

# **PSPNET adhocctl Library Reference**

© 2014 Sony Computer Entertainment Inc.  
All Rights Reserved.  
SCE Confidential

## Table of Contents

<b>Structures.....</b>	<b>3</b>
SceNetAdhocctlAdhocId .....	4
SceNetAdhocctlGroupName .....	5
SceNetAdhocctlNickname .....	6
SceNetAdhocctlPeerInfo .....	7
SceNetAdhocctlParameter .....	8
SceNetAdhocctlBSSId .....	9
<b>Initialization/Termination Functions.....</b>	<b>10</b>
sceNetAdhocctlInit .....	11
sceNetAdhocctlTerm .....	12
<b>Connection Information Getting/Setting Functions .....</b>	<b>13</b>
sceNetAdhocctlGetAdhocId .....	14
sceNetAdhocctlGetPeerList .....	15
sceNetAdhocctlGetPeerInfo .....	17
sceNetAdhocctlGetAddrByName .....	18
sceNetAdhocctlGetNameByAddr .....	20
sceNetAdhocctlGetParameter .....	21
sceNetAdhocctlGetEtherAddr .....	22

# Structures

000004892117

SCE CONFIDENTIAL

# SceNetAdhocctlAdhocId

## Ad hoc ID structure

### Definition

```
#include <pspnet_adhoc.h>
#define SCE_NET_ADHOCCTL_ADHOCID_LEN          9
#define SCE_NET_ADHOCCTL_ADHOCTYPE_PRODUCT_ID 0
#define SCE_NET_ADHOCCTL_ADHOCTYPE_RESERVED  1
#define SCE_NET_ADHOCCTL_ADHOCTYPE_SYSTEM     2
struct SceNetAdhocctlAdhocId {
    int type;
    SceChar8 data[SCE_NET_ADHOCCTL_ADHOCID_LEN];
    SceUChar8 padding[3];
};
```

### Members

*data*            Ad hoc ID data  
*type*            Ad hoc ID type  
*padding*        Padding data

### Description

This structure specifies an ad hoc ID. It is specified as an argument to the PSPNET adhocctl library initialization function `sceNetAdhocctlInit()`.

For *data*, as a rule, specify the title product number (four alphabetic characters and five numeric characters).

Further, contact SCE if there is a need to specify a value other than the title's product number, such as for communication between different applications.

The attributes of the ad hoc ID are indicated by *type*.

For *type*, specify one of the following values.

Macro	Description
<code>SCE_NET_ADHOCCTL_ADHOCTYPE_PRODUCT_ID</code>	Product ad hoc ID
<code>SCE_NET_ADHOCCTL_ADHOCTYPE_RESERVED</code>	Ad hoc ID for debugging
<code>SCE_NET_ADHOCCTL_ADHOCTYPE_SYSTEM</code>	Ad hoc ID for the system

### See Also

`sceNetAdhocctlInit()`, `sceNetAdhocctlGetAdhocId()`

SCE CONFIDENTIAL

---

# SceNetAdhocctlGroupName

---

## Group name structure

### Definition

---

```
#include <pspnet_adhoc.h>
#define SCE_NET_ADHOCCTL_GROUPNAME_LEN    8
struct SceNetAdhocctlGroupName {
    SceChar8 data[SCE_NET_ADHOCCTL_GROUPNAME_LEN];
};
```

### Members

---

*data*    Group name

### Description

---

This structure specifies a group name.

Multiple peers can participate in the same Independent Basic Service Set (IBSS) by using the same `SceNetAdhocctlGroupName`.

### See Also

---

`SceNetAdhocctlParameter`

SCE CONFIDENTIAL

---

# SceNetAdhocctlNickname

---

## Nickname structure

### Definition

---

```
#include <pspnet_adhoc.h>
#define SCE_NET_ADHOCCTL_NICKNAME_LEN      128
struct SceNetAdhocctlNickname {
    SceChar8 data[SCE_NET_ADHOCCTL_NICKNAME_LEN];
};
```

### Members

---

*data*    Nickname data

### Description

---

This structure stores the nickname which is specified in the system settings.  
When signed-up to Sony Entertainment Network, the Online ID will be used.

### See Also

---

```
sceNetAdhocctlGetParameter(), sceNetAdhocctlGetAddrByName(),
sceNetAdhocctlGetNameByAddr()
```

SCE CONFIDENTIAL

---

# SceNetAdhocctlPeerInfo

---

## Peer information structure

### Definition

---

```
#include <pspnet_adhoc.h>
struct SceNetAdhocctlPeerInfo {
    struct SceNetAdhocctlPeerInfo *next;
    struct SceNetAdhocctlNickname nickname;
    struct SceNetEtherAddr macAddr;
    SceUChar8 padding[6];
    SceUInt64 lastRecv;
};
```

### Members

---

<i>next</i>	Next entry in the list (NULL indicates end)
<i>nickname</i>	Nickname
<i>macAddr</i>	MAC address
<i>padding</i>	Padding data
<i>lastRecv</i>	System time when last packet was received (value obtained from <code>sceKernelGetProcessTimeWide()</code> )

### Description

---

This structure represents peer information.

### See Also

---

`sceNetAdhocctlGetPeerList()`, `sceNetAdhocctlGetAddrByName()`

SCE CONFIDENTIAL

---

# SceNetAdhocctlParameter

---

## PSPNET ad hoc communication mode parameter structure

### Definition

---

```
#include <pspnet_adhoc.h>

struct SceNetAdhocctlParameter {
    int channel;
    struct SceNetAdhocctlGroupName groupName;
    struct SceNetAdhocctlBSSID bssid;
    struct SceNetAdhocctlNickname nickname;
};
```

### Members

---

<i>channel</i>	Channel
<i>groupName</i>	Group name
<i>bssid</i>	BSSID
<i>nickname</i>	Nickname

### Description

---

This structure represents the parameters for PSPNET ad hoc communication mode.

### See Also

---

`sceNetAdhocctlGetParameter()`



SCE CONFIDENTIAL

---

# SceNetAdhocctlBSSID

---

## BSS ID structure

### Definition

---

```
#include <pspnet_adhoc.h>
#define SCE_NET_ADHOCCTL_BSSID_LEN 6
struct SceNetAdhocctlBSSID {
    SceUChar8 data[SCE_NET_ADHOCCTL_BSSID_LEN];
    SceUChar8 padding[2];
};
```

### Members

---

<i>data</i>	BSS ID data
<i>padding</i>	Padding data

### Description

---

This structure represents the BSS ID for IBSS.

### See Also

---

SceNetAdhocctlParameter, sceNetAdhocctlGetParameter()

# Initialization/Termination Functions

SCE CONFIDENTIAL

# sceNetAdhocctlInit

## Initialize library

### Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlInit(
    const struct SceNetAdhocctlAdhocId *adhoc_id
);
```

### Arguments

*adhoc\_id* Ad hoc ID

### 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_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_ALREADY_INITIALIZED	0x80410b07	Library already initialized

### Description

This function initializes the PSPNET adhocctl Library.

It creates an internal thread and starts it.

For *adhoc\_id*, specify the assigned ad hoc ID.

### Notes

This function is not multithread safe.

### Examples

```
#include <pspnet_adhoc.h>

int ret;
struct SceNetAdhocctlAdhocId adhoc_id;

adhoc_id.type = SCE_NET_ADHOCCTL_ADHOC_TYPE_PRODUCT_ID;
memcpy(adhoc_id.data, "123456789", SCE_NET_ADHOCCTL_ADHOCID_LEN);

ret = sceNetAdhocctlInit(&adhoc_id);
if(ret < 0){
    // Error handling
}
```

### See Also

sceNetAdhocctlTerm()

SCE CONFIDENTIAL

---

# sceNetAdhocctlTerm

---

## Terminate library

### Definition

---

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

### Arguments

---

None

### Return Values

---

Returns 0 for normal termination.

### Description

---

This function terminates the PSPNET adhocctl library.

### Notes

---

This function is not multithread safe.

### Examples

---

```
int ret;

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

### See Also

---

sceNetAdhocctlInit()

# Connection Information Getting/Setting Functions

SCE CONFIDENTIAL

# sceNetAdhocctlGetAdhocId

## Get ad hoc ID

### Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlGetAdhocId(
    struct SceNetAdhocctlAdhocId *adhoc_id
);
```

### Arguments

*adhoc\_id* Location where the obtained ad hoc ID 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_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED	0x80410b08	Library not initialized

### Description

This function gets the ad hoc ID which has been set in the PSPNET adhocctl library.

### Examples

```
int ret;
struct SceNetAdhocctlAdhocId adhoc_id;

ret = sceNetAdhocctlGetAdhocId(&adhoc_id);
if (ret < 0) {
    // Error handling
}
```

### See Also

SceNetAdhocctlAdhocId, sceNetAdhocctlInit()

# sceNetAdhocctlGetPeerList

Get list of information about other peers

## Definition

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

## Arguments

*buflen* Buffer size of *buf* (IN), list size (OUT)  
*buf* Pointer to the area where information on other peers 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_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED	0x80410b08	Library not initialized

## Description

This function gets all information on other peers which are present on the IBSS to which the library is connected.

An area of size *buflen* must be allocated in advance at the location indicated by *buf*.

This function exhibits two types of behavior depending on the value specified by *buf*.

- If NULL is specified for *buf*, then the buffer size necessary in order to obtain the structure list is returned to *buflen*.
- If an address value is specified for *buf*, then the `SceNetAdhocctlPeerInfo` structure data held in the library is copied into *buf*, up to the maximum buffer size specified by *buflen*. The list is linked using the *next* member, and is terminated with a NULL.

SCE CONFIDENTIAL

---

## Examples

---

```
int buflen;
struct SceNetAdhocctlPeerInfo *buf, *ptr;

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

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

ret = sceNetAdhocctlGetPeerList(&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) {
        // List processing
    }
}
free(buf);
```

## See Also

---

SceNetAdhocctlPeerInfo, sceNetAdhocctlGetPeerInfo(),  
sceNetAdhocctlGetAddrByName(), sceNetAdhocctlGetNameByAddr()



SCE CONFIDENTIAL

# sceNetAdhocctlGetPeerInfo

Get information about another peer

## Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlGetPeerInfo (
    const struct SceNetEtherAddr *addr,
    int size,
    struct SceNetAdhocctlPeerInfo *peer_info
);
```

## Arguments

*addr*            MAC address of peer for which information is to be obtained.  
*size*            Size of `SceNetAdhocctlPeerInfo`  
*peer\_info*       Pointer to structure where peer information 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
<code>SCE_ERROR_NET_ADHOC_NO_ENTRY</code>	0x80410716	Specified entry could not be found
<code>SCE_ERROR_NET_ADHOCCTL_INVALID_ARG</code>	0x80410b04	Invalid argument was specified
<code>SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED</code>	0x80410b08	Library not initialized

## Description

This function searches information on all peers connected to the IBSS that the library has connected, and obtains information on the peer with the specified MAC address. In *size* specify the size of the peer information structure specified by *peer\_info* (`SceNetAdhocctlPeerInfo`).

## Examples

```
struct SceNetEtherAddr addr;
struct SceNetAdhocctlPeerInfo peer_info;

// Set target MAC address in addr
...

ret = sceNetAdhocctlGetPeerInfo(&addr, sizeof(struct SceNetAdhocctlPeerInfo),
    &peer_info);
if (ret < 0) {
    // Error handling
}
```

## See Also

`SceNetAdhocctlPeerInfo`, `sceNetAdhocctlGetPeerList()`,  
`sceNetAdhocctlGetAddrByName()`, `sceNetAdhocctlGetNameByAddr()`

SCE CONFIDENTIAL

# sceNetAdhocctlGetAddrByName

Get MAC addresses corresponding to a nickname

## Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlGetAddrByName (
    const struct SceNetAdhocctlNickname *nickname,
    int *buflen,
    void *buf
);
```

## Arguments

*nickname*     Nickname  
*buflen*        Buffer size of *buf* (IN), list size (OUT)  
*buf*            Pointer to the area where information on other peers 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_NO_ENTRY	0x80410716	Specified entry could not be found
SCE_ERROR_NET_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED	0x80410b08	Library not initialized

## Description

This function searches for information on other peers present on the IBSS to which the library is connected, and obtains the MAC addresses which correspond to a nickname.

Nicknames can be assigned arbitrarily, so there will not necessarily be only one corresponding MAC address. Consequently, as with `sceNetAdhocctlGetPeerList()`, this information is obtained in the form of an `SceNetAdhocctlPeerInfo` structure list.

An area of size *buflen* must be allocated in advance at the location indicated by *buf*.

This function exhibits two types of behavior depending on the value specified by *buf*.

- If NULL is specified for *buf*, the buffer size necessary in order to obtain the structure list is returned to *buflen*.
- If an address value is specified for *buf*, then the `SceNetAdhocctlPeerInfo` structure data held in the library is copied into *buf*, up to the maximum buffer size specified by *buflen*. The list is linked using the *next* member, and is terminated with a NULL.

SCE CONFIDENTIAL

---

**Examples**

---

```

int buflen;
struct SceNetAdhocctlNickname nickname;
struct SceNetAdhocctlPeerInfo *buf, *ptr;

// Specify the user name to obtain in nickname
...

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

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

ret = sceNetAdhocctlGetAddrByName(&nickname, &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) {
        // List processing
    }
}

free(buf);

```

**See Also**

---

sceNetAdhocctlGetPeerList(), sceNetAdhocctlGetNameByAddr()

SCE CONFIDENTIAL

# sceNetAdhocctlGetNameByAddr

Get nickname corresponding to a MAC address

## Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlGetNameByAddr (
    const struct SceNetEtherAddr *addr,
    struct SceNetAdhocctlNickname *nickname
);
```

## Arguments

*addr*            MAC address  
*nickname*       Location where nickname 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_NO_ENTRY	0x80410716	Specified entry could not be found
SCE_ERROR_NET_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED	0x80410b08	Library not initialized

## Description

This function searches for information on other peers present on the IBSS to which the library is connected, and obtains the nickname which corresponds to a MAC address.

## Examples

```
struct SceNetEtherAddr addr;
struct SceNetAdhocctlNickname nickname;

// Specify the target MAC address in addr
...

ret = sceNetAdhocctlGetNameByAddr(&addr, &nickname);
if (ret < 0) {
    // Error handling
}
```

## See Also

sceNetAdhocctlGetPeerList(), sceNetAdhocctlGetAddrByName()

SCE CONFIDENTIAL

# sceNetAdhocctlGetParameter

Get various parameters of PSPNET ad hoc communication mode

## Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlGetParameter (
    struct SceNetAdhocctlParameter *parameter
);
```

## Arguments

*parameter*    Parameter structure

## 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_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED	0x80410b08	Library not initialized

## Description

This function obtains various types of parameters for the PSPNET ad hoc communication mode to which the library is connected.

## Examples

```
struct SceNetAdhocctlParameter parameter;

ret = sceNetAdhocctlGetParameter(&parameter);
if (ret < 0) {
    // Error handling
}
```

## See Also

SceNetAdhocctlParameter

SCE CONFIDENTIAL

# sceNetAdhocctlGetEtherAddr

Get own unit's Ethernet address

## Definition

```
#include <pspnet_adhoc.h>
int sceNetAdhocctlGetEtherAddr (
    struct SceNetEtherAddr *addr
);
```

## Arguments

*addr*                      Ethernet address

## 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_ADHOCCTL_INVALID_ARG	0x80410b04	Invalid argument was specified
SCE_ERROR_NET_ADHOCCTL_NOT_INITIALIZED	0x80410b08	Library not initialized

## Description

This function obtains your unit's Ethernet address.

On DevKits with **PS TV Emulation** set to **On** in **★Debug Settings** in the Settings application and on retail units, the address of the Wi-Fi device will return.

On DevKits with **PS TV Emulation** set **Off** and on TestKits with a USB Ethernet device connected, the Ethernet address of the USB Ethernet device will return. If a USB Ethernet device is not connected to a DevKit or TestKit, the address of the Wi-Fi device will return. This determination will be made upon a cold boot, so to change the behavior, change the connection status of the USB Ethernet device then reboot the DevKit or TestKit.

## Notes

This function is not multithread safe.

In SDKs earlier than 3.000, an error will occur for calls when PSPNET ad hoc communication is not connected.

## Examples

```
struct SceNetEtherAddr addr;

ret = sceNetAdhocctlGetEtherAddr(&addr);
if (ret < 0) {
    // Error handling
    return;
}
```

## See Also

SceNetEtherAddr

©SCEI