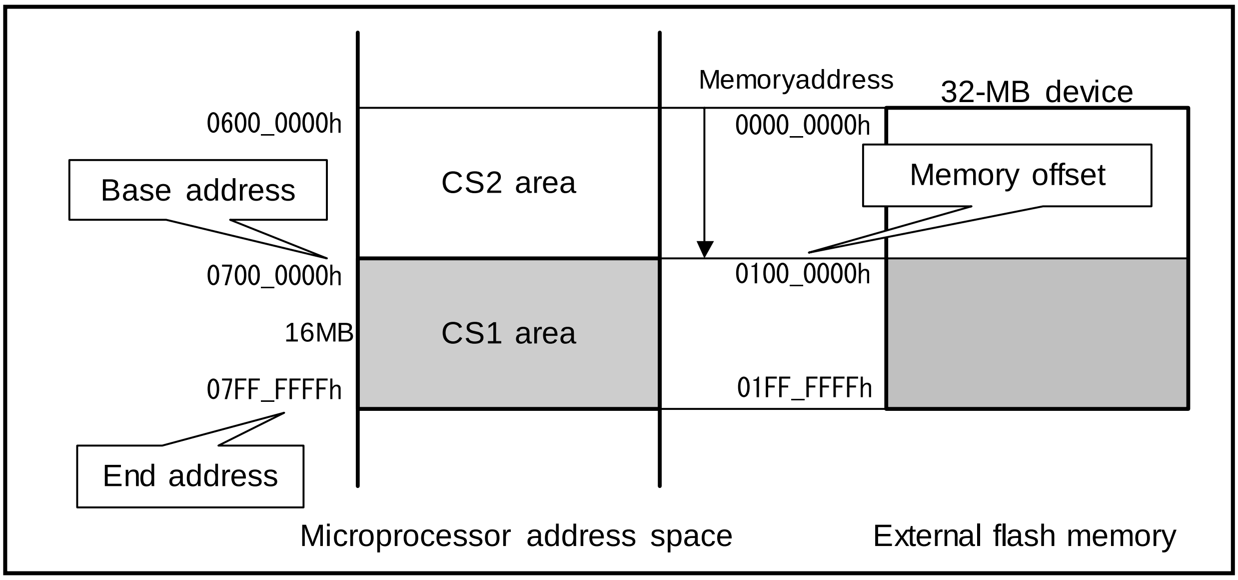
Figure 3.4 Example of Setting the End Address

If the end address exceeds the CS boundary, see Section 3.3.3, for details on how to set.

3.3.3 Memory Offset

When using large-capacity flash memory that may take up two or more of consecutive external CS areas of the microprocessor, you need to prepare separate USD files, one for each CS area.

In such a case, it is necessary to set a memory offset.



* Example for the case where the first half of 32-MB flash memory is mapped to the CS2 area and the latter half is mapped to the CS1 area by the RX610

* An external circuit is required that will input CS2# & CS1# to the CE# pin of the external flash and output a 0 to the most significant address A24 when CS2# = low or a 1 when CS1# = low.

Figure 3.5 Example of Using a Memory Offset

In the above example, CS2 and CS1 need to be set as follows:

■ CS2 area

[Base Address]:	"06000000"h	[End Address]:	"06FFFFFF"h
[Memory Offset]:	"00000000"h		

■ CS1 areas

[Base Address]:	"07000000"h	[End Address]:	"07FFFFFF"h
[Memory Offset]:	"01000000"h		

To the CS2 area, map the first half of the flash memory.

Change the initial value of the end address from 07FF_FFFFh to the end address of the CS2 area, 06FF_FFFFh

To the CS1 area, map the latter half of the flash memory.

For the memory offset, set 0100_0000h, an amount equal to the size of the CS2 area.

The end address is automatically changed from 08FF_FFFFh to 07FF_FFFFh simultaneously as the memory offset changes.

3.3.4 Mapping Information Window

This window shows the mapping start address and size per sector.

The content of this window is updated in real time according to the input values of [Base Address], [End Address], and [Memory Offset].

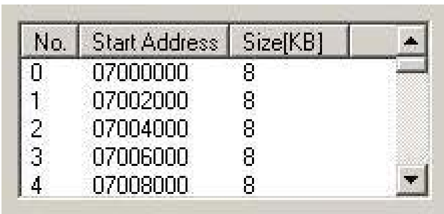


Figure 3.6 Mapping Information Window

3.3.5 Pre-Download Execution Script

Immediately before a download process, it is possible to execute a command script of HEW.

Here, specify the script file to be executed on that occasion.

By writing the external bus controller settings, etc. needed for accessing the external flash in a script beforehand, you can save the time that might otherwise be taken for various settings before executing a download.

If a download is executed while the CPU clock remains set to a low speed, it is possible that processing will take an extremely long time or will not terminate normally. We recommend that a setting to speed up the CPU clock be written in this script.

For details about the commands usable in a script file, see Section 4.3, "About the Script Command".

3.3.6 Post-Download Execution Script

Immediately after a download process, it is possible to execute a command script of HEW.

Here, specify the script file to be executed on that occasion.

By writing the external bus controller setting write-back processes, etc. in a script beforehand, you can save the time that might otherwise be taken for various settings after executing a download.

For details about the commands usable in a script file, see Section 4.3, "About the Script Command".

3.4 Work RAM

Set up the work RAM as needed for program to flash memory.

The work RAM data is automatically saved to the stack before a download is executed and automatically restored from the stack after the download is completed.

3.4.1 Base Address

Set the mapping start address of the work RAM.

Specify the internal RAM or one of RAMs connected to the external bus (hereafter called an external RAM).

3.4.2 Size

Displayed here is the necessary size of the work RAM that is automatically calculated.

3.4.3 External RAM

When you use an external RAM as the work RAM, check this checkbox.

3.4.4 Width of Access

Specify the width of access of the external RAM.

3.4.5 External RAM Pre-Use Execution Script

If you've chosen to use an external RAM, it is possible to automatically execute a command script of HEW immediately before use of the work RAM begins.

Here, specify the script file to be executed on that occasion.

By writing the external bus controller settings, etc. needed for accessing the external RAM in a script beforehand, you can save the time that might otherwise be taken for various settings before executing a download.

For details about the commands usable in a script file, see Section 4.3, "About the Script Command".

3.4.6 External RAM Post-Use Execution Script

If you've chosen to use an external RAM, it is possible to automatically execute a command script of HEW immediately after use of the work RAM is finished.

Here, specify the script file to be executed on that occasion.

By writing the external bus controller setting write-back processes, etc. in a script beforehand, you can save the time that might otherwise be taken for various settings after executing a download.

For details about the commands usable in a script file, see Section 4.3, "About the Script Command".

3.5 Comment Column

You can enter any character string as a comment here.

The comment you enter is recorded in the USD file.

3.6 Referring to the USD File

To re-edit the USD file you’ve created or create a new USD file based on the USD file you’ve created, specify the source USD file here that you want to edit or from which to create.

3.7 Save USD File

This button saves the USD file.

If the input content of the file is incomplete or erroneous, an error message is output and the save operation is canceled. In that case, correct the content according to the instructions of the error message.

4. Precautions

4.1 About the Generation of USD Files

The USD files should be created following the rules described below.

1. Basically, create one USD file per flash memory device.
   However, flash memory devices are used in parallel-connected form as in an 8-bit × 2, 8-bit × 4, or 16-bit × 2 form, handle the whole as one flash memory device and create one USD file for those devices.

2. For a type of flash memory device that takes up two or more external CS areas, create USD files separately, one for each CS area.
   (For details, see your emulator software manual.)
   In that case, use a memory offset as you create a USD file for each CS area.

4.2 When Using the USD File in Another PC

The script file you specify on the USD file creation tab is saved in the USD file in the form “absolute file path + file name.”

Therefore, if you want to use an existing USD file that has had a script file specified in another PC, you need to edit the path to the script file to make it suitable for the PC environment you use.

Here is the edit process.

1. Install EFE in the PC you use.
2. Launch EFE and open the USD file creating tab.
3. Use the [Load USD File...] button to load the USD file you’ve obtained.
4. In the script specification menu, edit the destination into which the script file is to be loaded.
5. Click the [Save USD File...] button, and a new USD file is created.

4.3 About the Script Command

On the USD file creation tab, you can specify one of the following script files:

- Pre-download execution script
- Post-download execution script
- External RAM pre-use execution script
- External RAM post-use execution script

In the above script files, the following command format is supported:

■ memory_fill command

Command format: MF <start> <end> <data> [<mode>]

start: start address（Can't be omitted）

end: end address（Can't be omitted)

data: fill data（Can't be omitted)

mode: [BYTE][WORD][LONG]; if omitted, [BYTE] is assumed

Follow the rules below when writing.

(1) Only use a single white space or a single tab to separate parameters.

(2) Start address must not be higher than the end address.

(3) Do not use the symbols to indicate the bases of numbers, such as 0x, for addresses and data.

Example: MEMORY_FILL 8C000 8C001 AA55 WORD

This changes the values at addresses 0x8C000 and 0x8C001 to the word 0xAA55.

Note:

Only part of the options of the memory_fill command usable in the command line window of HEW are supported.

If other options than those given above are specified, device operation cannot be guaranteed.

External Flash Definition Editor (EFE)
User's Manual

Publication Date:	Dec 12 2009	Rev.1.00
	Mar 3 2012	Rev.1.01

Published by: Renesas Electronics Corporation

Edited by: Renesas Solutions Corp.

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

External Flash Definition Editor

User's Manual

Renesas Electronics Corporation

R20UT1966EJ0101