libnetctl Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

Constants

SCE_NET_CTL_STATE_XXX

Constant for network connection state
(common to Internet communication and ad hoc communication)

Definition

Macro	Value	Description
SCE_NET_CTL_STATE_DISCONNECTED	0	Disconnected
SCE_NET_CTL_STATE_CONNECTING	1	Connecting (to cable or wireless device)
SCE_NET_CTL_STATE_IPOBTAINING	2	Obtaining IP address
SCE_NET_CTL_STATE_IPOBTAINED	3	IP address obtained

Description

These constants indicate the network connection state for both Internet communication and ad hoc communication.

The network connection state for Internet communication can be obtained with `sceNetCtlInetGetState()`.

The network connection state for ad hoc communication can be obtained with `sceNetCtlAdhocGetState()`.

See Also

`sceNetCtlInetGetState()`,`sceNetCtlAdhocGetState()`

SCE_NET_CTL_EVENT_TYPE_XXX

Constant indicating event that has occurred
(common to Internet communication and ad hoc communication)

Definition

Macro	Value	Description
SCE_NET_CTL_EVENT_TYPE_DISCONNECTED	1	Disconnection occurred
SCE_NET_CTL_EVENT_TYPE_DISCONNECT_REQ_FINISHED	2	Disconnection completed
SCE_NET_CTL_EVENT_TYPE_IPOBTAINED	3	IP address obtained. This event does not occur if after the IP address has been obtained, the state changes from IPOBTAINED to CONNECTING through reconnection with the IP address held as is, and the IPOBTAINED state is again returned to.

Description

These constants indicate main transition events of network connection state for both Internet communication and ad hoc communication.

Network connection state transition events can be obtained with the Internet communication callback function or the ad hoc communication callback function.

See Also

SceNetCtlCallback

SCE_NET_CTL_DEVICE_XXX

Constant for network connection device

Definition

Macro	Value	Description
SCE_NET_CTL_DEVICE_WIRELESS	0	Wireless connection
SCE_NET_CTL_DEVICE_WIRED	1	Cable connection
SCE_NET_CTL_DEVICE_PHONE	2	3G connection

Description

These constants indicate the network connection device during Internet communication.
Network connection device can be obtained from *info->device* by specifying SCE_NET_CTL_INFO_DEVICE in sceNetCtlInetGetInfo().

See Also

sceNetCtlInetGetInfo(DEVICE)

SCE_NET_CTL_LINK_XXX

Constant for link connection state

Definition

Macro	Value	Description
SCE_NET_CTL_LINK_DISCONNECTED	0	Disconnected
SCE_NET_CTL_LINK_CONNECTED	1	Connected

Description

These constants indicate the link connection state during Internet communication.

Link connection state can be obtained from *info->link* by specifying SCE_NET_CTL_INFO_LINK in sceNetCtlInetGetInfo().

See Also

sceNetCtlInetGetInfo(LINK)

SCE_NET_CTL_WIFI_SECURITY_XXX

Constant for security measure taken for wireless connection

Definition

Macro	Value	Description
SCE_NET_CTL_WIFI_SECURITY_NOAUTH	0	No security
SCE_NET_CTL_WIFI_SECURITY_WEP	1	WEP
SCE_NET_CTL_WIFI_SECURITY_WPAPSK_WPA2PSK	2	(Not used)
SCE_NET_CTL_WIFI_SECURITY_WPAPSK_TKIP	3	WPA-PSK(TKIP)
SCE_NET_CTL_WIFI_SECURITY_WPAPSK_AES	4	WPA-PSK(AES)
SCE_NET_CTL_WIFI_SECURITY_WPA2PSK_TKIP	5	WPA2-PSK(TKIP)
SCE_NET_CTL_WIFI_SECURITY_WPA2PSK_AES	6	WPA2-PSK(AES)
SCE_NET_CTL_WIFI_SECURITY_UNSUPPORTED	7	(Not used)

Description

These constants indicate the wireless security during Internet communication.

Security measure for a wireless connection can be obtained from *info->wifi_security* by specifying SCE_NET_CTL_INFO_WIFI_SECURITY in sceNetCtlInetGetInfo().

See Also

sceNetCtlInetGetInfo(WIFI_SECURITY)

SCE_NET_CTL_IP_XXX

Constant for IP setting

Definition

Macro	Value	Description
SCE_NET_CTL_IP_DHCP	0	DHCP
SCE_NET_CTL_IP_STATIC	1	Static IP
SCE_NET_CTL_IP_PPPOE	2	PPPoE

Description

These contents indicate the IP setting for Internet communication.

IP setting can be obtained from *info->ip_config* by specifying SCE_NET_CTL_INFO_IP_CONFIG in sceNetCtlInetGetInfo().

See Also

sceNetCtlInetGetInfo(IP_CONFIG)

SCE_NET_CTL_HTTP_PROXY_XXX

Constant for HTTP proxy server setting

Definition

Macro	Value	Description
SCE_NET_CTL_HTTP_PROXY_OFF	0	Setting off
SCE_NET_CTL_HTTP_PROXY_ON	1	Setting on

Description

These constants indicate the HTTP proxy server setting for Internet communication.
HTTP proxy setting can be obtained from *info->http_proxy_config* by specifying SCE_NET_CTL_INFO_HTTP_PROXY_CONFIG in sceNetCtlInetGetInfo().

See Also

sceNetCtlInetGetInfo(HTTP_PROXY_CONFIG)

SCE_NET_CTL_INFO_XXX

Codes representing connection information to be obtained

Definition

Macro	Value	Description
SCE_NET_CTL_INFO_CNF_NAME	1	Connection name (*info->cnf_name*)
SCE_NET_CTL_INFO_DEVICE	2	Device (*info->device*)
SCE_NET_CTL_INFO_ETHER_ADDR	3	Ethernet address (*info->ether_addr*)
SCE_NET_CTL_INFO_MTU	4	MTU (*info->mtu*)
SCE_NET_CTL_INFO_LINK	5	Link connection state (*info->link*)
SCE_NET_CTL_INFO_BSSID	6	BSSID (*info->bssid*)
SCE_NET_CTL_INFO_SSID	7	SSID (*info->ssid*)
SCE_NET_CTL_INFO_WIFI_SECURITY	8	Security measure for wireless connection (*info->wifi_security*)
SCE_NET_CTL_INFO_RSSI_DBM	9	Receive signal strength indicator (dBm) (*info->rssi_dbm*)
SCE_NET_CTL_INFO_RSSI_PERCENTAGE	10	Receive signal strength indicator (%) (*info->rssi_percentage*)
SCE_NET_CTL_INFO_CHANNEL	11	Channel (*info->channel*)
SCE_NET_CTL_INFO_IP_CONFIG	12	IP setting (*info->ip_config*)
SCE_NET_CTL_INFO_DHCP_HOSTNAME	13	DHCP hostname (*info->dhcp_hostname*)
SCE_NET_CTL_INFO_PPPOE_AUTH_NAME	14	PPPoE user ID (*info->pppoe_auth_name*)
SCE_NET_CTL_INFO_IP_ADDRESS	15	IP address (*info->ip_address*)
SCE_NET_CTL_INFO_NETMASK	16	Net mask (*info->netmask*)
SCE_NET_CTL_INFO_DEFAULT_ROUTE	17	Default route (*info->default_route*)
SCE_NET_CTL_INFO_PRIMARY_DNS	18	Primary DNS (*info->primary_dns*)
SCE_NET_CTL_INFO_SECONDARY_DNS	19	Secondary DNS (*info->secondary_dns*)
SCE_NET_CTL_INFO_HTTP_PROXY_CONFIG	20	HTTP proxy server setting (*info->http_proxy_config*)
SCE_NET_CTL_INFO_HTTP_PROXY_SERVER	21	Hostname of HTTP proxy server (*info->http_proxy_server*)
SCE_NET_CTL_INFO_HTTP_PROXY_PORT	22	Port number of HTTP proxy server (*info->http_proxy_port*)

Description

These constants are used to specify the connection information for Internet communication, which is obtained with `sceNetCtlInetGetInfo()`. Specify one of the above macro to *code*, and the corresponding connection information will be returned to an applicable member of the variable specified in *info*.

See Also

`sceNetCtlInetGetInfo()`

SCE_NETCHECK_DIALOG_MODE_XXX

Operation mode of Network Check Dialog

Definition

Macro	Value	Description
SCE_NETCHECK_DIALOG_MODE_INVALID	0	Non-operation state
SCE_NETCHECK_DIALOG_MODE_ADHOC_CONN	1	Ad hoc connection mode
SCE_NETCHECK_DIALOG_MODE_PSN	2	PSN℠ mode
SCE_NETCHECK_DIALOG_MODE_PSN_ONLINE	3	PSN℠ online mode
SCE_NETCHECK_DIALOG_MODE_PS3_CONNECT	4	PlayStation®3 connection mode
SCE_NETCHECK_DIALOG_MODE_PSP_ADHOC_CONN	5	PSPNET ad hoc connection mode
SCE_NETCHECK_DIALOG_MODE_PSP_ADHOC_CREATE	6	PSPNET ad hoc creation mode
SCE_NETCHECK_DIALOG_MODE_PSP_ADHOC_JOIN	7	PSPNET ad hoc join mode

Description

These constants indicate the operation mode of Network Check Dialog specified with `sceNetCheckDialogInit()`.

See Also

`sceNetCheckDialogInit()`

SCE_NETCHECK_DIALOG_PS3_CONNECT_ACTION_XXX

PlayStation®3 connection mode operation action of Network Check Dialog

Definition

Macro	Value	Description
SCE_NETCHECK_DIALOG_PS3_CONNECT_ACTION_ENTER	0	Connects to PlayStation®3
SCE_NETCHECK_DIALOG_PS3_CONNECT_ACTION_LEAVE	1	Disconnects from PlayStation®3

Description

This constant represents the operation of either connecting to the PlayStation®3 or disconnecting from it when the PlayStation®3 connection mode is specified with `sceNetCheckDialogInit()`.

See Also

`SceNetCheckDialogPS3ConnectParam`, `sceNetCheckDialogInit()`

libnetctl Error Codes

List of error codes returned by libnetctl

Definition

Macro	Value	Description
SCE_NET_CTL_ERROR_NOT_INITIALIZED	0x80412101	Library not initialized
SCE_NET_CTL_ERROR_NOT_TERMINATED	0x80412102	Library not terminated
SCE_NET_CTL_ERROR_CALLBACK_MAX	0x80412103	No more callbacks can be registered
SCE_NET_CTL_ERROR_ID_NOT_FOUND	0x80412104	ID specified in argument not registered
SCE_NET_CTL_ERROR_INVALID_ID	0x80412105	ID specified in argument invalid
SCE_NET_CTL_ERROR_INVALID_CODE	0x80412106	Code specified in argument invalid
SCE_NET_CTL_ERROR_INVALID_ADDR	0x80412107	Address specified in argument invalid
SCE_NET_CTL_ERROR_NOT_CONNECTED	0x80412108	Not connected
SCE_NET_CTL_ERROR_NOT_AVAIL	0x80412109	Not in state where information can be obtained
SCE_NET_CTL_ERROR_AUTO_CONNECT_DISABLED	0x8041210a	Intermittent connection is disabled
SCE_NET_CTL_ERROR_AUTO_CONNECT_FAILED	0x8041210b	Intermittent connection failed
SCE_NET_CTL_ERROR_ NO_SUITABLE_SETTING_FOR_AUTO_CONNECT	0x8041210c	No connection that can be used for intermittent connection exists
SCE_NET_CTL_ERROR_ DISCONNECTED_FOR_ADHOC_USE	0x8041210d	Disconnected because ad hoc communication occurred
SCE_NET_CTL_ERROR_DISCONNECT_REQ	0x8041210e	Disconnected because of explicit disconnection request. This error code may be returned upon retrieval of the SCE_NET_CTL_EVENT_TYPE_ DISCONNECTED event with sceNetCtlInetGetResult(), sceNetCtlAdhocGetResult()
SCE_NET_CTL_ERROR_INVALID_TYPE	0x8041210f	Type specified by argument is invalid
SCE_NET_CTL_ERROR_AUTO_DISCONNECT	0x80412110	Disconnected due to no-communication timeout during intermittent connection. This error code may be returned upon retrieval of SCE_NET_CTL_EVENT_TYPE_DIS CONNECTED event result with sceNetCtlInetGetResult(), sceNetCtlAdhocGetResult()
SCE_NET_CTL_ERROR_INVALID_SIZE	0x80412111	Structure size specified in *size* member of the structure is invalid
SCE_NET_CTL_ERROR_FLIGHT_MODE_ENABLED	0x80412112	Communication error occurred because the Flight Mode is set to ON
SCE_NET_CTL_ERROR_WIFI_DISABLED	0x80412113	Communication error occurred because Wi-Fi is set to OFF
SCE_NET_CTL_ERROR_WIFI_IN_ADHOC_USE	0x80412114	Communication error occurred because Wi-Fi is set to ad hoc mode

Macro	Value	Description
SCE_NET_CTL_ERROR_ETHERNET_PLUGOUT	0x80412115	Ethernet cable has been disconnected during use of the USB Ethernet for development
SCE_NET_CTL_ERROR_WIFI_DEAUTHED	0x80412116	Disconnected from the access point while in connected state during Internet communication mode
SCE_NET_CTL_ERROR_WIFI_BEACON_LOST	0x80412117	Became unable to receive the beacon from the access point while in connected state during Internet communication mode
SCE_NET_CTL_ERROR_DISCONNECTED_FOR_SUSPEND	0x80412118	Disconnection occurred due to system suspension while in connected state
SCE_NET_CTL_ERROR_COMMUNICATION_ID_NOT_EXIST	0x80412119	`SceNpCommunicationId` not specified during ad hoc connection
SCE_NET_CTL_ERROR_ADHOC_ALREADY_CONNECTED	0x8041211a	Further connection was attempted while in ad hoc connection state. Disconnect the ad hoc network by calling `sceNetCtlAdhocDisconnect()`
SCE_NET_CTL_ERROR_DHCP_TIMEOUT	0x8041211b	IP address acquisition in DHCP timed out (60 seconds)
SCE_NET_CTL_ERROR_PPPOE_TIMEOUT	0x8041211c	IP address acquisition in PPPoE timed out (60 seconds)
SCE_NET_CTL_ERROR_INSUFFICIENT_MEMORY	0x8041211d	The amount of memory of the libnet is insufficient. Increase the size of the memory specified with `sceNetInit()`
SCE_NET_CTL_ERROR_PSP_ADHOC_JOIN_TIMEOUT	0x8041211e	Timeout occurred in PSPNET ad hoc join mode of Network Check Dialog
SCE_NET_CTL_ERROR_UNKNOWN_DEVICE	0x80412188	Device specified by argument is invalid

Network Check Dialog Error Codes

List of error codes returned by Network Check Dialog

Definition

Macro	Value	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Calling another common dialog function
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified as the function's argument
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	Parameter error
SCE_COMMON_DIALOG_ERROR_NOT_RUNNING	0x80020404	Called during a period other than SCE_COMMON_DIALOG_STATUS_RUNNING
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during a period other than SCE_COMMON_DIALOG_STATUS_FINISHED
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceNetCheckDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM	0x80020433	There is an invalid setting in *commonParam.infobarParam*
SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR	0x80020434	There is an invalid setting in *commonParam.bgColor*
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_NETCHECK_DIALOG_ERROR_PARAM	0x80100c01	Parameter specified by argument is invalid
SCE_NETCHECK_DIALOG_ERROR_INVALID_MODE	0x80100c02	Start mode specified by argument is invalid
SCE_NETCHECK_DIALOG_ERROR_LACK_OF_LIBHTTP_POOL_SIZE	0x80100c03	Remaining amount of the libhttp memory pool is insufficient
SCE_NETCHECK_DIALOG_ERROR_LACK_OF_LIBSSL_POOL_SIZE	0x80100c04	Remaining amount of the libssl memory pool is insufficient
SCE_NETCHECK_DIALOG_ERROR_LATEST_PATCH_PKG_EXIST	0x80100c05	A new version update file exists
SCE_NETCHECK_DIALOG_ERROR_SIGN_OUT	0x80100c06	Sign-in processing failed
SCE_NETCHECK_DIALOG_ERROR_INVALID_PSPADHOC_PARAM	0x80100c07	Invalid parameter specified for PSPNET ad hoc connection mode
SCE_NETCHECK_DIALOG_ERROR_INVALID_TIMEOUT_PARAM	0x80100c08	Invalid timeout value specified for PSPNET ad hoc join mode
SCE_NETCHECK_DIALOG_ERROR_PSN_AGE_RESTRICTION	0x80100c09	Applicable to age restriction

Datatypes

SceNetCtlInfo

Network connection information union related to Internet communication

Definition

```
#include <net.h>
typedef union SceNetCtlInfo {
        char cnf_name[SCE_NET_CTL_CNF_NAME_LEN];
        SceUInt device;
        SceNetEtherAddr ether_addr;
        SceUInt mtu;
        SceUInt link;
        SceNetEtherAddr bssid;
        char ssid[SCE_NET_CTL_SSID_LEN];
        SceUInt wifi_security;
        uint8_t rssi_dbm;
        uint8_t rssi_percentage;
        uint8_t channel;
        SceUInt ip_config;
        char dhcp_hostname[SCE_NET_CTL_HOSTNAME_LEN];
        char pppoe_auth_name[SCE_NET_CTL_AUTH_NAME_LEN];
        char ip_address[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
        char netmask[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
        char default_route[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
        char primary_dns[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
        char secondary_dns[SCE_NET_CTL_IPV4_ADDR_STR_LEN];
        SceUInt http_proxy_config;
        char http_proxy_server[SCE_NET_CTL_HOSTNAME_LEN];
        uint16_t http_proxy_port;
} SceNetCtlInfo;
```

Members

cnf_name	Name assigned to network setting used for connection
device	Device used for Internet communication (cable or wireless)
ether_addr	Ethernet address of interface used for Internet communication
mtu	Interface MTU
link	Link connection state (disconnected or connected)
bssid	BSSID of connection-target-AP
ssid	SSID of connection-target-AP
wifi_security	Security measure for wireless connection
rssi_dbm	Receive signal strength indicator (dBm)
rssi_percentage	Receive signal strength indicator (%)
channel	Channel of access point
ip_config	Method to obtain IP address (automatic, manual, PPPoE)
dhcp_hostname	Hostname for DHCP
pppoe_auth_name	User ID for PPPoE
ip_address	IP address
netmask	Net mask
default_route	Default router
primary_dns	IP address of primary DNS
secondary_dns	IP address of secondary DNS
http_proxy_config	HTTP proxy server setting (use or don't use)
http_proxy_server	Hostname of HTTP proxy server
http_proxy_port	Port number of HTTP proxy server

Description

This datatype indicates the network connection information of Internet communication.

You can obtain network connection information from a variable of this type when calling `sceNetCtlInetGetInfo()`.

See Also

`sceNetCtlInetGetInfo()`

SceNetCtlAdhocPeerInfo

Terminal information for Ad hoc communication

Definition

```
#include <net.h>
typedef struct SceNetCtlAdhocPeerInfo {
        struct SceNetInAddr inetAddr;
        SceNpId npId;
        SceUInt64 lastRecv;
        int appVer;
        SceBool isValidNpId;
        char username[SCE_SYSTEM_PARAM_USER_NAME_MAXSIZE];
        char padding[7];
} SceNetCtlAdhocPeerInfo;
```

Members

inetAddr	IP address of the other connected terminal
npId	SceNpId of the other connected terminal
lastRecv	System time when the last packet was received (The value acquired with sceKernelGetProcessTimeWide())
appVer	Application version of the other connected terminal (In the current SDK, this is an invalid value.)
isValidNpId	Validity of *npId* of the other connected terminal The value will be SCE_TRUE if sign-up procedure has been completed
username	User name of the other connected terminal that can be acquired with the application utility library
padding	Padding data

Description

This datatype indicates the information of the terminal connected for ad hoc communication.

The information of the other ad hoc connected terminals is obtained by calling sceNetCtlAdhocGetPeerList().

User name that can be acquired with the application utility library is set to *username*. If the other connected terminal has already signed up, the value set in SceNpOnlineId will be set to *username*, and if not, a user name stored locally will be set.

See Also

sceNetCtlAdhocGetPeerList()

SceNetCtlNatInfo

NAT router information structure

Definition

```
#include <net.h>
typedef struct SceNetCtlNatInfo {
        SceSize size;
        int stun_status;
        int nat_type;
        SceNetInAddr mapped_addr;
} SceNetCtlNatInfo;
```

Members

size	The size of the structure An appropriate size must be stored by the caller.
stun_status	STUN status
nat_type	NAT type
mapped_addr	The IP address of the PlayStation®Vita system seen from the global Internet. Since *mapped_addr* is retrieved with STUN, 0 will return if STUN has not yet started. A valid value will return to *mapped_addr* while *stun_status* is SCE_NET_CTL_NATINFO_STUN_UNCHECKED.

Description

This is a structure indicating NAT router information. It is retrieved with sceNetCtlGetNatInfo().

Calling must be performed after storing the size of this structure in *size*.

One of the following values indicating the STUN status is returned to *stun_status*.

Since STUN results are added to the NAT type, normally it is not necessary for the application to get STUN status separately.

Macro	Value	Description
SCE_NET_CTL_NATINFO_STUN_UNCHECKED	0	STUN is not yet complete
SCE_NET_CTL_NATINFO_STUN_FAILED	1	STUN failed
SCE_NET_CTL_NATINFO_STUN_OK	2	STUN was successful

Either of the followings is passed to *nat_type*.

Macro	Value	Description
SCE_NET_CTL_NATINFO_NAT_TYPE_1	1	NAT type 1
SCE_NET_CTL_NATINFO_NAT_TYPE_2	2	NAT type 2
SCE_NET_CTL_NATINFO_NAT_TYPE_3	3	NAT type 3

See Also

sceNetCtlGetNatInfo()

SceNetCheckDialogPS3ConnectParam

PlayStation®3 connection mode start-up parameters of Network Check Dialog

Definition

```
#include <common_dialog.h>
#include <common_dialog/netcheck.h>
typedef struct SceNetCheckDialogPS3ConnectParam {
        SceNetCheckDialogPS3ConnectAction action;
        char ssid[SCE_NET_CTL_SSID_LEN];
        char wpaKey[SCE_NET_CTL_WIFI_SECURITY_KEY_LEN];
        char titleId[9 + 1];
} SceNetCheckDialogPS3ConnectParam;
```

Members

action	PlayStation®3 connection mode operation action Specify SCE_NETCHECK_DIALOG_PS3_CONNECT_ACTION_XXX
ssid	SSID of the PlayStation®3 access point to connect to
wpaKey	WPA key of the PlayStation®3 access point to connect to
titleId	TITLE ID of the title expecting connection running on the PlayStation®3

Description

These are the necessary parameters for running Network Check Dialog in PlayStation®3 connection mode. Set the necessary parameters, and specify the pointer to the structure in the ps3ConnectParam member of SceNetCheckDialogParam.

In *ssid*, *wpaKey* and *titleId*, specify the same values used when starting up the access point with CELL_SYSUTIL_AP_TYPE_USE_GAME_SETTING in the access point utility of the PlayStation®3.

See Also

PlayStation®3 "Access Point Utility Overview"

PlayStation®3 "Access Point Utility Reference"

SceNetCheckDialogAgeRestriction

Structure storing age restriction per country/region to be used by Network Check Dialog

Definition

```
#include <common_dialog.h>
#include <common_dialog/netcheck.h>
typedef struct SceNetCheckDialogAgeRestriction {
        char countryCode[SCE_NETCHECK_DIALOG_COUNTRY_CODE_LEN];
        SceInt8 age;
        SceInt8 padding;
} SceNetCheckDialogAgeRestriction;
```

Members

countryCode	Country/region code (2 bytes)
age	Age restriction
padding	Unused

Description

This datatype represents the age restriction per country/region to be used by Network Check Dialog.

Set this structure when you want to specify an individual age restriction for a country/region separate from the default age restriction specified with the *defaultAgeRestriction* member of SceNetCheckDialogParam.

For *countryCode*, specify the 2-character country/region code (excluding the NULL code).

The country/region code to specify is the same as SceNpCountryCode.

For details, refer to the "NP Library Reference" document.

For *age*, specify the age at which to set a restriction for the country specified in *countryCode*. A restriction will be placed on ages under the specified value.

See Also

sceNetCheckDialogInit()

SceNetCheckDialogParam

Startup parameter of Network Check Dialog

Definition

```
#include <common_dialog.h>
#include <common_dialog/netcheck.h>
typedef struct SceNetCheckDialogParam {
        SceUInt32 sdkVersion;
        SceCommonDialogParam commonParam;
        SceNetCheckDialogMode mode;
        SceNpCommunicationId npCommunicationId;
        SceNetCheckDialogPS3ConnectParam *ps3ConnectParam;
        SceNetAdhocctlGroupName *groupName;
        SceUInt32 timeoutUs;
        SceInt8 defaultAgeRestriction;
        SceInt8 padding[3];
        SceInt32 ageRestrictionCount;
        const SceNetCheckDialogAgeRestriction *ageRestriction;
        SceUInt8 reserved[104];
} SceNetCheckDialogParam;
```

Members

sdkVersion	Area to be used in the library. Applications need not refer this directly.
commonParam	Common parameters for Common Dialogs
mode	Operation mode of Network Check Dialog Specify SCE_NETCHECK_DIALOG_MODE_XXX.
npCommunicationId	NP Communication ID Specify the value issued on PlayStation®Vita Developer Network
ps3ConnectParam	Pointer to the start-up parameters of the PlayStation®3 connection Specify NULL if not using the PlayStation®3 connection mode.
groupName	Group name used with PSPNET ad hoc connection mode
timeoutUs	Timeout value used for PSPNET ad hoc join mode
defaultAgeRestriction	Default age restriction Set the default age restriction.
padding	Unused
ageRestrictionCount	Total number of age restrictions set by country/region Set the total number of age restrictions specified per country/region. Set 0 when not setting age restriction per country/region.
ageRestriction	List of age restrictions by country/region Set a pointer to the list of age restrictions specified per country/region. Set NULL when not specifying age restriction by country/region.
reserved	Unused

Description

This datatype indicates the startup parameter of Network Check Dialog.

It is initialized with `sceNetCheckDialogParamInit()` and specified with `sceNetCheckDialogInit()` upon Network Check Dialog starting.

If starting up Network Check Dialog in the ad hoc connection mode, specify the value issued through application on the PlayStation®Vita Developer Network (https://psvita.scedev.net/) in *`npCommunicationId`*. If you wish to create a group without interferences within the same title, specify an arbitrary value in the *`num`* member of *`npCommunicationId`*. Communication will only be enabled between terminals for which the same value has been specified.

Specify the common parameters for Common Dialogs in *`commonParam`*.

Given that it is not possible to perform specifications concerning the info bar in Network Check Dialog, always specify NULL in *`commonParam.infobarParam`*. If a value other than NULL is specified, `sceNetCheckDialogInit()` will return `SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM`. For details, refer to the "Common Dialog Reference" document.

When starting up in PlayStation®3 connection mode, be sure to set the pointer to the PlayStation®3 connection start-up parameters in *`ps3ConnectParam`*.

`groupName` is a parameter used with PSPNET ad hoc connection mode. PSPNET ad hoc communication can be performed between terminals with the same *`groupName`*. Specifying NULL is also possible.

`timeoutUs` is a parameter used with PSPNET ad hoc join mode. Be sure to specify the timeout value for participation processing. If this value is not specified, `SCE_NETCHECK_DIALOG_ERROR_INVALID_TIMEOUT_PARAM` will be returned by `sceNetCheckDialogInit()`.

For *`defaultAgeRestriction`*, specify the default age restriction.

To specify an individual age restriction for a country/region, specify the number of age restrictions to specify for countries/regions to *`ageRestrictionCount`* and specify the pointer to a list of age restrictions for each country/region to *`ageRestriction`*.

See Also

`sceNetCheckDialogParamInit()`

SceNetCheckDialogResult

Structure of Network Check Dialog result

Definition

```
#include <common_dialog.h>
#include <common_dialog/netcheck.h>
typedef struct SceNetCheckDialogResult {
        SceInt32 result;
        SceBool psnModeSucceeded;
        SceUInt8 reserved[124];
} SceNetCheckDialogResult;
```

Members

result	Exit status of Network Check Dialog
psnModeSucceeded	State that indicates whether the PSN℠ mode processing is complete at the time of exit. Specifically, whether being in sign-in state or not will return.
reserved	Not used

Description

This datatype indicates the processing result of Network Check Dialog.

It is specified with `sceNetCheckDialogGetResult()`.

See Also

`sceNetCheckDialogGetResult()`

Document serial number: 000004892117

SceNetCheckDialogPS3ConnectInfo

Structure storing PlayStation®3 connection information

Definition

```
#include <common_dialog/netcheck.h>
typedef struct SceNetCheckDialogPS3ConnectInfo {
        SceNetInAddr inaddr;
        SceUInt8 nickname[128];
        SceUInt8 macAddress[6];
        SceUInt8 reserved[6];
} SceNetCheckDialogPS3ConnectInfo;
```

Members

inaddr	IP address of the PlayStation®3
nickname	Nickname of the PlayStation®3
macAddress	MAC address of the PlayStation®3
reserved	Unused

Description

This is the data type storing PlayStation®3 information when Network Check Dialog is called successfully in PlayStation®3 connection mode.

It is specified with `sceNetCheckDialogGetPS3ConnectInfo()`.

See Also

`sceNetCheckDialogGetPS3ConnectInfo()`

SceNetCtlIfStat

Interface usage information structure

Definition

```
#include <libnetctl.h>
typedef struct SceNetCtlIfStat {
        SceSize size;
        SceUInt32 totalSec;
        SceUInt64 txBytes;
        SceUInt64 rxBytes;
        SceRtcTick resetTick;
        SceUInt32 reserved[8];
} SceNetCtlIfStat;
```

Members

size	Size of this structure It is necessary to store an appropriate size on the calling side
totalSec	Network usage time (seconds)
txBytes	Number of bytes sent
rxBytes	Number of bytes received
resetTick	Time of usage information reset within the settings application
reserved	Unused

Description

This is a data type representing usage information of the network interface.

It is retrieved with `sceNetCtlGetIfStat()`.

Notes

Note that there is no guarantee that the usage information retrievable with `sceNetCtlGetIfStat()` will be the exact information transmitted/received to/from the network; rather, it is merely indicative information counted when the interface is handled on the system side. Furthermore, the timing of statistic information updates on the system software side is undefined.

For this reason, reference of this usage information is limited to development purposes. In products, avoid programming that relies on this information.

Also, this information does not exist independently for each application. Instead, the usage information of all applications is added up and managed in the non-volatile area by the system software side. Therefore, it is not zero-initialized at the time of retrieval by the application, and the information accumulated as of that moment is retrieved instead.

See Also

`sceNetCtlGetIfStat()`

Functions for Initialization and Termination

sceNetCtlInit

Initialize libnetctl

Definition

```
#include <libnetctl.h>
int sceNetCtlInit(
        void
)
```

Arguments

None

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function initializes libnetctl.

If the libnetctl function is called without performing initialization, that function returns `SCE_NET_CTL_ERROR_NOT_INITIALIZED`.

Notes

Re-initialization can be done by calling this function again after terminating libnetctl by calling `sceNetCtlTerm()`.

This function is not multithread safe.

See Also

`sceNetCtlTerm()`

sceNetCtlTerm

Terminate libnetctl

Definition

```
#include <libnetctl.h>
void sceNetCtlTerm(
        void
)
```

Arguments

None

Return Values

None

Description

This function terminates libnetctl.

Notes

This function is not multithread safe.

See Also

`sceNetCtlInit()`

Functions for Obtaining Events

SceNetCtlCallback

Callback function to perform notification about event occurrences
(common to Internet communication and ad hoc communication)

Definition

```
#include <libnetctl.h>
typedef void (*SceNetCtlCallback)(
        int event_type,
        void *arg
);
```

Arguments

event_type	Event type
arg	Argument specified by application upon registering a callback function

Return Values

None

Description

This is a prototype of the callback function that is called when a transition occurs in connection state.

Either of the following macro is passed to *event_type*.

Macro	Value	Description	GetResult Function Call
SCE_NET_CTL_EVENT_TYPE_DISCONNECTED	1	Disconnection occurred	Possible
SCE_NET_CTL_EVENT_TYPE_DISCONNECT_REQ_FINISHED	2	Disconnection completed	Not possible
SCE_NET_CTL_EVENT_TYPE_IPOBTAINED	3	IP address obtained. This event does not occur if after the IP address has been obtained, the state changes from IPOBTAINED to CONNECTING through reconnection with the IP address held as is, and the IPOBTAINED state is again returned to.	Not possible

For *arg*, the application-defined argument specified upon registering the event handler with either `sceNetCtlInetRegisterCallback()` or `sceNetCtlAdhocRegisterCallback()` will be passed.

The result can be obtained with either `sceNetCtlInetGetResult()` or `sceNetCtlAdhocGetResult()` after callback notification is received among above events in which GetResult function calls are possible.

Notes

This function is not designed to guarantee the order of events notified by the libnetctl callback function.

In the case that multiple events occur from the time that `sceNetCtlCheckCallback()` was previously called until it is called this time, the call order of callback functions may be changed randomly.

Notes

This function is not multithread safe.

sceNetCtlCheckCallback

Get Internet communication connection state

Definition

```
#include <libnetctl.h>
int sceNetCtlCheckCallback(
        void
);
```

Arguments

None

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This is a trigger API for calling the callback function. If there are events that should be notified by the callback function when this function is called, the callback function is called within this function.

This function must be called periodically in situations when receiving event notification through the callback function is desired.

sceNetCtlInetGetResult

Get event result of Internet communication

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetResult(
        int event_type,
        int *error_code
);
```

Arguments

event_type	Event type for which result is to be obtained. Item for which event occurrences are notified by the callback function beforehand.
error_code	Pointer to area storing results. Error code of event result is returned.

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the results for those events among the events notified by the Internet communication callback function for which results can be obtained.

For *event_type*, specify the event type for which the result is to be obtained (notified beforehand by the callback function). An error code indicating the result is returned to **error_code*.

sceNetCtlAdhocGetResult

Get event result of ad hoc communication

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocGetResult(
        int event_type,
        int *error_code
);
```

Arguments

event_type	Event type for which result is to be obtained. Item for which event occurrences are notified with callback function beforehand.
error_code	Pointer to area storing results. Error code of event result is returned.

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the results for those events among the events notified by the ad hoc communication callback function for which results can be obtained.

For *event_type*, specify the event type for which the result is to be obtained (notified beforehand by the callback function). An error code indicating the result is returned to **error_code*.

Functions for Obtaining Internet Communication Connection State

sceNetCtlInetGetState

Get connection state of Internet communication

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetState(
        int *state
);
```

Arguments

state Pointer to variable where connection state is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the current network connection state of Internet communication and stores one of the following values to **state*.

Macro	Value	Description
SCE_NET_CTL_STATE_DISCONNECTED	0	Disconnected
SCE_NET_CTL_STATE_CONNECTING	1	Connecting (to cable or wireless device)
SCE_NET_CTL_STATE_IPOBTAINING	2	Obtaining IP address
SCE_NET_CTL_STATE_IPOBTAINED	3	IP address obtained

sceNetCtlInetRegisterCallback

Register callback for Internet communication connection state check

Definition

```
#include <libnetctl.h>
int sceNetCtlInetRegisterCallback(
        SceNetCtlCallback func,
        void *arg,
        int *cid
);
```

Arguments

func	Callback function to be registered
arg	Value to give the callback function as an argument
cid	Pointer to area where callback ID is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to register the callback that is called when the network connection state of Internet communication has changed.

It stores the callback ID in `*cid` and returns upon normal termination. This callback ID will be required when deleting the callback with `sceNetCtlInetUnregisterCallback()`.

See Also

`SceNetCtlCallback`, `sceNetCtlInetUnregisterCallback()`

sceNetCtlInetUnregisterCallback

Delete callback for Internet communication connection state check

Definition

```
#include <libnetctl.h>
int sceNetCtlInetUnregisterCallback(
        int cid
);
```

Arguments

cid ID of callback to be deleted

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to delete the callback that is called when the network connection state of Internet communication has changed.

For *cid*, specify the ID of the callback to be deleted.

See Also

SceNetCtlCallback, sceNetCtlInetRegisterCallback()

Functions for Obtaining Internet Communication Connection Information

sceNetCtlInetGetInfo

Get Internet communication connection information

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to obtain the network connection information for Internet communication.

For *code*, specify the information to be obtained using one of the following macro.

Macro	Value	Description
SCE_NET_CTL_INFO_CNF_NAME	1	Connection name (*info->cnf_name*)
SCE_NET_CTL_INFO_DEVICE	2	Device (*info->device*)
SCE_NET_CTL_INFO_ETHER_ADDR	3	Ethernet address (*info->ether_addr*)
SCE_NET_CTL_INFO_MTU	4	MTU (*info->mtu*)
SCE_NET_CTL_INFO_LINK	5	Link connection state (*info->link*)
SCE_NET_CTL_INFO_BSSID	6	BSSID (*info->bssid*)
SCE_NET_CTL_INFO_SSID	7	SSID (*info->ssid*)
SCE_NET_CTL_INFO_WIFI_SECURITY	8	Security measure for wireless connection (*info->wifi_security*)
SCE_NET_CTL_INFO_RSSI_DBM	9	Receive signal strength indicator (dBm) (*info->rssi_dbm*)
SCE_NET_CTL_INFO_RSSI_PERCENTAGE	10	Receive signal strength indicator (%) (*info->rssi_percentage*)
SCE_NET_CTL_INFO_CHANNEL	11	Channel (*info->channel*)
SCE_NET_CTL_INFO_IP_CONFIG	12	IP setting (*info->ip_config*)
SCE_NET_CTL_INFO_DHCP_HOSTNAME	13	DHCP hostname (*info->dhcp_hostname*)
SCE_NET_CTL_INFO_PPPOE_AUTH_NAME	14	PPPoE user ID (*info->pppoe_auth_name*)
SCE_NET_CTL_INFO_IP_ADDRESS	15	IP address (*info->ip_address*)
SCE_NET_CTL_INFO_NETMASK	16	Net mask (*info->netmask*)
SCE_NET_CTL_INFO_DEFAULT_ROUTE	17	Default route (*info->default_route*)
SCE_NET_CTL_INFO_PRIMARY_DNS	18	Primary DNS (*info->primary_dns*)
SCE_NET_CTL_INFO_SECONDARY_DNS	19	Secondary DNS (*info->secondary_dns*)
SCE_NET_CTL_INFO_HTTP_PROXY_CONFIG	20	HTTP proxy server setting (*info->http_proxy_config*)
SCE_NET_CTL_INFO_HTTP_PROXY_SERVER	21	Hostname of HTTP proxy server (*info->http_proxy_server*)

Macro	Value	Description
SCE_NET_CTL_INFO_HTTP_PROXY_PORT	22	Port number of HTTP proxy server (*info->http_proxy_port*)

For *info*, specify the address of the variable for receiving the obtained information. The information will then be returned to the member corresponding to *code*.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(CNF_NAME)

Get connection name

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_CNF_NAME
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_CNF_NAME
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to get the connection name of the connection used for network connection, by specifying SCE_NET_CTL_INFO_CNF_NAME for *code* in sceNetCtlInetGetInfo().

The obtained result is returned as a UTF-8 character string of up to 64 bytes to *info->cnf_name*.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(DEVICE)

Get information on networking device

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_DEVICE
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_DEVICE
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_DEVICE for *code* in sceNetCtlInetGetInfo() to obtain the device used for your network connection.

The obtained result will be returned to *info*->*device* as one of the following values.

Macro	Value	Description
SCE_NET_CTL_DEVICE_WIRELESS	0	Wireless connection
SCE_NET_CTL_DEVICE_WIRED	1	Cable connection
SCE_NET_CTL_DEVICE_PHONE	2	3G connection

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(ETHER_ADDR)

Get Ethernet address

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_ETHER_ADDR
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_ETHER_ADDR
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to get the Ethernet address of the device used for the network connection, by specifying SCE_NET_CTL_INFO_ETHER_ADDR for *code* in sceNetCtlInetGetInfo().

The obtained result will be returned to *info->ether_addr* of the SceNetEtherAddr type.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(MTU)

Get MTU

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_MTU
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_MTU
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_MTU for *code* in sceNetCtlInetGetInfo() to obtain the value of MTU (Maximum Transmission Unit).

The obtained information will be returned in *info*->*mtu* as an unsigned integer.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(LINK)

Get link connection state

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_LINK
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_LINK
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_LINK for *code* in sceNetCtlInetGetInfo() to obtain the state of the link connection.

The obtained result will be returned in *info*->*link* as one of the following values.

Macro	Value	Description
SCE_NET_CTL_LINK_DISCONNECTED	0	Disconnected
SCE_NET_CTL_LINK_CONNECTED	1	Connected

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(BSSID)

Get BSSID

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_BSSID
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_BSSID
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_BSSID for *code* in sceNetCtlInetGetInfo() to obtain BSSID of the connection-target access point.

The obtained result will be returned in *info*->*bssid* of the SceNetEtherAddr type.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(SSID)

Get SSID

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_SSID
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_SSID
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_SSID for *code* in sceNetCtlInetGetInfo() to obtain SSID of the connection-target access point.

The obtained information will be returned to *info*->*ssid* as a string with 32 characters maximum.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(WIFI_SECURITY)

Get security measure used for wireless connection

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_WIFI_SECURITY
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_WIFI_SECURITY
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_WIFI_SECURITY for *code* in sceNetCtlInetGetInfo() to obtain security measure used for wireless connection.

The obtained information will be returned to *info->wifi_security* as one of the following values.

Macro	Value	Description
SCE_NET_CTL_WIFI_SECURITY_NOAUTH	0	No security
SCE_NET_CTL_WIFI_SECURITY_WEP	1	WEP
SCE_NET_CTL_WIFI_SECURITY_WPAPSK_WPA2PSK	2	(Not used)
SCE_NET_CTL_WIFI_SECURITY_WPAPSK_TKIP	3	WPA-PSK(TKIP)
SCE_NET_CTL_WIFI_SECURITY_WPAPSK_AES	4	WPA-PSK(AES)
SCE_NET_CTL_WIFI_SECURITY_WPA2PSK_TKIP	5	WPA2-PSK(TKIP)
SCE_NET_CTL_WIFI_SECURITY_WPA2PSK_AES	6	WPA2-PSK(AES)
SCE_NET_CTL_WIFI_SECURITY_UNSUPPORTED	7	(Not used)

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(RSSI_DBM)

Get receive signal strength indicator (dBm)

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_RSSI_DBM
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_RSSI_DBM
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_RSSI_DBM for *code* in sceNetCtlInetGetInfo() to obtain the receive signal strength indicator in dBm.

The obtained information will be returned to *info->rssi_dbm* as an unsigned integer.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(RSSI_PERCENTAGE)

Get receive signal strength indicator (%)

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_RSSI_PERCENTAGE
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_RSSI_PERCENTAGE
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_RSSI_PERCENTAGE for *code* in sceNetCtlInetGetInfo() to obtain the receive signal strength indicator (%).

The obtained information will be returned to *info->rssi_percentage* as an unsigned integer. The return value can range from 0 to 100 (where 0 is the weakest and 100 is the strongest).

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(CHANNEL)

Get channel of the access point

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_CHANNEL
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_CHANNEL
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_CHANNEL for *code* in sceNetCtlInetGetInfo() to obtain the channel of the access point.

The obtained information will be returned to *info->channel* as an unsigned integer.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(IP_CONFIG)

Get IP setting

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_IP_CONFIG
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_IP_CONFIG
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_IP_CONFIG for *code* in sceNetCtlInetGetInfo() to obtain the method for obtaining the IP address.

The obtained information will be returned to *info->ip_config* as one of the following values.

Macro	Value	Description
SCE_NET_CTL_IP_DHCP	0	DHCP
SCE_NET_CTL_IP_STATIC	1	Static IP
SCE_NET_CTL_IP_PPPOE	2	PPPoE

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(DHCP_HOSTNAME)

Get DHCP hostname

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_DHCP_HOSTNAME
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_DHCP_HOSTNAME
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_DHCP_HOSTNAME for *code* in sceNetCtlInetGetInfo() to obtain the hostname to be used in DHCP.

The obtained information will be returned to *info->dhcp_hostname* as a string with 255 characters maximum.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(PPPOE_AUTH_NAME)

Get PPPoE user ID

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_PPPOE_AUTH_NAME
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_PPPOE_AUTH_NAME
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_PPPOE_AUTH_NAME for *code* in sceNetCtlInetGetInfo() to obtain the user ID to be used for PPPoE.

The obtained information will be returned to *info->pppoe_auth_name* as a string with 127 characters maximum.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(IP_ADDRESS)

Get IP address

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_IP_ADDRESS
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_IP_ADDRESS
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_IP_ADDRESS for *code* in sceNetCtlInetGetInfo() to obtain the IP address.

The obtained information will be returned to *info->ip_address* in a string format with breakpoints such as "192.168.1.1".

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(NETMASK)

Get net mask

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_NETMASK
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_NETMASK
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_NETMASK for *code* in sceNetCtlInetGetInfo() to obtain the net mask.

The obtained information will be returned to *info->netmask* in a string format with breakpoints such as "255.255.255.0".

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(DEFAULT_ROUTE)

Get default route

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_DEFAULT_ROUTE
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_DEFAULT_ROUTE
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_DEFAULT_ROUTE for *code* in sceNetCtlInetGetInfo() to obtain default route information.

The obtained information will be returned to *info->default_route* in a string format with breakpoints such as "192.168.1.1".

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(PRIMARY_DNS)

Get primary DNS

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_PRIMARY_DNS
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_PRIMARY_DNS
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_PRIMARY_DNS for *code* in sceNetCtlInetGetInfo() to obtain the primary DNS.

The obtained information will be returned to *info->primary_dns* in a string format with breakpoints such as "192.168.1.1".

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(SECONDARY_DNS)

Get secondary DNS

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_SECONDARY_DNS
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_SECONDARY_DNS
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_SECONDARY_DNS for *code* in sceNetCtlInetGetInfo() to obtain the secondary DNS.

The obtained information will be returned to *info->secondary_dns* in a string format with breakpoints such as "192.168.1.1".

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(HTTP_PROXY_CONFIG)

Get HTTP proxy server setting

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_HTTP_PROXY_CONFIG
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_HTTP_PROXY_CONFIG
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_HTTP_PROXY_CONFIG for *code* in sceNetCtlInetGetInfo() to obtain the HTTP proxy server setting.

The obtained information will be returned to *info->http_proxy_config* as one of the following values.

Macro	Value	Description
SCE_NET_CTL_HTTP_PROXY_OFF	0	HTTP proxy off
SCE_NET_CTL_HTTP_PROXY_ON	1	HTTP proxy on

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(HTTP_PROXY_SERVER)

Get the hostname of the HTTP proxy server

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_HTTP_PROXY_SERVER
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_HTTP_PROXY_SERVER
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_HTTP_PROXY_SERVER for *code* in sceNetCtlInetGetInfo() to obtain the hostname of the HTTP proxy server.

The obtained information will be returned to *info->http_proxy_server* as a string with 255 characters maximum.

See Also

SceNetCtlInfo

sceNetCtlInetGetInfo(HTTP_PROXY_PORT)

Get HTTP proxy server port number

Definition

```
#include <libnetctl.h>
int sceNetCtlInetGetInfo(
        int code,           // SCE_NET_CTL_INFO_HTTP_PROXY_PORT
        SceNetCtlInfo *info
);
```

Arguments

code	Code of connection information to be obtained SCE_NET_CTL_INFO_HTTP_PROXY_PORT
info	Pointer to variable where obtained information is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Specify SCE_NET_CTL_INFO_HTTP_PROXY_PORT for *code* in sceNetCtlInetGetInfo() to obtain the port number of the HTTP proxy server.

The obtained information will be returned to *info->http_proxy_port* as an unsigned integer.

See Also

SceNetCtlInfo

Functions for Getting NAT Router Information

sceNetCtlGetNatInfo

Get NAT router information

Definition

```
#include <libnetctl.h>
int sceNetCtlGetNatInfo(
        SceNetCtlNatInfo *natinfo
);
```

Arguments

natinfo NAT router information structure

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the connected NAT router information.

When calling, structure size must be stored in the *size* member of the NAT router information structure.

Functions for Retrieving Network Usage Information

sceNetCtlGetIfStat

Retrieve network usage information

Definition

```
#include <libnetctl.h>
int sceNetCtlGetIfStat(
        int device,         /* SCE_NET_CTL_DEVICE_XXX */
        SceNetCtlIfStat *ifstat
);
```

Arguments

device	Type of network device targeted for the retrieval of usage information Specify SCE_NET_CTL_DEVICE_XXX
ifstat	Usage information structure

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

Retrieves the usage information of the network device specified with *device*.

When calling, it is necessary to store the size of the structure in the *size* member of the usage information structure.

Notes

Note that there is no guarantee that the usage information retrievable with sceNetCtlGetIfStat() will be the exact information transmitted/received to/from the network; rather, it is merely indicative information counted when the interface is handled on the system side. Furthermore, the timing of statistic information updates on the system software side is undefined.

For this reason, reference of this usage information is limited to development purposes. In products, avoid programming that relies on this information.

Also, this information does not exist independently for each application. Instead, the usage information of all applications is added up and managed in the non-volatile area by the system software side. Therefore, it is not zero-initialized at the time of retrieval by the application, and the information accumulated as of that moment is retrieved instead.

Function for Obtaining the Maximum Downloadable Size Information

sceNetCtlGetPhoneMaxDownloadableSize

Get maximum downloadable size via a mobile network

Definition

```
#include <libnetctl.h>
#define SCE_NET_CTL_DOWNLOADABLE_SIZE_UNLIMITED   (0x7fffffffffffffLL)
int sceNetCtlGetPhoneMaxDownloadableSize(
        SceInt64 *maxDownloadableSize
);
```

Arguments

maxDownloadableSize	Maximum downloadable size

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function obtains the maximum downloadable size for contents when using a mobile network.

When downloadable size is unlimited, SCE_NET_CTL_DOWNLOADABLE_SIZE_UNLIMITED will return to *maxDownloadableSize*.

Notes

When the value of the network connection device is 3G connection (SCE_NET_CTL_DEVICE_PHONE), precautions must be taken to avoid downloading content that exceeds the maximum downloadable size of *maxDownloadableSize* all at once. This is a requirement of TRC (Technical Requirements Checklist) R3132.

When setting ★**Debug Settings -> Network -> Fake 3G Interface** to **Off** from Settings application on DevKit or TestKit; or when executing on a Wi-Fi- model of a retail unit, SCE_NET_CTL_DOWNLOADABLE_SIZE_UNLIMITED will return.

However, when **Fake 3G Interface** on DevKit or TestKit is set to **On**, 20 MiB will always return so that the behavior of the application when using a mobile network can be confirmed.

Functions for Disconnecting Ad Hoc Communication

sceNetCtlAdhocDisconnect

Disconnect from ad hoc communication mode

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocDisconnect(
        void
);
```

Arguments

None

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is a non-blocking function for notifying the internal thread about events.

Upon receiving an event, the internal thread disconnects the ad hoc communication.

After this function is successfully called, the `SCE_NET_CTL_EVENT_TYPE_DISCONNECT_REQ_FINISHED` event is notified to the callback function upon completion of the disconnection processing.

Functions for Obtaining Ad Hoc Communication Connection Information

sceNetCtlAdhocRegisterCallback

Register callback for ad hoc communication connection state check

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocRegisterCallback(
        SceNetCtlCallback func,
        void *arg,
        int *cid
);
```

Arguments

func	Callback function to be registered
arg	Value to give the callback function as an argument
cid	Pointer to area where callback ID is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to register the callback that is called when the network connection state of ad hoc communication has changed.

It stores the callback ID in **cid* and returns upon normal termination. This callback ID will be required when deleting the callback with `sceNetCtlAdhocUnregisterCallback()`.

See Also

`SceNetCtlCallback`, `sceNetCtlAdhocUnregisterCallback()`

sceNetCtlAdhocUnregisterCallback

Delete callback for ad hoc communication connection state check

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocUnregisterCallback(
        int cid
);
```

Arguments

cid ID of callback to be deleted

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function is used to delete the callback that is called when the network connection state of ad hoc communication has changed.

For *cid*, specify the ID of the callback to be deleted.

See Also

SceNetCtlCallback, sceNetCtlAdhocRegisterCallback()

sceNetCtlAdhocGetState

Get ad hoc communication connection state

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocGetState(
        int *state
);
```

Arguments

state Pointer to variable where connection state is to be stored

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the current network connection state of ad hoc communication and stores one of the following values to **state*.

Macro	Value	Description
SCE_NET_CTL_STATE_DISCONNECTED	0	Disconnected
SCE_NET_CTL_STATE_CONNECTING	1	Connecting (to cable or wireless device)
SCE_NET_CTL_STATE_IPOBTAINING	2	Obtaining IP address
SCE_NET_CTL_STATE_IPOBTAINED	3	IP address obtained

See Also

SCE_NET_CTL_STATE_XXX

sceNetCtlAdhocGetPeerList

Get other terminal information list for ad hoc communication

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocGetPeerList(
        SceSize *peerInfoNum,
        SceNetCtlAdhocPeerInfo *peerInfo
);
```

Arguments

peerInfoNum	Number of other terminal information
peerInfo	Pointer to area storing other terminal information

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the information of all other terminals that exist on the currently connected ad hoc communication network.

The area indicated by *peerInfo* must be secured for the number specified to *peerInfoNum*.

This function performs two different operations depending on the value that is specified to *peerInfo*.

- If NULL is specified to *peerInfo*, the number of other terminals is returned to *peerInfoNum*.
- If an address value is specified to *peerInfo*, the data of the SceNetCtlAdhocPeerInfo structure held in the library is copied to *peerInfo* with the number specified to *peerInfoNum* as the maximum value.

See Also

SceNetCtlAdhocPeerInfo

sceNetCtlAdhocGetInAddr

Get IP address of local terminal for ad hoc communication

Definition

```
#include <libnetctl.h>
int sceNetCtlAdhocGetInAddr(
        SceNetInAddr *inaddr
);
```

Arguments

inaddr IP address of local terminal

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function gets the IP address assigned to the local terminal on the currently connected ad hoc communication network.

If this function is called from other than `SCE_NET_CTL_STATE_IPOBTAINED` state, it returns an error.

Network Check Dialog

sceNetCheckDialogParamInit

Initialize startup parameter of Network Check Dialog

Definition

```
#include <netcheck_dialog.h>
static inline
void sceNetCheckDialogParamInit(SceNetCheckDialogParam *param)
{
        sceClibMemset( param, 0x0, sizeof(SceNetCheckDialogParam) );
        _sceCommonDialogSetMagicNumber( &param->commonParam );
        param->sdkVersion = SCE_PSP2_SDK_VERSION;
        param->defaultAgeRestriction = SCE_NETCHECK_DIALOG_INITIAL_AGE_RESTRICTION;
}
```

Arguments

param Startup parameter

Return Values

None.

Description

This function initializes the startup parameter specified with the Network Check Dialog start function.

The startup parameter must be initialized by this function before calling the Network Check Dialog start function.

The *mode* member of the startup parameter is not set by this function. Explicitly specify it following calling this function.

See Also

sceNetCheckDialogInit()

sceNetCheckDialogInit

Start Network Check Dialog

Definition

```
#include <netcheck_dialog.h>
SceInt32 sceNetCheckDialogInit(
        SceNetCheckDialogParam *param
);
```

Arguments

param Startup parameter

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function starts the Network Check Dialog processing. This function is a non-blocking function for requesting the start of the Network Check Dialog to the system and immediately returning.

After calling this function, get the connection status by calling `sceNetCheckDialogGetStatus()`. During the connection processing, the status is `SCE_COMMON_DIALOG_STATUS_RUNNING`. Thereafter, the status is updated to `SCE_COMMON_DIALOG_STATUS_FINISHED` when the connection succeeds or an error occurs. Whether the connection succeeded or failed can be known with `sceNetCheckDialogGetResult()`.

Regardless of whether the connection succeeded or failed, after the status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`, terminate the connection processing by calling `sceNetCheckDialogTerm()`.

Examples

```
/* Example when calling Network Check Dialog in the PSN℠ mode or PSN℠ online mode
*/
SceNetCheckDialogParam netcheckParam;
#define CONTENT_RATING_AGE         18 /* Age restriction */

sceNetCheckDialogParamInit(&netcheckParam);
/* Specify appropriate mode to netcheckParam.mode */
netcheckParam.defaultAgeRestriction = CONTENT_RATING_AGE;
/*-- Example of using the age restriction list for each country/region
#define AGE_RESTRICTION_COUNT_MAX   1
SceNetCheckDialogAgeRestriction ageRestrictions[AGE_RESTRICTION_COUNT_MAX] =
{ 'g', 'b', 1, 0 };
netcheckParam.ageRestrictionCount = AGE_RESTRICTION_COUNT_MAX;
--*/

SceInt32 ret = sceNetCheckDialogInit(&netcheckParam);
if(ret < 0){
        /* Error handling */
}
```

See Also

SCE_COMMON_DIALOG_STATUS_XXX, sceNetCheckDialogGetStatus(), sceNetCheckDialogGetResult(), sceNetCheckDialogTerm()

sceNetCheckDialogTerm

Terminate Network Check Dialog

Definition

```
#include <netcheck_dialog.h>
SceInt32 sceNetCheckDialogTerm(
        void
);
```

Arguments

None

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function terminates the Network Check Dialog.

Regardless of whether the connection succeeded or failed, be sure to call this function after `SCE_COMMON_DIALOG_STATUS_FINISHED` is returned.

This function does not cause disconnection.

See Also

`sceNetCheckDialogInit()`

sceNetCheckDialogGetStatus

Get processing status of Network Check Dialog

Definition

```
#include <netcheck_dialog.h>
SceCommonDialogStatus sceNetCheckDialogGetStatus(
        void
);
```

Arguments

None

Return Values

Returns a value of 0 or greater as the common processing status of Common Dialog upon normal termination.

Returns an error code upon error.

Description

This function returns the current status of the Network Check Dialog processing.

Macro	Value	Description
SCE_COMMON_DIALOG_STATUS_NONE	0	Not running
SCE_COMMON_DIALOG_STATUS_RUNNING	1	Running
SCE_COMMON_DIALOG_STATUS_FINISHED	2	Finished

After calling `sceNetCheckDialogInit()`, get the status of the connection processing by calling this function at regular intervals.

See Also

`SCE_COMMON_DIALOG_STATUS_XXX`, `sceNetCheckDialogInit()`

sceNetCheckDialogGetResult

Get result of Network Check Dialog

Definition

```
#include <netcheck_dialog.h>
SceInt32 sceNetCheckDialogGetResult(
        SceNetCheckDialogResult *result
);
```

Arguments

result Pointer to structure storing result

Return Values

Returns either 0 or a value greater than 0 as the common processing result of Common Dialog upon normal termination.

Returns an error code upon error.

Description

This function gets the processing result when it is called following completion of the Network Check Dialog processing.

Macro	Value	Description
SCE_COMMON_DIALOG_RESULT_OK	0	Normal termination
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	1	Termination through cancellation by user
SCE_COMMON_DIALOG_RESULT_ABORTED	2	Forced termination through sceNetCheckDialogAbort()
	Negative number	Error termination due to occurrence of error. The value is the error code of the error that occurred.

The result is passed to *result* specified by the argument.

See Also

SCE_COMMON_DIALOG_RESULT_XXX, SceNetCheckDialogResult

sceNetCheckDialogGetPS3ConnectInfo

Get information of connected PlayStation®3

Definition

```
#include <netcheck_dialog.h>
SceInt32 sceNetCheckDialogGetPS3ConnectInfo(
        SceNetCheckDialogPS3ConnectInfo *info
);
```

Arguments

info Pointer to the structure storing PlayStation®3 information

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

After Network Check Dialog is called in PlayStation®3 connection mode and processing is complete, call this function to retrieve PlayStation®3 information.

Before calling this function, be sure to call `sceNetCheckDialogGetResult()` to verify that the Network Check Dialog processing has been terminated normally.

See Also

`SceNetCheckDialogPS3ConnectInfo`, `sceNetCheckDialogGetResult()`

sceNetCheckDialogAbort

Abort Network Check Dialog

Definition

```
#include <netcheck_dialog.h>
SceInt32 sceNetCheckDialogAbort(
        void
);
```

Arguments

None

Return Values

Returns 0 upon normal termination.

Returns an error code upon error.

Description

This function aborts Network Check Dialog.

It can be called at any time between calling `sceNetCheckDialogInit()` and calling `sceNetCheckDialogTerm()`.

When calling is successful, Network Check Dialog will begin to abort processing.

When processing is completely aborted, the operation status will change to `SCE_COMMON_DIALOG_STATUS_FINISHED`.

For details on the operation status, refer to the `sceNetCheckDialogGetStatus()` section.

If Network Check Dialog has been closed with this function, `SCE_COMMON_DIALOG_RESULT_ABORTED` will return to *`result`* of the `SceNetCheckDialogResult` structure retrieved with `sceNetCheckDialogGetResult()`.