PSPNET adhoc Library Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Table of Contents

Structures

SceNetAdhocPollSd

Poll socket descriptor structure

Definition

```
#include <pspnet_adhoc.h>
struct SceNetAdhocPollSd {
        int id;
        int events;
        int revents;
};
```

Members

id	Socket ID
events	Target events
revents	Generated events

Description

This structure is used for specifying a socket ID and associated target events in order to perform synchronous I/O for multiple sockets. The result is returned in *revents*.

Types of *events* and *revents* are as follows:

Macro	Description
SCE_NET_ADHOC_EV_SEND	OK to send
SCE_NET_ADHOC_EV_RECV	OK to receive
SCE_NET_ADHOC_EV_CONNECT	Connection established (active open)
SCE_NET_ADHOC_EV_ACCEPT	Connection established (passive open)
SCE_NET_ADHOC_EV_FLUSH	Send complete
SCE_NET_ADHOC_EV_INVALID	Invalid socket ID (only used with generated event)
SCE_NET_ADHOC_EV_DELETE	Target socket has been deleted (only used with generated event)
SCE_NET_ADHOC_EV_ALERT	Target socket is in alert state (only used with generated event)
SCE_NET_ADHOC_EV_DISCONNECT	Connection terminated.

See Also

sceNetAdhocPollSocket()

SceNetAdhocPdpStat

PDP control block structure

Definition

```
#include <pspnet_adhoc.h>
struct SceNetAdhocPdpStat {
        struct SceNetAdhocPdpStat *next;
        int id;
        struct SceNetEtherAddr laddr;
        SceUShort16 lport;
        SceUInt32 rcv_sb_cc;
};
```

Members

next	Pointer to next entry in list (NULL indicates end)
id	Socket ID
laddr	Local address
lport	Local port number
rcv_sb_cc	Size of data in receive buffer

Description

This structure is used for obtaining a list of PDP control blocks

See Also

sceNetAdhocGetPdpStat()

SceNetAdhocPtpStat

PTP control block structure

Definition

```
#include <pspnet_adhoc.h>
struct SceNetAdhocPtpStat {
        struct SceNetAdhocPtpStat *next;
        int id;
        struct SceNetEtherAddr laddr;
        struct SceNetEtherAddr paddr;
        SceUShort16 lport;
        SceUShort16 pport;
        unsigned int snd_sb_cc;
        unsigned int rcv_sb_cc;
        int state;
};
```

Members

next	Pointer to next entry in list (NULL indicates end)
id	Socket ID
laddr	Local address
paddr	Peer address
lport	Local port number
pport	Peer port number
snd_sb_cc	Size of data in send buffer
rcv_sb_cc	Size of data in receive buffer
state	Connection state

Description

This structure is used for obtaining a list of PTP control blocks.

state can be one of the following types.

Macro	Description
SCE_NET_ADHOC_PTP_STATE_CLOSED	CLOSED state
SCE_NET_ADHOC_PTP_STATE_LISTEN	LISTEN state
SCE_NET_ADHOC_PTP_STATE_SYN_SENT	SYN_SENT state
SCE_NET_ADHOC_PTP_STATE_SYN_RCVD	SYN_RCVD state
SCE_NET_ADHOC_PTP_STATE_ESTABLISHED	ESTABLISHED state

See Also

sceNetAdhocGetPtpStat()

Initialization/Termination Functions

sceNetAdhocInit

Initialize library

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocInit(
        void
);
```

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_ALREADY_INITIALIZED	0x80410713	Library already initialized

Description

This function initializes the PSPNET adhoc library.

Examples

```
int ret;

ret = sceNetAdhocInit(void);
if (ret < 0) {
        // Error handling
}
```

See Also

sceNetAdhocTerm()

sceNetAdhocTerm

Terminate library

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocTerm(
        void
);
```

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_BUSY	0x80410714	Another API call is in progress Call this function after aborting all other API calls

Description

This function terminates the PSPNET adhoc library. If any APIs were called that have not yet completed, this function fails and returns `SCE_ERROR_NET_ADHOC_BUSY`. Before calling this function, be sure to abort all API calls in progress.

Examples

```
int ret;

ret = sceNetAdhocTerm();
if (ret < 0) {
        // Error handling
}
```

See Also

`sceNetAdhocInit()`

PSPNET Datagram Protocol (PDP) Functions

sceNetAdhocPdpCreate

Create PDP socket

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPdpCreate(
        const struct SceNetEtherAddr *saddr,
        SceUShort16 sport,
        int bufsize,
        int flag
);
```

Arguments

saddr	Local MAC address
sport	Local port number
bufsize	Socket buffer size
flag	Not used (set to 0)

Return Values

Returns the socket ID (>0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_PORT	0x80410703	Invalid port number was specified
SCE_ERROR_NET_ADHOC_PORT_IN_USE	0x8041070a	Specified port is already in use
SCE_ERROR_NET_ADHOC_SOCKET_ID_NOT_AVAIL	0x8041070f	No more sockets can be created
SCE_ERROR_NET_ADHOC_PORT_NOT_AVAIL	0x80410710	No port numbers are available
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function creates a PDP socket. If the local port number is set to 0, an arbitrary port number will be allocated. Only one socket can be created for each combination of one MAC address and one port number.

Examples

```
int ret, id;
struct SceNetEtherAddr addr;

// Get local MAC address
ret = sceNetAdhocctlGetEtherAddr(&addr);
if (ret < 0) {
        // Error handling
        return;
}

// Use arbitrary local port number
// Set socket buffer to 8192 bytes
id = sceNetAdhocPdpCreate(&addr, 0, 8192, 0);
if (id < 0) {
        // Error handling
}
```

Notes

Socket IDs range from 1 to 255.

This function will return an error if the PSPNET ad hoc communication has not been established. Be sure to call this function only after the PSPNET ad hoc communication is established.

See Also

```
sceNetAdhocPdpDelete()
```

sceNetAdhocPdpSend

Send PDP packet

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPdpSend(
        int id,
        const struct SceNetEtherAddr *daddr,
        SceUShort16 dport,
        const void *data,
        int len,
        unsigned int timeout,
        int flag
);
```

Arguments

id	Socket ID
daddr	Destination MAC address
dport	Destination port number
data	Pointer to send data
len	Length of send data
timeout	Timeout (μsec)
flag	Send options

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_INVALID_ADDR	0x80410702	Invalid address was specified
SCE_ERROR_NET_ADHOC_INVALID_PORT	0x80410703	Invalid port number was specified
SCE_ERROR_NET_ADHOC_INVALID_DATALEN	0x80410705	Invalid data length was specified
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred

Description

This function sends a PDP packet. If there is no room in the interface's send queue, the function call is blocked until there is room or until the interval specified by the *timeout* elapses. If 0 is specified for the *timeout*, the function will wait indefinitely. When a timeout occurs, SCE_ERROR_NET_ADHOC_TIMEOUT is returned.

If the sending option has been set to SCE_NET_ADHOC_F_NONBLOCK and if there is no room in the send queue, the function immediately returns SCE_ERROR_NET_ADHOC_WOULD_BLOCK regardless of the *timeout* specification.

The maximum data length which can be sent is 65,523 bytes.

flag types are as follows.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
void *data;
int ret, id, datalen;
struct SceNetEtherAddr daddr;
SceUShort16 dport;

// Provide datalen bytes of data to data
// Set addr and port
...

ret = sceNetAdhocPdpSend(id, &daddr, dport, data, len, 0, 0);
if (ret < 0) {
        // Error handling
}
```

Notes

The maximum fragment size is 1444 bytes. Data that exceeds 1444 bytes is sent by dividing (fragmenting) it into multiple packets. Data that has been fragmented and sent can be received once all fragments have been received. If one or more fragments are lost, this data is discarded.

See Also

sceNetAdhocPdpRecv()

sceNetAdhocPdpRecv

Receive PDP packet

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPdpRecv(
        int id,
        struct SceNetEtherAddr *saddr,
        SceUShort16 *sport,
        void *buf,
        int *len,
        unsigned int timeout,
        int flag
);
```

Arguments

id	Socket ID
saddr	Sender's MAC address
sport	Sender port number
buf	Pointer to receive buffer
len	Receive buffer size (IN), receive data length (OUT)
timeout	Timeout (μsec)
flag	Receive options

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_NOT_ENOUGH_SPACE	0x80410706	libnet memory is insufficient
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_WOULD_BLOCK	0x80410709	Wait state was entered
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred

Description

This function receives PDP packets. It returns the sender's MAC address, port number, and data length for the received packets. If no data can be received, the function call is blocked until a packet is received or until the interval specified by the *timeout* elapses. If 0 is specified for the *timeout*, the function will wait indefinitely. When a timeout occurs, SCE_ERROR_NET_ADHOC_TIMEOUT is returned.

If SCE_NET_ADHOC_F_NONBLOCK is set as a receive option and if no data can be received, the function immediately returns SCE_ERROR_NET_ADHOC_WOULD_BLOCK regardless of the *timeout* specification.

If the socket buffer size is exceeded when a packet is received, the packet is discarded without being written to the socket buffer.

flag can be the following type.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
char buf[1024];
int ret, id, len;
struct SceNetEtherAddr saddr;
SceUShort16 sport;

len = 1024;
ret = sceNetAdhocPdpRecv(id, &saddr, &sport, buf, &len, 0, 0);
if (ret < 0) {
        // Error handling
}
```

See Also

sceNetAdhocPdpSend()

sceNetAdhocPdpDelete

Delete PDP socket

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPdpDelete(
        int id,
        int flag
);
```

Arguments

id	Socket ID
flag	Not used (set to 0)

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function deletes a PDP socket. If the socket in question is in a waiting-to-send or waiting-to-receive state, this function returns an error (SCE_ERROR_NET_ADHOC_SOCKET_DELETED). Packets in the receive buffer are discarded.

Examples

```
int ret, id;

ret = sceNetAdhocPdpDelete(id, 0);
if (ret < 0) {
        // Error handling
}
```

See Also

sceNetAdhocPdpCreate()

sceNetAdhocGetPdpStat

Get PDP control blocks

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocGetPdpStat(
        int *buflen,
        void *buf
);
```

Arguments

buflen	Buffer size of *buf* (IN), list size (OUT)
buf	Pointer to area where the list of PDP control block structures will be stored

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function gets a list of PDP control block structures.

You must allocate *buflen* size in the memory area indicated by *buf*.

This function takes two kinds of action, depending on the value of *buf*.

- If *buf* is NULL, the function returns in *buflen* the size of the buffer needed to store the list of structures.
- If *buf* contains an address value, a list of SceNetAdhocPdpStat PDP control block structures is created up to the maximum buffer size specified in *buflen*, in the area specified by *buf*. The list has *buf* as its starting location, is linked using the *next* member, and is terminated by NULL.

Examples

```
int ret, buflen;
struct SceNetAdhocPdpStat *buf, *ptr;

ret = sceNetAdhocGetPdpStat(&buflen, NULL);
if (ret < 0) {
        // Error handling
        return;
}
else if (buflen == 0) {
        // The data did not exist
        return;
}

buf = malloc(buflen);
if (buf == NULL) {
        // Memory could not be allocated
        return;
}

ret = sceNetAdhocGetPdpStat(&buflen, buf);
if (ret < 0) {
        // Error handling
}
else if (buflen == 0) {
        // The data did not exist
}
else {
        for (ptr = buf; ptr != NULL; ptr = ptr->next) {
              // Process obtained control block
              ...
        }
}

free(buf);
```

See Also

SceNetAdhocPdpStat

PSPNET Transport Protocol (PTP) Functions

sceNetAdhocPtpOpen

Create PTP socket and open connection

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpOpen(
        const struct SceNetEtherAddr *saddr,
        SceUShort16 sport,
        const struct SceNetEtherAddr *daddr,
        SceUShort16 dport,
        SceUInt32 bufsize,
        SceUInt32 rexmt_int,
        int rexmt_cnt,
        int flag
);
```

Arguments

saddr	Local MAC address
sport	Local port number
daddr	Destination MAC address
dport	Destination port number
bufsize	Socket buffer size
rexmt_int	Retransmit interval (μsec)
rexmt_cnt	Retransmit count
flag	Not used (set to 0)

Return Values

Returns the socket ID (>0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_ADDR	0x80410702	Invalid address was specified
SCE_ERROR_NET_ADHOC_INVALID_PORT	0x80410703	Invalid port number was specified
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function creates a PTP socket and opens a connection to the specified destination MAC address and port number. If the local port number is set to 0, an arbitrary port number will be allocated. If the local port number is already in use, a new socket cannot be allocated to that port.

Examples

```
int ret, id;
struct SceNetEtherAddr saddr, daddr;
SceUShort16 dport;

// Get Local MAC address
ret = sceNetAdhocctlGetEtherAddr(&saddr);
if (ret < 0) {
        // Error handling
        return;
}

// Set destination address and port number (daddr, dport)
...

// Use an arbitrary local port number
// Set the socket buffer to 8192 bytes
// Set the retransmit interval to 500 ms, retransmit count to 20
id = sceNetAdhocPtpOpen(&saddr, 0, &daddr, dport, 8192, 500*1000, 20, 0);
if (id < 0) {
        // Error handling
}
```

Notes

This function will return `SCE_ERROR_NET_ADHOC_INVALID_ADDR` if the PSPNET ad hoc communication has not been established. Be sure to call this function only after the PSPNET ad hoc communication is established.

See Also

`sceNetAdhocPtpConnect()`

sceNetAdhocPtpConnect

Establish PTP connection

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpConnect(
        int id,
        unsigned int timeout,
        int flag
);
```

Arguments

id	Socket ID
timeout	Timeout (μsec)
flag	Option

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_WOULD_BLOCK	0x80410709	Wait state was entered
SCE_ERROR_NET_ADHOC_NOT_OPENED	0x8041070d	Socket has not been opened
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred
SCE_ERROR_NET_ADHOC_CONNECTION_REFUSED	0x80410718	Connection establishment was refused

Description

This function waits for a connection to be established on the socket created by `sceNetAdhocPtpOpen()`. If no connection is active, the function blocks until a connection is established or until the interval specified by the *timeout* elapses. If 0 is specified for *timeout*, the function will wait indefinitely. When a timeout occurs, this function returns `SCE_ERROR_NET_ADHOC_TIMEOUT`.

If `SCE_NET_ADHOC_F_NONBLOCK` is specified as an option and if no connection has been established, `SCE_ERROR_NET_ADHOC_WOULD_BLOCK` is returned immediately.

flag can be set to the following type.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
int ret, id;

while (1) {
        // Other tasks

        // Call in non-blocking mode (poll)
        ret = sceNetAdhocPtpConnect(id, 0, SCE_NET_ADHOC_F_NONBLOCK);
        if (ret == SCE_ERROR_NET_ADHOC_WOULD_BLOCK) {
              // No connection yet
              continue;
        }
        else if (ret < 0) {
              // Error handling
              break;
        }
        else {
              // Connection succeeded
              break;
        }
}
```

See Also

sceNetAdhocPtpOpen()

sceNetAdhocPtpListen

Create PTP socket and wait for connection

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpListen(
        const struct SceNetEtherAddr *saddr,
        SceUShort16 sport,
        unsigned int bufsize,
        unsigned int rexmt_int,
        int rexmt_cnt,
        int backlog,
        int flag
);
```

Arguments

saddr	Local MAC address
sport	Local port number
bufsize	Socket buffer size
rexmt_int	Retransmit interval (μsec)
rexmt_cnt	Retransmit count
backlog	Connection queue length
flag	Not used (set to 0)

Return Values

Returns the socket ID (>0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_ADDR	0x80410702	Invalid address was specified
SCE_ERROR_NET_ADHOC_INVALID_PORT	0x80410703	Invalid port number was specified
SCE_ERROR_NET_ADHOC_PORT_IN_USE	0x8041070a	Specified port is already in use
SCE_ERROR_NET_ADHOC_SOCKET_ID_NOT_AVAIL	0x8041070f	No socket IDs are available
SCE_ERROR_NET_ADHOC_PORT_NOT_AVAIL	0x80410710	No port numbers are available
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function creates a PTP socket and waits for a connection. If the local port number is set to 0, an arbitrary port number will be allocated. If the local port number is already in use, a new socket cannot be allocated to that port.

Examples

```
int ret, id;
struct SceNetEtherAddr saddr;
SceUShort16 sport;

// Get local MAC address
ret = sceNetAdhocctlGetEtherAddr(&addr);
if (ret < 0) {
        // Error handling
        return;
}
// Set the waiting port number to 80
sport = 80;

// Set the socket buffer to 8192 bytes
// Set the retransmit interval to 500 ms, the retransmit count to 20
// Set the connection queue to 5
id = sceNetAdhocPtpListen(&saddr, sport, 8192, 500*1000, 20, 5, 0);
if (id < 0) {
        // Error handling
}
```

Notes

This function will return `SCE_ERROR_NET_ADHOC_INVALID_ADDR` if the PSPNET ad hoc communication has not been established. Be sure to call this function only after the PSPNET ad hoc communication is established.

See Also

`sceNetAdhocPtpAccept()`

sceNetAdhocPtpAccept

Establish PTP connection

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpAccept(
        int id,
        struct SceNetEtherAddr *addr,
        SceUShort16 *port,
        SceUInt32 timeout,
        int flag
);
```

Arguments

id	Socket ID
addr	Peer MAC address
port	Peer port number
timeout	Timeout (μsec)
flag	Option

Return Values

Returns the socket ID (>0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_WOULD_BLOCK	0x80410709	Wait state was entered
SCE_ERROR_NET_ADHOC_NOT_LISTENED	0x8041070e	Socket is not being listened to
SCE_ERROR_NET_ADHOC_SOCKET_ID_NOT_AVAIL	0x8041070f	No socket IDs are available
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred

Description

This function waits for a connection on the socket created by `sceNetAdhocPtpListen()`. If no connection is active, the function call blocks until a connection is established or until the interval specified by the *`timeout`* elapses. If 0 is specified for *`timeout`*, the function will wait indefinitely. When a timeout occurs, this function returns `SCE_ERROR_NET_ADHOC_TIMEOUT`.

If `SCE_NET_ADHOC_F_NONBLOCK` is set as an option and if no connection is active, this function will immediately return `SCE_ERROR_NET_ADHOC_WOULD_BLOCK` regardless of the *`timeout`* specification.

Once a connection is established, this function returns the newly created socket ID.

The connected peer's MAC address and port number are returned in *`addr`* and *`port`*.

`flag` can be set to the following type.

Macro	Description
`SCE_NET_ADHOC_F_NONBLOCK`	Non-blocking mode

Examples

```
int id, newid;
struct SceNetEtherAddr addr;
SceUShort16 port;

while (1) {
        // Other tasks

        // Call in non-blocking mode (poll)
        newid = sceNetAdhocPtpAccept(id, &addr, &port, 0,
SCE_NET_ADHOC_F_NONBLOCK);
        if (newid == SCE_ERROR_NET_ADHOC_WOULD_BLOCK) {
              // No connection yet
              continue;
        }
        else if (newid < 0) {
              // Error handling
              break;
        }
        else {
              // Connection succeeded
              break;
        }
}
```

See Also

`sceNetAdhocPtpListen()`

sceNetAdhocPtpSend

Send data

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpSend(
        int id,
        const void *data,
        int *len,
        SceUInt32 timeout,
        int flag
);
```

Arguments

id	Socket ID
data	Pointer to send data
len	Size of send data (IN)
	Actual size of the socket buffer that is stored (OUT)
timeout	Timeout (μsec)
flag	Option

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_NET_ERROR_EINTR	0x80410104	Processing aborted by abort request
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_INVALID_DATALEN	0x80410705	Invalid data length was specified
SCE_ERROR_NET_ADHOC_NOT_ENOUGH_SPACE	0x80410706	libnet memory is insufficient
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_WOULD_BLOCK	0x80410709	Wait state was entered
SCE_ERROR_NET_ADHOC_NOT_CONNECTED	0x8041070b	Socket has not been connected
SCE_ERROR_NET_ADHOC_DISCONNECTED	0x8041070c	Socket was disconnected
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred

Description

This function sends data. If there is not enough space in the socket buffer, the function call blocks until space becomes available or until the interval specified by the *timeout* elapses. If 0 is specified for *timeout*, the function will wait indefinitely. When a timeout occurs, this function returns SCE_ERROR_NET_ADHOC_TIMEOUT.

If the option has been set to SCE_NET_ADHOC_F_NONBLOCK and if there is not enough space in the socket buffer, this function immediately returns SCE_ERROR_NET_ADHOC_WOULD_BLOCK regardless of the *timeout* specification.

When the return value indicates normal completion or is SCE_ERROR_NET_ADHOC_NOT_ENOUGH_SPACE, SCE_ERROR_NET_ADHOC_TIMEOUT, SCE_ERROR_NET_ADHOC_SOCKET_ALERTED, SCE_ERROR_NET_ADHOC_WOULD_BLOCK, SCE_ERROR_NET_ADHOC_DISCONNECTED, or SCE_NET_ERROR_EINTR, then the actual size of the socket buffer that is stored is returned in *len*. When this function is used to send data successively, move the send-data position in *len*-sized increments before calling it again.

flag can be set to the following type.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
void *data;
int ret, id, datalen, offset, len;

// Provide datalen bytes of data to data
...

offset = 0;
while (offset < datalen) {
        // Other tasks

        len = datalen - offset;
        // Send in non-blocking mode (poll)
        ret = sceNetAdhocPtpSend(id, data + offset, &len, 0,
SCE_NET_ADHOC_F_NONBLOCK);
        if (ret < 0 && ret != SCE_ERROR_NET_ADHOC_WOULD_BLOCK) {
              // Error handling
        }
        offset += len;
}
```

See Also

sceNetAdhocPtpRecv()

sceNetAdhocPtpRecv

Receive data

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpRecv(
        int id,
        void *buf,
        int *len,
        unsigned int timeout,
        int flag
);
```

Arguments

id	Socket ID
buf	Pointer to receive buffer
len	Receive buffer size (IN) receive data size (OUT)
timeout	Timeout (μsec)
flag	Option

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_WOULD_BLOCK	0x80410709	Wait state was entered
SCE_ERROR_NET_ADHOC_DISCONNECTED	0x8041070c	Socket was disconnected
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred

Description

This function receives data. If there is no data to be received, the function call blocks until data is received or until the interval specified by the *timeout* elapses. If 0 is specified for *timeout*, the function will wait indefinitely. When a timeout occurs, this function returns SCE_ERROR_NET_ADHOC_TIMEOUT.

If SCE_NET_ADHOC_F_NONBLOCK is specified as a receive option, and if there is no data to be received, SCE_ERROR_NET_ADHOC_WOULD_BLOCK is returned immediately regardless of the *timeout* specification.

When the function completes normally, the actual size of the contents of the receive buffer is returned in *len*.

flag can be set to the following type.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
char buf[1024];
int ret, id, len;
struct SceNetEtherAddr addr;
SceUShort16 port;

// receive 1024 bytes
needlen = 1024;
if (offset < needlen) {
        // Other tasks

        len = needlen - offset;
        // Receive in non-blocking mode (poll)
        ret = sceNetAdhocPtpRecv(id, buf + offset, &len, 0,
SCE_NET_ADHOC_F_NONBLOCK);
        if (ret < 0) {
              // Error handling
              break;
        }
        offset += len;
}
```

sceNetAdhocPtpFlush

Flush send buffer

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpFlush(
        int id,
        unsigned int timeout,
        int flag
);
```

Arguments

id	Socket ID
timeout	Timeout (μsec)
flag	Option

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_SOCKET_DELETED	0x80410707	Socket was deleted
SCE_ERROR_NET_ADHOC_SOCKET_ALERTED	0x80410708	Alert was set
SCE_ERROR_NET_ADHOC_WOULD_BLOCK	0x80410709	Wait state was entered
SCE_ERROR_NET_ADHOC_NOT_CONNECTED	0x8041070b	Socket has not been connected
SCE_ERROR_NET_ADHOC_DISCONNECTED	0x8041070c	Socket was disconnected
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred

Description

This function waits for data in the send buffer to be sent. If the data in the send buffer has not been completely sent, this function call blocks until the transmission has completed or until the interval specified by the *timeout* elapses. If 0 is specified for *timeout*, the function will wait indefinitely. When a timeout occurs, this function returns SCE_ERROR_NET_ADHOC_TIMEOUT.

If SCE_NET_ADHOC_F_NONBLOCK is specified as an option, and if the data in the send buffer has not been completely sent, SCE_ERROR_NET_ADHOC_WOULD_BLOCK is returned immediately regardless of the *timeout* specification.

flag can be set to the following type.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
int ret, id;

while (1) {
        // Other tasks

        // Call in non-blocking mode (poll)
        ret = sceNetAdhocPtpFlush(id, 0, SCE_NET_ADHOC_F_NONBLOCK);
        if (ret == SCE_ERROR_NET_ADHOC_WOULD_BLOCK) {
              // Sending
              continue;
        }
        else if (ret < 0) {
              // Error handling
              break;
        }
        else {
              // Send complete
              break;
        }
}
```

sceNetAdhocPtpClose

Break connection and delete PTP socket

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPtpClose(
        int id,
        int flag
);
```

Arguments

id	Socket ID
flag	Not used (set to 0)

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function breaks the PTP connection and deletes the socket. Any data present in either the send buffer or receive buffer is discarded.

If the relevant socket is in a wait state, the function will return an SCE_ERROR_NET_ADHOC_SOCKET_DELETED error.

Examples

```
int ret, id;

ret = sceNetAdhocPtpClose(id, 0);
if (ret < 0){
        // Error handling
}
```

sceNetAdhocGetPtpStat

Get PTP control blocks

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocGetPtpStat(
        int *buflen,
        void *buf
);
```

Arguments

buflen	Buffer size of *buf* (IN), list size (OUT)
buf	Pointer to area for storing the list of PTP control block structures.

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function gets a list of PTP control block structures.

buflen size in the area specified by *buf* must be allocated.

This function takes two kinds of action, depending on the value of *buf*.

- If *buf* is NULL, the function returns in *buflen* the size of the buffer needed to store the list of structures.
- If *buf* contains an address value, a list of SceNetAdhocPtpStat PTP control block structures is created up to the maximum buffer size specified in *buflen*, in the area specified by *buf*. The list has *buf* as its starting location, is linked using the *next* member, and is terminated by NULL.

Examples

```
int ret, buflen;
struct SceNetAdhocPtpStat *buf, *ptr;

ret = sceNetAdhocGetPtpStat(&buflen, NULL);
if (ret < 0) {
        // Error handling
        return;
}
else if (buflen == 0) {
        // The data did not exist
        return;
}

buf = malloc(buflen);
if (buf == NULL) {
        // Memory could not be allocated
        return;
}

ret = sceNetAdhocGetPtpStat(&buflen, buf);
if (ret < 0) {
        // Error handling
}
else if (buflen == 0) {
        // The data did not exist
}
else {
        for (ptr = buf; ptr != NULL; ptr = ptr->next) {
              // Process obtained control block
              ...
        }
}

free(buf);
```

See Also

SceNetAdhocPtpStat

PTP/PDP Common Functions

sceNetAdhocPollSocket

Synchronous I/O over multiple sockets

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocPollSocket(
        struct SceNetAdhocPollSd *sds,
        int nsds,
        SceUInt32 timeout,
        int flag
);
```

Arguments

sds	Pointer to poll socket descriptor array
nsds	*sds* array count
timeout	Timeout (μsec)
flag	Option

Return Values

Returns a value equal to or greater than 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized
SCE_ERROR_NET_ADHOC_TIMEOUT	0x80410715	Timeout occurred
SCE_ERROR_NET_ADHOC_EXCEPTION_EVENT	0x80410717	Exception event was generated

Description

This function performs synchronous I/O across multiple PDP sockets. The socket ID to be synchronized and the target events are specified through the array of poll socket descriptors.

The function call blocks until at least one event is generated. If SCE_NET_ADHOC_F_NONBLOCK is set as an option, the function returns 0 without waiting for an event to be generated. The generated events are stored in the *revents* member of the poll socket descriptor.

flag can be the following type.

Macro	Description
SCE_NET_ADHOC_F_NONBLOCK	Non-blocking mode

Examples

```
int ret, id0, id1, nsds;
struct SceNetAdhocPollSd sds[2];

...

// Wait for a receive event on sockets id0 and id1
sds[0].id = id0;
sds[0].events |= SCE_NET_ADHOC_EV_RECV;
sds[1].id = id1;
sds[1].events |= SCE_NET_ADHOC_EV_RECV;

nsds = 2;

while (1) {
        ret = sceNetAdhocPollSocket(sds, nsds, 0, 0);
        if (ret < 0) {
              // Error handling
              break;
        }

        for (i = 0; i < nsds; i++) {
        if (sds[i].revents & SCE_NET_ADHOC_EV_RECV) {
                    // Perform receive processing for sds[i].id
              }
        }
}
```

Notes

Even when immediately making a synchronous I/O call for an event returned with a poll socket descriptor array, processing is not necessarily handled without blocking. To call synchronous I/O without blocking, always specify the non-blocking mode.

If the target socket is in the alert state or if it is deleted, the function returns `SCE_ERROR_NET_ADHOC_EXCEPTION_EVENT`. The socket for which the event occurred can be confirmed by referencing the `SCE_NET_ADHOC_EV_ALERT` or `SCE_NET_ADHOC_EV_DELETE` flag to *`revents`*.

See Also

`SceNetAdhocPollSd`

sceNetAdhocSetSocketAlert

Set alert state

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocSetSocketAlert(
        int id,
        int flag
);
```

Arguments

id Socket ID
flag Alert setting option

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function sets the target socket to an alert state. The desired type of call can be set using the alert-setting option. An error (SCE_ERROR_NET_ADHOC_SOCKET_ALERTED) is returned until the alert state has been cleared for the specified call. Blocking is removed on blocked calls.

flag can be set to the following type.

Macro	Description
SCE_NET_ADHOC_F_ALERTSEND	Send
SCE_NET_ADHOC_F_ALERTRECV	Receive
SCE_NET_ADHOC_F_ALERTCONNECT	Connection established (active open)
SCE_NET_ADHOC_F_ALERTACCEPT	Connection established (passive open)
SCE_NET_ADHOC_F_ALERTFLUSH	Send complete
SCE_NET_ADHOC_F_ALERTPOLL	Polling
SCE_NET_ADHOC_F_ALERTALL	All

Examples

```
int ret, id;

// Set all alert flags on the socket ID
ret = sceNetAdhocSetSocketAlert(id, SCE_NET_ADHOC_F_ALERTALL);
if (ret < 0) {
        // Error handling
}
```

See Also

sceNetAdhocGetSocketAlert()

sceNetAdhocGetSocketAlert

Get alert state

Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocGetSocketAlert(
        int id,
        int *flag
);
```

Arguments

id	Socket ID
flag	Alert setting option

Return Values

Returns 0 for normal termination.

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Macro	Value	Description
SCE_ERROR_NET_ADHOC_INVALID_SOCKET_ID	0x80410701	Invalid socket ID was specified
SCE_ERROR_NET_ADHOC_INVALID_ARG	0x80410711	Invalid argument was specified
SCE_ERROR_NET_ADHOC_NOT_INITIALIZED	0x80410712	Library not initialized

Description

This function gets the specified alert state for a socket.

flag can be set to the following type.

Macro	Description
SCE_NET_ADHOC_F_ALERTSEND	Send
SCE_NET_ADHOC_F_ALERTRECV	Receive
SCE_NET_ADHOC_F_ALERTCONNECT	Connection established (active open)
SCE_NET_ADHOC_F_ALERTACCEPT	Connection established (passive open)
SCE_NET_ADHOC_F_ALERTFLUSH	Send complete
SCE_NET_ADHOC_F_ALERTPOLL	Polling

Examples

```
int ret, id, flag;

// Store alert flag for socket id in flag
ret = sceNetAdhocGetSocketAlert(id, &flag);
if (ret < 0) {
        // Error handling
}
```

See Also

sceNetAdhocSetSocketAlert()