dsedb/dsidb Command Reference

Table of Contents

About This Manual

This is the Runtime Library Release 2.4 version of the *dsedb/dsidb Command Reference* manual.

It provides detailed information regarding dsedb and dsidb commands and options.

Changes Since Last Release

None

Related Documentation

The “dsnet Overview” document describes the dsnet package, which contains a debugger, driver, manager, and other components for controlling a target (DTL-T10000).

Note: the Developer Support Web site posts current developments regarding the Libraries and also provides notice of future documentation releases and upgrades.

Typographic Conventions

Certain Typographic Conventions are used throughout this manual to clarify the meaning of the text:

Convention	Meaning
`courier`	Indicates literal program code.
italic	Indicates names of arguments and structure members (in structure/function definitions only).
medium bold	Indicates data types and structure/function names (in structure/function definitions only).
blue	Indicates a hyperlink.

Developer Support

Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
In North America:	*In North America:*
Attn: Developer Tools Coordinator Sony Computer Entertainment America 919 East Hillsdale Blvd. Foster City, CA 94404, U.S.A. Tel: (650) 655-8000	E-mail: PS2_Support@playstation.sony.com Web: http://www.devnet.scea.com/ Developer Support Hotline: (650) 655-5566 (Call Monday through Friday, 8 a.m. to 5 p.m., PST/PDT)

Sony Computer Entertainment Europe (SCEE)

SCEE developer support is available to licensees in Europe only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
In Europe:	*In Europe:*
Attn: Production Coordinator Sony Computer Entertainment Europe 30 Golden Square London W1F 9LD, U.K. Tel: +44 (0) 20 7859-5000	E-mail: ps2_support@scee.net Web: https://www.ps2-pro.com/ Developer Support Hotline: +44 (0) 20 7859-5777 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT)

List of Commands

dsedb and dsidb commands are listed below. For more information about each command, see "Command Details."

Table 1

Command	Explanation	Remark
dt [-[abcdefhrqsuvw]*] [<tid>]	Display threads	E
ds [-v] [<sid>]	Display semaphores	E
intr	Display interrupt handler	E
dr [-<cpuid>] [-[hfxw]*] [<reg>]...	Display register	E
dr [<reg>]...	Display register	I
sr [<reg> [<val>]]...	Set register	
rload <fname>	Load register	
rsave <fname> [<reg>]...	Save register	
bload <fname> <adr>	Load binary data	
bsave <fname> <adr> <cnt>	Save binary data	
di [-m <mark>] [<adr> [<cnt>]]	Disassemble and display	
vdi [-<cpuid>] [-m <mark>] [<adr> [<cnt>]]	Disassemble and display VU microinstruction	E
as <adr> <inst>	Simple line-unit assembler	
list [<adr> [<cnt> [<back>]]]	Display source line	
bt [<cnt>]	Display backtrace	
bp [<adr>[,<cnt>]]...	Set breakpoint	
ub [all|<adr>]...	Cancel breakpoint	
be [<adr>]...	Enable breakpoints	
bd [<adr>]...	Disable breakpoints	
hbp [pc][<uskx>] [:<adr>[,<amsk>]]... hbp [da|dr|dw][<uskx>] [:<adr>[,<amsk>] =<val>[,<vmsk>]]...	Set hardware breakpoint	E
hbp [pc|da|dr|dw][<uk>] [:<adr>[,<amsk>]]	Set hardware breakpoint	I
hub [pc|da|dr|dw]...	Cancel hardware breakpoint	
pload <fname> [<args>]...	Load target program	E
mload <fname> [<args>]...	Load target program module	I
mlist [-m]	Display loaded modules list for target	I
sload [<fname>]	Load only program symbols	E
sload [-id <id>] [-b <base>] [<fname>]	Load only program module symbols	I
run [<fname> [<args>]...]	Execute target program	E
mstart [-d] [<fname> [<args>]...]	Execute target program module	I
cont [<cnt>]	Continue target program	

Command	Explanation	Remark
dt [-[abcdefhrqsuvw]*] [<tid>]	Display threads	E
ds [-v] [<sid>]	Display semaphores	E
intr	Display interrupt handler	E
step [<cnt>]	Step-execute target program	
next [<cnt>]	Next-execute target program	
lstep [<cnt>]	Step-execute in source-line units	
lnext [<cnt>]	Next-execute in source-line units	
until <adr>...	Continue execution according to temporary breakpoints	
break	Interrupt target program	
wait	Wait for target program to stop	
reset [-i] [<ebootp> [<ibootp>]	Reset target	
xgkt <fname> <cnt> [<off>]	Start XGKICK trace	E
dbgctl [vu0|vu1] [on|off]	Set debug mode	E
storeimage <fname> <bp> <bw> <psm> <x><y> <w> <h>	Store GS image data	E
bpfunc <adr>	Set breakpoint function	
set [all] [<var>[=<val>]]	Display and set option/user variables	
alias [-r] [<name> [<value>]]	Display, set, or delete alias	
subst [-r] [<pattern> <replace>...]	Display, set, or delete command input substitution	
source <fname>	Execute command from file	
![<cmd> [<args>]...]	Call shell	
cd [<directory>]	Change current directory	
printf "<fmt>" [<args>]...	Output formatted message	
record [[-a] <fname>]	Start and end recording of screen output contents	
repeat [-c <cnt>] [-i <sec>] [-while <expr>] <cmd>...	Repeatedly execute command	
show [log|status|history| dbconf |symbol|mdebug] ...	Display internal information	
help [<keyword>]...	Display help message	
quit	Quit dsedb	

*E: dsedb-specific

I: dsidb-specific

Line Input

When entering commands in dsedb/dsidb the following editing keys can be used besides the arrow keys:

Table 2

Editing key	Function
CTRL-P	Move to previous line
CTRL-N	Move to next line
CTRL-B	Move back to the previous character (left)
CTRL-F	Move forward to the next character (right)
CTRL-A	Move to the start of the current line
CTRL-E	Move to the end of the current line
BS	Delete the previous character
DEL	Delete the previous character
CTRL-D	Delete the current character
CTRL-X	Delete the entire current line
CTRL-U	Delete the entire current line
CTRL-K	Delete from the current character to the end of the current line
CTRL-L	Redisplay the current line
CTRL-C	Suspend input (with ^C echo)
CTRL-G	Suspend input (without ^G echo)
LF, CR	End input and execute current line as a command
TAB	Complete command name, filename, symbol name, etc.
CTRL-R	Search in reverse direction
CTRL-S	Search in forward direction
CTRL-Y	Insert the character string deleted by CTRL-K

Completion using TAB is only enabled when a TAB is entered at the end of the line. The following four cases are supported.

1. Strings with no spaces

   Attempts completion as a command name.
2. set command argument

   Attempts completion as an option name
3. help command argument

   Attempts completion as a help keyword.
4. Other command arguments

   Attempts completion as a filename if it contains a '/' or a '.'. Otherwise, attempts completion as a symbol.

During program execution, entering CTRL-C sends a break command to the target. Entering CTRL-G does not sent a break command, but simply returns control to dsedb. The RETURN key enables the previously executed command (such as display memory contents or perform step execution) to be reexecuted. At this time, the command (such as display memory contents) is reexecuted after incrementing the address.

Expressions

The following expressions (expr) can be used as arguments for commands that take generic values other than a filename.

Table 3

Expression	Description
hexadecimal number	0x can be omitted if the value does not duplicate a symbol
symbol	Symbol
<fname>:<lineno>	Source file name/address using line number
<fname>:<func>	Source file name/function names that overlap in static
.	Current address (dot address)
.<digit>	Numeric value (octal, decimal, or hexadecimal) in any of the formats used in the C programming language
$<reg>	Current register or user variable value
(expr)	Parentheses used as a formula
+expr	Explicit plus sign
-expr	Minus sign
~expr	Bit inversion
*expr	Address indirect specification (the specified destination address is always a word)
expr1[expr2]	Equivalent to *(expr1 + ((expr2) << 2))
!expr	Logical not
expr1 || expr2	Logical OR
expr1 && expr2	Logical AND
expr1 | expr2	Bit OR
expr1 ^ expr2	Bit XOR
expr1 & expr2	Bit AND
expr1 == expr2	1 if values equal, 0 otherwise
expr1 != expr2	1 if values not equal, 0 otherwise
expr1 < expr2	Unsigned compare (less than)
expr1 <= expr2	Unsigned compare (less than or equal)
expr1 > expr2	Unsigned compare (greater than)
expr1 >= expr2	Unsigned compare (greater than or equal)
expr1 >> expr2	Shift expr1 to the right by expr2 bits (both values are unsigned)
expr1 << expr2	Shift expr1 to the left by expr2 bits (both values are unsigned)
expr1 + expr2	Arithmetic addition of expr1 and expr2 (both values are unsigned)
expr1 - expr2	Arithmetic subtraction of expr1 and expr2 (both values are unsigned)

Expression	Description
expr1 * expr2	Arithmetic multiplication of expr1 and expr2 (both values are unsigned)
expr1 / expr2	Arithmetic division of expr1 and expr2 (both values are unsigned)
expr1 % expr2	Arithmetic remainder of expr1 and expr2 (both values are unsigned)

The priority order of parentheses and of unary and binary operators is the same as it is in the C programming language, as shown below.

Table 4

Parentheses, unary and binary operators	Priority order
() []	High
! + - ~ * (unary operators)	↑
* / %	↑
+ - (binary operators)	↑
<< >>	↑
< <= > >=	↑
== !=	↑
&	↑
^	↑
|	↑
&&	↑
||	Low

Unless otherwise specified, a numeric value within an expression (expr) is interpreted as a hexadecimal number. Exceptions are the "10" of $10 or a line number within a symbol.

The definition of a hexadecimal number is [0[xX]][0-9a-fA-F]+. No distinction is made between uppercase and lowercase letters. To specify a decimal number at a location where a hexadecimal number can be used, place a "." before the digits. To specify an octal number, prefix it with ".0". There should not be any space between "." and the digit.

Either of the following formats can be used for specifying symbols.

a. <symbol>
b. :<symbol>

A value such as "a123" can be interpreted either as a hexadecimal number or as a symbol. If the symbol "a123" actually exists, an error will occur. In this case, explicitly indicate either a hexadecimal number by prefixing the value with "0x", or prefixing it with ":" as in ":a123", if it is a symbol.

A register can be specified as a decimal number such as "$4", or as a symbolic expression such as "$a0".

A "." within an expression is treated as a special symbol. During the execution of various commands such as di or as, or commands for displaying or setting memory, the "." assumes the value of the "current address." After the command completes, the value of "." is set to the "next address."

The following are examples of expressions.

```
db 80100000
di main+0x100
di (main + 100 + *(1002 + 4) * 5)
di $epc-8
db $t0+0x124+$v0+buf
```

List of Options

The dsedb options are as follows:

Table 5

Option Name	Value Format	Meaning	Remark
target_name	Character string	Hostname and port of dsnetm	*
tty_mask	Numerical value	Target TTYP I/O setting	*
atty_mask	Numerical value	IOP/EE TTYP display settings	*
file_priority	Numerical value	Priority of Remote File Access	*
reset_on_start	Boolean	Whether to reset target on starting	*
check_manager_version	Boolean	Whether to check manager version	*
target_exec_ctl_config	Numerical value	Target execution control status	
target_exec_ctl_override	Numerical value	Target execution control overwrite	
dr_default_di	Character string	di execution content specification at dr command	
ex_default_dr	Character string	dr execution content specification at exception generation	
dr0_default_di	Character string	di execution content specification at dr -vu0 command	
ex0_default_dr	Character string	dr execution content specification at VU0 exception generation	
dr1_default_di	Character string	di execution content specification at dr -vu1 command	
ex1_default_dr	Character string	dr execution content specification at VU1 exception generation	
lstep_default_list	Character string	Execute list after lstep, lnext	
lstep_stop_at_no_line	Numerical value	Stop if no line number for lstep, lnext	
source_directories	Character string	source search directory list specification	
initial_ebootp	Character string	Initial value of boot parameter for EE	

Option Name	Value Format	Meaning	Remark
current_ebootp	Character string	Current value of boot parameter for EE	
initial_ibootp	Character string	Initial value of boot parameter for IOP	
current_ibootp	Character string	Current value of boot parameter for IOP	
automatic_prefix_breakpoint	Boolean	Automatic prefix placement to breakpoint number format	
describe_ub_all	Boolean	Perform a clear all only when all is specified with ub	
di_address	Numerical value	di command address display format	
di_instruction_word	Numerical value	di command instruction word display format	
di_branch_address	Numerical value	di command branch address display format	
di_macro	Numerical value	Macro instruction control for di/as commands	
help_lang	Character string	Help display language	
help_pager	Character string	Pager to be used when displaying help	
hex_radix	Boolean	Numerical value regarded as hex without 0x	
log_total_size	Numerical value	Max total bytes of transmitting/receiving log	
log_packet_size	Numerical value	Max bytes of transmitting/receiving log per packet	
histfile	Character string	History save file name	
histfilesize	Numerical value	Number of histories to be saved	
histsize	Numerical value	Upper limit of histories	
tty_max_size	Numerical value	Number of max pending bytes in TTY packet	

* Start-up specific

On startup, dsedb checks for the existence of ~/.dsedbrc and ./.dsedbrc, and if these files exist, it executes them as auto-execution-on-startup files in the order they are detected. startup-specific options can be set only by using an option statement within an autorun file.

They cannot be changed after start-up using the set command. The option statement has the same arguments as the set command.

Command Details

dt [-[abcdefhrqsuvw]*] [<tid>] (for dsedb)

Displays a list of threads. If <tid> is specified, a detailed display for that thread ID will be performed. If neither <tid>, state, nor the thread ready queue option is specified, dt -a is assumed.

Options that specify thread states.

Table 6

Option name	Description
-a	Displays threads of all states.
-e	Displays RUN state threads.
-r	Displays READY state threads.
-w	Displays WAIT state threads.
-s	Displays SUSPEND state threads.
-u	Displays WAIT-SUSPEND state threads.
-d	Displays DORMANT state threads.

This option is used when displaying the thread ready queue. If combined with an option that specifies a thread state, the option that specifies the thread state will take priority.

Table 7

Option name	Description
-q	Display threads in the thread ready queue. Threads are displayed in the order they entered the queue.

This option clears the number of times threads entered RUN state.

Table 8

Option name	Description
-c	Initialize count to 0, the number of times threads entered RUN state.

This option provides a detailed display of threads. Can be used in combination with the option that specifies a thread state or the thread ready queue option.

Table 9

Option name	Description
-v	Displays details of threads.

Specify this option to display the thread backtrace. The pseudo-variables $BT<n>,$SFA<n>,$SFS<n> of the bt command are set in the back trace displayed last.

Table 10

Option name	Description
-b	Displays the thread backtrace.

This option controls the display format for the detailed display.

Table 11

Option name	Description
-h	Displays 32-bit values as hexadecimal numbers.(fpr,facc)
-f	Displays 32-bit values as floating numbers. (fpr,facc) [default]

Displays the current register value when the thread is in RUN state. There may be a problem with displaying the current register values in an interrupt handler. This problem is scheduled to be fixed.

ds [-v] [<sid>] (for dsedb)

Displays a semaphore list. Specifying <sid> will display details for that semaphore ID. If <sid> is not specified, the entire list will be displayed. When details of a semaphore are displayed, the thread IDs of the threads in WAIT state for that semaphore will be displayed in the order in which they entered WAIT state.

Table 12

Option name	Description
-v	Displays details of semaphores.

intr (for dsedb)

Displays a list of interrupt handlers registered by the EE kernel API.

dr [-<cpuid>] [-[hfxw]*] [<reg>]... (for dsedb)

For <reg>, specify a register name or register group name. The following register group names can be specified. Register number format (for example, $a0 or $4) can also be used for general purpose registers.

Table 13

Register Group Name	Contents
gpr	General Purpose Registers
hls	HI, LO, HI1, LO1, SA
scr	System Co-processor Registers
pcr	Performance Counter Registers
hdr	Hardware Breakpoint Registers
fpr	Floating Point Registers
fpc	Floating Point Control Registers
vu0f	VU0 Floating Registers
vu0l	VU0 Integer Registers
vu1f	VU1 Floating Registers
vu1l	VU1 Integer Registers

Any of the following can be specified to the -<cpuid> option:

Table 14

Option name	Description
-cpu	Displays the CPU register when <reg> is omitted. [default]Displays the lower 32 bits of general purpose registers, $lo, $hi, $sa, $PC, $cause, $status, $badvaddr and $badpaddr,and also disassembles and displays the addresses before and after $PC.
-vu0	Displays the VU0 register when <reg> is omitted.
-vu1	Displays the VU1 register when <reg> is omitted. Displays VI00-VI16, TPC, FBRST and VPU-STAT registers, and also disassembles and displays the addresses before and after the TPC.$PC is a pseudo register that holds a program counter value that was calculated from the values of the four registers $epc, $errorepc, $status, and $cause.

With the -[hfxw]* option, display of the fpr, vu0f and vu1f register values can be specified.

Table 15

Option name	Description
-h	Displays 32-bit value in hex. (fpr,vu0f,vu1f) [default]
-f	Displays 32-bit value as a floating-point number.(fpr,vu0f,vu1f)
-x	Displays vu0f and vu1f in XYZW order. (vu0f,vu1f)
-w	Displays vu0f and vu1f in WZYX order. (vu0f,vu1f) [default]

dr [<reg>]... (for dsidb)

Specifies a register name or a register group name to <reg>. The following are register group names which can be registered:

Table 16

Register Group Name	Contents
gpr	General Purpose Registers
hl	HI,LO
scc	System Coprocessor Registers
c2r	coprocssor 2 (GTE) registers
c2c	coprocssor 2 (GTE) control registers
gte	all GTE registers (cop2,cp2)

For general purpose registers, the register number format (e.g. $a0 is $4) can also be used.

If <reg> is omitted, the general-purpose registers, $lo, $hi, $PC, $bada,and reverse assembly of a section around $PC are displayed.

$PC is a pseudo-register holding the program counter value calculated from the register values of $epc and $status.

sr [<reg> [<val>]]...

The value <val> is set for the register specified by <reg>.

rload <fname>

This command loads the register value from the specified file <fname>. The file name saved with the rsave command is usuallyspecified.

rsave <fname> [<reg>]...

This command saves the register name and value of the specified register <reg>... in the specified file <fname>

When <reg>... is omitted, the same register that is saved in the case of dr omission is saved.

dx [<adr> [<cnt>]]

Memory contents are displayed beginning with the specified address <adr>.

Table 17

Command name	Description
dq [<adr> [<cnt>]]	Display memory contents (quad)
dd [<adr> [<cnt>]]	Display memory contents (double)
dw [<adr> [<cnt>]]	Display memory contents (word)
dh [<adr> [<cnt>]]	Display memory contents (half)
db [<adr> [<cnt>]]	Display memory contents (byte)
df [<adr> [<cnt>]]	Display memory contents (float)

The default <adr> is the dot address (current address).

<cnt> indicates the number of bytes to be displayed at one time. If a fraction is specified with the dq command, for example, it is rounded up. If an I/O register that must be accessed in 128-bit units was accessed by a command other than dq, the value cannot be guaranteed.

sx <adr> <val>...

The specified data values <val>... are set in memory beginning with the specified address <adr>. There are six commands for the different data widths, as indicated below.

Table 18

Command name	Description
sq <adr> <val>...	Set memory contents (quad)
sd <adr> <val>...	Set memory contents (double)
sw <adr> <val>...	Set memory contents (word)
sh <adr> <val>...	Set memory contents (half)
sb <adr> <val>...	Set memory contents (byte)
sf <adr> <fval>...	Set memory contents (float)

The argument <fval> of the sf command is an expression that contains a floating point value. Only the following expressions are permitted.

```
<floating_point_number>
(<fexpr>)
+<fexpr>
-<fexpr>
*<fexpr>
<fexpr1> + <fexpr2>
<fexpr1> - <fexpr2>
<fexpr1> * <fexpr2>
<fexpr1> / <fexpr2>
```

ix [<adr>] [<cnt>]

The debugger inputs data from the device in the specified address <adr> and displays the results. There are five commands for the different data widths, as indicated below.

Table 19

Command name	Description
iq [<adr> [<cnt>]]	# Input from device (quad)
id [<adr> [<cnt>]]	# Input from device (double)
iw [<adr> [<cnt>]]	# Input from device (word)
ih [<adr> [<cnt>]]	# Input from device (half)
ib [<adr> [<cnt>]]	# Input from device (byte)

The default address of <adr> is the last-specified address.

<cnt> is the number of data. It becomes the last-specified number when omitted together with the address. It becomes 1 when the address is specified. Unlike dq/dd/dw/dh/db, the dot address is not updated.

ox <adr> <val>...

The debugger outputs the specified data <val>... to the devices after the specified address <adr>. There are five commands for the different data widths, as indicated below.

Table 20

Command name	Description
oq <adr> <val>...	# Output to device (quad)
od <adr> <val>...	# Output to device (double)
ow <adr> <val>...	# Output to device (word)
oh <adr> <val>...	# Output to device (half)
ob <adr> <val>...	# Output to device (byte)

Unlike sq/sd/sw/sh/sb, the dot address is not updated.

bload <fname> <adr>

All of the contents of the file named <fname> are loaded as binary data at the area specified by the starting address <adr>.

bsave <fname> <adr> <cnt>

The contents of the area having a size in bytes of <cnt> beginning at the address <adr> are saved as binary data in the file named <fname>.

di [-m <mark>] [<adr> [<cnt>]]

Instructions are disassembled and displayed beginning with the specified address <adr>. The default <adr> is the dot address (current address). <cnt> indicates the number of instructions to be displayed at one time. The default is 20 instructions.

If -m <mark> is specified, the marker "->" is displayed for the line corresponding to address <mark>.

vdi [-<cpuid>] [-m <mark>] [<adr>] [<cnt>]]

Every eight bytes of data are disassembled and displayed as VU micro instructions beginning with the specified address <adr>.

Any of the following can be specified to the -<cpuid> option:

Table 21

Option name	Function
-cpu	Sets CPU main memory space as the specified address space. [default]
-vu0	Sets uMEM0 space as the specified address space.
-vu1	Sets uMEM1 space as the specified address space.

The default <adr> is the dot address (current address).

<cnt> indicates the number of instructions to be displayed at one time. The default is 20 instructions.

If -m <mark> is specified, the marker "->" is displayed for the line corresponding to address <mark>.

as <adr> <inst>

The instruction <inst> is written at the specified address <adr>. For example, the following can be specified.

```
dsedb > as 10000 lw $v0,10($s0)
```

list [<adr> [<cnt> [<back>]]]

Source lines above and below the line corresponding to the <adr> address argument are displayed. The default value for <adr> is $PC.

The <cnt> line number argument indicates the number of source lines to be displayed. Default is 21 lines.

<back> is the difference between the first line displayed and the source line corresponding to the address argument. Default is 10 lines (display begins from 10 lines back).

Pressing RET by itself after executing the list command will display the same number of lines last displayed starting with the line after the last line that was displayed.

A "->" to the left of a source line indicates that the line corresponds to $PC.

To enable source line displays, create the executable using gcc -g.

bt [<cnt>]

Starting with the current program counter, go up the stack frame and display the program counter values in the stack frame.

If a stack frame count <cnt> is specified, display that number of values. If no count is specified, the entire stack frame will be displayed.

When a backtrace is displayed, the following pseudo-variables are set and can be referenced with the commands shown below.

Table 22

Psuedo- variable	Description
$BT<n>	backtrace address If <n> is 1 or more, the address of the function call instruction from the stack frame. If <n> is 0, the value of $PC.
$SFA<n>	stack frame address The smallest value of the <n>-th stack frame
$SFS<n>	stack frame size The smallest value of the <n>-th stack frame subtracted from the largest value.

For example, to set a breakpoint at the position back to the second stack frame position, enter bp $BT2+0x8. As shown below, the subst command can be used to define a dsf instruction that will dump the <n>-th stack frame as dw.

```
dsedb > subst "^dsf  *\([0-9][0-9]*\) *$" dw $SFA\1 $SFS\1
dsedb > dsf 2
```

bp [<adr>[,<cnt>]...

A breakpoint is set at the specified address <adr>. If the argument is omitted, all breakpoints that are currently set will be displayed.The count value <cnt> is the specified number of passing through the breakpoint. When it is omitted, 1 is set.

When a breakpoint is set, a corresponding breakpoint pseudo-variable $BP<n> is set to that address, and expressions using $BP<n> can be used for subsequent commands such as ub.

ub [all | <adr>]...

The breakpoint at the specified address <adr> is canceled. When the argument is omitted or all is specified, the debugger cancels all the breakpoints.

be [<adr>]...

This command validates the breakpoint of the specified address <adr>. When <adr> is omitted, all the breakpoints are validated.

bd [<adr>]...

This command invalidates the breakpoint of the specified address <adr>. When <adr> is omitted, all the breakpoints are invalidated.

hbp [pc][uskx]*[:<adr>[,<amsk>]]...
hbp [da|dr|dw][uskx]*[:<adr>[,<amsk>]=<val>[,<vmsk>]]... (for dsedb)

These commands set hardware breakpoints to the specified address <adr>.

pc, da, dr and dw are hardware breakpoint access types and have the following meanings:

Table 23

Access type name	Description
pc	program counter
da	data access
dr	data read access
dw	data write access

Only one setting can be made at at time for da, dr and dw.

<amsk> is a mask value of the specified address.

<uskx> is the CPU operation mode specification, and has the following meaning:

Table 24

u - user only
s - supervisor only
k - kernel only
x - EXL mode only
uk - user or kernel

uskx: all operation modes

The [pc|da|dr|dw] part cannot be omitted. When <adr> is omitted, the last address of the same type is set.

When <uskx> is omitted, the operation mode returns to the previous mode.

For da, dr and dw, the specified data value <val> and its mask value <vmsk> can also be specified.

When all the arguments are omitted, the current hardware breakpoint is displayed.

Specification example: With a data read to address 0x08000000

```
dsedb > hbp druskx: 0x08000000
```

hbp [pc|da|dr|dw][ku]*[:<adr>[,<amsk>]]... (for dsidb)

These commands set hardware breakpoints to the specified address <adr>. pc, da, dr and dw are hardware breakpoint access types and have the following meanings:

Table 25

Access type name	Description
pc	program counter
da	data access
dr	data read access
dw	data write access

Only one setting can be made at at time for da, dr and dw.

<amsk> is a mask value of the specified address.

<uk> is the specification of CPU operation mode, and has the following meaning:

Table 26

u - user only
k - kernel only
uk - user or kernel

The [pc|da|dr|dw] part cannot be omitted.

When <adr> is omitted, the last address of the same type is set.

When <uk> is omitted, the operation mode of the same type is set.

Specification example:

```
dsedb > hbp druk:0x08000000
```

hub [pc|da|dr|dw]...

This command cancels the hardware breakpoint of the specified type [pc|da|dr|dw]. When all the arguments are omitted, all the hardware breakpoints are cancelled.

pload <fname> [<args>]... (only for dsedb)

The specified executable file <fname> is loaded in the memory of the target, and information such as the starting address is maintained internally by the debugger. <args>... are passed as arguments to the target program during execution.

mload <fname> [<args>]... (only for dsidb)

The debugger loads the specified execution file <fname> into the memory of the target and stores information such as the start address inside the debugger. <args>... is passed to the target program as an argument during execution.

mlist [-m] (only for dsidb)

Displays a chart of the program module loaded to the target memory. Normally, only modules loaded with the mload/mstart commands are displayed. However, by using the -m option, all program modules existing in the memory can be displayed.

sload [<fname>] (for dsedb)

The symbol information of the specified executable file <fname> is added to the internal symbol information of the debugger. This command differs from pload in that no executable file is loaded into the target memory. If <fname> is omitted, all symbol information in the debugger will be cleared.

sload [-id <id>] [-b <base>] [<fname>] (for dsidb)

Adds symbol information from the specified executable file <fname> to the internal symbol information in the debugger. Unlike pload, the executable is not loaded into target memory.

If <fname> is omitted, all internal symbol information in the debugger is cleared. Combinations of the -id <id> option and the -b <base> option are used as follows.

sload <fname>

Queries the target for the module name and version in the module file <fname> and determines a module base address.

sload -id <id> <fname>

Uses the module base address based on the specified module ID <id>. Will result in an error if the module name and version in the module file <fname> do not match the information of the target.

sload -b <base> <fname>

Uses the module base address <base> without querying the target.

The -id option cannot be used at the same time as the -b option.

run [<fname> [<args>]...] (only for dsedb)

The program is executed. If pload has already been executed, this command can be executed without arguments.If pload has not been executed, specifying a filename for the argument will start execution of the target program.

Standard output from the target program will be displayed on the screen.

To suspend the program, enter CTRL-C. To return control to the debugger without suspending the program, enter CTRL-G.

Refer to the description of dsecons for details on TTYP handling during the execution of a program.

mstart [-d] [<fname> [<args>]...] (only for dsidb)

Calls the entry point of the program module. When mload is already executed, it can be called without an argument. When mload is not executed, specifying the file name to the argument results in calling the program module load and the entry point.

If a -d option is added, an entry point is called in debug mode. Specifically, it sets the breakpoint and calls the entry point and the TTYP output is displayed on a screen until returning from the program module. To suspend the target program, enter CTRL-C. To return the control to the debugger without stopping, enter CTRL-G.

If the -d option is not added, display the prompt immediately after booting the program module.

Refer to the description of dsecons for details on TTYP handling during the execution of a program.

cont [<cnt>]

Continuous execution of the target program is performed starting with the current state.

The debugger waits until a breakpoint is reached or some kind of exception is generated. The argument <cnt> specifies the continuation count. The default is 1. Standard output from the target program will be displayed on the screen.

To suspend the target program, enter CTRL-C. To return control to the debugger without suspending the target program, enter CTRL-G.

Refer to the description of dsecons for details on TTYP handling during the execution of a program.

step [<cnt>]

Step execution is performed the number of times specified by <cnt> (default is 1) beginning with the current $PC.

The step count matches the number of CPU instructions. Even if delayed branch instructions are included, step execution will be performed directly in the order that the CPU executes them.

next [<cnt>]

Next execution is performed the number of times specified by <cnt> (default is 1) beginning with the current $PC.

The next count matches the number of CPU instructions. Even if delayed branch instructions are included, next execution will be performed directly in the order that the CPU executes them.

The only difference from step execution is in whether or not execution will be traced into the target function call of the jal or jalr instruction.

lstep [<cnt>]

Step execution is performed for the specified number of lines <cnt>, from the current $PC (default value of <cnt> is 1).

If there is no line number for $PC during command execution or after an internal single-instruction execution, a message will be displayed and the command will be terminated. (Default)

If the lstep_stop_at_no_line option variable is set, errors will not be reported for locations that do not have line number information. Step execution will continue up to where internal line number information is available. To enable lstep, create the executable using gcc -g.

lnext [<cnt>]

Executes the next <cnt> number of lines starting with the current $PC (default value of <cnt> is 1).

If there is no line number for $PC during command execution or after an internal single-instruction execution, a message will be displayed and the command will be terminated. (Default)

If the lstep_stop_at_no_line option variable is set, errors will not be reported for locations that do not have line number information. Step execution will continue up to where internal line number information is available. To enable lstep, create the executable using gcc -g.

until <adr>...

Temporary breakpoints are used to perform continuous execution from the current $PC until the specified addresses <adr>... in the order they are specified. The until command does not affect breakpoints that were set by the bp command.

break

Program execution is suspended.

wait

The debugger waits for an exception to be generated while the program is being executed.

reset [-i] [<ebootp> [<ibootp>]

This command resets the whole target after invalidating all the breakpoints and clearing symbolinformation.

When the -i option is specified, the initial_[ei]bootp value is set to the current_[ei]bootp variable.When the boot parameter for the EE <ebootp> is specified, the value is set to the current_ebootp.

When the boot parameter for the IOP <ibootp> is specified, the value is set to the current_ibootp.The boot parameter for the target uses the current_[ei]bootp value.

xgkt <fname> <cnt> [<off>]

This command records the GIF packet data indicated by the XGKICK instruction into the specified file <fname> from the specified offset <off> for the specified number <cnt> by using the D-bit interrupt of the VU1. When <off> is omitted, the offset becomes 0.

To use this function in the current version, it is necessary to set the D-bit in the upper instruction of the XGKICK instruction in advance.

dbgctl [vu0|vu1] [on|off]

This command sets the D-bit interrupt of the VU0 and VU1.

Specify VU0 or VU1 in [vu0|vu1]. Then specify "on" to enable the D-bit interrupt, and specify "off" to disable it. When executing the xgkt command, "dbgctl vu1 on" is automatically executed.In the current version, to stop the microprogram by the D-bit , it is necessary to previously set the D-bit in the upper instruction of the instruction to be stopped.

storeimage <fname> <bp> <bw> <psm> <x> <y> <w> <h>

Reads GS image data using the specified parameters and saves to the specified file <fname>

<bp>

Base address of transfer buffer (actual address is bp x 64)

<bw>

Width of transfer buffer (actual width is bw x 64)

<psm>

Pixel format of transfer data

Table 27

Input number	Pixel format
0	PSMCT32 (pixel size:32bit)
1	PSMCT24 (pixel size:24bit)
2	PSMCT16 (pixel size:16bit)
10	PSMCT16S (pixel size:16bit)
19	PSMT8 (pixel size:8bit)
20	PSMT4 (pixel size:4bit)
27	PSMT8H (pixel size:8bit)
36	PSMT4HL (pixel size:4bit)
44	PSMT4HH (pixel size:4bit)
48	PSMZ32 (pixel size:32bit)
49	PSMZ24 (pixel size:24bit)
50	PSMZ16 (pixel size:16bit)
58	PSMZ16S (pixel size:16bit)

<x><y>

Upper left point of transfer area

<w><h>

Width and height of transfer area (in pixels)

The image data size (w x h x pixel size) must be a multiple of 16 bytes, and it must be less than 32767 X 16 bytes. Also, when the pixel size is 8 bits, x and w must both be multiples of 2. Similarly, when the pixel size is 4 bits, x and w must both be multiples of 4.

bpfunc <adr>

Sets function <adr> to be executed when the breakpoint is passed. When <adr> is 0, the setting is cleared.

set [all] [<var>[=<val>]]

To display only the options that have been changed from their default values, specify only set. To display all options, specify set all.

If an option has a string or numeric value as its value, to set the value of option name <var> to <val>, specify set <var>=<val>. If an option has a Boolean value, to set the option to the value true, specify set <var>. To set the option to the value false, specify set no<var>.

For the meaning of each option, see "4. dsedb Options."

The set command can also be used to display and set user variables. Names of user variables must not contain option names and must not be dsedb register names. User variables are always 32 bit numeric values and will always be displayed even if 'all' is not specified.

User variables are referenced using the $<var> notation within an expression.

If <val> is omitted when setting a user variable (e.g. set <var>=), the variable will be deleted.

alias [-r] [<name> [<value>]]

If the specified alias <name> has been specified as a command name, the command string <value> is executed.

To display all aliases that are currently set, specify only alias. To delete all aliases that are currently set, specify alias -r. To display only the setting of alias <name>, specify alias <name>. To set the new alias <name> to <value>, specify alias <name> <value>.

The specifications ${*} and ${N}, which are only available with this alias command, can be used within <value>. ${*} is expanded to all of the specified arguments at execution time, and ${N} is expanded up to the Nth specified argument at execution time.

To specify more than one command in <value>, use a construction such as alias dr2 (dr;dr).

subst [-r] [<pattern> <replace>...]

This command handles a command input line as a character string, and replaces all the patterns <pattern> included in the character string with a replacement character string <replace>...

To display all the currently-set replacements, specify the subst only. To delete all the currently-set replacements, specify the subst -r. To delete the currently-set replacement of <pattern> with <replace>..., specify the subst -r <pattern> <replace>... Unlike the alias, this command does not delete the previous setting even when the replacement to the same <pattern> has been set.

Also, when <replace>... is omitted, replacement to an empty character string (i.e. deletion) is set.In the case of two or more replacement definitions, all are applied in the order they have been specified.

The <pattern> is normally specified by the character string format surrounded by ' " '. When no space is included, however, the ' " ' surrounding the character string can be omitted.

The <pattern> is specified according to the formal expressions below.

Table 28

Character string	Description
X	Character X (when X is excluded from the expressions below.)
\X	Character X (when X is not '(',')'.)
.	One optional character
X*	Repetition of X (including 0-time repetition)
[abc]	One of 'a', 'b' and 'c' characters
[x-y]	One of characters from 'x' to 'y'
[^abc]	One character other than 'a','b' and 'c'
^	Start of line (corresponds only when it is at the start of <pattern>.)
$	End of line (corresponds only when it is at the end of <pattern>.)
\(	Start record of tag
\)	End record of tag

<replace>... can be specified with the following special characters as well as the ordinary character strings.

Table 29

Character string	Description
\X	Character X (when X is not '1'-'9')
&	The whole character strings which currently correspond
\1	1st tag
\2	2nd tag
...	
\9	9th tag

The following are specification examples:

Enable the breakpoint number to be input with $bp<n>.

```
dsedb> subst "$bp\([0-9]*\)" $BP\1
```

Set the show history to be executable only with history.

```
dsedb> subst "^ *history *$" show history
```

When the start of the command input line is '\', no replacement processing is performed. When a setting such as subst ".*" is made by mistake, for example, quit is always executable by inputting \quit.

source <fname>

The contents of the specified file <fname> are executed as a command string.

A '\' at the end of a line in a file indicates that the line continues on to the next line.

![<cmd> [<args>]...]

The command <cmd> and arguments <args>... that follow "!" are passed to the shell and executed. <cmd> and <args> are directly passed to the shell without any evaluation being performed. The shell that is used is determined by the environment variable SHELL. If SHELL has not been set, "/bin/sh" will be used. If <cmd> is omitted, only the shell is called.

cd [<directory>]

The current directory is changed to <directory>. If the argument is omitted, the current directory is changed to the directory indicated by the environment variable HOME.

printf "<fmt>" [<args>]...

A message is output to the screen.

The same formats as those that are used with printf(3) can be used for <fmt>. However, the floating-related formats (e, E, f, and g) and %n are not supported. <args> is always evaluated as an expression and its value is used.

record [[-a] <fname>]

The subsequent screen output contents are saved in the specified file <fname>.

If <fname> is omitted, recording ends.

If the -a option is specified, the screen output contents are written to the file in append mode.If the next record command is specified without ending recording, the previous file is automatically closed.

repeat [-c <cnt>] [-i <sec>] [-while <expr>] <cmd>...

The specified commands <cmd>... are repeatedly executed.

When a repetition count <cnt> is specified with the -c option, the commands are repeated only the specified number of times. If no repetition count is specified, the commands are repeated indefinitely until an error occurs or CTRL-C is entered. Processing such as sending and receiving packets is performed once between repeated executions. If no send/receive packet is performed or no keyboard entry is made within the interval in seconds specified by <sec> (default value is 1 [second]) with the -i option, the next repetition begins.

If a send/receive packet is performed or a keyboard entry is made, the next repetition is executed without waiting for the interval specified with the -i option.

When the -while <expr> option is specified, <expr> will be evaluated each time through, before execution of <cmd>... . If the value of <expr> is 0, the repeat command will be terminated.

When the -while option is specified without the -i option, the timeout value will be 0.

To repeatedly execute more than one command, use a construction such as repeat (dr;dr).

show log [[-]<option>] ...

The DECI2 send/receive packet log is displayed by the show log ... command.

Table 30

Option name	Description
l[ong]	Also display time differences, IP addresses, and other information.
n[um]	Display all DECI2 packets in hexadecimal.
r[emove]	Delete all log data.
a[ll]	Display all data (in contrast to t[ail] and h[ead])
t[ail] <N>	Display only the last <N> data.
h[ead] <N>	Display only the first <N> data.
i[d] <ID>	Display only data related to <ID>.
m[sg] <msg>	Display only data related to <msg>.
p[roto] <proto>	Display only data related to protocol <proto>.
s[rc] <src>	Display only data related to source node <src>.
d[st] <dst>	Display only data related to destination node <dst>.
v[erbose]	Display data in verbose mode.

long and num specify display formats, remove specifies data deletion, and all, tail, and head specify the number of data values to be displayed. The other options specify display conditions. Multiple specifications are always interpreted as being combined by a logical AND.

show status [-m] [-d]

Displays the internal status.

- If -m is specified, memory status is displayed.
- If -d is specified, the statuses of all connections that handle DECI2 packets are displayed.

If both -m and -d are omitted, both types of status information are displayed.

show history

Displays past input lines.

show dbconf

Displays the setting information returned from the target.

show symbol [<str>]

Displays symbol information.

When <str> is omitted, all information is displayed. When <str> is specified, only information which includes <str> in the name is displayed.

show mdebug [<str>]

Displays mdebug information.

When <str> is omitted, all information is displayed. When <str> is specified, only information with <str> in the name are displayed.

help [<keyword>]...

If <keyword> is omitted, a list of dsedb commands will be displayed. If <keyword> is specified, help information for that particular keyword will be displayed. The following can be specified for <keyword>.

Table 31

Keyword name	Description
<cmd>	help information for command name <cmd> will be displayed.
reg	accessible register groups will be displayed.
edit	keys available for line editing will be displayed.
var	a list of option variables that can be set using the set command will be displayed.
<var>	Information about the option variable <var>, which can be set with the set command, will be displayed.
expr	Information about expressions that can be used will be displayed.
keyword	All keywords that can be specified with the help command will be displayed.

quit

dsedb is terminated.

Option Descriptions

target_name

Any of the following three methods can be used to set the host name and port of dsnetm.

1. Value of environment variable "DSNETM"
2. option statement within auto-execution-on-startup file
3. dsedb startup option -d <host>[:<port>]

If more than one of these is specified, the setting is overwritten in the order 1)->2)->3). The default value of this option is "" (null) and means the default port specification of the host which executed the dsedb.

This is a startup-specific option that can be executed only by using an option statement within an autorun file.

tty_mask

This option sets whether or not the target's TTYP input/output is to be handled by dsedb. If the target's TTYP input/output is not to be handled by dsedb, specify 0. There are 11 TTYP types from E0TTYP to E9TTYP and EKTTYP. Bit 0 of the tty_mask corresponds to EOTTYP and Bit 9 corresponds to E9TTYP.EKTTYP can be specified with Bit 15. The default value of this option is 0x83ff.

This is a startup-specific option that can be executed only by using an option statement within an autorun file. For the TTYP output of dsedb, only the display of individual lines using LF, is supported. Note that this is different from dsecons.

atty_mask

For dsedb, atty_mask sets whether or not to display the TTYP of the IOP, and for dsidb, it sets whether or not to display the TTYP of the EE. The bit value definitions are similar to those of tty_mask.

The default value of this option is 0x0. This is a startup-specific option that can be executed only by using an option statement within an autorun file.

file_priority

This option sets whether or not Remote File Access from the target is to be handled by dsedb. If Remote File Access from the target is not to be handled by dsedb, specify -1. The default value of this option is 0xd0.

This is a startup-specific option that can be executed only by using an option statement within an autorun file.

reset_on_start

This option specifies whether or not to reset the target when dsedb is started up. This specification is effective when neither the -r nor the -nr command line option has been specified. The default value of this option is true.

This is a startup-specific option that can be executed only by using an option statement within an autorun file.

check_manager_version

The dsedb specifies whether to check the version information on the dsnetm on starting. When true, checking is performed, and when not, the operation is terminated with an error. When false, the version information is not checked.

The default value for this option is true. If the -ncmv is specified in the command line, it is overwritten by false. This is a startup-specific option that can be executed only by using an option statement within an autorun file.

target_exec_ctl_config

This shows the execution control state of an execution command such as cont, step, next, etc. 0 means control on the host side and anything else means control on the target side.

target_exec_ctl_override

This specifies overwriting to the value of target_exec_ctl_config. Values of 0 or more are overwritten.

dr_default_format

The default value is:

```
option dr_default_format=""
```

which means that a fixed, internal display format should be used.

To explicitly set the equivalent format, specify the following in an autorun file.

For ~/.dsidbrc

```
option dr_default_format="\
at=%at  v0-1=%v0,%v1  a0-3=%a0,%a1,%a2,%a3\n\
t0-7=%t0,%t1,%t2,%t3, %t4,%t5,%t6,%t7\n\
s0-7=%s0,%s1,%s2,%s3, %s4,%s5,%s6,%s7\n\
t8=%t8 t9=%t9   k0=%k0 k1=%k1   gp=%gp sp=%sp\n\
fp=%fp ra=%ra   lo=%lo hi=%lo   PC=%PC bada=%bada\n\
$cr=0x%cr [%symcr]\n\
$sr=0x%sr [%symsr]\n"
```

For ~/.dsedbrc

```
option dr_default_format="\
at=%at  v0-1=%v0,%v1  a0-3=%a0,%a1,%a2,%a3\n\
t0-7=%t0,%t1,%t2,%t3, %t4,%t5,%t6,%t7\n\
s0-7=%s0,%s1,%s2,%s3, %s4,%s5,%s6,%s7\n\
t8=%t8 t9=%t9   k0=%k0 k1=%k1   gp=%gp sp=%sp\n\
fp=%fp ra=%ra   lo=%lo hi=%lo   sa=%sa PC=%PC\n\
badvaddr=%badvaddr badpaddr=%badpaddr\n\
$cause   = 0x%cause [%symcause]\n\
$status  = 0x%status [%symstatus]\n"
```

Conversion targets are limited to the following.

%%	This means '%'.
%<reg>	The register name <reg> is displayed according to printf("%08x",...).

You cannot specify any of the following for <reg>.

- Register group name
- Value corresponding to numeric representation of $<n>
- Pseudo register name
- '$' prefix
- '_' prefix
- cr -> cause, sr -> status alias (dsedb)
- cause -> cr alias (dsidb)

If a name that cannot be specified has been specified, %<reg> is displayed without being converted.

For the symbolic display of the cause or status register, %symcr can be specified in dsib and %symstatus, %symcause can be specified in dsedb as in the example above, and it is always preceded and followed by a single space.

Also, 128-bit registers other than %08 and 16-bit displays are not supported, and no setting option exists for dr -vu[01] in dsedb.

dr_default_di

When the dr command is specified without an argument, the instructions before and after the current $PC are disassembled and displayed. The display method to be used is determined by the value of this option. The default value used when this option is not specified is "\\di -m $PC $PC-8 7" (where the leading "\" indicates that no alias expansion is to be performed).

ex_default_dr

When execution that was started by a command such as cont, step, or next is stopped due to a breakpoint or the generation of an exception, dr information is displayed. The display method to be used is determined by the value of this option.

The default value used when this option is not specified is "\\dr".

dr0_default_di

This specifies the disassembly display method at the execution of dr -vu0.

The default value when there is no setting is "\\vdi -vu0 -m $_vu0vi26 $_vu0vi26-3 7".

ex0_default_dr

This specifies the dr display method when a VU0 exception occurs.

The default value when there is no setting is "\\dr -vu0".

dr1_default_di

This specifies the disassembly display method at the execution of dr-vu0. The default value when there is no setting is

"\\vdi -vu1 -m $_vu1vi26 $_vu1vi26-3 7".

ex1_default_dr

This specifies the dr display method when a VU1 exception is generated.

The default value when there is no setting is "\\dr -vu1".

lstep_default_list

Specifies the list display method used after successful completion of lstep, lnext.

If no method is specified, the default value is "\\list".

lstep_stop_at_no_line

Specifies whether command is to be terminated if there is no line number for $PC when starting or running lstep, lnext.

lstep_stop_at_no_line

Specifies whether command is to be terminated if there is no line number for $PC when starting or running lstep, lnext.

Table 32

Specification	Display
0	continues internally performing step, next until an address having a line number is reached.
1	displays a message and terminates the command. (default)

source_directories

This variable specifies the source file search directories to be referenced by the di and list commands.

When the beginning of a source file name contained in .mdebug of an executable file is '/' (absolute path) or the source_directories variable is "", the source filename is only referenced as it is specified.

Otherwise, references are attempted by placing one or more directory names (using ':' as the delimiter) in front of the source filename. A null delimiter means that null is placed in front of the source filename. For example, if ".:" is specified, "" is placed in front of the source filename after a reference is made with "." placed in front of the name.

When the specified directory name starts with '@', the '@' part is replaced by the directory name that was the result of converting the absolute path of the object filename.

The default value of this option is "".

- Example of specifying only the directory containing the object:

  set source_directories="@"
- Example of specifying the sequence Current directory -> Object location -> ABC -> DEF

  set source_directories=".:@:ABC:DEF"

initial_ebootp

This is the initial value of the EE boot parameter that is referenced by the reset command. The default value is "-1", which means "do not change the current value".

current_ebootp

This is the current value of the EE boot parameter that is referenced by the reset command. The default value is "-1", which means "do not change the current value".

initial_ibootp

This is the initial value of the IOP boot parameter that is referenced by the reset command. The default value is "-1", which means "do not change the current value".

current_ibootp

This is the current value of the IOP boot parameter that is referenced by the reset command. The default value is "-1", which means "do not change the current value".

automatic_prefix_breakpoint

The argument of the ub, be and bd command is the address. In the expression specifying the address, a pseudo register in the $BP<n> format can be specified.

If the automatic_prefix_breakpoint variable is true, the "$BP" is automatically placed in front of the "$BP" when all the arguments of the ub, be and db commands are composed of numerical values in decimal number only. The default value is false.

describe_ub_all

Specifies how clearing of all breakpoints will be performed when argumentsare omitted for the ub command. When the describe_ub_all variable is true,all breakpoints will not be cleared in this case. The default value is false.

di_address

Sets di command address(label) display format.

Table 33

Specification	Display
0x0	Both the hex value and the symbol are displayed. (default)
0x1	Only the hex value is displayed.
0x2	Only the symbol, if it exists, otherwise the hex value.
0x3	Displays the same symbol only once on a separate line.
0x4	Displays the line number in addition to 0x0
0x5	Displays the line number in addition to 0x1
0x6	Displays the line number in addition to 0x2
0x7	Displays the line number in addition to 0x3
0x8	Displays the source line in addition to 0x4
0x9	Displays the source line in addition to 0x5
0xa	Displays the source line in addition to 0x6
0xb	Displays the source line in addition to 0x3.
	If the source file cannot be accessed, same as 0x7. (default).

To enable display of line numbers and source lines, create the executable using gcc -g.

di_instruction_word

Sets di command instruction word display format.

0: Display hex value of instruction word. (default)
1: Do not display hex value of instruction word.

di_branch_address

Sets di command branch address display format.

0: Display both hex value and symbol. (default)
1: Display only hex value.
2: Display only the symbol, if it exists, otherwise display the hex value.

di_macro

Specifies whether the di/as commands will handle macro instructions such as li, move.

0: no macro handling
1: handling of li, move, b, bal, dmove macros. (default)

help_lang

Specifies the language used for help command displays.

Table 34

Specification	Display
"eng"	English (default)
"euc"	Japanese (EUC code)
"jis"	Japanese (JIS code)
"sjis"	Japanese (shift-JIS code)

help_pager

Specifies the pager used by the help command to display information.

Default is "" (empty string), i.e., no pager.

hex_radix

This option sets the method of interpreting a numeric value in dsedb.

The default value is true, and a numeric value without a 0x prefix is interpreted as a hexadecimal number. If the option is set to false, a hexadecimal number must be prefixed by 0x as in the C programming language.

If symbols are not used, it is more convenient to set this option true. However, if symbols are present and the command "db abc" is specified, it is ambiguous whether this refers to symbol abc or hexadecimal value

0xabc, and in the latter case must be reentered as "db 0xabc." If the option is set to false, the command "db abc" will refer to the symbol abc and there is no need to reenter the specification depending on the situation.

log_total_size

By default, the DECI2 packet log is saved with a maximum size of 128 kilobytes. If the send/receive log is unnecessary, set this option to 0.

log_packet_size

By default, up to 128 bytes starting at the beginning of a packet is saved in the log. To save all data in the log, set the value of this option to at least 65535.

histfile

By default, history is saved in the file indicated by the "~/.dsedb_history" character string when dsedb is finished. The history is loadded from this file on startup

histfilesize

This is the number of histories to be saved in the histfile.

The default value is 256.

histsize

This shows the upper limit of the number of histories recorded when a command is in operation. The default value is 256.

tty_max_size

This shows the maximum number of pending bytes of the TTY packet transferred by dsedb. The default value is 64KB.