FUJITSU Semiconductor Devices

CONTROLLER MANUAL

F²MC-8L Family
Compact ICE Debugger
for SOFTUNE Workbench
Command Reference

Windows95/98/NT4.0 Version

1. Circuit diagrams utilizing Fujitsu products are included as a mean of illustrating typical semiconductor applications. Complete information sufficient for construction proposes is not necessarily given.
2. The information contained in this document has been carefully checked and is believed to be reliable. However, Fujitsu assumes no responsibility for inaccuracies.
3. The information contained in this document does not convey any license under the copy right, patent right to trademarks claimed and owned by Fujitsu.
4. Fujitsu reserved the right to change products or specifications without notice.
5. No part of this publication may be copied or reproduced in any form or by any means, or transferred to any third party without prior written consent of Fujitsu.
6. The products described in this document are not intended for use in equipment requiring high reliability, such as marine relays and medical life-support systems. For such applications, contact your Fujitsu sales representative.

PREFACE

■ What is the Softune Workbench?

Softune Workbench is support software for developing programs for the FR, F^2MC-16, and F^2MC-8L families of microcontrollers.

It is a combination of a development manager, simulator debugger, emulator debugger, monitor debugger, and an integrated development environment for efficient development.

This dynamic link library is a debugger to control the compact ICE in cooperation with F^2MC-8L Softune Workbench

■ Organization of Manual

This manual consists of 14 chapters and one appendix.

Chapter 1 Environment Setup Commands
Chapter 2 Program Execution Commands
Chapter 3 Break/Event Control Commands
Chapter 4 Program Execution Analysis Commands
Chapter 5 Memory/Register Operation Commands
Chapter 6 Line Assemble and Disassemble Commands
Chapter 7 Load and Save Commands
Chapter 8 Source File/Symbol Commands
Chapter 9 Command Procedure Commands
Chapter 10 Replacement Commands
Chapter 11 Utility Commands
Chapter 12 Task Debug Commands
Chapter 13 Control Commands
Chapter 14 Built-in Variables and Functions
Appendix

Contents

Command Reference Notation Format

The command reference notation format is given below.

Command name (debuggers)

"Format"

"Description"

"Example"

"See Also"

Command name:

Name of command to be explained

Debuggers

Usable commands depend on the debugger type. Which debugger can use the command and which debugger cannot use it are described.

◎: Can use command

○: Can use command except when instruction being executed

×: Cannot use command

－: There is no debugger

Format

The format, parameters, and command qualifiers of the command are explained. Enter the command in this format.

Description

The command function is explained.

Example

Command coding example. This example may differ slightly from the actual coding.

1 Environment Setup Commands

1.1 INITIALIZE

	SIM	EML	MON
[8L Compact ICE]			
INITIALIZE	—	◎	—
[8L] INITIALIZE	◎	◎	—

“Description”

The **INITIALIZE** command initializes the debugger.

This initialization nullifies all settings other than macro, alias and debug variable.

“Example”

```
>INITIALIZE
```

1.2 EXIT

	SIM	EML	MON
[8L Compact ICE]			
EXIT	—	◎	—
[8L] EXIT	◎	◎	—

“Description”

The **EXIT** command terminates the debugger.

“Example”

>EXIT

1.3 RESET

	SIM	EML	MON
[8L Compact ICE]			
RESET	—	◎	—
[8L] RESET	◎	◎	—

“Description”

The **RESET** command inputs the reset signal to the MCU.

“Example”

>RESET

1.4 SET MODE

	SIM	EML	MON
[8L Compact ICE]			
SET MODE	—	○	—
[8L] SET MODE	×	○	—

"Format"

- Command qualifiers

[MB2140 ICE]

/NORMAL (default at start-up)

Sets event mode to NORMAL.

/MULTITRACE

Sets event mode to MULTITRACE.

/PERFORMANCE

Sets event mode to PERFORMANCE.

[Compact ICE]

/OR

Sets event mode to OR mode.

/SEQUENCE

Sets event mode to SEQUENCE mode.

"Description"

The **SET MODE** command sets the event mode as follows:

- NORMAL mode
  The event function is used for trace control by a sequencer. Command setting related to **SEQUENCE**, **DELAY**, and **TRACE** is enabled.
- MULTITRACE mode
  The event function is used for multitracing. Command setting related to **MULTITRACE** is enabled.
- PERFORMANCE mode
  The event function is used for measuring performance. Command setting related to **PERFORMANCE** is enabled.

The commands related to **EVENT** can be used in all modes, each of which has different values. If a mode is changed, the value will return to the value previously set in the mode.
A mode change will also clear the single-trace, multitrace, and performance buffers.
The default is "/**NORMAL**".

The **SET MODE** command sets the event mode as follows:

- OR mode
  The event function is used for trace control by a sequencer. Command setting related to **DELAY** and **TRACE** is enabled.

- SEQUENCE mode
  The event function is used for a sequencer. Command setting related to **DELAY** and **TRACE** is enabled.

The default is "/**OR**".

“Example”

```
>SET MODE /MULTITRACE
```

1.5 SHOW MODE

	SIM	EML	MON
[8L Compact ICE]			
SHOW MODE	—	◎	—
[8L] SHOW MODE	×	◎	—

“Description”

The **SHOW MODE** command displays the setting state of the event mode.

“Example”

```
>SHOW MODE
event mode : normal
```

1.6 SET RADIX

	SIM	EML	MON
[8L Compact ICE]			
SET RADIX	◎	◎	—
[8L] SET RADIX	◎	◎	—

"Format"

- Command qualifiers

/BINARY

Sets default base number to binary number.

/OCTAL

Sets default base number to octal number.

/DECIMAL

Sets default base number to decimal number.

/HEXADECIMAL (default)

Sets default base number to hexadecimal number.

"Description"

The **SET RADIX** command sets default base number.

"Example"

```
>SET RADIX/HEXADECIMAL
```

1.7 SHOW RADIX

	SIM	EML	MON
[8L Compact ICE]			
SHOW RADIX	—	◎	—
[8L] SHOW RADIX	◎	◎	—

“Description”

The **SHOW RADIX** command displays the current base number.

“Example”

```
>SHOW RADIX
default radix : hexadecimal
```

1.8 SET SOURCE

	SIM	EML	MON
[8L Compact ICE]			
SET SOURCE	—	◎	—
[8L] SET SOURCE	◎	◎	—

"Format"

- Command qualifiers

/DISPLAY (default at start-up)

Sets mode in which source lines displayed.

/NODISPLAY

Sets mode in which source lines not displayed.

"Description"

When the disassemble list is displayed, the **SET SOURCE** command sets whether to display the added source line.

When the debugger is started, the mode in which source lines are displayed is set.

"Example"

```
>SET SOURCE/DISPLAY
```

1.9 SHOW SOURCE

	SIM	EML	MON
[8L Compact ICE]			
SHOW SOURCE	—	◎	—
[8L] SHOW SOURCE	◎	◎	—

“Description”

The **SHOW SOURCE** command displays the source line display mode set by the **SET SOURCE** command.

“Example”

```
>SHOW SOURCE
source mode : display
```

1.10 SHOW SYSTEM

	SIM	EML	MON
[8L Compact ICE]			
SHOW SYSTEM	—	◎	—
[8L] SHOW SYSTEM	◎	◎	—

"Description"

The **SHOW SYSTEM** command displays system information.

"Example"

```
FR Family Softune Workbench V30L00R01
Debugger type        = Emulator Debugger
MCU type             = MB91101
DSU type             = DSU2
Monitor version      = V01L01
Communication device = LAN
Host name            = efr11
```

1.11 SET MAP

			SIM	EML	MON
(Format 1)					
[8L Compact ICE]					
	SET MAP	{address | address-range}		○	—
[8L]	SET MAP	{address | address-range}	○	○	—
(Format 2)					
[8L Compact ICE]					
	SET MAP	{/GUARD | /NOGUARD}	—	○	—
[8L]	SET MAP	{/GUARD | /NOGUARD}	×	○	—

"Format"

- Parameters

address (address formula)

Specify the memory address where access attribute to be set.

address-range (address formula)

Specify the memory area where access attribute to be set.

- Command qualifiers

Specifying access attribute (used in format 1)

/READ

Enables data read access.

[8L] 8L assumes even code access to be READ.

/WRITE

Enables data write access.

/CODE

Enables code read access.

If command qualifier is omitted, **/READ/WRITE** is set.

Specifying area type (used in format 1)

/USER (only for EML)

Sets area to user memory area

/EMULATION (only for EML) (default when omitted)

Sets area to emulation area

Access attribute of undefined area (used in format 2)

/GUARD (only for EML) (default when internal ROM provided)

Disables access to undefined area.

/NOGUARD (only for EML) (default when internal ROM not provided)

Enables full access to undefined area.

"Description"

The **SET MAP** command sets a memory space area type and access attribute.

(SIM)

Up to 31 memory areas can be set.

When the load module file is loaded by the **LOAD** command, appropriate access attributes are automatically set according to the file information.

For the FR simulator, up to 128 Mbytes can be specified in the total of each area.

(EML)

- User memory area (**/USER**)
  Up to 20 areas including the emulation area can be specified in bytes. There is no restriction on the size of one area.
- Emulation area (**/EMULATION**)
  Up to 20 areas including the user memory area can be specified in bytes. There is no restriction on the size of one area.
- Undefined area
  Either the command qualifier for allowing any access (**/NOGUARD**) or for disallowing access (**/GUARD**) can be specified for an undefined area.

"Example"

```
>SET MAP/READ/WRITE 1000..1FFF
```

1.12 SHOW MAP

	SIM	EML	MON
[8L Compact ICE]			
SHOW MAP	—	◎	—
[8L] SHOW MAP	◎	◎	—

“Description”

The **SHOW MAP** command displays the set memory space access attributes.

“Example”

```
>SHOW MAP
          address                attribute
00000000    ..  000011FF        read write
00001200    ..  FFFEFFFF        undefined
FFFF0000    ..  FFFFFFFF        read code
```

1.13 CANCEL MAP

		SIM	EML	MON
[8L Compact ICE]				
	CANCEL MAP { address | address-range }	—	○	—
[8L]	CANCEL MAP { address | address-range }	○	○	—

"Format"

- Parameters

  address (address formula)

  Specify the address where undefined attribute to be assigned.

  address-range (address formula)

  Specify the address range where undefined attribute to be assigned.

- Command qualifier

  /ALL

  Assigns undefined attribute to all set maps.

"Description"

The **CANCEL MAP** command assigns the undefined attribute to the specified address area.

"Example"

```
>CANCEL MAP/ALL
```

1.14 ENABLE VERIFYMODE

	SIM	EML	MON
[8L Compact ICE]			
ENABLE VERIFYMODE	—	○	—
[8L] ENABLE VERIFYMODE	×	○	—

“Description”

The **ENABLE VERIFYMODE** command enables the verify mode used when memory is written by a command.

The verify mode is enabled when the debugger is started.

“Example”

>ENABLE VERIFYMODE

1.15 DISABLE VERIFYMODE

	SIM	EML	MON
[8L Compact ICE]			
DISABLE VERIFYMODE	—	○	—
[8L] DISABLE VERIFYMODE	×	○	—

"Description"

The **DISABLE VERIFYMODE** command disables the verify mode used when memory is written by a command.

The verify mode is enabled when the debugger is started.

"Example"

```
>DISABLE VERIFYMODE
```

1.16 SHOW VERIFYMODE

	SIM	EML	MON
[8L Compact ICE]			
SHOW VERIFYMODE	—	◎	—
[8L] SHOW VERIFYMODE	×	◎	—

“Description”

The **SHOW VERIFYMODE** command displays the status of the verify mode (mode in which verify operation is enabled or displayed when memory is written by a command).

“Example”

```
>SHOW VERIFYMODE
verify mode : enable
```

1.17 SET VECTOR

		SIM	EML	MON
[8L Compact ICE]				
	SET VECTOR vector-number, address-value	—	○	—
[8L]	SET VECTOR vector-number, address-value	○	○	—

"Format"

- Parameters

vector-number

Specify the number of vector to be set.

address-value

Specify the starting address of routine corresponding to specified vector number.

"Description"

The **SET VECTOR** command sets the address value of the vector number set in the specified area.

"Example"

```
>SET VECTOR 11, 0FF100
>SHOW VECTOR 11..11
VectorNo.   Address    Symbol
11          00FF100
```

1.18 SHOW VECTOR

		SIM	EML	MON
[8L Compact ICE]				
	SHOW VECTOR [vector-number-range]	—	◎	—
[8L]	SHOW VECTOR [vector-number-range]	◎	◎	—

“Format”

- Parameter

  vector-number-range

  Specify the range of vector numbers to be displayed.
  Specify range in "[starting-number..ending-number]" format.

“Description”

The **SHOW VECTOR** command displays vector number data.
If vector-number-range specifying is omitted, vector number display is started from the next vector number.

“Example”

```
>SHOW VECTOR 6..8
VectorNo.   Address    Symbol      Factor
6           00000000               System Reserved
7           FF201000   co_1000     Co-processor Absence
8           FF110000   CO_ERROR    Co-processor error
```

1.19 ENABLE WATCHDOG

	SIM	EML	MON
[8L Compact ICE]			
ENABLE WATCHDOG	—	○	—
[8L] ENABLE WATCHDOG	×	○	—

"Description"

The **ENABLE WATCHDOG** command enables a watchdog timer.

"Example"

>ENABLE WATCHDOG

1.20 DISABLE WATCHDOG

	SIM	EML	MON
[8L Compact ICE]			
DISABLE WATCHDOG	—	○	—
[8L] DISABLE WATCHDOG	×	○	—

"Description"

The **DISABLE WATCHDOG** command enables a watchdog timer.

"Example"

>DISABLE WATCHDOG

1.21 SHOW WATCHDOG

	SIM	EML	MON
[8L Compact ICE]			
SHOW WATCHDOG	—	◎	—
[8L] SHOW WATCHDOG	×	◎	—

"Description"

The **SHOW WATCHDOG** command displays the enabled/disabled state of a watchdog timer.

"Example"

```
>SHOW WATCHDOG
watchdog : enable
```

Chapter 2 Program Execution Commands

2.1 GO

		SIM	EML	MON
[8L Compact ICE]				
	GO [starting-address] [, break-address1] [, break-address2]	—	○	—
[8L]	GO [starting-address] [, break-address1] [, break-address2]	○	○	—

"Format"

- Parameters

starting-address (address formula)

Specify the address at which program execution started.

break-address (address formula)

Specify the address at which program execution stopped.

- Command qualifiers

Return setting

/RETURN

Executes program from function currently being executed to parent function return location.

Only programs coded in C can use this function.

The optimized program may not be stopped normally.

Setting interrupt mask

/MASK

Masks interrupt.

/NOMASK

Does not mask interrupt.

[MB2140 ICE only] Trace control

/ENABLETRACE

Enables trace function at start of program execution.

/DISABLETRACE

Disables trace function at start of program execution.

"Description"

The **GO** command executes the program from the specified starting address.

If starting-address specifying is omitted, the program is executed from the address indicated by the current program counter.

The break address set by the **GO** command is automatically deleted when program execution is stopped.

The command qualifiers, **/ENABLETRACE** and **/DISABLETRACE**, are specified for trace control by a sequencer.

If a command qualifiers is omitted, program execution will start as set by the **SET GO** command.

"Example"

```
>GO power$20
Break at main$10
>GO power$20, main$5
```

2.2 SET GO

	SIM	EML	MON
[8L Compact ICE]			
SET GO	—	○	—
[8L] SET GO	○	○	—

"Format"

- Command qualifiers

Setting interrupt mask

/MASK

Masks interrupt.

/NOMASK (default at start-up)

Does not mask interrupt.

[MB2140 ICE only] Trace control

/ENABLETRACE (default at start-up)

Enables trace function at start of program execution.

/DISABLETRACE

Disables trace function at start of program execution.

"Description"

The **SET GO** command specifies the execution conditions for the **GO** command as command qualifiers are omitted.

"Example"

```
>SET GO /MASK
>GO
```

2.3 SHOW GO

	SIM	EML	MON
[8L Compact ICE]			
SHOW GO	—	◎	—
[8L] SHOW GO	◎	◎	—

“Description”

The **SHOW GO** command displays the current execution conditions (**SET GO** command settings) for the **GO** command.

“Example”

```
>SHOW  GO
go mode : nomask, enabletrace
```

2.4 STEP

	SIM	EML	MON
[8L Compact ICE]			
STEP [step count]	—	○	—
[8L] STEP [step count]	○	○	—

"Format"

- Parameter

step-count (default: decimal number)

Specify the count of times **STEP** command executed (H'1 to H'FFFFFFFF).

If step-count specifying is omitted, the count of times is 1.

- Command qualifiers

/INSTRUCTION

Executes program in units of machine instructions.

/LINE

Executes program in units of source lines.

/AUTOMATIC (default at start-up)

Automatically changes execution unit according to source window display mode as follows:

- When the source window display mode is the source line display mode, the program is executed in units of source lines (**/LINE**).
- When the source window display mode is another display mode, the program is executed in units of machine instructions (**/INSTRUCTION**).

/INTO

Executes program for each step in called function, subroutine, or interrupt handler.

/OVER

Executes function call and subroutine call instructions (i.e., **CALL**) and software interrupt instructions (i.e., **INT**) as one step.

Function call is valid when **/LINE** is specified. Subroutine call instructions and software interrupt instructions are valid when **/INSTRUCTION** is specified.

/MASK

Masks interrupt while the MCU running.

/NOMASK (default at start-up)

Does not mask interrupt while the MCU running.

"Description"

The **STEP** command executes the program in units of source lines or machine

instructions according to the condition set by the **SET STEP** command.
The condition set by the **SET STEP** command can be ignored by specifying a command qualifier.

"Example"

```
>STEP
>STEP/INSTRUCTION
```

2.5 SET STEP

	SIM	EML	MON
[8L Compact ICE]			
SET STEP	—	○	—
[8L] SET STEP	○	○	—

"Format"

- Command qualifiers

/INSTRUCTION

Executes program in units of machine instructions.

/LINE

Excutes program in units of source lines.

/AUTOMATIC (default at start-up)

Automatically changes execution unit according to source window display mode as follows:

- When the source window display mode is the source line display mode, the program is executed in units of source lines (**/LINE**).
- When the source window display mode is another display mode, the program is executed in units of machine instructions (**/INSTRUCTION**).

/INTO (default at start-up)

Executes program for each step in called function, subroutine, or interrupt handler.

/OVER

Executes function call and subroutine call instructions (i.e., **CALL**) and software interrupt instructions (i.e., **INT**) as one step.

Function call is valid when **/LINE** is specified. Subroutine call instructions and software interrupt instructions are valid when **/INSTRUCTION** is specified.

"Description"

The **SET STEP** command specifies the step execution condition when no command qualifier is specified in the **STEP** command.

When the Softune workpbench is started, the step execution condition is **AUTOMATIC**, **INTO**.

"Example"

```
>SET STEP/INSTRUCTION
```

2.6 SHOW STEP

	SIM	EML	MON
[8L Compact ICE]			
SHOW STEP	—	◎	—
[8L] SHOW STEP	◎	◎	—

"Description"

The **SHOW STEP** command displays the step execution condition of the current **STEP** command.

"Example"

```
>SHOW STEP
step mode : insruction, into
```

2.7 CALL

	SIM	EML	MON
[8L Compact ICE]			
CALL function name ([argument [, ...]])	—	○	—
[8L] CALL function name ([argument [, ...]])	○	○	—

"Format"

- Parameters

function-name

Specify the name of function to be called.

argument

Compiles with C arguments.

However, structures and unions cannot be specified as variable names.

"Description"

The **CALL** command executes the specified function and displays a return value. (However, if the return value is of structure or union type, an error occurs).

The **CALL** command can be used only when the program coded in C is compiled with debug information.

If a breakpoint is reached when a function is being executed by the **CALL** command, the program breaks at that position.

CALL command execution is continued by subsequently restarting program execution with the **GO** command.

To suspend **CALL** command execution, use the **CLEAR CALL** command.

The **CALL** command cannot be nested.

The register and flag values before the function is called are retained. These values are restored to the original values after the function has been executed.

The argument of the specified function is evaluated and executed in dummy argument type.

If the count of specified actual arguments is greater than that of dummy arguments, extra actual arguments are evaluated in int type.

The return value is set in built-in variable **%CALL**.

The **CALL** command sets a breakpoint at the address indicated by the current program counter and sets the return address at the address so that control will return to the breakpoint. The command then calls the function.

For this reason, if the function executed by the **CALL** command accidentally passes the address indicated by the current program counter, the program will break in the middle of the function.

In this case, the following message is displayed:

Break at address by Invalid call termination

CALL command execution is continued by restarting program execution with the **GO** command.

“Example”

```
>CALL debug (cmd, p)
return value is H'0001
```

2.8 CLEAR CALL

	SIM	EML	MON
[8L Compact ICE]			
CLEAR CALL	—	○	—
[8L] CLEAR CALL	○	○	—

"Description"

The **CLEAR CALL** command cancels the **CALL** command and restores the status set before the register is called.

"Example"

```
>CALL debug (cmd, p)
Break at 00FF0F20 by breakpoint
>CLEAR CALL
```

2.9 SHOW STATUS

	SIM	EML	MON
[8L Compact ICE]			
SHOW STATUS	—	◎	—
[8L] SHOW STATUS	◎	◎	—

“Description”

The **SHOW STATUS** command displays the MCU execution status.

If the MCU breaks, the command displays the break factor of the immediately-preceding program execution.

“Example”

```
>SHOW STATUS
MCU status : executing
>SHOW STATUS
break at 0000FF00 by breakpoint
```

Chapter 3 Break/Event Control Commands

3.1 SET BREAK

		SIM	EML	MON
[8L Compact ICE]				
	SET BREAK break-address [, pass-count]	—	○	—
[8L]	SET BREAK break-address [, pass-count]	○	○	—

"Format"

- Parameters

break-address (address formula)

Specify the address at which breakpoint set.

pass-count (default: decimal number)

Specify the number of times breakpoint to be hit (1 to 65535).

Program execution specifying is stopped when this number of times is reached.

If pass-count is omitted, 1 is assumed.

Pass-count is valid only in the simulator debugger; it is ignored in the emulator and compact ICE debuggers.

"Description"

The **SET BREAK** command sets a breakpoint at the specified break address.

The count of breakpoints to be specified is as follows:

[8L-SIM]	65535
[8L-EML]	65535
[8L-Compact ICE]	65535

"Example"

```
>SET BREAK 00ff0200
>SET BREAK 00ff0300, 3
```

3.2 SHOW BREAK

	SIM	EML	MON
[8L Compact ICE]			
SHOW BREAK [breakpoint-number [, ...]]	—	◎	—
[8L] SHOW BREAK [breakpoint-number [, ...]]	◎	◎	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number.

- Command qualifier

  /ALL (default when omitted)

  Displays all breakpoints.

"Description"

The **SHOW BREAK** command displays the breakpoints set by the **SET BREAK** command.

"Example"

```
>SHOW BREAK
no.   en/dis   address    pass-count      symbol
1     enable   00FF0F00   1 (     1)
4     disable  00FF20DE   65535 ( 1234)
```

3.3 CANCEL BREAK

	SIM	EML	MON
[8L Compact ICE]			
CANCEL BREAK [breakpoint-number [, ...]]	—	○	—
[8L] CANCEL BREAK [breakpoint-number [, ...]]	○	○	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number.

  Use the **SHOW BREAK** command to reference the set breakpoint numbers.

- Command qualifier

  /ALL

  Cancels all breakpoints.

"Description"

The **CANCEL BREAK** command cancels the specified breakpoint(s).

"Example"

```
>CANCEL BREAK 1
>CANCEL BREAK 3
```

3.4 ENABLE BREAK

	SIM	EML	MON
[8L Compact ICE]			
ENABLE BREAK [breakpoint-number [, ...]]	—	○	—
[8L] ENABLE BREAK [breakpoint-number [, ...]]	○	○	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number.

  Use the **SHOW BREAK** command to reference the set breakpoint numbers.

- Command qualifier

  /ALL

  Enables all breakpoints.

"Description"

The **ENABLE BREAK** command enables the specified breakpoint(s).

"Example"

```
>ENABLE BREAK 2
>ENABLE BREAK 3, 4
```

3.5 DISABLE BREAK

	SIM	EML	MON
[8L Compact ICE]			
DISABLE BREAK [breakpoint-number [, …]]	—	○	—
[8L] DISABLE BREAK [breakpoint-number [, …]]	○	○	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number.

  Use the **SHOW BREAK** command to reference the set breakpoint numbers.

- Command qualifier

  /ALL

  Disables all breakpoints.

"Description"

The **DISABLE BREAK** command disables the specified breakpoint(s).

"Example"

```
>DISABLE BREAK 2
>DISABLE BREAK 3, 4
```

3.6 SET DATABREAK

		SIM	EML	MON
[8L Compact ICE]				
	SET DATABREAK data-access-address [, pass-count]	—	○	—
[8L]	SET DATABREAK data-access-address [, pass-count]	○	○	—

"Format"

- Parameters

data-access-address (address formula)

Specify the address at which data access breakpoint set.

pass-count (default: decimal number)

Pass-count is valid only in the simulator debugger.
Specify the number of times data access breakpoint hit (1 to 65535).
Program execution is stopped when this number of times is reached.
If pass-count is specifying omitted, 1 is assumed.

"Description"

The **SET DATABREAK** command breaks the program when data at the specified address is accessed.
The pass count value is set each time the program is executed.
Data breakpoints can be specified are shown below.

[8L-SIM] 65535
[8L-EML] 65535
[8L Compact ICE] 65535

If an automatic variable in the function is specified, the current address at which the variable is stored is set as the data access address (take care when using automatic variables).
To break the program when a C variable is accessed, specify "&" before the variable as the variable address.

"Example"

```
>SET DATABREAK &checkflg, 3
```

3.7 SHOW DATABREAK

		SIM	EML	MON
[8L Compact ICE]				
	SHOW DATABREAK [breakpoint-number [, ...]]	—	◎	—
[8L]	SHOW DATABREAK [breakpoint-number [, ...]]	◎	◎	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number.

- Command qualifier

  /ALL (default when omitted)

  Displays all breakpoints.

"Description"

The **SHOW DATABREAK** command displays the breakpoints set by the **SET DATABREAK** command.

"Example"

```
>SHOW DATABREAK
no.   en/dis   address    read/write   pass-count      symbol
1     enable   00002000   read only    1 (    0)       \trac
4     disable  00002052   write only   65535 ( 2345)
```

3.8 CANCEL DATABREAK

		SIM	EML	MON
[8L Compact ICE]				
	CANCEL DATABREAK [breakpoint-number [, ...]]	—	○	—
[8L]	CANCEL DATABREAK [breakpoint-number [, ...]]	○	○	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number.

  Use the **SHOW DATABREAK** command to reference the set breakpoint numbers.

- Command qualifier

  /ALL

  Cancels all data access breakpoints.

"Description"

The **CANCEL DATABREAK** command cancels the specified data access breakpoint(s).

"Example"

```
>CANCEL DATABREAK 1
>CANCEL DATABREAK 3
```

3.9 ENABLE DATABREAK

		SIM	EML	MON
[8L Compact ICE]				
	ENABLE DATABREAK [breakpoint-number [, ...]]	—	○	—
[8L]	ENABLE DATABREAK [breakpoint-number [, ...]]	○	○	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the breakpoint number

  Use the **SHOW DATABREAK** command to reference the set breakpoint numbers.

- Command qualifier

  /ALL

  Enables all data breakpoints.

"Description"

The **ENABLE DATABREAK** command enables the specified data breakpoint(s).

"Example"

```
>ENABLE DATABREAK 2
>ENABLE DATABREAK 3, 4
```

3.10 DISABLE DATABREAK

		SIM	EML	MON
[8L Compact ICE]				
	DISABLE DATABREAK [breakpoint-number [, ...]]	—	○	—
[8L]	DISABLE DATABREAK [breakpoint-number [, ...]]	○	○	—

"Format"

- Parameter

  breakpoint-number (default: decimal number)

  Specify the a breakpoint number.

  Use the **SHOW DATABREAK** command to reference the set breakpoint numbers.

- Command qualifier

  /ALL

  Disables all data breakpoints temporarily.

"Description"

The **DISABLE DATABREAK** command disables the specified data breakpoint(s).

"Example"

```
>DISABLE DATABREAK 2
>DISABLE DATABREAK 3, 4
```

3.11 SET EVENT

	SIM	EML	MON
[8L Compact ICE] SET EVENT event number, address [& = mask] [, [!] d = data [& = mask] [, p = pass count]	—	○	—
[8L] SET EVENT event number, address [& = mask] [, [!] d = data [& = mask] [, e = external probe data [& = mask]]	×	○	—

"Format"

- Parameters

event-number

Specify the event number (1 to 8).

In case of the compact ICE, specify the event number (1 to 2).

address [& = mask] (address type, data type)

Specify a memory location taken as an event generating condition. If a mask is specified, only one portion where the bit of the mask is 1 will be valid and the others will be "don't care".

If mask data is omitted, all the bits will be valid.

Automatic variables in C cannot be specified.

d = data [& = mask] (data type, data type)

Specify the data taken as an event generating condition. If a mask is specified, only one portion where the bit of the mask is 1 will be valid and the others will be "don't care".

If mask data is omitted, all the bits will be valid.

If **!** is specified, the specified data and mask will be assumed to be "not".

Incase of the compact ICE, "not" cannot be used.

e = external probe data [& = mask] (data type, data type)

Specify the external probe data (8 bits in length) taken as an event generating condition. If a mask is specified, only one portion where the bit of the mask is 1 will be valid and the others will be "don't care".

If mask data is omitted, all the bits will be valid.

In case of the compact ICE, the external probe data cannot be used.

p = pass count (Compact ICE only)

Specify the event hit coount. .

- Command qualifiers
- **Access attributes**

/CODE

Takes code access to specified address as event generating condition.

/READ

Takes read access to specified address as event generating condition.

/WRITE

Takes write access to specified address as event generating condition.

/MODIFY (MB2140 ICE only)

Takes changing of data at specified address as event generating condition.

/MODIFY cannot be specified together with other qualifiers for specifying access attributes. If **/MODIFY** is given, the address mask will be disabled. **/CODE** and **/WRITE** cannot be specified.
When **/MODIFY** is omitted, **/CODE** is assumed to be specified.

"Description"

[MB2140 ICE]The **SET EVENT** command sets the event that triggers a sequencer, multitrace, and performance. If data and external probe data are omitted, they will be all "don't care".
If an address is assumed to be an event condition, it will be affected by a prefetch by the MCU. Setting should be performed considering the prefetch by the MCU.
The NATIVE mode has the following restrictions:

- Only **/CODE** can be specified. If other access attributes are specified, an error is assumed.
- No data specifying (d =) is allowed in parameters.

The event is set in each mode set by the **SET MODE** command. Up to 8 events can be specified in each mode and information about them is independent in each mode. If a mode is changed, event information in a mode before the change will be saved and event information previously set in a mode after the change will be restored.
In the normal mode, the event will trigger a sequencer. To avoid this, only an event should be set.
In the MULTITRACE mode, all the set events will trigger a multitrace.
In the PERFORMANCE mode, each event number has the following meaning.

Event 1 → Starting event in section 1
Event 2 → Ending event in section 1
Event 3 → Starting event in section 2
Event 4 → Ending event in section 2
Event 5 → Starting event in section 3
Event 6 → Ending event in section 3
Event 7 → Starting event in section 4
Event 8 → Ending event in section 4

Use of the **SET RUNMODE** command will clear all the event settings.

"Example"

```
>SET EVENT /READ1, func1
>SET EVENT /WRITE   2,&data[2],!d=h'10
```

3.12 SHOW EVENT

		SIM	EML	MON
[8L Compact ICE]				
	SHOW EVENT [event-number [, …]]	—	◎	—
[8L]	SHOW EVENT [event-number [, …]]	×	◎	—

"Format"

- Parameter

  event-number

  Specify the event number (1 to 8).

  In case of the compact ICE, specify the event number (1 to 2).

- Command qualifier

  /ALL (default when omitted)

  Specifies all events.

"Description"

The **SHOW EVENT** command shows the contents set by the **SET EVENT** command.

"Example"

```
>SHOW EVENT
```

3.13 CANCEL EVENT

		SIM	EML	MON
[8L Compact ICE]				
	CANCEL EVENT [event-number [, …]]	—	◎	—
[8L]	CANCEL EVENT [event-number [, …]]	×	◎	—

"Format"

- Parameter

  event-number

  Specify the event number (1 to 8).
  In case of the compact ICE, Specify the event number (1 to 2).

- Command qualifier

  /ALL (default when omitted)

  Specifies all events.

"Description"

The **CANCEL EVENT** command cancels the event corresponding to a specified event number.

"Example"

```
>CANCEL EVENT
```

3.14 ENABLE EVENT

	SIM	EML	MON
[8L Compact ICE]			
ENABLE EVENT [event-number [, …]]	—	◎	—
[8L] ENABLE EVENT [event-number [, …]]	×	◎	—

"Format"

- Parameter

  event-number

  Specify the event number (1 to 8).

  In case of the compact ICE, specify the event number (1 to 2).

- Command qualifier

  /ALL (default when omitted)

  Specifies all events.

"Description"

The **ENABLE EVENT** command enables the event temporary disabled.

"Example"

```
>ENABLE EVENT
```

3.15 DISABLE EVENT

		SIM	EML	MON
[8L Compact ICE]				
	DISABLE EVENT [event-number [, …]]	—	◎	—
[8L]	DISABLE EVENT [event-number [, …]]	×	◎	—

"Format"

- Parameter

  event-number

  Specify the event number (1 to 8).
  In case of the compact ICE, specify the event number (1 to 2).

- Command qualifier

  /ALL (default when omitted)

  Specifies all events.

"Description"

The **DISABLE EVENT** command temporarily disables the event.

"Example"

```
>DISABLE EVENT
```

3.16 SET DELAY

	SIM	EML	MON
[8L Compact ICE]			
SET DELAY [delay-count]	—	○	—
[8L] SET DELAY [delay-count]	×	○	—

"Format"

- Parameter

delay-count (default: decimal number)

Specify the value (0 to 65,535) of the delay between when a sequencer terminates and when a trace terminates.
The delay count is in machine cycles.
In case of the compact ICE, specify the value (0 to 255) of the delay count.

- Command qualifiers

/BREAK (default at start-up)

Specifies stopping of MCU execution when delay count terminates.

/NOBREAK

Specifies no stopping of MCU execution when delay count terminates.

"Description"

The **SET DELAY** command sets the delay count as a sequencer terminates and specifies whether to cause a break at the end of a delay count.

"Example"

```
>SET DELAY /NOBREAK 200
```

3.17 SHOW DELAY

	SIM	EML	MON
[8L Compact ICE]			
SHOW DELAY	—	○	—
[8L] SHOW DELAY	×	○	—

"Description"

The **SHOW DELAY** command displays the setting state of a delay count and the setting state of a break as a delay count terminates.

"Example"

```
>SHOW DELAY
```

Chapter 4 Program Execution Analysis Commands

4.1 SHOW CALLS

	SIM	EML	MON
[8L Compact ICE]			
SHOW CALLS [call-frame-count]	—	○	—
[8L] SHOW CALLS [call-frame-count]	○	○	—

"Format"

- Parameter

  call-frame-count (default: decimal number)

  Specifies count of call frames requiring information (D'1 to D'256).

"Description"

The **SHOW CALLS** command displays the calling history until current function.

When call-frame-count is not specified, the command displays up to 255 frames.

When the function to be displayed contains an argument, the command displays the argument as a hexadecimal number.

If there is no C debug information, the command displays the function address.

The command analyzes accumulated stack data and determines which data to display according to the analysis result. It analyzes accumulated stack data according to the stack format used when C function called.

Note the following when using the **SHOW CALLS** command:

- The command cannot be used in the programs coded in assembler.
- In the optimized program, the command may be unable to display data normally.
- If the program is not compiled with debug information, the command displays the address instead of the function name. However, if the program breaks at the beginning of the function, the command cannot display data normally.

"Example"

```
>SHOW CALLS
cheker (12, 8)
main (3, 4)
```

4.2 SET TRACE

	SIM	EML	MON
[8L Compact ICE]			
SET TRACE	—	○	—
[8L] SET TRACE	○	○	—

"Format"

- Command qualifiers

  Trace buffer-full break specified

  /BREAK

  Enables trace buffer-full break.

  /NOBREAK (default at start-up)

  Disables trace buffer-full break.

"Description"

The **SET TRACE** command clears trace memory to enable the trace function.

Enabling the trace buffer-full break, suspends program execution when the trace buffer becomes full.

"Example"

```
>SET TRACE/BREAK
```

4.3 SHOW TRACE

						SIM	EML	MON
(Format 1)								
[8L Compact ICE]								
	SHOW TRACE	[/DATA]	[trace-number	[.. trace-number]]		—	○	—
[8L]	SHOW TRACE	[/DATA]	[trace-number	[.. trace-number]]		○	◎	—
(Format 2)								
[8L Compact ICE]								
	SHOW TRACE	/STATUS				—	○	—
[8L]	SHOW TRACE	/STATUS				○	○	—

"Format"

- Command qualifiers classified by function

/STATUS

Displays trace measurement conditions, enabled/disabled state of trace function, and storage status of trace buffer.

This qualifiers is specified in Format 2.

/DATA (default when omitted)

Displays traced data.

This qualifiers is specified in Format 1.

- Parameters

trace-number (default: decimal number)

Specify the number of trace data to be displayed with decimal number.

- Command qualifiers

/CYCLE

Displays data in valid bus cycle. For the DSU3, this qualifier cannot be specified.

/INSTRUCTION (default when omitted)

Executes disassemble display. For the DSU3, this qualifier cannot be specified.

/SOURCE

Displays trace result in units of source lines. For the DSU3, this qualifier cannot be specified.

/ONEFRAME

Displays trace data only by one line.

/NEXT

Displays from the frame which next level of sequencer being traced.

"Description"

The **SHOW TRACE** command displays the trace data stored in the trace buffer.

Sampled trace data is assigned numbers. Trace data in the execution stop location (trigger point) is assigned number 0. The sampled trace data is assigned negative numbers until the execution stop location is reached. These numbers are called frame numbers.

When **/INSTRUCTION** is specified, the command executes disassemble display according to the sampled trace data.

When **/SOURCE** is specified, the command displays source lines according to the sampled trace data.

When trace-number is omitted, the command starts trace data display from the oldest trace data or the trace data of the trace number next to the previously-displayed last trace number.

When only the display start trace number is specified, the command displays 12 trace data, starting from the display start trace number.

When the display start trace number is less than the trace number of the oldest trace data, the command starts trace data displays from the oldest trace data.

When **/STATUS** is specified, the command displays the current trace status.

"Example"

```
SHOW TRACE/SOURCE -65
frame no.  source
-00065 :   demo3.c$489                  if (sy->str[0] == ab1)
-00059 :   demo3.c$491          }
-00055 :   demo3.c$487          for ( i = 0 ; i < 12 ; i++ ) {
-00052 :   demo3.c$492          ackdat += 5;
-00047 :   demo3.c$493          nckdat = ackdat;
-00043 :   demo3.c$494          return (ab1);
-00042 :   demo3.c$495 }
-00038 :   demo3.c$464                  if (rc != 0)
-00035 :   demo3.c$465                          sy->dat1 = 0x21;
-00031 :   demo3.c$467          }
-00027 :   demo3.c$460          for ( i = 0 ; i < NUM ; i++) {
-00024 :   demo3.c$468          return (0);
```

4.4 CLEAR TRACE

	SIM	EML	MON
[8L Compact ICE]			
CLEAR TRACE	—	○	—
[8L] CLEAR TRACE	○	◎	—

"Description"

The **CLEAR TRACE** command clears the trace buffer.

"Example"

```
>CLEAR TRACE
```

4.5 ENABLE TRACE

	SIM	EML	MON
[8L Compact ICE]			
ENABLE TRACE	—	○	—
[8L] ENABLE TRACE	○	◎	—

“Description”

The **ENABLE TRACE** command enables the trace function.

“Example”

>ENABLE TRACE

4.6 DISABLE TRACE

	SIM	EML	MON
[8L Compact ICE]			
DISABLE TRACE	—	○	—
[8L] DISABLE TRACE	○	◎	—

"Description"

The **DISABLE TRACE** command disables the trace function.

However, for the DSU3 chip, this command cannot specify the trace function disabled in the internal-trace mode and external-trace mode.

"Example"

```
>DISABLE TRACE
```

4.7 SEARCH TRACE

	SIM	EML	MON
(Format 1)			
[8L Compact ICE]			
SEARCH TRACE [address [& = mask-data]] [, f = search-start number]	—	○	—
[8L] SEARCH TRACE [address [& = mask-data]] [, f = search-start number]	○	◎	—
(Format 2)			
[8L Compact ICE]			
SEARCH TRACE [d = data [& = mask-data]] [, f = search-start number]	—	○	—
[8L] SEARCH TRACE [d = data [& = mask-data]] [, f = search-start number]	○	◎	—

"Format"

- Parameters

address (address formula)

Specify the address to be searched.

data (data formula)

Specify the data transmission register access data to be searched.

This parameter is valid only when the debugger type is an emulator debugger.

mask-data (data formula)

Specify the masking and searching of address and data.

Only bits set to 1 are to be compared for search.

Search-start-number (default: decimal number)

Specify the search start frame number with decimal number

When this parameter is omitted, the command starts data search from the beginning of the trace buffer.

- Command qualifiers

/ALL (default when omitted)

Searches for all associated frames.

/ONEFRAME

Terminates trace data search when one frame found.

/CYCLE (default when omitted)

Searches for trace data in units of valid bus cycles.

/INSTRUCTION

If trace data cannot be rearranged in machine instruction execution units, it is searched for in machine cycles.

/LONG

Specifies handling of event condition data as data of 4-byte length.

/WORD

Specifies that event condition data to be treated as 2-byte data.

/BYTE

Specifies that event condition data to be treated as 1-byte data.

Data length specifying is valid only for Format 2.

When this specifying is omitted, event condition data is a data length.

/READ

Searches trace frame or step where read access made to specified address.

/WRITE

Searches trace frame or step where write access made to specified address.

/LEVEL (only for MB2140 ICE, except compact ICE)

Searches for point where level of sequencer changes.

If this qualifier is given, parameters other than the search start number cannot be specified.

"Description"

The **SEARCH TRACE** command searches for trace data according to the specified condition.

When the trace data matching the condition is found, the command displays it in the same format as the **SHOW TRACE** command.

When **/ONEFRAME** is specified, the debugger terminates this command when one frame is found.

"Example"

```
>SEARCH TRACE/INSTRUCTION 0xF0AE6
frame no.    address      mnemonic
-00010   :   000F0AE6     ENTER    #004
-00009   :   000F0AE8     LEAVE
-00008   :   000F0AEA     LD       @R15+, RP
-00007   :   000F0AEC     RET
-00006   :   000F0ADE     LEAVE
-00005   :   000F0AE0     LD       @R15+, RP
-00004   :   000F0AE2     RET
-00003   :   000F0ACE     LD       @(R14,-4), R4
-00002   :   000F0AD0     LEAVE
-00001   :   000F0AD2     LD       @R15+, RP
00000    :   000F0AD4     ADDSP    #4
```

Chapter 5 Memory/Register Operation Commands

5.1 EXAMINE

	SIM	EML	MON
[8L Compact ICE]			
EXAMINE FORMULA [, ...]	—	○	—
[8L] EXAMINE FORMULA [, ...]	◎	◎	—

"Format"

- Parameter

  expression (address formula)

  Specify the expression to be analyzed.

- Command qualifiers

  /BINARY

  Specifies that the value to be examined to be displayed as binary number.

  /OCTAL

  Specifies that value to be examined to be displayed as octal number.

  /DECIMAL

  Specifies that value to be examined to be displayed as decimal number.

  /HEXADECIMAL

  Specifies that value to be examined to be displayed as a hexadecimal number.

"Description"

The **EXAMINE** command analyzes the specified C expression and displays the result.
When a variable is specified, the command displays the data.
When a variable of structure or union type is specified, the command displays all the member values. When only an array name is specified, the command displays all the data of that array.
When the display base number of a command qualifier is omitted, the base number specified by the **SET RADIX** command is assumed.

“Example”

```
>EXAMINE strsym
strsym =     {
             a = H'20
             b = H'4A30
             c = H'3012
}
>EXAMINE strsym.a
strsym.a = H'20
>EXAMINE flags [0]
flags [0] = H'03
>EXAMINE flags
flags [0] = H'05
flags [1] = H'50
flags [2] = H'10
flags [3] = H'2A
>EXAMINE/DECIMAL count
count = D'12
>EXAMINE/HEXADECIMAL count
count = H'0C
>EXAMINE/DECIMAL fwork
fwork = 2.36S+1
```

5.2 ENTER

	SIM	EML	MON
[8L Compact ICE]			
ENTER variable = data	—	○	—
[8L] ENTER variable = data	◎	◎	—

"Format"

- Parameters

variable (address formula)

Specify the variable where data to be stored.

data (data formula)

Specify the data to be stored.

- Command qualifiers

/BYTE

Stores specified value in specified memory location as 1-byte data.

/WORD

Stores specified value in specified memory location as 2-byte data.

/LONG

Stores specified value in specified memory location as 4-byte data.

/SINGLE

Stores specified value in specified memory location as single-precision floating-point number.

/DOUBLE

Stores specified value in specified memory location as double-precision floating-point number.

"Description"

The **ENTER** command assigns the specified data to the specified variable.

Specifying the type of command qualifier enables data to be assigned at the specified size.

“Example”

```
>ENTER tmcnt = 10
>ENTER work = 6A5
>ENTER tmp = 1DF2BF
>ENTER fsymbol = F'10.55S+2
>ENTER/WORD work = 12345678
```

5.3 SET MEMORY

		SIM	EML	MON
[8L Compact ICE]				
	SET MEMORY [storage-address] = data [, ...]	—	○	—
[8L]	SET MEMORY [storage-address] = data [, ...]	◎	◎	—

"Format"

- Parameters

storage-address (address formula)

Specify the memory location where specified data to be stored.

data (data formula)

Specify the value to be stored.

- Command qualifiers

/BIT

Stores specified value in specified memory location as bit-length data.

/BYTE (default when omitted)

Stores specified value in specified memory location as 1-byte data.

/WORD

Stores specified value in specified memory location as 2-byte data.

/LONG

Stores specified value in specified memory location as 4-byte length.

/SINGLE

Stores specified value in specified memory location as single-precision floating-point number.

/DOUBLE

Stores specified value in specified memory location as double-precision floating-point number.

/STRING

Stores value specified in character string in specified memory location as ASCII code data.

"Description"

The **SET MEMORY** command stores the specified data in the specified memory location according to the type of the specified command qualifier.

When storage-address is omitted, the command stores the specified data in the memory location next to the memory location last accessed by the **SHOW MEMORY** or **SHOW MEMORY** commands. The type of the data to be stored is the same as that of the last accessed memory data.

When only a period (.) is specified in storage-address, the command stores the data in the memory location last accessed by the **SHOW MEMORY** or **SET MEMORY** commands. In this case, the type of the data to be stored is also the same as that of the last accessed memory data.

If the type of command qualifier is omitted, **/BYTE** is assumed.

"Example"

```
>SET MEMORY/BYTE 1000 = 10
>SET MEMORY/HALFWORD 1030 = 6A5
>SET MEMORY/WORD 1050 = 1DF2BF
>SET MEMORY/STRING 2000 = "ST"
>SET MEMORY . = 45
>SET MEMORY/BIT 8000:3 = 1
>SET MEMORY/SINGLE 2050 = F'10.55S+2
```

5.4 SHOW MEMORY

		SIM	EML	MON
[8L Compact ICE]				
	SHOW MEMORY [{address | address-range} [, ...]]	—	○	—
[8L]	SHOW MEMORY [{address | address-range} [, ...]]	◎	◎	—

"Format"

- Parameters

address (address formula)

Specify the address in memory location to be checked.

address-range (address formula)

Specify the memory area range to be checked.

- Command qualifiers

/BIT

Specifies that value to be checked to be displayed as 1-bit data.

/BYTE (default when omitted)

Specifies that value to be checked to be displayed as 1-byte data.

/WORD

Specifies that value to be checked to be displayed as 2-byte data.

/LONG

Specifies that value to be checked to be displayed as 4-byte data.

/SINGLE

Specifies that value to be checked to be displayed as single-precision floating-point number.

/DOUBLE

Specifies that value to be checked to be displayed as double-precision floating-point number.

/ASCII

Specifies that value to be checked to be displayed as ASCII characters.

/STRING

Specifies that value to be checked to be displayed as character string.

/BINARY

Specifies that value to be checked to be displayed as binary number.

/OCTAL

Specifies that value to be checked to be displayed as octal number.

/DECIMAL

Specifies that value to be checked to be displayed as decimal number.

/HEXADECIMAL

Specifies that value to be checked to be displayed as hexadecimal number.

"Description"

The **SHOW MEMORY** command displays data in the memory location, specified by address or address-range, according to the type of specified data. However, when /BIT is specified, address-range cannot be specified.

When address and address-range are omitted, the command displays data in the memory location next to the memory location last accessed by the **SHOW MEMORY** or **SET MEMORY** commands.

The type of the data to be displayed is the same as that of the last-accessed memory data.

When only a period (.) is specified, the command displays the data in the memory location last accessed by the **SHOW MEMORY** or **SET MEMORY** commands.

In this case, the type of data to be displayed is also the same as that of the last accessed memory data.

If the command qualifier type is omitted, **/BYTE** is assumed.

If the display base number of a command qualifier is omitted, the base number specified by the **SET RADIX** command is assumed.

"Example"

```
>SHOW MEMORY/DECIMAL 1000
00001000 = D'12
>SHOW MEMORY/BINARY 1000
00001000 = B'00001100
>SHOW MEMORY/HEXADECIMAL 1000..1001
00001000 = H'0C
00001001 = H'41
>SHOW MEMORY/HEXADECIMAL/HALFWORD 1000
00001000 = D'410C
>SHOW MEMORY/HEXADECIMAL/WORD 1000
00001000 = H'0030410C
>SHOW MEMORY/HEXADECIMAL 1000, 1020
00001000 = H'0C
00001020 = H'E3
>SHOW MEMORY/ASCII 1001
00001000 = 'A'
>SHOW MEMORY/SINGLE/DECIMAL 1030
00001030 = 2.36S+1
>SHOW MEMORY/BYTE 1000
00001000 = H'0C
>SHOW MEMORY .
00001000 = H'0C
>SHOW MEMORY
00001001 = H'41
```

5.5 SEARCH MEMORY

			SIM	EML	MON
[8L Compact ICE]					
	SEARCH MEMORY	address-range = data [, …] [, S = skip-byte-count]	—	○	—
[8L]	SEARCH MEMORY	address-range = data [, …] [, S = skip-byte-count]	◎	◎	—

"Format"

- Parameters

address-range (address formula)

Specify the memory area to be searched.

data (data formula)

Specify the data to be searched.

skip-byte-count (data formula)

Specify the number of bytes to be skipped.

H'1 to H'FFFF can be specified.

If this parameter is omitted, the data length is assumed.

- Command qualifiers

/BYTE (default when omitted)

Searches for specified data as byte-length data.

/WORD

Skips 2 bytes and searches for specified data as 2-byte data.

/LONG

Searches for specified data as 4-byte-length data.

/ASCII

Searches for specified data as ASCII character strings.

"Description"

The **SEARCH MEMORY** command searches the specified memory for the specified data and displays the address matching the data.

"Example"

```
>SEARCH MEMORY 2000..3000 = 88
found at = 00002050
found at = 00002577
found at = 00002BDF
```

5.6 SET REGISTER

		SIM	EML	MON
[8L Compact ICE]				
	SET REGISTER register-name = data	—	○	—
[8L]	SET REGISTER register-name = data	○	○	—

"Format"

- Parameters

register-name

Specify the name of register or flag to be modified.

data (data formula)

Specify the value to be set in specified register or flag.

"Description"

The **SET REGISTER** command sets the specified value in the specified register or flag.

"Example"

```
>SET REGISTER PC = 1000
>SET REGISTER C = 1
```

5.7 SHOW REGISTER

		SIM	EML	MON
[8L Compact ICE]				
	SHOW REGISTER [register-name]	—	○	—
[8L]	SHOW REGISTER [register-name]	○	○	—

"Format"

- Parameter

  register-name

  Specify the name of register or flag to be checked.

- Command qualifier

  /ALL (default when omitted)

  Displays values of all registers and flags.

"Description"

The **SHOW REGISTER** command displays the values of the specified register or flag in hexadecimal notation.

When not set, each flag in the CCR register displays "–". When set, it displays the flag name.

"Example"

```
>SHOW REGISTER PC
PC = 00FF0000
>SHOW REGISTER
R0 = 00000000    R1 = 00000000    R2 = 00000000    R3 = 00000000
R4 = 00000000    R5 = 00000000    R6 = 00000000    R7 = 00000000
R8 = 00000000    R9 = 00000000    R10 = 00000000   R11 = 00000000
R12 = 00000000   R13 = 00000000   R14 = 00000000   R15 = 0000FFC0
MDH = 00000000   MDL = 00000000   RP = 00000000    PS = FFFFFFFF
PC = 000FF000    USP = 0000E000   SSP = 0000FFC0   CCR = --SINZVC
SCR = ++T  ILM = 1F  TBR = 000FFC00
```

5.8 COMPARE

			SIM	EML	MON
[8L Compact ICE]					
	COMPARE	compare-origin-address-range, comparison-destination-address	—	○	—
[8L]	COMPARE	compare-origin-address-range, comparison-destination-address	◎	◎	—

"Format"

- Parameters

compare-origin-address-range (address formula)

Specify the memory area of compare origin.

comparison-destination-address (address formula)

Specify the comparison destination address.

"Description"

The **COMPARE** command compares memory data.

When no error is found as a result of the comparison, the **COMPARE** command displays "Not found".

When an error is found, the command displays (in hexadecimal notation) the memory location of the compare origin and the data to the left and the memory location of the comparison destination and the data to the right.

"Example"

```
>COMPARE 2000..3000, 4000
address     source   destination  address
00002050    35       10           00004050
00002051    40       00           00004051
```

5.9 FILL

		SIM	EML	MON
[8L Compact ICE]				
	FILL address-range = data [, ...]	—	○	—
[8L]	FILL address-range = data [, ...]	◎	◎	—

"Format"

- Parameters

address-range (address formula)

Specify the memory range to be filled with data.

data (data formula)

Specify the data filling specified memory area.

- Command qualifiers

/BYTE (default when omitted)

Specifies filling of memory area with 1-byte data.

/WORD

Specifies filling of memory area with 2-byte data.

/LONG

Specifies filling of memory area with 4-byte data.

/ASCII

Specifies filling of memory area with ASCII character string data.

"Description"

The **FILL** command fills the specified memory area with any data.

"Example"

```
>FILL 2000..2FFF = 23
```

5.10 MOVE

			SIM	EML	MON
[8L Compact ICE]					
	MOVE	transfer-source-address-range, transfer-destination-address	—	○	—
[8L]	MOVE	transfer-source-address-range, transfer-destination-address	◎	◎	—

"Format"

- Parameters

transfer-source-address-range (address formula)

Specify the memory area from where data transferred.

transfer-destination-address (address formula)

Specify the memory location to where data to be transferred.

"Description"

The **MOVE** command transfers data from the specified memory area to the specified transfer destination.

"Example"

```
>MOVE 2000..3000, 4000
```

5.11 DUMP

			SIM	EML	MON
[8L Compact ICE]					
	DUMP	[{starting-address | address-range}]	—	○	—
[8L]	DUMP	[{starting-address | address-range}]	◎	◎	—

"Format"

- Parameters

starting-address (address formula)

Specify the memory address where dump to be started.

address-range (address formula)

Specify the memory area range to be dumped.

- Command qualifiers

/BIT

Dumps data in bits.

/BYTE (default when omitted)

Dumps data in bytes.

/WORD

Dumps data in 2 bytes.

/LONG

Dumps data in 4 bytes.

"Description"

The **DUMP** command dumps data in the specified memory area.

When only start-address is specified, the **DUMP** command dumps the first 16 lines in the output window.

When no parameter is specified, the command starts dumping from the memory location next to the memory location last-displayed as a result of previous command execution.

“Example”

```
>DUMP   100..118
address   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ---ascii --
00000100 00  00  41  42  43  00  00  00  00  00  00  00  00  00  00  00 ..ABC.....
00000110 53  49  4D 55  4C 41  54  4F 52                          SIMULATOR
>
>DUMP/HALFWORD   100..118
address    +0     +2     +4     +6     +8     +A     +C     +E     ---ascii--
00000100   0000   4241   0043   0000   0000   0000   0000   0000   ..ABC.....
00000110   4953   554D   414C   4F54   0052                        SIMULATOR
>
>DUMP/BIT   5
address    :7   :6   :5   :4   :3   :2   :1   :0   HEX
00000005   0    1    1    0    1    0    1    0    6A
00000006   1    1    1    1    0    1    0    0    F4
       .                   .                   .
       .                   .                   .
```

5.12 COPY

		SIM	EML	MON
[8L Compact ICE]				
	COPY transfer-source-address-range	—	○	—
[8L]	COPY transfer-source-address-range	×	○	—

"Format"

- Parameter

  transfer-source address range

  Specify the transfer-source memory area.

"Description"

The **COPY** command copies data in user memory corresponding to a specified memory area to emulation memory corresponding to a specified memory area. The specified memory area must be mapped as emulation memory.

"Example"

```
>COPY  600000..60FFFF
```

5.13 VERIFY

	SIM	EML	MON
[8L Compact ICE]			
VERIFY collating address range	—	○	—
[8L] VERIFY collating address range	×	○	—

"Format"

- Parameters

collating address range

Specify the collating memory area.

"Description"

The **VERIFY** command collates data in user memory corresponding to a specified memory area with data in emulation memory corresponding to a specified memory area. If there is no difference as a result of collating, the system waits for completion of command execution. The collating area must be mapped as emulation memory.

"Description"

```
>VERIFY  600000..60FFFF
```

Chapter 6 Line Assemble and Disassemble Commands

6.1 ASSEMBLE

		SIM	EML	MON
[8L Compact ICE]				
	ASSEMBLE [starting-address] = assemble-character-string	—	○	—
[8L]	ASSEMBLE [starting-address] = assemble-character-string	○	○	—

"Format"

- Parameters

starting-address (address formula)

Specify a starting address of memory containing line-assembled codes.

assemble-character-string (character string)

Specify a character string to be line-assembled. Please the string enclose in double quotation marks ' " ' (character).

"Description"

The **ASSEMBLE** command line-assembles the entered mnemonic and operand, and stores the instruction code in the specified memory location.

When starting-address is omitted, the memory location containing the next address of the last executed instruction code is assumed.

"Example"

```
>ASSEMBLE 1000 = "RET"
>ASSEMBLE 1006 = "ADD #1, R1"
>DISASSEMBLE 1000
00001000    RET
00001002    LDI #0, R0
00001004    LDUB @R0, R1
00001006    ADD #1, R1
00001008    STB R1, @R0
0000100A
```

6.2 DISASSEMBLE

		SIM	EML	MON
[8L Compact ICE]				
	DISASSEMBLE [{starting-address | address-range}]	—	◎	—
[8L]	DISASSEMBLE [{starting-address | address-range}]	◎	◎	—

"Format"

- Parameters

starting-address (address formula)

Specify a starting address of memory to be disassembled.

address-range (address formula)

Specify a range of memory to be disassembled.

"Description"

The **DISASSEMBLE** command disassembles data in the specified memory location and displays it in the output window.

When only starting-address is specified, the command disassembles and displays data by 16 lines.

When only a period (.) is specified in starting-address or address-range, the command starts disassembled data display from the address indicated by the current program counter.

When starting-address and address-range are omitted, the command displays disassembled data by 16 lines, starting from the line next to the last displayed line.

When **/DISPLAY** is specified in the **SET SOURCE** command and the memory location corresponds to the source line, the **DISASSEMBLE** command also displays the source line.

"Example"

```
>DISASSEMBLE 1000..1002
00001000    9720      RET
00001002    C000      LDI:8    #0, R0
>DISASSEMBLE .
000FF000    1781      ST       RP, @-R15
000FF002    0F07      ENTER    #01C
000FF004    C010      LDI:8    #1, R0
000FF006    7FF0      STB      R0, @ (R14, -1)
                .
                .
                .
```

Chapter 7 Load and Save Commands

7.1 LOAD

					SIM	EML	MON
[8L Compact ICE]							
	LOAD	file-name	[, address]	[, file-offset [, byte-count]]	—	○	—
[8L]	LOAD	file-name	[, address]	[, file-offset [, byte-count]]	○	○	—

"Format"

- Parameters

file-name

Specify a name of file to be loaded.

The default extension depends on the command qualifier to be specified.

address (address formula)

Specify a memory location (address) where memory image file to be loaded.

This parameter is valid only when command qualifier **/BINARY** is specified.

Specifying other command qualifiers results in an error.

file-offset (data formula)

Specify an offset of read start data in specified file.

When file-offset is omitted, data is read from the beginning of the file.

This parameter is valid only when command qualifier **/BINARY** is specified.

Specifying other command qualifiers results in an error.

byte-count (data formula)

Specify a count of data to be read.

When byte-count is omitted, all data is read.

This parameter is valid only when command qualifier **/BINARY** is specified.

Specifying other command qualifiers results in an error.

- Command qualifiers

/OBJECT (default when omitted)

Loads load module file.

The default extension is "**.abs**".

/DEBUG

Loads only debug information from load module file.

The default extension is "**.abs**".

/BINARY

Loads binary format memory image file.

The default extension is "**.bin**".

Addressing cannot be omitted.

/COVERAGE (only for EML in 8L, except compact ICE)

Loads coverage data file.
The default extension is "**.cov**".

/ALIAS

Loads alias file (command alias definition, macro command definition).
The default extension is "**.lst**".

/AUTOMATIC (default when omitted)

For simulator debugger, automatically sets map area at loading.

/MANUAL

For simulator debugger, does not automatically set map area at loading.
A map area must be set by the **SET MAP** command.

/READ

For simulator debugger, sets ROM area for data segment as **/READ** attribute if **AUTOMATIC** qualifier valid.
If this qualifier is omitted, the **/READ/CODE** attribute will be set.

"Description"

The **LOAD** command loads the specified file.
This command can load the following files:

- Load module file
  Absolute-format object file created by linker.
- Memory image file
  Memory image file saved by SAVE command.
  (An address should be always specified to load the files.)
- Coverage data file
  Coverage data file saved by SAVE command.
- Alias file
  File containing command alias definition and macro command definition.
  (If a file extension is omitted, the default extension is added and the file is opened.)

"Example"

```
>LOAD debug
>LOAD/BINARY data.bin, FE0000
```

7.2 SAVE

		SIM	EML	MON
[8L Compact ICE]				
	SAVE file-name [, address-range]	—	○	—
[8L]	SAVE file-name [, address-range]	○	○	—

"Format"

- Parameters

file-name

Specify a name of file where memory data to be saved.

When the file name extension is omitted, any of the following extensions is added:

- "**.bin**" (valid when memory data saved in memory image)
- "**.cov**" (valid when coverage data saved)
- "**.lst**" (valid when command alias definition or macro command definition saved)

address-range (address formula)

Specify a memory area to be saved.

Address-range is valid only when command qualifier **/BINARY** is specified.

Specifying other command qualifiers results in an error.

- Command qualifiers

/BINARY (default when omitted)

Saves memory data to memory image file in binary format.

The default extension is "**.bin**".

Address-range specification cannot be omitted.

/COVERAGE (only for EML in 8L, except compact ICE)

Saves coverage data in all areas specified by **SET COVERAGE** command.

The default extension is "**.cov**".

Address range specification is invalid.

/ALIAS

Saves command alias definition and macro command definition to alias file.

The default extension is "**.lst**".

Address-range specification is invalid.

"Description"

When all command qualifiers are omitted or when **/BINARY** is specified, the **SAVE** command saves data in the specified memory to the memory image file (binary format of data only).

In this case, address-range specification cannot be omitted.

If **/COVERAGE** is specified, this command will save coverage measurement data in all areas specified by the **SET COVERAGE** command.

When /ALIAS is specified, the command saves command alias definition and macro command definition to the alias file.

"Example"

```
>SAVE memo.bin, 0..0fff
```

Chapter 8 Source File/Symbol Commands

8.1 LIST

		SIM	EML	MON
[8L Compact ICE]				
	LIST [{ [file-name] line-number [.. line-number] | address}]	—	◎	—
[8L]	LIST [{ [file-name] line-number [.. line-number] | address}]	◎	◎	—

"Format"

- Parameters

file-name

Specify a name of source file to be displayed.

When file-name is omitted, the previously-specified file name is assumed.

line-number

Specify a number of source line to be displayed.

"$" must always precede a line number.

When line numbers are delimited by "**..**", the source lines within the specified range are displayed.

address (address formula)

Specify an address (memory location) where code attribute stored.

Specify this parameter when displaying the source line corresponding to the address (memory location).

"Description"

The **LIST** command displays the source line corresponding to the specified line number.

When only a period (.) is specified in file-name, line-number, or address, the command displays source lines of the count of lines in the output window, starting from the source line corresponding to the current program counter.

If the value in the program counter is rewritten due to program execution when all parameters are omitted, the command starts source line display from the source line corresponding to the current program counter.

In other cases, the command displays 19 source lines, starting from the line next to the previously-displayed last line.

“Example”

```
>LIST PROGRAM.C$2..$3
2:          x = x+1 ;
3:          printf ("%d\n", x) ;
>LIST subdisp
30:   subdisp ( )
31:   {
32:         int i;
33:
34:         for (i = p; i > = 1; i--)
35:               printf ( "data [%d] = %d \n", i, data [i] );
36:
                    .
                    .
                    .
>LIST.
53:         switch (*s) {
54:           case '0' : z = " "; return (z) ;
55:           case '1' : z = "a"; return (z) ;
56:           case '2' : z = "b"; return (z) ;
                    .
                    .
                    .
```

8.2 SET PATH

		SIM	EML	MON
[8L Compact ICE]				
	SET PATH [source-search-directory-name [, ...]]	—	◎	—
[8L]	SET PATH [source-search-directory-name [, ...]]	◎	◎	—

"Format"

- Parameter

  source-search-directory-name

  Specify a directory for which source file to be searched.

- Command qualifier

  /APPEND

  Appends specified search directory to current setting.

"Description"

The **SET PATH** command specifies the directories used to search for the source file.
The command searches the specified directories for the source file in sequence from the left.
When source-search-directory-name is omitted, the current directory is assumed.

"Example"

```
>SET PATH A: \
```

8.3 SHOW PATH

	SIM	EML	MON
[8L Compact ICE]			
SHOW PATH	—	◎	—
[8L] SHOW PATH	◎	◎	—

“Description”

The **SHOW PATH** command displays currently-enabled source file search directories.

“Example”

```
>SHOW PATH
source file search path = a: \
```

8.4 SHOW SCOPE

	SIM	EML	MON
[8L Compact ICE]			
SHOW SCOPE	—	○	—
[8L] SHOW SCOPE	○	○	—

“Description”

The **SHOW SCOPE** command displays the module and function names including the memory location indicated by the current program counter.

“Example”

```
>SHOW SCOPE
current scope = SIEVE\sub_main\
```

8.5 UP

	SIM	EML	MON
[8L Compact ICE]			
UP	—	○	—
[8L] UP	○	○	—

“Description”

The **UP** command moves the scope to the parent function.

UP/DOWN information is cleared when the MCU is executed, RESET is performed, or the program counter is updated.

“Example”

```
>UP
Current Scope = demo\sort\
```

8.6 DOWN

	SIM	EML	MON
[8L Compact ICE]			
DOWN	—	○	—
[8L] DOWN	○	○	—

"Description"

The **DOWN** command moves the scope to the child function.

UP/DOWN information is cleared when the MCU is executed, RESET is performed, or the program counter is updated.

"Example"

```
>DOWN
Current Scope = demo\check\
```

Chapter 9 Command Procedure Commands

9.1 BATCH

		SIM	EML	MON
[8L Compact ICE]				
	BATCH file-name [, actual-parameter [, ...]]	—	◎	—
[8L]	BATCH file-name [, actual-parameter [, ...]]	◎	◎	—

"Format"

- Parameters

file-name

Specify a name of file where command procedure to be executed written.
The default extension is "**.prc**".

actual-parameter

Specify an actual parameter required for command procedure.

- Command qualifier

/ICON

Converts debugger to icon and executes it when command procedure executed.
When command procedure execution terminates, the icon is restored to the original size.

"Description"

The **BATCH** command executes the commands in the specified command procedure file.
Batch processing (procedure file call) can be nested for up to 8 levels.
Actual parameters are replaced with temporary parameters (%P0 to %P9) in the order they were specified.
When the count of temporary parameters is greater than that of the specified actual parameters, the remaining temporary parameters are replaced by empty strings.
When the count of temporary parameters is less than that of the specified actual parameters, the remaining parameters are ignored.
The count of the specified actual parameters can be referenced by means of %NP.

"Example"

```
>BATCH TST.PRC, 0, 0FFF, BRK
```

9.2 QUIT

	SIM	EML	MON
[8L Compact ICE]			
QUIT	—	◎	—
[8L] QUIT	◎	◎	—

“Description”

Executing the **QUIT** command when the command procedure is being executed quits command procedure processing.

If the **QUIT** command is executed during execution of a control command, the program exits all the control command loops.

When this command is executed in the command wait status, nothing is executed.

“Example”

Data in command procedure file

```
IF %NP < 2
    QUIT
ENDIF
SET VARIABLE I = 0
SET VARIABLE ADDR = %P0
WHILE %I<%P1
    SET MEMORY %ADDR = %I
    SET VARIABLE I = %I+1
    IF %ADDR == H'FFFFFF
        QUIT
    ELSE
        SET VARIABLE ADDR = %ADDR+1
    ENDIF
ENDW
```

Chapter 10 Replacement Commands

10.1 SET ALIAS

		SIM	EML	MON
[8L Compact ICE]				
	SET ALIAS alias = command-character-string	—	◎	—
[8L]	SET ALIAS alias = command-character-string	◎	◎	—

"Format"

- Parameters

alias (identifier)

Specify a command alias.

command-character-string

Specify a command character string (command name, command qualifier, and parameter) to be replaced with specified alias, enclosed in double quotation marks (").

"Description"

The **SET ALIAS** command defines a command alias.

It is convenient to define command aliases for frequently-used commands.

No command alias can be nested.

Other command aliases cannot be included in command alias definition.

"Example"

```
>SET ALIAS BP = "SET BREAK 00FF0300,3"
>SET ALIAS E = "ENTER"
>SET ALIAS R = "SHOW REGISTER"
```

10.2 SHOW ALIAS

	SIM	EML	MON
[8L Compact ICE]			
SHOW ALIAS	—	◎	—
[8L] SHOW ALIAS	◎	◎	—

“Description”

The **SHOW ALIAS** command displays the defined command alias list.

“Example”

```
>SHOW ALIAS
T  : STEP
D  : EXAMINE
PC : SHOW REGISTER PC
>
```

10.3 CANCEL ALIAS

	SIM	EML	MON
[8L Compact ICE]			
CANCEL ALIAS [alias [,...]]	—	◎	—
[8L] CANCEL ALIAS [alias [,...]]	◎	◎	—

"Format"

- Parameter

  alias (identifier)

  Specify a command alias to be canceled.

- Command qualifier

  /ALL

  Cancels aliases of all command character strings.

"Description"

The **CANCEL ALIAS** command cancels the alias of the specified command character string.

"Example"

```
>CANCEL ALIAS BP
>
```

10.4 SET VARIABLE

			SIM	EML	MON
[8L Compact ICE]					
	SET VARIABLE	debug-variable-name = replacing-character-string	—	◎	—
[8L]	SET VARIABLE	debug-variable-name = replacing-character-string	◎	◎	—

"Format"

- Parameters

debug-variable-name (identifier)

Specify a debug variable to be defined.

replacing-character-string

Specify a character string replacing debug variable.

"Description"

The **SET VARIABLE** command defines a debug variable.

The defined debug variable can be used as part of the parameter field when the command is specified.

The used debug variable is replaced with the replacing character string defined by this command as is.

All the variables that can be specified in the parameter field can be defined.

For example, a character string and an expression can be defined as they are.

"Example"

```
>SET VARIABLE ADDR = 0309+12
>SET VARIABLE STR = "ABCDEF"
>SET MEMORY/STRING %ADDR = %STR
(can be replaced with SET MEMORY/STRING 0309+12 = "ABCDEF")
>SET VARIABLE CNT = 1
>WHILE %CNT<5
*PRINTF "val [%d] = %d\n", %CNT, %CNT
*SET VARIABLE CNT = %EVAL (%CNT+1)
(The %EVAL function is defined so that the CNT character string will not exceed the limit.)
*ENDW
```

10.5 SHOW VARIABLE

		SIM	EML	MON
[8L Compact ICE]				
	SHOW VARIABLE [debug-variable-name [,...]]	—	◎	—
[8L]	SHOW VARIABLE [debug-variable-name [,...]]	◎	◎	—

"Format"

- Parameter

  debug-variable-name (identifier)

  Specify a debug variable name to be displayed.

- Command qualifier

  /ALL (default when omitted)

  Displays all debug variables.

"Description"

The **SHOW VARIABLE** command displays the definition of the specified debug variable.

"Example"

```
>SET VARIABLE CNT = 1
>WHILE %CNT<5
*SHOW VARIABLE CNT
*PRINTF "CNT = %d\n", %CNT
*SET VARIABLE CNT = %CNT+1
*ENDW
CNT : 1
CNT = 1
CNT : 1+1
CNT = 2
CNT : 1+1+1
CNT = 3
CNT : 1+1+1+1
CNT = 4
>
```

10.6 CANCEL VARIABLE

		SIM	EML	MON
[8L Compact ICE]				
	CANCEL VARIABLE [debug-variable-name [,...]]	—	◎	—
[8L]	CANCEL VARIABLE [debug-variable-name [,...]]	◎	◎	—

"Format"

- Parameter

  debug-variable-name (identifier)

  Specify a debug variable name to be canceled.

- Command qualifier

  /ALL

  Cancels all debug variables.

"Description"

The **CANCEL VARIABLE** command cancels the specified debug variable.

"Example"

```
>CANCEL VARIABLE CHKADR, X, Y
>
```

Chapter 11 Utility Commands

11.1 SET LOGGING

	SIM	EML	MON
[8L Compact ICE]			
SET LOGGING [file-name]	—	◎	—
[8L] SET LOGGING [file-name]	◎	◎	—

"Format"

- Parameter

  file-name

  Specify a log file name.
  The default extension is "**.LOG**".
  When file-name is omitted, the **DEBUG.LOG** file is used to log data.

- Command qualifiers

  /OPEN (default when omitted)

  Newly opens specified file.

  /APPEND

  Appends log data to end of specified file.

  /EXPANSION (default when omitted)

  Logs command list and its result.

  /COMMAND

  Logs only user-entered data.

"Description"

The **SET LOGGING** command opens the specified logging file and starts logging.
When command qualifier **/APPEND** is specified, data in the previous file is not lost.
The data to be logged can be selected.
Specifying command qualifier **/COMMAND** enables the entered command list to be used as the command procedure file because only the list is logged.

"Example"

```
>SET LOGGING filename.log
>
>SET LOGGING/COMMAND filename.log
```

11.2 SHOW LOGGING

	SIM	EML	MON
[8L Compact ICE]			
SHOW LOGGING	—	◎	—
[8L] SHOW LOGGING	◎	◎	—

“Description”

The **SHOW LOGGING** command displays the logging status.

“Example”

```
>SHOW LOGGING
en/dis        : ENABLE
logging file  : logfile.log
logging data  : EXPANSION
```

11.3 CANCEL LOGGING

	SIM	EML	MON
[8L Compact ICE]			
CANCEL LOGGING	—	◎	—
[8L] CANCEL LOGGING	◎	◎	—

“Description”

The **CANCEL LOGGING** command cancels the logging setup and closes the logging file.

“Example”

```
>CANCEL LOGGING
```

11.4 ENABLE LOGGING

	SIM	EML	MON
[8L Compact ICE]			
ENABLE LOGGING	—	◎	—
[8L] ENABLE LOGGING	◎	◎	—

“Description”

The **ENABLE LOGGING** command enables logging again.

“Example”

```
>ENABLE LOGGING
```

11.5 DISABLE LOGGING

	SIM	EML	MON
[8L Compact ICE]			
DISABLE LOGGING	—	◎	—
[8L] DISABLE LOGGING	◎	◎	—

"Description"

The **DISABLE LOGGING** command temporarily disables logging.

The **ENABLE LOGGING** command can be used to enable logging again.

"Example"

```
>DISABLE LOGGING
```

11.6 PRINTF

		SIM	EML	MON
[8L Compact ICE]				
	PRINTF "format-control-string" [, expression [, ...]]	—	◎	—
[8L]	PRINTF "format-control-string" [, expression [, ...]]	◎	◎	—

"Format"

- Parameters

format-control-string

Specify character strings to be displayed on screen and format for expression value display.
Enclose format specification in double quotation marks (" ").
"% [flag] [width] [.precision] [l] type"

%

Specify this parameter when displaying data according to format specification.
The **PRINTF** command displays characters that are not format specification after % as they are.

flag

Specify whether to right- or left-justify display, o (octal number) or 0 (hexadecimal number), and 0x, 0X output control.
When flag is omitted, the display is right-justified.
This parameter is invalid when the conversion display format is b or f.

- \- : Left-justification
- # : Adds 0, 0x, or 0X before numeric value 0 is added when the conversion display format is o. 0x is added when the format is x. 0X is added when the format is X.

width

Specify minimum count of digits of integer to be output.
When the conversion result is less than the specified count of digits, the remaining areas are padded with 0s.
To pad with 0s at right-justification, add 0 to the beginning and specify the digits count.
When the conversion display format is b or f, width is invalid.

precision

Specify minimum count of digits of integer to be output.
When the conversion result is less than the specified count of digits, the remaining areas are padded with 0s.
When the conversion display format is b or f, precision is invalid.

l

Specify whether to display the language expression value as the long, unsigned long type when the conversion display format is d, u, o, x, or X.
When l is omitted, the language expression value is assumed to be the short, unsigned short type.

type

Specify one of following conversion display formats:

- d : Signed decimal number
- u : Unsigned decimal number
- o : Unsigned octal number
- x : Unsigned hexadecimal number (Lower-case characters a to f represent 10 to 15, respectively.)
- X : Unsigned hexadecimal number (Upper-case characters A to F represent 10 to 15, respectively.)
- c : One character
- b : Unsigned binary number
- s : Character string (Only addressing is valid. The maximum number of characters is 128 bytes.)

expression

Specify the expression to be displayed.

"Description"

The **PRINTF** command displays the specified character string and the expression value of the specified format on the screen.

"Example"

```
>PRINTF "ABC = %d\n", datflg
ABC = 3
```

11.7 SET OUTPUT

	SIM	EML	MON
[8L Compact ICE]			
SET OUTPUT	—	◎	—
[8L] SET OUTPUT	◎	◎	—

"Format"

- Command qualifiers

/SOURCE (default when omitted)

Opens source window in mixed mode, even if no file.

/INSTRUCTION

Opens source window as disassembly window, even if no file.

"Description"

When the user program stops, the **SET OUTPUT** command opens the source window according to the debug information at the position indicated by the PC. In this case, the operation that is performed when no target source file can be found is set.

"Example"

>SET OUTPUT /SOURCE

11.8 SHOW OUTPUT

	SIM	EML	MON
[8L Compact ICE]			
SHOW OUTPUT	—	◎	—
[8L] SHOW OUTPUT	◎	◎	—

"Description"

The **SHOW OUTPUT** command shows the display mode set by the **SET OUTPUT** command.

"Example"

```
>SHOW OUTPUT
source mode:  source
```

Chapter 12 Task Debug Commands

12.1 SHOW OBJECT

	SIM	EML	MON
[8L Compact ICE]			
SHOW OBJECT [object-number]	○	○	○
[8L] SHOW OBJECT [object-number]	○	○	—

"Format"

- Parameter

  object-number (valid only for /TSK, /SEM, /FLG, /MBX, and /CYC)

  Specify an object number to be displayed.

  When object-number is omitted, the outlines of all the objects are displayed.

- Command qualifiers

  /TSK

  Displays task data.

  /SEM

  Displays semaphore data.

  /FLG

  Displays an event flag value.

  /MBX

  Displays mailbox data.

  /CYC

  Displays cyclic handler data.

  /TMRQ

  Displays data in timer queue.

"Description"

The **SHOW OBJECT** command displays specified object data.

"Example"

```
>SHOW OBJECT/TSK 1
[tskid]   0001      [tcbadr]   0051413c   [exinf]    00000001
[tskpri]  0001      [itskpri]  0001
[status]  TTS_RDY
[wupcnt]  0000      [suscnt]   0000       [tmocnt]   ffffffff
[stack]   0051661c  [stkarea]  0051656c..0051666b
```

Chapter 13 Control Commands

13.1 IF

	SIM	EML	MON
[8L Compact ICE]			
IF ~ ELSEIF ~ ELSE ~ ENDIF	—	◎	—
[8L] IF ~ ELSEIF ~ ELSE ~ ENDIF	◎	◎	—

"Format"

```
IF formula
    command-list
[ELSEIF formula
    command-list]
[ELSE
    command-list]
ENDIF
```

- Parameters

formula

Specify the execution condition formula of specified command list.

command-list

Specify the commands to be executed.

"Description"

When formula is evaluated as true, the command list immediately after **IF** is executed. When formula is evaluated as false, the command list after **ELSE** is executed.

If formula is false when **ELSE** is omitted, nothing is executed.

Only macros or batch can use the **IF** command.

"Example"

```
IF %R0 == 0
  print "OK!!"
else
  print "NG!!"
```

13.2 REPEAT

	SIM	EML	MON
[8L Compact ICE]			
REPEAT ~ UNTIL	—	◎	—
[8L] REPEAT ~ UNTIL	◎	◎	—

“Format”

REPEAT
command-list
UNTIL formula

- Parameters

command-list

Specify the commands to be executed.

formula

Specify the execution condition formula of specified command list.

“Description”

The **REPEAT** command evaluates the UNTIL formula after the command list specified by command-list has been executed. This command repeats execution of the command list while the formula is false.

Only macros or batch can use the **REPEAT** command.

“Example”

```
REPEAT
  STEP
UNTIL  %PC == main
```

13.3 WHILE

	SIM	EML	MON
[8L Compact ICE]			
WHILE ~ ENDW	—	◎	—
[8L] WHILE ~ ENDW	◎	◎	—

"Format"

```
WHILE formula
  command-list
ENDW
```

- Parameters

formula

Specify the execution condition formula of specified command list.

command-list

Specify the commands to be executed.

"Description"

When the specified formula is evaluated as true, the **WHILE** command repeats execution of the specified command list.

Only macros or batch can use the **WHILE** command.

"Example"

```
WHILE %PC != function
  STEP
ENDW
```

13.4 BREAK

	SIM	EML	MON
[8L Compact ICE]			
BREAK	—	◎	—
[8L] BREAK	◎	◎	—

“Description”

The **BREAK** command enables the program to exit the control structure.
This command is valid only in the **REPEAT** and **WHILE** command lists.
Only macros or batch can use the **BREAK** command.

“Example”

```
WHILE     1
if %PC == main
BREAK
ENDIF
STEP
ENDW
```

Chapter 14 Built-in Variables and Functions

14.1 %CALL

	SIM	EML	MON
[8L Compact ICE]			
%CALL	—	○	—
[8L] %CALL	○	○	—

"Description"

%CALL returns the return value for the last-executed **CALL** command. If the function return values are **void** and **double**, 0 is returned.

"Example"

```
>CALL  func(100,200)
return value is H'40
>ENTER val=%CALL+0x80
```

14.2 %ERRNUM

	SIM	EML	MON
[8L Compact ICE]			
%ERRNUM	—	◎	—
[8L] %ERRNUM	◎	◎	—

“Description”

%ERRNUM replaces the error number with the last error number executed from the Command Window.

0 indicates that there is no error.

“Example”

```
>PRINTF "ERROR NO. = %d\n", %ERRNUM
ERROR NO. = 5
```

14.3 %ENTRY

	SIM	EML	MON
[8L Compact ICE]			
%ENTRY	—	◎	—
[8L] %ENTRY	◎	◎	—

“Description”

%ENTRY replaces the execution starting address with the execution starting address of the load module being loaded.

0 indicates that there is no execution starting entry.

“Example”

```
>PRINTF "ENTRY = 0x%X\n", %ENTRY
ENTRY = 0x10000
```

14.4 %STKTOP

	SIM	EML	MON
[8L Compact ICE]			
%STKTOP	—	○	—
[8L] %STKTOP	○	○	—

“Description”

%STKTOP replaces the starting address with the starting address of the stack area of the load module being loaded.

0 indicates that there is no stack area.

“Example”

```
>PRINTF "STACK = 0x%X\n", %STKTOP
STACK = 0x80000
```

14.5 %RADIX

	SIM	EML	MON
[8L Compact ICE]			
%RADIX	—	◎	—
[8L] %RADIX	◎	◎	—

“Description”

%RADIX replaces the base number with the currently-set base number (BINARY, OCTAL, DECIMAL, or HEXADECIMAL).

“Example”

```
>PRINTF "base-number = "
>PRINTF %TOSTR (%RADIX)
base-number = HEXADECIMAL
```

14.6 %SCPADR

	SIM	EML	MON
[8L Compact ICE]			
%SCPADR	—	○	—
[8L] %SCPADR	○	○	—

“Description”

%SCPADR replaces the scope address with the current scope address.

“Example”

```
>PRINTF " scope = 0x%X\n", %SCPADR
scope = 0x18300
```

14.7 %LOADNUM

	SIM	EML	MON
[8L Compact ICE]			
%LOADNUM	—	○	—
[8L] %LOADNUM	○	○	—

“Description”

%LOADNUM replaces the size with the size of the last loaded binary file.

“Example”

```
>PRINTF "byte-count = %d\n", %LOADNUM
byte-count = 584
```

14.8 %BIT, %B, %W, %L, %S, %D

		SIM	EML	MON
[8L Compact ICE]				
	%BIT, %B, %W, %L, %S, %D	—	○	—
[8L]	%BIT, %B, %W, %L, %S, %D	◎	◎	—

“Format”

- Parameter

  address

  Specify the address from where memory data to be read.

“Description”

%BIT, **%B**, **%W**, **%L**, **%S**, or **%D** replaces the data with any of the following memory data read from the specified address:

%BIT : Bit data
%B : Byte data
%W : Word data
%L : Long word data
%S : Single-precision floating-point number data
%D : Double-precision floating-point number data

“Example”

```
>PRINTF "10000 = 0x%X\n", %W (10000)
10000 = 0xAABBAACC
```

14.9 %STRGET

		SIM	EML	MON
[8L Compact ICE]				
	%STRGET (character-string, character-position, character-count)	—	◎	—
[8L]	%STRGET (character-string, character-position, character-count)	◎	◎	—

"Format"

- Parameters

character-string

Specify a replacing character string.

character-position

Specify a character position where get processing to be started (character position relative to first character).

character-count

Specify a count of characters to be gotten.

"Description"

%STRGET replaces the character string in the specified count of characters, starting from the specified character position in the specified character string.

"Example"

```
>PRINTF %TOSTR (%STRGET ("abcdefghijklmn", 3, 4) )
cdef
```

14.10 %STRSTR

		SIM	EML	MON
[8L Compact ICE]				
	%STRSTR (character-string-1, character-string-2)	—	◎	—
[8L]	%STRSTR (character-string-1, character-string-2)	◎	◎	—

"Format"

- Parameters

character-string-1

Specify a character string including character string to be retrieved.

character-string-2

Specify a character string to be retrieved.

"Description"

$STRSTR checks whether character-string-1 includes character-string-2.

When character-string-1 includes character-string-2, **$STRSTR** replaces the character position number with the character position number in character-string-1.

When character-string-1 does not include character-string-2, **%STRSTR** replaces the character position number with 0.

"Example"

```
>PRINTF "%d\n", %STRSTR ("abcdefghijklmn", "fg")
6
```

14.11 %STRCMP

	SIM	EML	MON
[8L Compact ICE]			
%STRCMP (character-string-1, character-string-2)	—	◎	—
[8L] %STRCMP (character-string-1, character-string-2)	◎	◎	—

"Format"

- Parameter

  character-string-1, character-string-2

  Specify a character strings to be compared.

"Description"

%STRCMP compares character-string-1 with character-string-2.

When the character strings match, **%STRCMP** sets 0. When they do not match, **%STRCMP** sets 1.

"Example"

```
>PRINTF "%d\n", %STRCMP ("abcde", "fg")
1
>PRINTF "%d\n", %STRCMP ("abcde", "abcde")
0
```

14.12 %STRLEN

	SIM	EML	MON
[8L Compact ICE]			
%STRLEN (character-string)	—	◎	—
[8L] %STRLEN (character-string)	◎	◎	—

“Format”

- Parameter

  character-string

  Specify a replacing character string.

“Description”

%STRLEN replaces character string with the count of characters.

“Example”

```
>PRINTF "%d\n", %STRLEN ("abcde")
5
```

14.13 %STRCAT

		SIM	EML	MON
[8L Compact ICE]				
	%STRCAT (character-string-1, character-string-2)	—	◎	—
[8L]	%STRCAT (character-string-1, character-string-2)	◎	◎	—

"Format"

- Parameter

  character-string-1, character-string-2

  Specify a character strings to be linked.

"Description"

%STRCAT replaces character string with character string created by linking character-string-1 and character-string-2.

"Example"

```
>PRINTF %TOSTR (%STRCAT ("abcde", "fg") )
abcdefg
```

14.14 %SYMLEN

	SIM	EML	MON
[8L Compact ICE]			
%SYMLEN (symbol-name)	—	◎	—
[8L] %SYMLEN (symbol-name)	◎	◎	—

“Format”

- Parameter

  symbol-name

  Specify a symbol.

“Description”

%SYMLEN returns the size of a specified symbol.

“Example”

```
>PRINTF "%d\n", %SYMLEN ("abcde")
2
```

14.15 %TOVAL

	SIM	EML	MON
[8L Compact ICE]			
%TOVAL (character-string)	—	◎	—
[8L] %TOVAL (character-string)	◎	◎	—

"Format"

- Parameter

  character-string

  Specify a character string.

"Description"

%TOVAL deletes double quotation marks (") from both ends of the specified character string.

This function is used when the character string enclosed in double quotation marks is specified in a field where only parameters other than character strings can be written.

"Example"

```
>SET BREAK %TOVAL ("main")
```

14.16 %TOSTR

	SIM	EML	MON
[8L Compact ICE]			
%TOSTR (character-string)	—	◎	—
[8L] %TOSTR (character-string)	◎	◎	—

“Format”

- Parameter

  character-string

  All parameter types can be specified.

“Description”

%TOSTR encloses the specified character string in double quotation marks (").
This function is used when the specified character string is specified in a field where only character strings can be written as parameters.

“Example”

```
>PRINTF %TOSTR (main)
main
```

14.17 %EVAL

	SIM	EML	MON
[8L Compact ICE]			
%EVAL (expression)	—	◎	—
[8L] %EVAL (expression)	◎	◎	—

"Format"

- Parameter

expression

Specify an expression to be evaluated.

"Description"

%EVAL evaluates specified expression.

"Example"

```
>PRINTF "%d\n", %TOSTR (%EVAL (10+20+30) )
60
```

14.18 %BNUM

	SIM	EML	MON
[8L Compact ICE]			
%BNUM (address)	—	◎	—
[8L] %BNUM (address)	◎	◎	—

“Format”

- Parameter

address

Specify an address.

“Description”

%BNUM returns the number of the breakpoint corresponding to a specified address. If there is no corresponding breakpoint, 0 is returned.

“Example”

```
>PRINTF "%d\n", %BNUM(1000)
0
```

14.19 %DBNUM

	SIM	EML	MON
[8L Compact ICE]			
%DBNUM (address)	—	◎	—
[8L] %DBNUM (address)	◎	◎	—

"Format"

- Parameter

address

Specify an address.

"Description"

%DBNUM returns the number of the data breakpoint corresponding to a specified address. If there is no corresponding data breakpoint, 0 is returned.

"Example"

```
>PRINTF "%d\n", %DBNUM(1000)
0
```

Appendix A Manager-Related Messages

E4002W Insufficient memory.

"Explanation" System memory is insufficient.
"Operator response" Terminate another program and execute this program.

E4011W Registration not possible.

"Explanation" Data cannot be written to the system registry.
"Operator response" Terminate another program and execute this program.

E4012W Function call failed. Exefile is old.

"Explanation" The version of the program file does not correspond to that of the DLL file.
"Operator response" Install the latest version of ***Softune Workbench***.

E4013W Failed function call. DLLfile is old.

"Explanation" The version of the program file does not correspond to that of the DLL file.
"Operator response" Install the latest version of ***Softune Workbench***.

E4020W CPU information file version is different. Contains uninterpretable information.

"Explanation" The CPU information file is old and does not contain the required information.
"Operator response" Get the latest CPU information file.

E4021W Chip type in CPU information file is not applicable.

"Explanation" Information for a different CPU is specified.
"Operator response" Specify the correct CPU information file.

E4022W Please enter CPU infomation file.

"Explanation" The CPU information file cannot be found.
"Operator response" Enter the CPU information file directory.

E4023W Illegal tool option data. Default data is set.

"Explanation" The project file has illegal tool option data.
"Operator response" Reset the tool option data.

E4024W Invalid CPU infomation. Set default value.

"Explanation" The CPU information file has illegal data.
"Operator response" Get the latest CPU information file.

E4100W Access was denied.

"Explanation" The file cannot be accessed.
"Operator response" The file may be write- or read-disabled. Check the file attributes.

E4110W Too many open files.

"Explanation" The maximum number of files that can be opened is exceeded.
"Operator response" Close other files.

E4120W Directory does not exist.

"Explanation" The directory cannot be found.
"Operator response" Enter the correct directory name.

E4121W Drive is not ready.

"Explanation" The drive cannot be accessed.
"Operator response" Check the drive.

E4122W Path is invalid.

"Explanation" The directory cannot be found.
"Operator response" Enter the correct directory name.

E4123W Unable to create directory.

"Explanation" The directory cannot be created.
"Operator response" The directory may be write-disabled, or a file in the directory may be in use by another process.

E4124W Unable to delete directory.

"Explanation" The directory cannot be deleted.
"Operator response" The directory may be write-disabled, or a file in the directory may be in use by another process.

E4125W Destination disk is full.

"Explanation" The remaining capacity of the disk is insufficient.
"Operator response" Delete unnecessary files.

E4126W Could not be removed because it is the current directory.

"Explanation" An attempt was made to delete the current directory.
"Operator response" Move from the current directory to delete another directory.

E4127W This directory cannot be access.

"Explanation" Access to the directory is denied.
"Operator response" Permission to access the directory may be denied.

E4130W File cannot be open.

"Explanation" The file cannot be opened.
"Operator response" Permission to access the file or directory may be denied.

E4131W File cannot be close.

"Explanation" The file cannot be closed.
"Operator response" Permission to access the file or directory may be denied.

E4132W File cannot be read.

"Explanation" The file cannot be read.
"Operator response" Permission to access the file or directory may be denied.

E4133W File cannot be written.

"Explanation" The file cannot be written.
"Operator response" Permission to access the file or directory may be denied.

E4134W File cannot be create.

"Explanation" The file cannot be created.
"Operator response" Permission to access the file or directory may be denied.

E4135W File cannot be delete.

"Explanation" The file cannot be deleted.
"Operator response" Permission to access the file or directory may be denied.

E4136W File cannot be change name.

"Explanation" The file cannot be renamed.
"Operator response" Permission to access the file or directory may be denied.

E4137W File cannot be copied.

"Explanation" The file cannot be copied.
"Operator response" Permission to access the file or directory may be denied.

E4138W File not found.

"Explanation" The file cannot be found.
"Operator response" Check the file name.

E4140W File not found. Do you create this file?

"Explanation" The file cannot be found.
"Operator response" To create a new file, click the OK button.

E4142W A sharing violation occurred while accessing.

"Explanation" The same file is being used by another process.
"Operator response" Terminate the other program. In some rare cases, the file may remain in use even after the program is terminated. In this case, reboot Windows.

E4143W A locking violation occurred while accessing.

"Explanation" The same file is being used by another process.
"Operator response" Terminate the other program. In some rare cases, the file may remain in use even after the program is terminated. In this case, reboot windows.

E4200W The project file format is illegal.

"Explanation" The projection file cannot be read properly.
"Operator response" The projection file may be different from that for ***Softune Workbench*** or may be damaged. Create a new project file.

E4201W Project file cannot be opened – CPU type is different.

"Explanation" The projection file is different from that for the MCU.
"Operator response" Create a new project file for the MCU.

E4202W Unable to save project file.

"Explanation" An error occurred at writing to the project file.
"Operator response" The remaining disk capacity may be insufficient or the project file may be write-disabled.

E4204W Illegal CPU information of projectfile. Setting default value.

"Explanation" CPU information in the project file is illegal, and is substituted for the default.
"Operator response" Check the set value for CPU information in the project file.

E4205W Target file directory not found. Create a directory?

"Explanation" The target project file directory is not specified.
"Operator response" Click the OK button to create a directory.

E4206W List file directory not found. Create a directory?

"Explanation" The target list file directory is not specified.
"Operator response" Click the OK button to create a directory.

E4207W Object file directory not found. Create a directory?

"Explanation" The target object file directory is not specified.
"Operator response" Click the OK button to create a directory.

E4210W Please specify the project name.

"Explanation" The project name is not specified.
"Operator response" Enter the project name.

E4211W Please specify the project directory.

"Explanation" The project directory is not specified.
"Operator response" Enter the project directory name.

E4212W Please specify the target file name.

"Explanation" The target file name is not specified.
"Operator response" Enter the target file name.

E4213W Includes characters that cannot be designated. \ / : , ; * ? " " < > |

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4214W Includes characters that cannot be designated. , ; * ? " " < > |

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4215W Includes characters that cannot be designated. , ; * ? " " < > |

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4220W Please specify the target file name.

"Explanation" The target file name is not specified.
"Operator response" Enter the target file name.

E4221W Directory not found. Do you create this directory?

"Explanation" The directory is not specified.
"Operator response" Enter the directory name.

E4222W Unable to create directory.

"Explanation" The directory cannot be created.
"Operator response" The file may be write-disabled.

E4223W Changed target MCU. CPU information changed to default value.

"Explanation" When the target MCU is changed, the preset CPU information returns to the default.
"Operator response" Reset the CPU information.

E4224W Specify target MCU.

"Explanation" The target MCU is not specified.
"Operator response" Enter the target MCU name.

E4225W Specify project type.

"Explanation" The project type is not specified.
"Operator response" Specify the project type.

E4226W Includes characters that cannot be designated. , ; * ? " " < > |

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4227W Please specify Object File Directory.

"Explanation" The target object file directory is not specified.
"Operator response" Enter the directory name.

E4228W Please specify List File Directory.

"Explanation" The target list file directory is not specified.
"Operator response" Enter the directory name.

E4230W Double specification.

"Explanation" The same specification is already in use.
"Operator response" Change the specification.

E4232W Setup file is not registered. Registered automatically.

"Explanation" Starting the debugger requires a setup file. If a setup file is not specified, create it with the same name as that of the project file.
"Operator response" Use **[Project]-[Basic Set]-[Debugger]** to set the items required for the automatically-created setup file.

E4233W Available setup file is not registered. Registered automatically.

"Explanation" Starting the debugger requires a setup file. If a setup file is not specified, create it with the same name as that of the project file.
"Operator response" Use **[Project]-[Basic Set]-[Debugger]** to set the items required for the automatically-created setup file.

E4234W Please specify the title.

"Explanation" The title is not specified.
"Operator response" Specify the title.

E4240W Already a registered member.

"Explanation" The specified file is already saved in the project.
"Operator response" Check the file name.

E4241W This file name has already been registered.

"Explanation" The specified file is already saved in the project.
"Operator response" Check the file name.

E4242W File not found. Do you registered?

"Explanation" An attempt was made to save a non-existent file in the project.
"Operator response" If the file name is correct, save the file. An inquiry is made as to whether to create a new file when starting the editor.

E4243W Too many select files.

"Explanation" The count of selected files exceeds the maximum value.
"Operator response" Decrease the count of selected files.

E4301W Unable to create command line.

"Explanation" The option file to start the language tool cannot be created.
"Operator response" Check the access permission for the OPT subdirectory under the project directory, or the disk capacity.

E4302W Failed during start.

"Explanation" The tool cannot be started.
"Operator response" The tool name may be incorrect. Check the tool settings.

E4303W Command Line too long.

"Explanation" The command line is too long (max. 2048 characters).
"Operator response" Check the option parameters.

E4304W Failed during start editor.

"Explanation" The saved external editor cannot be started.
"Operator response" Check the executable file name of the editor.

E4305W Compiler/Assembler is started.

"Explanation" An attempt is made to close the project during tool start up.
"Operator response" Use the Suspend button to terminate the tool and close the project.

E4306W Make function is started.

"Explanation" An attempt is made to close the project during tool start up.
"Operator response" Use the Suspend button to terminate the tool and close the project.

E4307W Build function is started.

"Explanation" An attempt is made to close the project during tool start up.
"Operator response" Use the Suspend button to terminate the tool and close the project.

E4308W Include Dependencies is started.

"Explanation" An attempt is made to close the project during tool start up.
"Operator response" Use the Suspend button to terminate the tool and close the project.

E4309W Tool is started.

"Explanation" An attempt is made to close the project during tool start up.
"Operator response" Use the Suspend button to terminate the tool and close the project.

E4400W Setup file is read only. Setup information is not saved.

"Explanation" The setup file cannot be written.
"Operator response" Set the setup file to the write-enabled state.

E4420W Maximum of address is xxxx.

"Explanation" The address exceeds the maximum value.
"Operator response" Check the address specification.

E4421W The start address exceeds the end address.

"Explanation" The specified address range is incorrect.
"Operator response" Check the address range specification.

E4422W The designated address is already designated.

"Explanation" The specified address range has already been saved.
"Operator response" Check the address range.

E4601W Double specification.

"Explanation" The specified item has already been saved.
"Operator response" Check the specification contents.

E4603W Illegal tool option data.

"Explanation" The tool option data does not have the necessary data.
"Operator response" Open the Tool Option Check dialog and click the OK button. When the control data is displayed, input the necessary data.

E4604W There is no control data.

"Explanation" Unspecified control data is found.
"Operator response" Specify the control data.

E4605W Includes characters that cannot be designated. ! " " # $ % & ' ( ) - = ^ ~ \ | @ ` [{ ; + : *] } , < . > / ? [SP] [TAB]

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4606W Includes characters that cannot be designated. ! " " # $ % & ' ( ) - =^ ~ | @ ` [{ ; + *] } , < > / ?

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4607W Includes characters that cannot be designated. ! " " # $ % & '( ) - = ^ ~ \ | @ ` [{ ; + :] } , < . > / ? [SP] [TAB]

"Explanation" These characters cannot be used.
"Operator response" Change the name.

E4610W The range of the number of lines is 20-255.

"Explanation" The count of lines exceeds the limit.
"Operator response" Change the count of lines.

E4611W The range of the number of columns is 80-1023.

"Explanation" The count of lines exceeds the limit.
"Operator response" Change the count of lines.

E4612W The range of the number of columns is 70-1023.

"Explanation" The count of lines exceeds the limit.
"Operator response" Change the count of lines.

E4613W The range of the number of tabs is 0-32.

"Explanation" The count of lines exceeds the limit.
"Operator response" Change the count of lines.

E4614W Please specify the macro name.

"Explanation" The macro name is not specified.
"Operator response" Specify the macro name.

E4615W Please specify the include path.

"Explanation" The install path is not specified.
"Operator response" Specify the install path.

E4616W Already a registerd macro name. Do you change contents?

"Explanation" The specified macro name has already been saved.
"Operator response" To change the setting, click the OK button.

E4620W Please specify the start address.

"Explanation" The start address is not specified.
"Operator response" Specify the start address.

E4621W Please specify the end address.

"Explanation" The end address is not specified.
"Operator response" Specify the end address.

E4622W The start address is larger than the end address.

"Explanation" The address range is incorrect.
"Operator response" Specify the address range.

E4623W Please specify a correct start address.

"Explanation" The start address is incorrect.
"Operator response" Specify the correct start address.

E4624W Please specify a correct end address.

"Explanation" The end address is incorrect.
"Operator response" Specify the correct end address.

E4625W Please specify the ROM/RAM area name.

"Explanation" The ROM/RAM area name is not specified.
"Operator response" Specify the ROM/RAM area name.

E4626W Please specify the section name.

"Explanation" The section name is not specified.
"Operator response" Specify the section name.

E4627W Maximum of address is 0xFFFFFFFF.

"Explanation" The address exceeds the maximum value.
"Operator response" Check the address specification.

E4628W Maximum of address is 0xFFFFFF.

"Explanation" The address exceeds the maximum value.
"Operator response" Check the address specification.

E4629W Maximum of address is 0xFFFF.

"Explanation" The address exceeds the maximum value.
"Operator response" Check the address specification.

E4630W Cannot specify address over bank.

"Explanation" The specified address crosses several banks.
"Operator response" Specify an address within one bank.

E4631W Specify symbol name.

"Explanation" The symbol name is not specified.
"Operator response" Specify the symbol name.

E4632W Specify set value.

"Explanation" The set value is not specified.
"Operator response" Specify the set value.

E4635W This symbol name has already been registered. Change the setting?

"Explanation" The specified symbol name has already been saved.
"Operator response" To change the setting, click the OK button.

E4636W This ROM/RAM area name has already been registered. Change the setting?

"Explanation" The specified ROM/RAM area name has already been saved.
"Operator response" To change the setting, click the OK button.

E4637W This section name has already been registered. Change the setting?

"Explanation" The specified section name has already been saved.
"Operator response" To change the setting, click the OK button.

E4638W Address must be specified to leader section name.

"Explanation" The address is not specified in the leading section name.
"Operator response" Specify the address.

E4639W This section name has already been specified in another ROM/RAM area.

"Explanation" The specified ROM/RAM area name has already been saved.
"Operator response" Check the ROM/RAM area name.

E4640W Specify exact address.

"Explanation" The address specification is incorrect.
"Operator response" Specify the correct address.

E4701W Specified directory does not exist. Specify?

"Explanation" A non-existent directory is specified.
"Operator response" If there is no error, click the OK button.

E4702W Cannot specify multiple directories.

"Explanation" Only one directory can be specified.
"Operator response" Specify only one directory.

E4703W Illegal Environment Variable.

"Explanation" The set value is illegal.
"Operator response" Check the set value.

E4740W This executable file does not exist. Register in the list?

"Explanation" The file in the execution file name cannot be found.
"Operator response" Check the file name.

E4741W Title is not specified.

"Explanation" The title is not specified.
"Operator response" Specify the title.

E4742W Executable file is not specified.

"Explanation" An execution file name is not specified.
"Operator response" Specify an execution file name.

E4743W The registration count is maximum. You cannot register any more.

"Explanation" No more settings can be saved.
"Operator response" Delete unnecessary settings.

E4744W Syntax error. \r\r Illegal macro is specified.

"Explanation" An undefined option and macro description are found in the execution directory.
"Operator response" Check the syntax.

E4745W Title is too long.

"Explanation" The title is too long.
"Operator response" Shorten the title.

E4746W Execute file name is too long.

"Explanation" The execution file name is too long.
"Operator response" Shorten the file name.

E4747W Option too long.

"Explanation" The option string is too long.
"Operator response" Shorten the option string.

E4748W The executing directory too long.

"Explanation" The directory name is too long.
"Operator response" Shorten the directory name.

E4749W Directory not found. Create this directory?

"Explanation" The specified directory cannot be found.
"Operator response" If the directory is correct, click the OK button.

E4750W Already a registerd title. Do you change contents?

"Explanation" The specified title has already been saved.
"Operator response" To change the setting, click the OK button.

E4752W Start tool does not exist.

"Explanation" The tool to be started cannot be found.
"Operator response" Check the saved tool name and directory name.

E4760W The registered error syntax format cannot be converted.

"Explanation" The error message in the output window cannot be analyzed.
"Operator response" Check the setting in the syntax list in **[Setup]-[Set Error Jump]**.

E4761W Syntax error. Undefined Macro.

"Explanation" An undefined macro is specified.
"Operator response" Check the syntax.

E4762W Syntax error. Undefined separete of '%f', '%*'.

"Explanation" The delimiter indicating the end of %f and %* is not input.
"Operator response" The description of the macros, %f and %*, needs the delimiter to identify the end of %f and %*. The next character in the macro description is regarded as the delimiter.

E4763W Syntax error. Duplicate Macro syntax.

"Explanation" The macros, %f, %l, and %h, are duplicated.
"Operator response" Check the syntax.

E4764W Syntax error. Invalid '\' syntax .

"Explanation" \ is used for other than \t, \], and \\.
"Operator response" Check the syntax.

E4765W Syntax error. Invalid '%[]' syntax.

"Explanation" The description of the macro, %[], is illegal.
"Operator response" There may be no correspondence in []. Check the syntax.

E4766W Syntax error. Donot describe '%f'.

"Explanation" The macro, %f or %h, is not described.
"Operator response" Always describe %f or %h in the error jump setting syntax.

E4767W Syntax error. Invalid Macro into '%[...]'.

"Explanation" An illegal macro is described in the macro, %[].
"Operator response" Only the macro, %% or %] can be described in the macro, %[].

E4768W Already a registerd syntax. Do you change contents?

"Explanation" The same syntax has already been saved.
"Operator response" To change the contents, click the OK button.

E4769W Syntax not specified.

"Explanation" The syntax is not specified.
"Operator response" Specify the syntax.

E4771W Syntax too long.

"Explanation" The character string in the syntax is too long.
"Operator response" The maximum character string is 255 characters.

E4772W Comment too long.

"Explanation" The comment is too long.
"Operator response" The maximum comment is 255 characters.

E4773W The registration count is maximum. You cannot register any more.

"Explanation" The count of saved settings exceeds the maximum value.
"Operator response" Check unnecessary settings.

E4774W The same syntax has already been set in the SYSTEM. It cannot be changed.

"Explanation" The same syntax has already been set in the SYSTEM.
"Operator response" Syntax that has already been saved in the SYSTEM cannot be changed.

E4780W Title not specified.

"Explanation" The title is not specified.
"Operator response" Specify the title.

E4781W Execute filename not specified.

"Explanation" The execution file name is not specified.
"Operator response" Specify the execution file name.

E4782W Option not specified.

"Explanation" The option is not specified.
"Operator response" Specify the option.

E4783W Already a registerd title. Do you change contents?

"Explanation" The specified title has already been saved.
"Operator response" To change the setting, click the OK button.

E4784W Syntax error. Undefined Macro.

"Explanation" An undefined macro is specified.
"Operator response" Check the syntax.

E4785W Syntax error. Duplicate Macro syntax.

"Explanation" The macros, %f, %l, and %h, are duplicated.
"Operator response" Check the syntax.

E4786W Syntax error. Donot describe '%f'.

"Explanation" The macro, %f or %h, is not described.
"Operator response" Always describe %f or %h in the error jump setting syntax.

E4789W The registration count is maximum. You cannot register any more.

"Explanation" The count of saved settings exceeds the maximum value.
"Operator response" Delete unnecessary settings.

E4790W Editor in list not selected.

"Explanation" The editor to be operated is not specified.
"Operator response" Select the required editor from the editor list and operate it.

E4791W The standard editor cannot delete and change.

"Explanation" An attempt was made to delete or change the standard editor.
"Operator response" The standard editor is built into ***Softune Workbench***. It cannot be deleted or changed.

E4792W This executable file does not exist. Register in the list?

"Explanation" The specified execution file cannot be found.
"Operator response" If the execution file name or directory name has no error, save it as it is.

E4793W The valid editor cannot delete.

"Explanation" An attempt was made to delete the editor selected as the "editor to be used."
"Operator response" Change the "editor to be used" to another before deleting it.

E4794W Directory not found. Create this directory?

"Explanation" The specified director cannot be found.
"Operator response" To create a directory, click the OK button.

E4795W Title too long.

"Explanation" The title exceeds the maximum count of characters.
"Operator response" Shorten the title.

E4796W Execute file name too long.

"Explanation" The execution file name is too long.
"Operator response" Shorten the execution file name.

E4797W Option string too long.

"Explanation" The option string is too long.
"Operator response" Shorten the option string.

E4798W The executing directory too long.

"Explanation" The directory name is too long.
"Operator response" Shorten the directory name.

Appendix B Error Message for Debuggers

F9201S Invalid setup file (not found).

"Explanation" The specified setup file could not be found.
"Operator response" Check that the file specified in the startup option setup file specification exists.

F9202S Invalid command or parameter (in setup file).

"Explanation" An invalid command or parameter exists in the setup file.
"Operator response" Use the Setup Wizard to restart the Softune debugger.

F9203S Invalid machine program (execution error).

"Explanation" The machine program is already executed or it cannot be executed because the system resources are insufficient.
"Operator response" Check the execution state of the machine program. If the machine program is not executed, close the View Window or terminate another startup program.

F9401S Invalid emulation pod or MCU cable (unmatch or no-connected).

"Explanation" The emulation pod or the MCU cable is incorrect. Alternatively, the MCU cable is not connected correctly.
"Operator response" Turn off the emulator, then check the emulation pod and MCU cable. If the cable is not connected correctly, connect it correctly, then restart the Softune debugger.

F9402S Invalid emulator hardware monitor program (unmatch).

"Explanation" The monitor program loaded into the emulator is incorrect.
"Operator response" Start the loader program attached to this product to load the monitor program, then restart the Softune debugger. For details, refer to the ***Installation Manual***.

F9403S Emulator hardware error.

"Explanation" The emulator hardware cannot be normaly operated.
"Operator response" Check if the MCU performs normal operation. Reset and restart the main body of the emulator. If this error occurred frequently, the emulator hardware or MCU target system may be out of order.

F9404S Invalid emulator hardware monitor program version (old).

"Explanation" Invalid install commands or parameters are found in the install file.
"Operator response" Use the monitor loader program to load the monitor attached to this product. For details, refer to the ***Installation Manual***.

F9601S Invalid communication status (or cable connection).

"Explanation" The state of communication line is abnormal, or the cable is not connected correctly.
"Operator response" Check the state of communication line.

F9602S Invalid communication device name (or not specified).

"Explanation" The specified communication device name is incorrect.
"Operator response" Check the communication device name in install file.

F9603S Invalid INTERFACE (not specified in install file).

"Explanation" Invalid **INTERFACE** (not specified in install file).
"Operator response" Check the install file.

F9604S Cannot initialize "WINSOCK.DLL".

"Explanation" "WINSOCK.DLL" cannot be initialized.
"Operator response" "WINSOCK.DLL" should correspond to the LAN software to be used. Store "WINSOCK.DLL" to the Windows drectory or PATH-set directory, referring to the manual for LAN software.

F9901S Memory allocation error.

"Explanation" Sufficient memory for startup cannot be allocated.
"Operator response" Increase the host machine empty memory and then restart this program.

F9902S System error.

"Explanation" This program could not startup normally because of system error.
"Operator response" Restart the system and then restart this program.

F9903S A necessary DLL file was not found.

"Explanation" The required DDL file cannot be loaded.
"Operator response" Re-install ***Softune Workbench***.

F9904S The version of CPU information file is an old version.

"Explanation" The version of the CPU information file is old, so information cannot be set properly.
"Operator response" Update the CPU information file to a new version.

W1001S Invalid data value (underflow).

"Explanation" Data underflowed the specified precision.
"Operator response" Recheck the precision or data.

W1002S Invalid data value (overflow).

"Explanation" Data overflowed the specified precision.
"Operator response" Recheck the precision or data.

W1101S Invalid symbol (multiple).

"Explanation" Duplicate symbols are found.
"Operator response" If this message is output when the **LOAD** command is executed, recheck the source file corresponding to the load module.

W1102S Invalid code section or entry data (not found in load module).

"Explanation" The code section and input data are not in the loaded load module. The program counter (PC) is not set.
"Operator response" Set the program counter (PC) and then execute the program.

W1103S Command history buffer allocation error (in host memory).

"Explanation" Buffer memory for the command history cannot be allocated to an internal memory area in the host machine.
"Operator response" Expand the internal memory area in the host machine. If the Softune debugger is used as is, the command history function cannot be used.

W1104S Invalid address (mis-alignment).

"Explanation" In the FR family MCU, 16-bit data must be aligned on a 16-bit boundary and 32-bit data on a 32-bit boundary, respectively.
"Operator response" Review the specified address.

W1201S Invalid HELP command file (not found).

"Explanation" The **HELP** command file is not placed in a correct location.
"Operator response" Place the **HELP** command file in a correct location.

W1202S Loaded different series's file.

"Explanation" The load module file made with the tool of a series different from the chip specification of the installation file was loaded.
"Operator response" Do the reload after the file is confirmed when the specified load module file is not a file of the purpose.

W1401S Invalid timer (overflow).

"Explanation" The execution-time timer overflowed during program execution.
"Operator response" Shorten the measurement time.

W1402S Invalid performance measuring data (buffer full).

"Explanation" The buffer that stores performance measuring data became full during program execution. Performance is not subsequently measured.
"Operator response" Reduce the measurement count.

W1403S Invalid pass count (overflow).

"Explanation" The pass count overflowed.
"Operator response" Check the term in the expression, then re-enter the command.

W1404S User reset.

"Explanation" An user reset is specified in MCU during command execution.
"Operator response" Re-enter the **GO** command in Run menu.

W1901S The setup file is read-only. The change in setup information cannot be preserved.

"Explanation" The setup file is read-only. Changes to the setup information cannot be saved.
"Operator response" Remove the read-only attribute from the attributes for the setup file corresponding to the setup file name.

W1902S Invalid CPU information data.

"Explanation" Data in the CPU information file is invalid.
"Operator response" Obtain the latest CPU information file.

W1903S There is a possibility with an old version of DLL.

"Explanation" The version of the program does not match that of the DLL file.
"Operator response" Install the latest DLL file.

E4001S Command error.

"Explanation" The command or line assembler syntax is incorrect.
"Operator response" Check the syntax and parameters, then re-enter the command.

E4002S Command qualifier error.

"Explanation" The specified command qualifier is incorrect or it does not exist in the command.
"Operator response" Check the command qualifier, then re-enter the command.

E4003S Syntax error.

"Explanation" An error is found in the command or line assembler syntax.
"Operator response" Check the syntax and parameters and then re-enter.

E4004S Invalid parameter count (over limit).

"Explanation" The parameter count is too large.
"Operator response" Check the command syntax and then re-enter.

E4005S Invalid parameter omission.

"Explanation" A required parameter is omitted.
"Operator response" Check the command syntax and then re-enter the parameter.

E4006S Parameter error.

"Explanation" Illegal parameters are specified. The parameter name is illegal or parameters cannot be recognized as numeric values.
"Operator response" Check the command syntax or input radix and then re-enter.

E4007S Invalid operand.

"Explanation" There are invalid operands in the expression. Attempts were made to perform arithmetic operations using floating-point numbers. Arithmetic operations using floating-point numbers cannot be performed.
"Operator response" Check the operands in the statement and then re-enter.

E4008S Invalid operator.

"Explanation" There are invalid operators in the expression.
"Operator response" Check the operators in the expression and then re-enter.

E4009S Syntax error (operand not found).

"Explanation" The operand is not found in the polynomial operator in the expression.
"Operator response" Check the expression and then input the operand correctly.

E4010S Syntax error (' " or ' " not found).

"Explanation" ' " or ' " on the right side of ' "or ' " is not found in the expression.
"Operator response" Check the expression and then input quotation marks correctly.

E4011S Invalid nest level (over limit).

"Explanation" The nest level of ( ), *, and [ ] in the expression exceeds 16. Or, the nest level of the structure or union exceeds 16.
"Operator response" Check the expression.

E4012S Syntax error (dividing by zero).

"Explanation" Division by 0 is found in the expression.
"Operator response" Check the operand in the expression and then re-enter.

E4013S Invalid address specifying.

"Explanation" The ending address may be less than the starting address or the specified address range may extend over two or more areas.
"Operator response" Check the addresses, then re-enter the command.

E4014S Invalid bit pattern (over 0x1 to 0xff).

"Explanation" The value of the specified bit pattern is other than 0x1 to 0xff.
"Operator response" Check the bit pattern and then re-enter.

E4015S Invalid bit offset (over 0 to 31).

"Explanation" The specified bit offset is not 0 to 31.
"Operator response" Check the bit offset, then re-enter the command.

E4016S Invalid register or flag name (not found).

"Explanation" The specified register or flag name is not found.
"Operator response" Check the register or flag name and then re-enter.

E4017S Invalid symbol (not found).

"Explanation" The specified symbol is not found in the symbol table. Or, the specified symbol is a local variable and the symbol path name is not entered in the current scope.
"Operator response" Check whether the invalid symbol name is specified or whether the symbol data in the module to which the symbol belongs is entered in the symbol table, and then reenter. If the symbol data in the module to which the symbol belongs is entered in the symbol table, specify the data with the symbol path name assigned, or enter the symbol path name in the current scope.

E4018S Invalid command alias (not found).

"Explanation" The specified command alias does not exist.
"Operator response" Check the command alias, then re-enter the command.

E4019S Invalid line number (not found).

"Explanation" The specified line number is not found in the source file. Or, the load module file (line number data) corresponding to the source file is not loaded.
"Operator response" Check the source file and then reenter. Or, load the load module file corresponding to the source file.

E4020S Invalid starting display line number (over ending line number).

"Explanation" The source line start line number is larger than the display end line number.
"Operator response" Check the line number and then re-enter.

E4021S Invalid cycle count (0).

"Explanation" 0 was specified as the cycle count.
"Operator response" Check the cycle count, then re-enter the command.

E4022S Invalid break point number (not found).

"Explanation" The specified break point number is not found.
"Operator response" Check the break point number.

E4023S Invalid data break point number (not found).

"Explanation" The specified data break point number is not found.
"Operator response" Check the data break point number.

E4024S Invalid watch point number (not found).

"Explanation" The specified watch point number is not found.
"Operator response" Check the watch point number.

E4025S Invalid starting display trace number (over ending number).

"Explanation" The starting display trace number is larger than the display ending trace number.
"Operator response" Check the trace number, then re-enter.

E4026S Invalid format statement characters.

"Explanation" The specified format statement character string is incorrect.
"Operator response" Check the format statement character string, then re-enter the command.

E4027S Invalid symbol (not found) path name.

"Explanation" The specified symbol path name is not found.
"Operator response" Check the symbol path name and then re-enter.

E4028S Invalid function (not found, or argument error).

"Explanation" The specified function is not found. Or, the invalid argument of the function is specified.
"Operator response" Check the function or argument and then re-enter.

E4029S Invalid expression (used variable of structure or union type).

"Explanation" The structure or union variable cannot be used as the operand in the language expression.
"Operator response" Recheck the data format. Prefix the operator & to the variable.

E4030S Invalid address (not found).

"Explanation" The corresponding address is not found in the line number.
"Operator response" Recheck the line number.

E4031S Invalid automatic variable reference.

"Explanation" Attempts are made to reference the automatic variable out of the function in which the variable is set.
"Operator response" The automatic variable can be referenced only within the function in which the variable is set.

E4032S Invalid variable specifying.

"Explanation" The specified variable is not the member of the structure or union variable.
"Operator response" Check the structure or union member.

E4033S Floating point data format error.

"Explanation" The floating-point data format is illegal.
"Operator response" Recheck the floating-point data format.

E4034S Invalid macro command definition (not found).

"Explanation" The specified macro command name is not found.
"Operator response" Check the macro command name, then re-enter the command.

E4101S Invalid command list nest level (over 8).

"Explanation" The nesting level of command list of the command procedure, command macro, or break point exceeds 8.
"Operator response" Review the execution of the command.

E4102S Symbol definition error.

"Explanation" The free area allocated in host machine memory is insufficient to execute commands. This error occurs when too many device drivers are incorporated under the MS-DOS (PC) environment.
"Operator response" Expand the free area allocated in host machine memory, then restart this program.

E4103S OS command error.

"Explanation" An OS command cannot be executed. The command shell format is incorrect.
"Operator response" Start the command shell of correct format.

E4104S Invalid command shell (not found).

"Explanation" The command shell could not be found.
"Operator response" Review the environment variable, etc., so that the command shell can be started.

E4105S Invalid alias string.

"Explanation" The command alias includes an unregisterable character.
"Operator response" Review command alias registration, then re-enter the command.

E4106S Invalid macro command name (registered already).

"Explanation" The same macro command is already registered.
"Operator response" Review the macro command name, then re-enter the command.

E4107S Invalid memory map definition.

"Explanation" Memory mapping is too complex to define the area. When setting the memory area attributes, the areas with different attributes are excessive, causing the internal table to overflow.
"Operator response" Simplify the memory mapping.

E4108S Memory allocation error.

"Explanation" There is insufficient memory space for command execution by the host machine. This error occurs when there are too many device drivers in the PC operating environment (MS-DOS).
"Operator response" Increase the memory space in the host machine to restart this program.

E4109S Object loading error.

"Explanation" The object load destination goes beyond the address Hxffffff.
"Operator response" Check the object size and object load destination or the specified address.

E4110S Log file open error (already).

"Explanation" The log file is already open.
"Operator response" Close the current log file, then open a new log file.

E4111S Memory access error.

"Explanation" Attempts were made to access undefined memory. The address where access causing an error is made is displayed in the address part.
"Operator response" Check the memory mapping.

E4112S Invalid nest level of structure or union (over 16).

"Explanation" The debug data table could not be created in the host machine memory.
"Operator response" Increase the memory space in the host machine memory, then restart this program.

E4113S Debug data table creation error.

"Explanation" The debug data table cannot be created in memory of the host machine or in the directory specified in TMP in the install file.
"Operator response" Increase the memory space in the host machine and restart the program. Or, check the condition of the directory specified in TMP in the install file.

E4114S Logging control command error.

"Explanation" The log file was operated although it is not open.
"Operator response" Check that the log file is open.

E4115S Invalid alias name (registered already).

"Explanation" The same command alias is already registered.
"Operator response" Review the command alias, then re-enter the command.

E4116S Invalid alias name (not found).

"Explanation" The specified command alias does not exist.
"Operator response" Check the command alias, then re-enter the command.

E4117S Data type error.

"Explanation" The data type is unmatched.
"Operator response" Check the data type and then re-enter.

E4118S Invalid member name (not specified).

"Explanation" The structure or union name cannot be specified.
"Operator response" Specify the structure or union name together with the member name.

E4119S Break point and data break point setting error.

"Explanation" Breakpoints and data breakpoints cannot be set.
"Operator response" Check the setting of breakpoints and the number of breakpoints set.

E4120S CALL command error.

"Explanation" The **CALL** command is already executing; it cannot be nested.
"Operator response" Suspend the **CALL** command with a **CLEAR CALL** command. Alternatively, execute the **GO** or **STEP** command until the call operation terminates, then execute the **CALL** command.

E4121S Invalid function (at the top).

"Explanation" There is no higher-level function than this function or this function is called from a program other than a C program.
"Operator response" Check the current function.

E4122S Invalid function (at the bottom).

"Explanation" There is no lower-level function than this function or this function is called from a program other than a C program.
"Operator response" Check the current function.

E4123S Invalid coverage map (over-full).

"Explanation" The coverage area cannot be set any more.
"Operator response" Simplify coverage area specification.

E4124S Coverage area setting error.

"Explanation" The coverage area is not set.
"Operator response" Set the coverage area.

E4125S Invalid coverage area.

"Explanation" An area outside the coverage area was specified.
"Operator response" Check and specify the coverage area.

E4126S Invalid coverage file

"Explanation" A file other than the coverage file was specified.
"Operator response" Check file data.

E4127S Invalid debug data (not loaded).

"Explanation" The debug data file has not loaded.
"Operator response" Load the debug data file, then specify a coverage.

E4128S Mapping overlap.

"Explanation" The specified map area overlaps another area.
"Operator response" Check map specification, then re-enter the command.

E4129S Invalid address (mis-alignment).

"Explanation" In the FR family MCU, 16-bit data must be aligned on a 16-bit boundary and 32-bit data on a 32-bit boundary, respectively.
"Operator response" Review the specified address.

E4130S Cannot open current source window.

"Explanation" The source window that displays the current location could not be found in the set source search directory.
"Operator response" Set the directory containing the source file.

E4131S Cannot be used in current mode of debugger.

"Explanation" The current debugger type cannot be used. The functions that can be used depend on the type of the debugger.
"Operator response" Check the type of debugger.

E4132S Task debugging cannot be used.

"Explanation" The task debug function cannot be used if an object with that function is not loaded.
"Operator response" Load the object with the task debug function.

E4201S File access error.

"Explanation" The file cannot be accessed.
"Operator response" Check the condition of the disk in the host.

E4202S File close error.

"Explanation" The file cannot be closed.
"Operator response" Check the condition of the disk in the host.

E4203S File open error.

"Explanation" The file cannot be opened.
"Operator response" Check the file name or the condition of the disk in the host. Or, check the file and directory.

E4204S Data write error.

"Explanation" Data cannot be written to the file.
"Operator response" Check the condition of the disk in the host.

E4205S Invalid line number (not found).

"Explanation" The corresponding source line is not found at the specified address. Even if the corresponding source line is not found, the source line is displayed in the source window.
"Operator response" Or, load the load module with debug data.

E4206S Alias file load error.

"Explanation" The specified alias file cannot be loaded.
"Operator response" Check the alias file name or the disk state of the host machine. Alternatively, check the directory containing the alias file.

E4207S Alias file save error.

"Explanation" The specified alias file cannot be saved.
"Operator response" Check the condition of the disk in the host.

E4208S Invalid file format.

"Explanation" The format of the file to be loaded is illegal.
"Operator response" Check the file.

E4209S Open file read error.

"Explanation" An error occurred during reading of the opened file.
"Operator response" Check the file (drive) being read.

E4301S Invalid interrupt factor number.

"Explanation" The specified interrupt factor number does not exist.
"Operator response" Check the interrupt factor number and enter it once again (IRQ0 to IRQ47).

E4302S Invalid I/O buffer number.

"Explanation" The specified I/O buffer number does not exist.
"Operator response" The simulator provides 0 to 3 I/O buffers.

E4303S Invalid port.

"Explanation" An address was specified beyond the port address range.
"Operator response" A port address can be specified only in the MCU I/O area. Specify an address in the MCU I/O area.

E4304S Invalid output destination.

"Explanation" A data output destination, which is already in use as the data output destination, was specified.
"Operator response" Specify a data output destination not in use.

E4305S Invalid port simulation (over 16).

"Explanation" The count of specified ports exceeds 16.
"Operator response" Specify 16 ports or less.

E4306S Simulation memory allocation error.

"Explanation" Simulation memory cannot be allocated to an internal memory area in the host machine.
"Operator response" Expand the internal memory area in the host machine.

E4307S Invalid input data file.

"Explanation" The file name assigned to the input port is incorrect or the file does not exist.
"Operator response" Check the general format of the file.

E4401S Verify error.

"Explanation" A verify error occurred when data was being written to memory by a command.
"Operator response" Check that data was written to the I/O area where values change and that memory is mounted. Also check whether or not a memory error occurred.

E4402S Parity error (at emulation memory).

"Explanation" A parity error occurred at accessing to the emulation memory.
"Operator response" Reset the emulator body, then restart this. If the error occurs frequently, it may be an emulation memory malfunction.

E4403S Parity error (at debug memory).

"Explanation" A parity error occurred at accessing to the memory for emulator operation.
"Operator response" Reset the emulator body, then restart this. If the error occurs frequently, it may be a malfunction of the memory for emulator operation.

E4404S Command error (MCU is busy).

"Explanation" An not executable command was tried to execute during MCU execution.
"Operator response" Check the command.

E4405S Invalid event number (not found).

"Explanation" The specified event number ix not found.
"Operator response" Check the event number.

E4406S Invalid level number (not found).

"Explanation" The specified level number is not found.
"Operator response" Check the level number.

E4407S Command error (event mode violation).

"Explanation" A command was specified that violates the event mode.
"Operator response" Check the event mode setting with the **Debug Environment** in the Setup menu.

E4408S Invalid latch number (not found).

"Explanation" The specified latch number is not found.
"Operator response" Check the latch number.

E4409S Invalid supply voltage.

"Explanation" The supply voltage supplied from the user system is found abnormal.
"Operator response" Review the supply voltage of the user system.

E4410S Invalid system clock.

"Explanation" The system clock supplied from the user system is found abnormal.
"Operator response" Review the system clock of the user system.

E4411S MCU reset error.

"Explanation" The MCU reset cannot be executed normally.
"Operator response" The mode data and the reset vector read at reset may be an incorrect value. Set a correct value and retry this command. When this error is occurred if the mode data is read from the user memory, the user memory cannot be read. Therefore, map it in the emulation memory before executing the **Reset of MCU** command in the Run menu.

E4412S Invalid MCU.

"Explanation" Commands cannot be executed because MCU is not an operational state.
"Operator response"

1. Set the reset vector and the mode data, then execute the **Reset of MCU** command.
2. Release the SLEEP, STOP or HOLD state on the user system side, or set the reset vector and the mode data, then execute the **Reset of MCU** command. Note that the HOLD state cannot be released by the **Reset of MCU** command in the Run menu.
3. Check the execution result of the command.

E4413S Invalid jump level number.

"Explanation" The jump destination level number of the sequencer is incorrect.
"Operator response" Review the jump destination level number. The sequencer cannot jump to the same level as the level to be specified.

E4414S Command error (on internal ROM real-time mode).

"Explanation" The command cannot be executed because the MCU execution mode is native.
"Operator response" Change the MCU execution mode to debug.

E4415S Command error (user reset).

"Explanation" This command cannot be executed because user reset is specified. This error occurs even if user reset is already released.

"Operator response" Release user reset, execute the **Reset of MCU** command in the Run menu, then execute this command.

E4416S Abort command error.

"Explanation" The **ABORT** command cannot be executed due to the SLEEP or STOP state.

"Operator response" Release the SLEEP or STOP state.

E4417S Command error (hardware standby).

"Explanation" This command cannot be executed due to the hardware standby state. This error occurs even if the hardware standby state is already released.

"Operator response" Release the hardware standby state, execute the **Reset of MCU** command in the Run menu, then execute this command.

E4418S Command error (timer-mode violation).

"Explanation" When the timer mode is "timer", the **SHOW CYCLE** command and **CLEAR CYCLE** command cannot be executed. When the timer mode is "cycle", the **SHOW TIMER** command and **CLEAR TIMER** command cannot be executed.

"Operator response" Check the timer mode, then re-enter the command.

E4419S Invalid break point (not found).

"Explanation" The software break point became invalid because data in the address where the software break point is set was rewritten by program execution. Alternatively, the software break point remained in memory because an error occurred when the point was being reburied. In this case, data in the program being loaded and setting data at the software break point are not guaranteed.

"Operator response" Delete all software breaks, then review the program data. If some software breaks still remain in memory, reload the program.

E4420S Monitor hit stack-check function.

"Explanation" A stack-check exception occurred within the monitor at returning to the user-program.

"Operator response" Invalidate a stack-check function or increase usable stack area.

E4501S Verify error.

"Explanation" A verify error occurred when data was being written to memory by a command.

"Operator response" Check that data was written to the I/O area where values change and that memory is mounted. Also check whether or not a memory error occurred.

E4502S Illegal stack area.

"Explanation" The stack area used by the monitor debugger cannot be accessed.

"Operator response" Secure the correct stack area.

E4503S System call error (cannot execute).

"Explanation" In this state, a system call cannot be executed normally.

"Operator response" Execute a system call in the state in which system calls can be issued. Interrupts may be disabled.

E4504S Command not suported.

"Explanation" The associated function is not built in a target side.
"Operator response" Built the associated function in the target-side program.

E4601S Invalid communication status (or cable connection).

"Explanation" The communication line state is abnormal or the cable connection is incorrect.
"Operator response" Check the line connection state.

E4602S Communication: Parallel adapter not connected.

"Explanation" The parallel communication adapter is not connected.
"Operator response" Connect the parallel communication adapter to the MB2141 correctly, then re-execute this program.

E4603S Communication: Mismatch parallel adapter version.

"Explanation" Communication cannot be performed because the version of the parallel communication adapter is old.
"Operator response" Use the latest parallel communication adapter.

E4604S Communication: Cannot find host name.

"Explanation" The specified host name is not registered in the hosts file.
"Operator response" Please register the host name in the hosts file.
For details, refer to the Operation Manual [Setting LAN Interface].

E4605S Communication: Cannot find port number.

"Explanation" The port number of ICE is not defined in the services file.
"Operator response" Please register the port number in the services file.
For details, refer to the Operation Manual [Setting LAN Interface].

E4606S Communication: Cannot open device.

"Explanation" Abnormality is found in the specified device or not connected correctly.
"Operator response" Please confirm whether the specified device is correctly connected.

E4607S Communication: Time out.

"Explanation" Reception information on transmission information was not received within the fixed time.
"Operator response" Please confirm whether the specified device is correctly connected.

E4901S Not enough timer resource.

"Explanation" The timer resource of Windows cannot be used.
"Operator response" End other applications, then re-execute this command.

E4902S The key code cannot be defined.

"Explanation" The key code cannot be defined.
"Operator response" Define another key code.

Appendix C Execution Suspension Messages List

Break at *address* by breakpoint

"Explanation" This message is displayed when a break is caused by a software breakpoint. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by hardware breakpoint

"Explanation" This message is displayed when a break is caused by a hardware breakpoint (including breakpoint specified by **GO** command).
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by code event break (No. code-event-number)

"Explanation" This message is displayed when a break is caused by a code event.
Address indicates the address of the next instruction to be executed where execution was suspended.
Code-event-number indicates the number of the code event that caused the break.

Break at *address* by code event break (sequential)

"Explanation" This message is displayed when a sequential break is caused by code event 1 or 2.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by data event break (No. data-event-number)

"Explanation" This message is displayed when a break is caused by a data event.
Address indicates the address of the next instruction to be executed where execution was suspended.
Data-event-number indicates the number of the data event that caused the break.

Break at *address* by data event break (sequential)

"Explanation" This message is displayed when a sequential break is caused by data event 1 or 2.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by trace buffer full

"Explanation" This message is displayed when a break is caused by a trace buffer full.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by alignment error break (code)

"Explanation" This message is displayed when a break is caused by a code fetch alignment error.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by alignment error break (data)

"Explanation" This message is displayed when a break is caused by a data access alignment error.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by external trigger break

"Explanation" This message is displayed when a break is caused by the input of an external signal to the TRIG pin of the emulator.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by trace lost break

"Explanation" This message is displayed when a break is caused by the trace data loss.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by data break at access-address

"Explanation" This message is displayed when a break is caused by a data breakpoint.
Address indicates the address of the next instruction to be executed where execution was suspended.
Access-address indicates the address where the access that caused the break was made.

Break at *address* by guarded access access-type at access-address

"Explanation" This message is displayed when a break is caused by code fetch access to a code fetch inhibited area, read access to a read-inhibited area, or write access to a write-inhibited area.
There may be an error in the memory attribute or the program.
Address indicates the address of the next instruction to be executed where execution was suspended.
Access-type indicates the type of the access that caused the break.
Access-address indicates the address where the access that caused the break was made.

Break at *address* by dispatch task from task ID=<dispatch-source-task-ID> to task ID=<dispatch-destination-task-ID>

"Explanation" This message is displayed when a break is caused by task dispatch.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by system call <system-call-name> on task ID=<task-ID>

"Explanation" This message is displayed when a break is caused by a system call.
Address indicates the address of the next instruction to be executed where execution was suspended.
System-call-name indicates the name of the system call that caused the break.
Task-ID indicates the ID of the task that issued the system call.

Break at *address*

" " This message is displayed whe **ABORT** command on

Address indicates the address of the next instruction to be executed where execution was suspended.

address by output file overflow

"Explanation" This message is displayed when a break occurs because data could not be written to the data output file of an output port.
Check the data output file of the output port.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by stop abnormal action

"Explanation" This message is displayed when a break occurs because a non-executable instruction was added after a prefix instruction.
Check the program because it may be incorrect.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by invalid call termination

"Explanation" The **CALL** command is executed after a breakpoint is set in the address indicated by the current PC and the RP register is set so that control will return to the address. For this reason, a break occurs if the address of the original PC is executed during execution of the **CALL** command.
In this way, this message is displayed when a break occurs before execution of the **CALL** command is completed.
Restart execution of the **CALL** command with the **GO** command as is or suspend execution with the **CLEAR CALL** command.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by EIT

"Explanation" This message is displayed when a break is caused by EIT.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by step command

"Explanation" This message is displayed by the **SHOW STATUS** command when a break is caused by step (INTO) execution.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by call command

"Explanation" This message is displayed by the **SHOW STATUS** command when a break is caused by step (OVER) execution.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at *address* by unknown break factor

"Explanation" This message is displayed when a break factor is undefined.
Address indicates the address of the next instruction to be executed where

execution was suspended.