

# **DBGP Specification (for EE Thread Extension)**

© 2001 Sony Computer Entertainment Inc.

Publication date: October 2001

Sony Computer Entertainment Inc.  
1-1, Akasaka 7-chome, Minato-ku  
Tokyo 107-0052, Japan

Sony Computer Entertainment America  
919 E. Hillsdale Blvd.  
Foster City, CA 94404, U.S.A.

Sony Computer Entertainment Europe  
30 Golden Square  
London W1F 9LD, U.K.


The *DBGP Specification (for EE Thread Extension)* manual is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® license agreements.

The *DBGP Specification (for EE Thread Extension)* manual is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® license agreements.

Unauthorized reproduction, distribution, lending, rental or disclosure to any third party, in whole or in part, of this book is expressly prohibited by law and by the terms of the Sony Computer Entertainment PlayStation® license agreements.

Ownership of the physical property of the book is retained by and reserved by Sony Computer Entertainment. Alteration to or deletion, in whole or in part, of the book, its presentation, or its contents is prohibited.

The information in the *DBGP Specification (for EE Thread Extension)* manual is subject to change without notice. The content of this book is Confidential Information of Sony Computer Entertainment.

 and PlayStation are registered trademarks of Sony Computer Entertainment Inc. All other trademarks are property of their respective owners and/or their licensors.

# Table of Contents

<b>About This Manual</b>	<b>v</b>
Changes Since Last Release	v
Related Documentation	v
Typographic Conventions	v
Developer Support	v
<b>Overview</b>	<b>1</b>
Message Format	1
<b>Messages</b>	<b>2</b>
Thread List(THREADLIST/THREADLISTR)	2
Thread Control Block(TCB/TCBR)	4
Semaphore Block(SEMABLOCK/SEMABLOCKR)	8
Handler List(HANDLERLIST/HANDLERLISTR)	9



## About This Manual

This is the Runtime Library Release 2.4 version of the *DBGP Specification (for EE Thread Extension)* manual.

It describes the EE thread extension of DBGP.

## Changes Since Last Release

None

## Related Documentation

**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
<code>courier</code>	Indicates literal program code.
<i>italic</i>	Indicates names of arguments and structure members (in structure/function definitions only).
<b>medium bold</b>	Indicates data types and structure/function names (in structure/function definitions only).
<a href="#">blue</a>	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
<i>In North America:</i>	<i>In North America:</i>
Attn: Developer Tools Coordinator	E-mail: <a href="mailto:PS2_Support@playstation.sony.com">PS2_Support@playstation.sony.com</a>
Sony Computer Entertainment America	Web: <a href="http://www.devnet.scea.com/">http://www.devnet.scea.com/</a>
919 East Hillsdale Blvd.	Developer Support Hotline: (650) 655-5566
Foster City, CA 94404, U.S.A.	(Call Monday through Friday,
Tel: (650) 655-8000	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
<i>In Europe:</i> Attn: Production Coordinator Sony Computer Entertainment Europe 30 Golden Square London W1F 9LD, U.K. Tel: +44 (0) 20 7859-5000	<i>In Europe:</i> E-mail: ps2_support@scee.net Web: <a href="https://www.ps2-pro.com/">https://www.ps2-pro.com/</a> Developer Support Hotline: +44 (0) 20 7859-5777 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT)

## Overview

This document describes the EE thread extension of DBGP. These functions are supported in DBGP Version 3.10 and later. In versions which do not support these functions, DBGP\_RESULT\_INVALIDREQ will be returned in the DBGP header result field.

## Message Format

Values set in each field of a DBGP header are as follows.

### id

DBGP\_CPUID\_CPU     0     // CPU (ESDBGP)

### group

DBGP\_GROUP\_EE\_THREAD   1

### type

DBGP_EE_THREAD_TYPE_THREADLIST	0x30 //Get Thread List Request
DBGP_EE_THREAD_TYPE_THREADLISTR	0x31 //Get Thread List Reply
DBGP_EE_THREAD_TYPE_TCB	0x32 //Get TCB Request
DBGP_EE_THREAD_TYPE_TCBR	0x33 //Get TCB Reply
DBGP_EE_THREAD_TYPE_SEMABLOCK	0x34 //Get Semaphore Block Request
DBGP_EE_THREAD_TYPE_SEMABLOCKR	0x35 //Get Semaphore Block Reply
DBGP_EE_THREAD_TYPE_HANDLERLIST	0x36 //Get Handler List Request
DBGP_EE_THREAD_TYPE_HANDLERLISTR	0x37 //Get Handler List Reply

code, count are described in the description of type.

### result

Result codes shown in the following table are entered in messages sent from a target to a host. 0 is always set in messages sent from a host to a target.

Common to all groups

DBGP_RESULT_GOOD	0x00 // Good
DBGP_RESULT_INVALIDREQ	0x01 // Invalid Request
DBGP_RESULT_UNIMPREQ	0x02 // Unimplemented Request
DBGP_RESULT_ERROR	0x03 // Error

If it was determined that an invalid request occurred after the target recognized a type, or if a message of an unimplemented type is received, messages of those response types will be returned for each appropriate type.

In addition, the sender must always set fields to 0 that are unspecified in this document. If the receiver obtains a value that is not 0, it must be handled as an error to the best extent possible.

Messages

Thread List(THREADLIST/THREADLISTR)

Messages which request valid thread lists in the EE kernel and their responses.

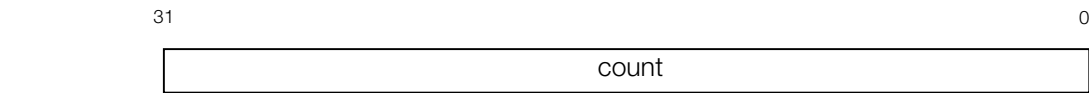
The messages consist of the following types of headers and data.

DBGP\_EE\_THREAD\_TYPE\_THREADLIST: DECI2 header + DBGP header  
DBGP\_EE\_THREAD\_TYPE\_THREADLISTR: DECI2 header + DBGP header +  
DBGP\_EE\_THREADLIST\_HDR + DBGP\_EE\_THREADLIST\_DATA X  
DBGP\_EE\_THREADLIST\_HDR.count

Set the code field of the DBGP header by combining the codes below.

DBGP_EE_THREAD_CODE_ALLTHREADLIST	0x3f
List of all threads	
DBGP_EE_THREAD_CODE_THREADREADYQUEUELIST	0x0
Thread ready queue thread list	
DBGP_EE_THREAD_CODE_RUN	0x1
RUN state thread list	
DBGP_EE_THREAD_CODE_READY	0x2
READY state thread list	
DBGP_EE_THREAD_CODE_WAIT	0x4
WAIT state thread list	
DBGP_EE_THREAD_CODE_SUSPEND	0x8
SUSPEND state thread list	
DBGP_EE_THREAD_CODE_WAITSPEND	0x20
WAITSPEND state thread list	
DBGP_EE_THREAD_CODE_DORMANT	0x10
DORMANT state thread list	
DBGP_EE_THREAD_CODE_RESET	0x80
Initialize to 0 the number of times the threads have gone to RUN state.	

Figure 1: Thread List Header(DBGP\_EE\_THREADLIST\_HDR)



count

Number of threads included in the thread list



Figure 2: Thread List Data(DBGP\_EE\_THREADLIST\_DATA)

31	0
id	
priority	
status	
cause	
waitid	
wakeupcount	
count	
pc	
sp	
func	
ra	
reserved [0]	
reserved [1]	

**id**

Thread ID

**priority**

Thread priority

**status**

Thread state

THS_RUN	0x01	RUN state
THS_READY	0x02	READY state
THS_WAIT	0x04	WAIT state
THS_SUSPEND	0x08	SUSPEND state
THS_WAITSPEND	0xc	WAIT-SUSPEND state
THS_DORMANT	0x10	DORMANT state

**cause**

WAIT factor (None = 0, Sleep = 1, Sema = 2)

**waitid**

Semaphore ID

**wakeupcount**

Number of wakeup requests

**count**

Number of times the thread has gone to RUN state

**pc**

Program counter

**sp**

Stack pointer \$29

**func**

Execution start address

**ra**

Return address \$31

**reserved[2]**

Reserved area

A `DBGP_EE_THREAD_TYPE_THREADLISTR` response returns a 1 in the count field of a DBGP header. When the thread is in RUN state, the current values are placed in pc,sp,ra.

**Limitations**

When there is a break in the interrupt handler, the register of the thread in RUN state is obtained and the current value is returned.

**Thread Control Block(TCB/TCBR)**

Messages which specify valid thread IDs in the EE kernel and their responses. The messages consist of the following types of headers and data.

`DBGP_EE_THREAD_TYPE_TCB`: DECI2 header + DBGP header +

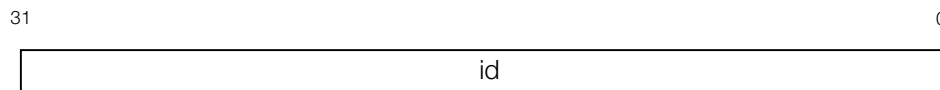
`DBGP_EE_THREADID_DATA`

`DBGP_EE_THREAD_TYPE_TCBR`: DECI2 header + DBGP header +

`DBGP_EE_TCB_DATA`

The format for `DBGP_EE_THREADID_DATA` is as follows.

**Figure 3: Thread Id Data(DBGP\_EE\_THREADID\_DATA)**

**id**

Valid thread ID to be specified

Figure 4: TCB Data (DBGP\_EE\_THREADTCB\_DATA)

31

0

id
priority
status
cause
waitid
wakeupcount
count
pc
stackpointer
func
args
argc
stack
stacksize
endofheap
option
gpReg
initpriority
sa
fcr31
gpr [0] [0]
gpr [0] [1]
gpr [0] [2]
gpr [0] [3]
:
gpr [31] [2]
gpr [31] [3]
hi [0]
hi [1]
lo [0]
lo [1]
hi1 [0]
hi1 [1]
lo1 [0]
lo1 [1]
fpr [0]
:
fpr [31]
facc
reserved [0]
reserved [1]
reserved [2]

**id**

Thread ID

**priority**

Thread priority

**status**

Thread state

THS_RUN	0x01	RUN state
THS_READY	0x02	READY state
THS_WAIT	0x04	WAIT state
THS_SUSPEND	0x08	SUSPEND state
THS_WAITSPEND	0xc	WAIT-SUSPEND state
THS_DORMANT	0x10	DORMANT state

**cause**

WAIT factor (None = 0, Sleep = 1, Sema = 2)

**waitid**

Semaphore ID during semaphore WAIT

**wakeupcount**

Number of wakeup requests

**count**

Number of times the thread has gone to RUN state

**pc**

Program counter

**stackpointer**

Top address of user stack (= \$29(sp) + 0x280)

**func**

Execution start address

**args**

Arguments of StartThread

**argc**

Number of arguments of StartThread

**stack**

Bottom address of stack

**stacksize**

Stack size

**endofheap**

End address of heap

**option**

Cannot be used

**gpReg**

Address of global pointer

**initpriority**

Initial value of priority

**sa**

sa register

**fcr31**

fcr31 register

**gpr[32]**

GPR

**hi**

hi register

**lo**

lo register

**hi1**

hi1 register

**lo1**

lo1 register

**fpr[32]**

fpr register

**facc**

facc register

**reserved[3]**

Reserved area

A DBGP\_EE\_THREAD\_TYPE\_TCBP response returns a 0 in the count field of a DBGP header when the thread corresponding to the specified thread ID does not exist. 1 is returned when the thread does exist. When the thread is in RUN state, the current register values are placed in gpr, fprt, facc, pc, sa, hi, lo, hi1, lo1.

**Limitations**

When there is a break in the interrupt handler, the register of the thread in RUN state is obtained and the current value is returned.

**Semaphore Block(SEMABLOCK/SEMABLOCKR)**

Messages which request valid semaphore information in the EE kernel and response messages which return semaphore information.

The messages consist of the following types of headers and data.

DBGP\_EE\_THREAD\_TYPE\_SEMABLOCK: DECI2 header + DBGP header +  
 DBPG\_EE\_SEMAID\_DATA  
 DBGP\_EE\_THREAD\_TYPE\_SEMABLOCK R: DECI2 header + DBGP header +  
 DBGP\_EE\_SEMABLOCKHDR + (DBGP\_EE\_SEMABLOCK\_DATA +  
 DBGP\_EE\_THREADID\_DATA X DBGP\_SEMABLOCK\_DATA.numWaitThreads ) X  
 DBGP\_EE\_THREADBLOCK\_HDR.count

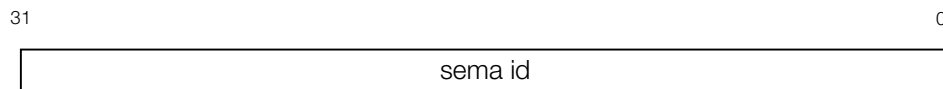
In the code field of the DBGP header of DBGP\_EE\_THREAD\_TYPE\_SEMABLOCK

DBGP\_EE\_THREAD\_CODE\_SEMAID 0 // set the semaphore ID in DBGP\_EE\_SEMAID\_DATA and get semaphore information.

DBGP\_EE\_THREAD\_CODE\_SEMAALL 1 // get all semaphore information.

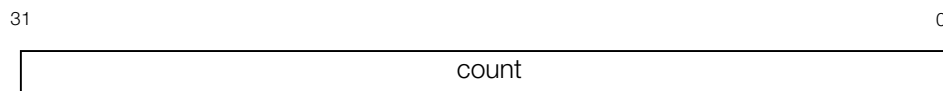
Set DBGP\_EE\_SEMAID\_DATA to 0.

**Figure 5: Semaphore ID data (DBGP\_EE\_SEMAID\_DATA)**

**sema id**

Semaphore ID

**Figure 6: Semaphore block data (DBGP\_EE\_SEMABLOCK\_HDR)**

**count**

Number of semaphores included in response message

Figure 7: Semaphore information (DBGP\_EE\_SEMABLOCK\_DATA + DBGP\_EE\_THREADID\_DATA X DBGP\_EE\_SEMABLOCK\_DATA.numWaitThreads)

31	0
	sema id
	count
	maxcount
	attr
	option
	numWaitThreads
	thread id
	thread id
	thread id
	:

**sema id**

Semaphore ID

**count**

Number of semaphore resources

**maxcount**

Maximum number of semaphore resources

**attr**

Semaphore attributes

**option**

Additional user defined information

**numWaitThreads**

Number of threads waiting for semaphore

**thread id**

Thread ID of thread waiting for semaphore. This ID is returned in order of semaphore waiting.

When the respective semaphore exists, a 1 is returned in the count field of the DBGP header of a DBGP\_EE\_THREAD\_TYPE\_SEMABLOCKR response. Returns a 0 if no semaphore exists.

## Handler List(HANDLERLIST/HANDLERLISTR)

Messages that request interrupt handler lists registered using the EE kernel API and their responses.

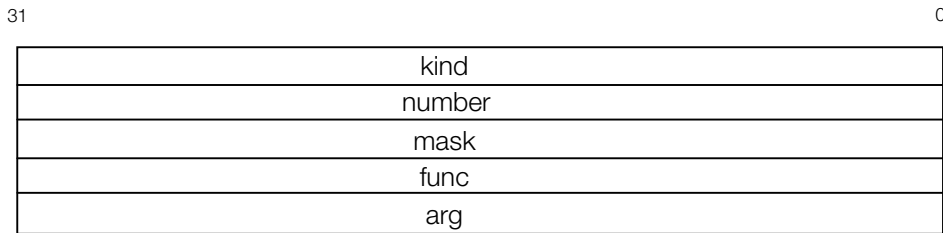
The messages consist of the following types of headers and data.

DBGP\_EE\_THREAD\_TYPE\_HANDLERLIST

DBGP\_EE\_THREAD\_TYPE\_HANDLERLISTR: DECI2 header + DBGP header +

DBGP\_EE\_HANDLER\_DATA X DBPG header.count

Figure 8: Interrupt handler data (DBGP\_EE\_HANDLER\_DATA)

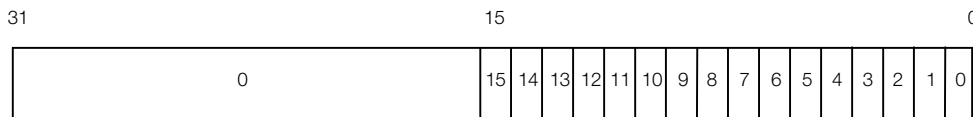


**kind**  
Handler type  
DBGP\_EE\_INTC\_HANDLER 0 // INTC interrupt  
DBGP\_EE\_DMAC\_HANDLER 1 // DMAC interrupt

**number**  
Interrupt handler number (0-15)

**mask**  
Checks whether the interrupt handler is valid

Figure 9: DBGP\_EE\_HANDLER\_DATA.mask



If the handler is valid, the bits of that number change to 1. If invalid, 0.

**func**  
Entry point of interrupt handler function

**arg**  
Argument to interrupt handler

The DBGP\_EE\_THREAD\_TYPE\_HANDLERLISTR response returns the number of DBGP\_EE\_HANDLER\_DATA that is returned in the count field of the DBGP header.