

PlayStation®2 EE Library Overview

Release 2.4

Inet Libraries

© 2001 Sony Computer Entertainment Inc.
Publication date: October 2001

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

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

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

The *PlayStation®2 EE Library Overview - Inet Libraries* manual is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® license agreements.

The *PlayStation®2 EE Library Overview - Inet Libraries* manual is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® license agreements.

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

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

The information in the *PlayStation®2 EE Library Overview - Inet Libraries* manual is subject to change without notice. The content of this book is Confidential Information of Sony Computer Entertainment.

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

Summary Table of Contents

About This Manual	v
Changes Since Last Release	v
Related Documentation	v
Typographic Conventions	v
Developer Support	v
Chapter 1: Network Socket Library Overview	1-1
Chapter 2: Network Configuration GUI Library	2-1

About This Manual

This is the Runtime Library Release 2.4 version of the *PlayStation®2 EE Library Overview - Inet Libraries* manual.

The purpose of this manual is to provide overview-level information about the PlayStation®2 EE inet libraries. For related descriptions of the PlayStation®2 EE inet library structures and functions, refer to the *PlayStation®2 EE Library Reference - Inet Libraries*.

Changes Since Last Release

Chapter 2: Network Configuration GUI Library

New

Related Documentation

Library specifications for the IOP can be found in the *PlayStation®2 IOP Library Reference* manuals and the *PlayStation®2 IOP Library Overview* manuals.

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

Typographic Conventions

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

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

Developer Support

Sony Computer Entertainment America (SCEA)

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

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

Sony Computer Entertainment Europe (SCEE)

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

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

Chapter 1:

Network Socket Library Overview

Library Overview	1-3
Related Files	1-3
Sample Programs	1-4
Using libinsck Together With the libnet API	1-4
Initialization Procedure	1-4
Initialize SIFRPC	1-4
Load Each Module	1-4
Initialize MSIFRPC	1-4
Initialize the libnet Library	1-4
Load Network Settings on Stack	1-4
errno and h_errno	1-5
Relationship to I/O Functions Such as sceRead() and sceWrite()	1-5

Library Overview

libinsck is a socket API-compatible wrapper library that runs on top of the network library (inet). It enables a general socket API to be used on top of inet that can handle the network. Although inet runs on the IOP, a library for transparently using the inet API from the EE is provided as sample code in ee/sample/inet/libnet. libinsck is an EE library that runs on top of libnet.

Each socket API function in libinsck has a function name which starts with the prefix `scelInsock`. General API names are provided as aliases by `#define` in the include file.

Example: For `bind()`

API: `scelInsockBind()`

Alias: `#define bind scelInsockBind()`

However, libinsck does not provide all socket API functions that are available under UNIX. For information about the functions that are provided, see the corresponding reference.

Related Files

The following files are required to use libinsck.

Table 1-1

Category	File Name
Header files	sample/inet/libnet/libnet.h
	include/libmrpc.h
	include/libinsck.h
	(the following five files are included internally)
	include/ libinsck /netdb.h
	include/ libinsck /arpa/inet.h
	include/ libinsck /netinet/in.h
	include/ libinsck /netinet/tcp.h
Library files	include/ libinsck /sys/socket.h
Module files	libinsck.a
	libnet.a
	netcnf.irx
	inet.irx
	inetctl.irx
	libmrpc.irx
	libnet.irx

For compatibility with the socket API, the header files also provide socket API functions in the `include/libinsck/` directory. Normally, you need only include `include/libinsck.h`, which internally includes all of these socket API functions.

Since libinsck uses libnet internally, libnet.a and libnet.irx must be created in advance. libnet is provided in the sample code.

Sample Programs

The following sample programs that use libinsck were obtained by changing the samples that were provided with inet to use the socket API.

- ee/sample/inet/socket/http_get Sample program for getting index.html from a specified web server
- ee/sample/inet/socket/echo_server Sample TCP server that runs on the echo port

Using libinsck Together With the libnet API

Since libinsck is provided on top of libnet, the API provided by libnet (the inet API) can also be used at the same time as the socket API. However, the inet functions must be applied to a socket that has already been created; no API is provided otherwise.

Initialization Procedure

libinsck uses the libnet initialization procedure as is.

The initialization procedure is as follows. For details, refer to the individual samples.

Initialize SIFRPC

```
sceSifInitRpc(0);
```

Load Each Module

```
#define MAIN_MOD_ROOT          "host0:/usr/local/sce/iop/modules/"
#define IOP_MOD_INET          MAIN_MOD_ROOT "inet.irx"

while(sceSifLoadModule(IOP_MOD_INET, 0, NULL) < 0)
    printf("inet.irx cannot be loaded.\n");

...
```

Initialize MSIFRPC

```
sceSifMInitRpc(0);
```

Initialize the libnet Library

```
sceSifMClientData cd;

...

libnet_init(&cd);
```

Load Network Settings on Stack

Wait for network interface initialization

```
#define ERR_STOP    while(1)
#define NETBUFSIZE 512
#define NET_DB      "host0:/usr/local/sce/conf/net/net.db"
```

```

#define USR_CONF_NAME      "Combination4"

static u_int net_buf[NETBUFSIZE]  __attribute__((aligned(64)));

...

int if_id;
struct sceInetAddress myaddr;

if((ret = reg_handler(&cd, net_buf)) < 0){
    printf("reg_handler() failed.\n");
    ERR_STOP;
}
if((ret = load_set_conf(&cd, net_buf, NET_DB, USR_CONF_NAME)) < 0)
{
    printf("load_inet_cnf() failed.\n");
    ERR_STOP;
}
if((ret = wait_get_addr(&cd, net_buf, &if_id, &myaddr)) < 0){
    printf("wait_get_addr() failed.\n");
    ERR_STOP;
}

```

errno and h_errno

A problem occurs when an errno that is provided by the current EE libc.a is directly accessed from multiple threads of the EE kernel. As a result, libinsck keeps internally an errno for each thread independently, and these are used for errno or h_errno by the socket API. Access to these independent errno or h_errno is performed using the sceInsockErrno and sceInsockHErrno macros.

When libinsck is used by a single thread, the EE errno may be used.

Relationship to I/O Functions Such as sceRead() and sceWrite()

With a socket API such as that available under UNIX, a general I/O function such as read() or write can be used for input from or output to a socket. However, sceRead(), sceWrite(), sceClose() and other functions provided on the EE cannot be used with a socket that was created using libinsck.

Chapter 2:

Network Configuration GUI Library

Library Overview	2-3
Related Files	2-3
Functions That Must be Defined	2-3
Background Files	2-4
Library Startup and Shutdown	2-5
Initialization Processing Needed Before sceNetGuiCnf_Do() is Called	2-5
Termination Processing Needed After Calling sceNetGuiCnf_Do()	2-7
Library Functions	2-8
Selecting a Network Configuration	2-8
Saving to a Memory Card	2-9
Saving to a Hard Disk Drive	2-9
Adding Hardware Settings	2-9
Editing Hardware Settings	2-9
Deleting Hardware Settings	2-10
Copying Hardware Settings	2-10
Adding Network Service Provider Settings	2-10
Editing Network Service Provider Settings	2-11
Deleting Network Service Provider Settings	2-11
Copying Network Service Provider Settings	2-12
Creating New Settings	2-12
Adding and Editing Combinations	2-12
Deleting Combinations	2-13
Testing the Connection	2-13
Determining Errors	2-13
USB Hardware Device Driver Autoload	2-14
USB Mouse	2-14
USB Keyboard	2-14
Supported Controllers	2-14
Supported Connection Hardware	2-15
Changing the Background and Colors	2-15
Network Application Operating Procedure	2-15
Adding Hardware Settings	2-15
Editing Hardware Settings	2-18
Deleting Hardware Settings	2-19
Copying Hardware Settings	2-21
Adding Network Service Provider Settings	2-22
Editing Network Service Provider Settings	2-24
Deleting Network Service Provider Settings	2-26
Copying Network Service Provider Settings	2-27
Creating New Settings	2-29
Adding or Editing Combinations	2-30
Deleting Combinations	2-31
Selecting a Network Configuration	2-32

Connection Test	2-34
Quitting Network Configuration Processing	2-35
Library Restrictions	2-36
TCP/IP Stack	2-36
Specifying the Modem Driver	2-36
Graphic System	2-36
Sound System	2-36
IOP Modules	2-36
User Interface	2-36
Character Codes	2-36
Future Version Upgrade Plans	2-37

Library Overview

The Network Configuration GUI library (ntguicnf) is a program for adding, editing, deleting and copying common network configuration files used by title applications that use the network. ntguicnf is designed to conform to the TRC (Technical Requirements Checklist), including the user interface. By incorporating ntguicnf in a title application, the effort required to develop code to support the "Network Configuration Screen" can be reduced.

Related Files

The following files are required by ntguicnf.

Category	Filename
Library files	libkernel.a
	libgraph.a
	libvu0.a
	libpad.a
	libmc.a
	ntguicnf.a
Header file	ntguicnf.h
Module files	sio2man.irx
	padman.irx
	mcmman.irx
	mcserv.irx
	netcnf.irx (required for Release 2.3.1 and later)
	inet.irx (required for Release 2.3.3 and later)
	inetctl.irx (required for Release 2.3.3 and later)
	ppp.irx (required for Release 2.3.3 and later)
	pppoe.irx (required for Release 2.3.3 and later)
	usbd.irx
	usbmload.irx (only if the USB hardware device driver is autoloaded)
	dev9.irx (only if using a hard disk drive)
	atad.irx (only if using a hard disk drive)
	hdd.irx (only if using a hard disk drive)
	pfs.irx (only if using a hard disk drive)
	smapi.irx (only if using a hard disk drive)
	ntguicnf.irx

Note: The only memory card that is supported by ntguicnf is the PS2 Memory Card.

Functions That Must be Defined

A program that uses ntguicnf must define the following callback functions. For details about each callback function, refer to the ntguicnf library reference.

Memory Allocation Functions

sceNetGuiCnfCallback_Malloc
 sceNetGuiCnfCallback_Memalign
 sceNetGuiCnfCallback_Realloc
 sceNetGuiCnfCallback_Free

Software Keyboard Functions

sceNetGuiCnfCallback_SKBInit
 sceNetGuiCnfCallback_SKBDestroy
 sceNetGuiCnfCallback_SKBGetVirPktTopAddr
 sceNetGuiCnfCallback_SKBGetStatus
 sceNetGuiCnfCallback_SKBSendMouseMessage
 sceNetGuiCnfCallback_SKBEnableKey
 sceNetGuiCnfCallback_SKBEveryFrame

Character Conversion Functions

sceNetGuiCnfCallback_SJIStoUTF8
 sceNetGuiCnfCallback_UTF8toSJIS

USB Mouse Function (only if using a USB Mouse)

sceNetGuiCnfCallback_UsbMouseRead

USB Keyboard Function (only if using a USB Keyboard)

sceNetGuiCnfCallback_UsbKbRead

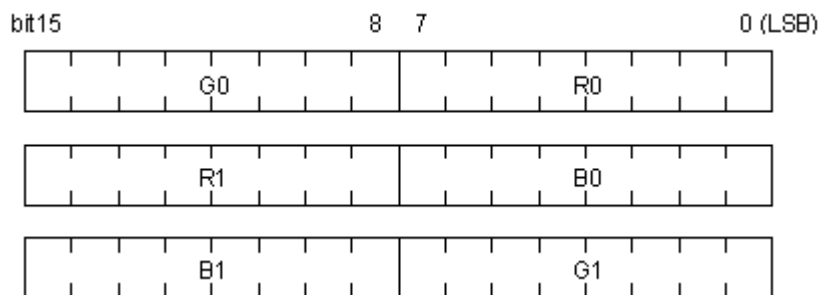
Controller Function

sceNetGuiCnfCallback_PadRead

Background Files

ntguicnf can swap backgrounds that correspond to application images by reading in backgrounds as files. A background file is limited to 640 x 448 pixels consisting of R (8 bits) + G (8 bits) + B (8 bits) binary data as shown below.

Figure 1



R0, R1: Red components (8 bits)
 G0, G1: Green components (8 bits)
 B0, B1: Blue components (8 bits)

Library Startup and Shutdown

The main processing for ntguicnf is performed by calling the `sceNetGuiCnf_Do()` function. Once the appropriate initialization is performed, `sceNetGuiCnf_Do()` should be called and a screen for configuring the network will be displayed. Subsequently, a network configuration can be selected, added, edited, or deleted and a connectivity test performed by the user.

The initialization processing that is needed before `sceNetGuiCnf_Do()` is called by the title application, and the required termination processing after `sceNetGuiCnf_Do()` has finished are explained below.

Initialization Processing Needed Before `sceNetGuiCnf_Do()` is Called

1. Load required IOP modules

Required IOP modules are not loaded by `sceNetGuiCnf_Do()`. The title application must load IOP modules that are needed by `sceNetGuiCnf_Do()` before the function is called. For information about required modules, see "Related Files" in the Library Overview.

2. Define required callback functions

`sceNetGuiCnf_Do()` uses arguments to obtain functions that are used as callbacks, such as a memory allocation function or software keyboard function. Callback functions allow behavior to change depending on the title application's purpose or image. Consequently, the following types of functions must be defined by the title application.

- Memory allocation function
- Software keyboard function
- Character conversion function
- USB mouse function (only if using a USB mouse)
- USB keyboard function (only if using a USB keyboard)
- Controller function

For information about these types of functions and their definitions, see "Functions That Must be Defined" in the Library Overview. Also, refer to the Library Reference.

3. Create a semaphore that performs `SignalSema` when a v-blank occurs

After one frame's worth of processing has completed, `sceNetGuiCnf_Do()` will issue `WaitSema()` and sleep. Since the semaphore that is reported to `WaitSema()` is not created internally by `sceNetGuiCnf_Do()`, the title application must use `CreateSema()` to create a semaphore and pass the ID of that semaphore to `sceNetGuiCnf_Do()`. Also, `SignalSema` is not invoked within `sceNetGuiCnf_Do()` so the title application must include a mechanism for invoking `SignalSema` when a v-blank occurs.

4. Terminate libpad

Internally, `sceNetGuiCnf_Do()` initializes libpad with the following procedure.

```
scePadInit( 0 );
scePadPortOpen( 0, 0, ... );
```

As a result, the title application must perform the following sort of termination processing for libpad before invoking `sceNetGuiCnf_Do()`.

```
scePadPortClose( 0, 0 )
scePadEnd( );
```


5. Configure the background file path

sceNetGuiCnf_Do() reads in the background from a file. As a result, the title application must pass the path where the background file resides as an argument to sceNetGuiCnf_Do(). Moreover, since the file path is accessed as a string by a pointer reference in sceNetGuiCnf_Do(), the title application must allocate this string in a static area.

6. Configure a buffer for receiving the configuration

The title application must provide a buffer that sceNetGuiCnf_Do() will use to receive the user-selected configuration. A static area must be allocated for the buffer, and it will be referenced by a pointer in sceNetGuiCnf_Do().

The buffer is defined by the sceNetGuiCnfEnvData structure.

7. Set default values to be used when adding (optional)

To set default values to be used when adding, the title application must define a sceNetGuiCnfEnvData structure where the default values can be set. A static area must be allocated for this structure, and it will be referenced by a pointer in sceNetGuiCnf_Do().

When setting default values to be used when adding, all of the following members of the sceNetGuiCnfEnvData structure must be set. Members other than those listed below are ignored.

<i>address</i>	IP address
<i>netmask</i>	Netmask
<i>gateway</i>	Default router
<i>dns1_address</i>	Primary DNS
<i>dns2_address</i>	Secondary DNS
<i>phone_numbers1</i>	Tel. Number1
<i>phone_numbers2</i>	Tel. Number2
<i>phone_numbers3</i>	Tel. Number3
<i>auth_name</i>	User ID
<i>auth_key</i>	Password
<i>chat_additional</i>	Additional AT command
<i>outside_set</i>	Outside number
<i>dhcp_host_name</i>	DHCP host name
<i>dialing_type</i>	Dialing type
<i>idle_timeout</i>	Line timeout (minutes)
<i>phy_config</i>	Ethernet hardware operating mode
<i>dhcp</i>	DHCP used/unused setting
<i>pppoe</i>	PPPoE (PPP over Ethernet) used/unused setting

For details about the values that can be set for each member, refer to the "Guidelines for Creating a Network Configuration Application" document. To not configure a string-format member, set '\0' at the str[0] position.

8. Configure startup options

To change the startup method as needed by the title application, you must configure startup options. Startup options are defined with the flag member of the sceNetGuiCnf_Arg structure. The following startup options are available.

Table 1

Constant	bit	Meaning
SCE_NETGUICNF_FLAG_USE_HDD	0	Use hard disk drive
SCE_NETGUICNF_FLAG_USE_USB_MOUSE	1	Use USB mouse
SCE_NETGUICNF_FLAG_USE_USB_KB	2	Use USB keyboard
SCE_NETGUICNF_FLAG_USE_SELECT_OPTION	3	Enable the bit 3 startup option
SCE_NETGUICNF_FLAG_SELECT_ONLY	4	0: Skip configuration selection 1: Select configuration only
SCE_NETGUICNF_FLAG_MC_SLOT1_ONLY	5	Only use memory card slot 1

When a hard disk drive is used and the SCE_NETGUICNF_FLAG_USE_HDD bit is set to 1, the hdd0:__sysconf partition will be mounted on pfs0 by sceNetGuiCnf_Do(). Therefore, the title application must delete any mount on pfs0 before using sceNetGuiCnf_Do().

9. Initialize USB device driver autoloading

No setting is performed by sceNetGuiCnf_Do() to autoload the USB device driver. The title application must perform USB device driver autoload initialization processing when usbmload.irx is loaded.

Termination Processing Needed After Calling sceNetGuiCnf_Do()

1. Reset the IOP (optional)

The IOP is not reset by sceNetGuiCnf_Do(). To delete IOP modules that are no longer needed after using sceNetGuiCnf_Do(), you must use the following procedure to reset the IOP (BOOT_FROM_CD is the procedure for starting up from a CD).

```
#if BOOT_FROM_CD
    sceCdInit( SCECdEXIT );
#endif /* BOOT_FROM_CD */
    sceSifExitCmd( );
#if BOOT_FROM_CD
    while( sceSifRebootIOP( "cdrom0:"IOP_IMAGE_FILE";1" ) == 0 );
#else /* BOOT_FROM_CD */
    while( sceSifRebootIOP( "host0:"IOP_IMAGE_file" ) == 0 );
#endif /* BOOT_FROM_CD */
    sceSifInitRpc( 0 );
    sceSifLoadFileReset( );
    sceFsReset( );
    sceSifInitIopHeap( );
#if BOOT_FROM_CD
    sceCdInit( SCECdINIT );
    sceCdMmode( SCECdCD );
#endif /* BOOT_FROM_CD */
```

An application that subsequently reuses sceNetGuiCnf_Do() must reload the required IOP modules.

2. Reinitialize the graphic system

sceNetGuiCnf_Do() will internally initialize the graphic system. If the title application is to display a graphic again after using sceNetGuiCnf_Do(), the graphic system must be reinitialized.

3. Reinitialize libpad

When processing ends, sceNetGuiCnf_Do() performs libpad termination processing with the following procedure.

```
scePadPortClose( 0, 0 )
scePadEnd( );
```

If the title application is to reuse libpad after using sceNetGuiCnf_Do(), libpad must be reinitialized.

4. Initialize libmc (optional)

The sceNetGuiCnf_Do() internally initializes libmc with the following procedure.

```
sceMcInit( );
```

If the title application is to reuse libmc after using sceNetGuiCnf_Do(), libmc must be reinitialized with sceMcInit(), as necessary.

Library Functions

Selecting a Network Configuration

1. Selecting the configuration during startup

When sceNetGuiCnf_Do() starts, an automatic selection of the configuration is performed as follows.

- If combinations are present, the combination that was added / selected last is selected in priority order of PS2 memory card, hard disk drive.
- If no combination is present, the hardware settings and network service provider settings that were added / selected last are selected in priority order of PS2 memory card, hard disk drive.

2. Selecting with a combination

sceNetGuiCnf_Do() can select a combination that combines previously registered hardware settings and network service provider settings. When sceNetGuiCnf_Do() is terminated, the configuration is read in from the PS2 memory card or hard disk drive and its contents are returned in the buffer pointed to by the result_env_data member of the sceNetGuiCnf_Arg structure.

Example:

Combination	Combination 1
Hardware	SCE/Ethernet (Network Adaptor)
Network service provider	Local Area Network settings

3. Selecting hardware and network service provider settings individually

sceNetGuiCnf_Do() can select hardware and network service provider settings individually without being aware of the device where the settings were saved. When sceNetGuiCnf_Do() is terminated, the settings are individually read in from the PS2 memory card or hard disk drive and the contents are returned in the buffer pointed to by the result_env_data member of the sceNetGuiCnf_Arg structure. Settings that have not been saved as files can also be selected in a similar manner.

Example:

Combination	(Unselected)
Hardware	SCE/Ethernet (Network Adaptor)
Network service provider	Local Area Network settings

Saving to a Memory Card

Among memory cards, only the PS2 memory card is supported by `sceNetGuiCnf_Do()`. The slots that are supported are given below.

If the startup option `SCE_NETGUICNF_FLAG_MC_SLOT1_ONLY` bit is 1

- Only slot 1 is supported.

If the startup option `SCE_NETGUICNF_FLAG_MC_SLOT1_ONLY` bit is 0

- Both slots 1 and 2 are supported. However, slot 1 takes precedence when automatic slot recognition is performed.

When the latter startup option is selected, that condition must be described in the reference manual of the title application. For other information about saving the network configuration to a memory card, refer to the "Guidelines for Creating a Network Configuration Application" document.

Saving to a Hard Disk Drive

When the startup option `SCE_NETGUICNF_FLAG_USE_HDD` bit is 1, `sceNetGuiCnf_Do()` allows the network configuration to be saved to the hard disk drive. The network configuration is saved in the `/etc/network` directory in the `hdd0:__sysconf` partition which was mounted on `pfs0`. For information about saving the network configuration to a hard disk drive, refer to the "Guidelines for Creating a Network Configuration Application" document.

Adding Hardware Settings

`sceNetGuiCnf_Do()` can be used to add hardware settings. The user can set the following information when hardware settings are added.

For a modem or TA

- Dialing type
- Additional AT command
- Outside number
- Line timeout (minutes)

For USB Ethernet hardware or a network adaptor

- Ethernet hardware operating mode

The title application can also set default values independently when adding hardware settings. The hardware devices that can be added are limited to devices that are physically connected and do not have existing settings (the settings don't exist on either a PS2 memory card or the hard disk drive, when both are used). Hardware settings can be saved either on a PS2 memory card or the hard disk drive. The user can also specify that settings should not be saved.

Editing Hardware Settings

`sceNetGuiCnf_Do()` can edit hardware settings. The user can set the following information when hardware settings are edited.

For a modem or TA

- Dialing type
- Additional AT command
- Outside number
- Line timeout (minutes)

For USB Ethernet hardware or a network adaptor

- Ethernet hardware operating mode

Any hardware settings that exist can be edited. Edited hardware settings can be saved either on a PS2 memory card or the hard disk drive. However, the edited settings must be saved on the same device where the selected hardware settings reside. The user can also specify that edited settings should not be saved.

Deleting Hardware Settings

sceNetGuiCnf_Do() can delete hardware settings. Any hardware settings that exist can be deleted.

Copying Hardware Settings

sceNetGuiCnf_Do() can copy hardware settings provided that both the PS2 memory card and hard disk drive are available. Any hardware settings that exist can be copied. Copied hardware settings can be saved either on a PS2 memory card or the hard disk drive. However, the selected hardware settings cannot be copied if they already exist on the destination device.

Adding Network Service Provider Settings

sceNetGuiCnf_Do() can be used to add network service provider settings. The user can set the following information when network service provider settings are added.

For a modem or TA

- User ID
- Password
- Tel. Number1
- Tel. Number2
- Tel. Number3
- Primary DNS (if you set the DNS server manually and it is not automatically obtained)
- Secondary DNS (if you set the DNS server manually and it is not automatically obtained)
- Setting name

For USB Ethernet hardware or a network adaptor

- User ID (when PPPoE (PPP over Ethernet) is used)
- Password (when PPPoE (PPP over Ethernet) is used)
- IP address (when neither PPPoE (PPP over Ethernet) nor DHCP is used)
- Netmask (when neither PPPoE (PPP over Ethernet) nor DHCP is used)
- Default router (when neither PPPoE (PPP over Ethernet) nor DHCP is used)
- Primary DNS (when PPPoE (PPP over Ethernet) is used and if you set the DNS server manually and it is not automatically obtained, and when neither PPPoE (PPP over Ethernet) nor DHCP is used)
- Secondary DNS (when PPPoE (PPP over Ethernet) is used and if you set the DNS server manually and it is not automatically obtained, and when neither PPPoE (PPP over Ethernet) nor DHCP is used)
- Setting name

Except for the setting name, the title application can set default values independently when adding network service provider settings. The network service provider settings that can be added are limited to those with setting names that do not already exist. The added network service provider settings can

be saved either on a PS2 memory card or on the hard disk drive. However, the added settings must be saved on the same device where the selected network provider settings reside. The user can also specify that settings should not be saved.

Editing Network Service Provider Settings

sceNetGuiCnf_Do() can edit hardware settings. The user can set the following information when network service provider settings are edited.

Network service provider settings that are created when a modem or TA was selected as the hardware device, when settings are added

- User ID
- Password
- Tel. Number1
- Tel. Number2
- Tel. Number3
- Primary DNS (if you set the DNS server manually and it is not automatically obtained)
- Secondary DNS (if you set the DNS server manually and it is not automatically obtained)
- Setting name

Network service provider settings created when USB Ethernet hardware or a network adaptor was selected as the hardware device, when settings are added and PPPoE (PPP over Ethernet) is used

- User ID
- Password
- Primary DNS (if you set the DNS server manually and it is not automatically obtained)
- Secondary DNS (if you set the DNS server manually and it is not automatically obtained)
- Setting name

Network service provider settings that are created when USB Ethernet hardware or a network adaptor was selected as the hardware device, when settings are added and PPPoE (PPP over Ethernet) is not used

- IP address (when DHCP is not used)
- Netmask (when DHCP is not used)
- Default router (when DHCP is not used)
- Primary DNS (when DHCP is not used)
- Secondary DNS (when DHCP is not used)
- Setting name

Any network service provider settings that exist can be edited. Edited network service provider settings can be saved either on a PS2 memory card or the hard disk drive. However, the edited settings must be saved on the same device where the selected network service provider settings reside. The user can also specify that edited settings should not be saved.

Deleting Network Service Provider Settings

sceNetGuiCnf_Do() can delete network service provider settings. Any network service provider settings that exist can be deleted.

Copying Network Service Provider Settings

sceNetGuiCnf_Do() can copy network service provider settings provided that both the PS2 memory card and hard disk drive are available. Any network service provider settings that exist can be copied. Copied network service provider settings can be saved either on a PS2 memory card or the hard disk drive. However, the selected network service provider settings cannot be copied if they already exist on the destination device.

Creating New Settings

The "New Settings" item enables sceNetGuiCnf_Do() to add hardware settings and network service provider settings at the same time. New settings can be added in the following two situations.

- When an existing hardware setting is selected
- When a physically connected hardware device that does not have an existing setting is selected

When an existing hardware setting is selected, the hardware device that is defined by that setting must be physically connected.

As soon as an existing setting is selected, processing will shift to the addition of a network service provider setting and no hardware setting will be added.

For the items that the user can set when adding a network service provider setting, see "Adding Network Service Provider Settings."

The title application can independently set default values when adding network service provider settings. The network service provider settings that can be added are limited to those with setting names that do not already exist. The added network service provider settings can be saved either on a PS2 memory card or on the hard disk drive. However, the added settings must be saved on the same device where the selected network provider settings reside. The user can also specify that settings should not be saved.

When a hardware device that is physically connected but does not have an existing setting is selected, processing will shift to the addition of a hardware setting. For the items that the user can set when adding a hardware setting, see "Adding Hardware Settings."

The title application can independently set default values when adding hardware settings.

After completing the hardware settings, processing will shift to the addition of a network service provider setting. For the items that the user can set when adding a network service provider setting, see "Adding a Network Service Provider Setting."

The title application can independently set default values when adding network service provider settings. The network service provider settings that can be added are limited to those with setting names that do not already exist. The added network service provider settings can be saved either on a PS2 memory card or on the hard disk drive. The user can also specify that settings should not be saved.

After the network service provider settings have been saved, the hardware settings will also be saved. If processing for saving the network service provider setting fails, the hardware settings will not be saved. However, the network service provider settings are saved even if saving of the hardware settings fails.

Adding and Editing Combinations

sceNetGuiCnf_Do() can add hardware and network service provider settings to a combination. When a combination that has been added is selected, it can also be edited to combine different hardware and network provider settings. The hardware settings and network service provider settings that can be added and edited are limited to settings that satisfy the following conditions.

- The settings must be saved as files (a setting that was created by selecting "Do not save" cannot be added to a combination)

- The settings must reside on the same device (for example, a hardware setting that resides on a PS2 memory card can only be combined with a network service provider setting that also resides on the memory card)

If the above conditions are satisfied, the "Next" button for performing the add or edit can be pressed.

Deleting Combinations

sceNetGuiCnf_Do() can delete a combination. The combinations that can be deleted are limited to the following.

- Combinations that were created by adding a hardware setting and a network provider setting as a combination
- Combinations that were previously created by adding a hardware setting and a network service provider setting as a combination, but from which only the hardware setting, only the network service provider setting, or both the hardware setting and the network service provider setting were later deleted

If the above conditions are satisfied, the "Delete" button for performing the delete can be pressed.

Testing the Connection

sceNetGuiCnf_Do() can perform a connection test when a configuration that satisfies the following conditions is selected.

- A combination is selected, the hardware and network provider settings that have been added to the combination exist, and the hardware device specified in the hardware setting is physically connected
- A hardware setting and network service provider setting are selected and the hardware device specified in the hardware setting is physically connected

When the above conditions are satisfied, the "Connection Test" button for performing the connection test can be pressed. If the connection test succeeds, the connection is disconnected and a message indicating that the connection test succeeded is displayed. If the connection test failed, a message indicating the failure is displayed.

Determining Errors

When reading a network configuration that was saved by another PlayStation 2

When sceNetGuiCnf_Do() reads a network configuration that was saved by another PlayStation 2, it executes the following process.

1. First, a message is displayed indicating that the network configuration was saved by another PlayStation 2, then another message is displayed asking whether or not the entire network configuration of the device where that configuration resides should be deleted.
2. If "Delete" is selected, the deletion is reconfirmed. If "Do not delete" is selected, a message is displayed indicating that the entire network configuration of the device where that configuration resides cannot be used (it is unavailable).
3. If "Delete" is selected on the deletion reconfirmation screen, the entire network configuration of the device where that configuration resides is deleted. If "Do not delete" is selected, a message is displayed indicating that the entire network configuration of the device where that configuration resides cannot be used (it is unavailable).
4. After confirming that deletion has completed, initialization processing for the displayed screen is performed again.

When a network configuration that has been saved is corrupted

When `sceNetGuiCnf_Do()` reads a corrupted network configuration, it displays a message indicating that the network configuration is corrupted, then performs the same processing as if it were reading a network configuration that was saved by another PlayStation 2.

Errors during addition, editing, deletion, or copying

When an error occurs during addition, editing, deletion, or copying, `sceNetGuiCnf_Do()` displays a message indicating that an error has occurred. The following types of errors may occur.

- The settings cannot be saved because the maximum number of settings that can be saved have been exceeded.
- There is insufficient free space available.
- A setting having the same name already exists.
- The memory card is not a PS2 memory card.
- Some other problem caused addition, editing, deletion, or copying to fail.

Errors while testing a connection

When an error occurs while testing a connection, `sceNetGuiCnf_Do()` displays a message indicating that an error has occurred. The following types of errors may occur.

- The connection test failed.
- The connection could not be made using the specified combination of hardware settings and network service provider settings.

USB Hardware Device Driver Autoload

`sceNetGuiCnf_Do()` can autoload the USB hardware device driver. The driver can be autoloaded by the title application when it loads and initializes `usbmload.irx`. The autoload function can also be disabled by not loading `usbmload.irx`. However, in that case, the title application must explicitly load the USB hardware device driver.

USB Mouse

A USB mouse can be used with `sceNetGuiCnf_Do()` by setting the `SCE_NETGUICNF_FLAG_USE_USB_MOUSE` bit to 1 in the *flag* member of the `sceNetGuiCnf_Arg` structure during startup. and setting the *cb_mouse_read* member of the `sceNetGuiCnf_Arg` structure appropriately.

USB Keyboard

A USB keyboard can be used with `sceNetGuiCnf_Do()` by setting the `SCE_NETGUICNF_FLAG_USE_USB_KB` bit to 1 in the *flag* member of the `sceNetGuiCnf_Arg` structure during startup and setting the *cb_kb_read* member of the `sceNetGuiCnf_Arg` structure appropriately.

Supported Controllers

The controllers that are supported by `sceNetGuiCnf_Do()` are not specifically defined. The title application can specify the supported controllers by specifying a function for reading controller button information in the *cb_pad_read* member of the `sceNetGuiCnf_Arg` structure.

Supported Connection Hardware

sceNetGuiCnf_Do() supports the following types of connection hardware. Vendor support depends on the configuration of the title application.

- Modem or TA
- USB Ethernet hardware
- Network adaptor

Changing the Background and Colors

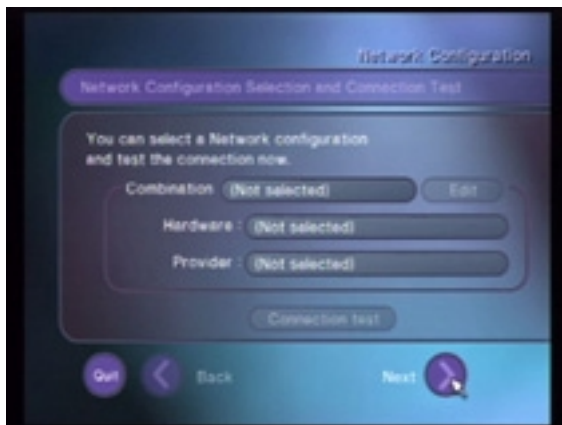
sceNetGuiCnf_Do() can change the background image of the network configuration application and the colors of the following components.

- Color of the title bar that is always displayed at the top of the screen
- Background color of the window that is always displayed in the center of the screen
- Color of each of the Quit, Back, and Next buttons that are always displayed at the bottom of the screen
- Color of the title bar of a one-choice dialog box (* In the current version, the color specified here will also be used for the title bar of an error dialog box)
- Color of the title bar of a two-choice dialog box
- Color of the title bar of an error dialog box (* Unused in the current version)
- Color of the title bar of a non-selectable dialog box

Network Application Operating Procedure

Adding Hardware Settings

The following procedure is used to add a hardware setting.



After starting up the "Network Configuration Selection and Connection Test" screen, click "Next." Depending on the setting of the flag member of the SceNetGuiCnf_Arg structure, this screen may be skipped and it may not be possible to click the "Next" button. For details, see "Configure startup options" in "Initialization Processing Needed Before sceNetGuiCnf_Do() is Called" of the "Library Startup and Shutdown" chapter in this document.



When the display changes to the "Connection Settings" screen, select "Hardware Settings."



When the display changes to the "Hardware Settings" screen, select "Add Hardware Settings."



The display will change to the "Add Hardware Settings" screen where hardware devices that are connected to the PlayStation 2 that do not have existing settings on the PS2 memory card or hard disk drive will be shown. Select the hardware device for which you wish to add a hardware setting, then click "Next."



The display will change to the setting screen corresponding to the selected hardware. (The above figure shows the screen when USB Ethernet hardware or a network adaptor is selected.) After setting the required item(s), click "Next."



The display will change to the "Save Hardware Settings" screen. Select the destination for saving the hardware settings. After the settings are saved, the display will change back to the "Network Configuration Selection and Connection Test" screen. The hardware settings that were added will be shown selected on this screen.

Editing Hardware Settings

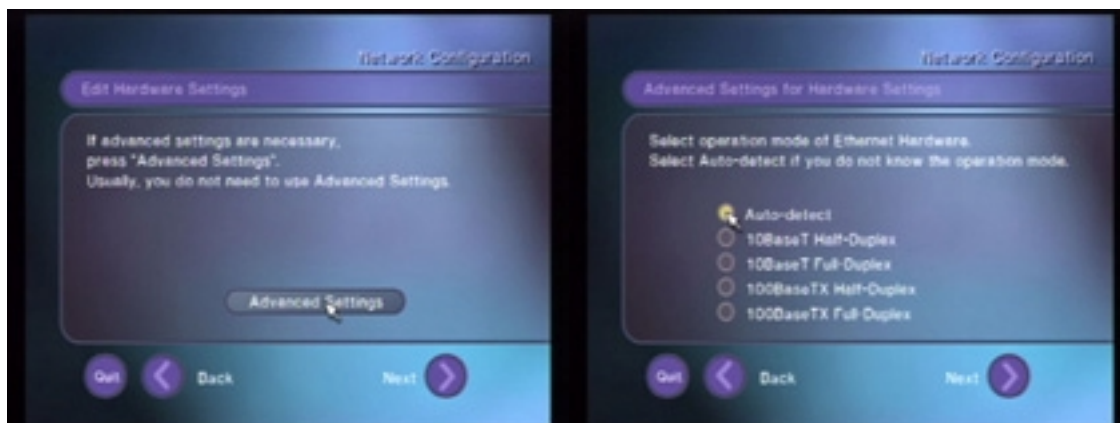
The following procedure is used to edit hardware settings. The operations for changing to the "Hardware Settings" screen are the same as those described for adding hardware settings.



After the display changes to the "Hardware Settings" screen, select "Edit Hardware Settings."



The display will change to the "Edit Hardware Settings" screen. The hardware settings that are present on the PS2 memory card and hard disk drive will be listed on this screen. Select the hardware setting that you wish to edit, then click "Next."



The display will change to the setting screen corresponding to the selected hardware. (The above figure shows the screen when USB Ethernet hardware or a network adaptor is selected.) After setting the required item(s) (the above figure shows the screen when Advanced Settings is selected and the operating mode of the Ethernet hardware is set), click "Next."



The display will change to the "Overwrite Hardware Settings" screen. The hardware settings will be saved to the device where the selected hardware settings reside. (If the connection state of the device changes during the procedure described above, the display is forcibly returned to the screen for selecting hardware settings.) After the settings are saved, the display will change back to the "Network Configuration Selection and Connection Test" screen. The hardware settings that were edited will be shown selected on this screen.

Deleting Hardware Settings

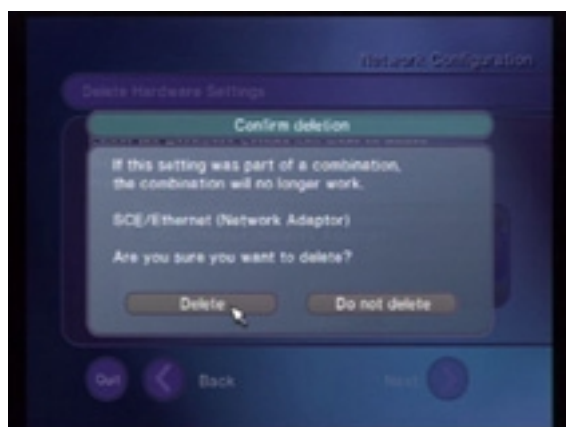
The following procedure is used to delete hardware settings. The steps for displaying the "Hardware Settings" screen are the same as those described for adding hardware settings.



After the display changes to the "Hardware Settings" screen, select "Delete Hardware Settings."



The display will change to the "Delete Hardware Settings" screen. The hardware settings present on the PS2 memory card and hard disk drive will be listed on this screen. Select the hardware settings that you wish to delete, then click "Delete."



When a dialog box for confirming the deletion appears, click "Delete." After the delete has completed, the display will change back to the "Hardware Settings" screen.

Copying Hardware Settings

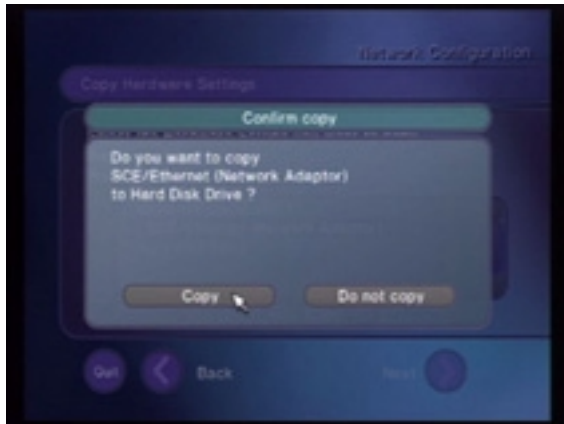
The following procedure is used to copy hardware settings. The steps for displaying the "Hardware Settings" screen are the same as those described for adding a hardware setting.



After the display changes to the "Hardware Settings" screen, select "Copy Hardware Settings." (This item cannot be selected unless the PS2 memory card and hard disk drive are connected. Copying can be executed only from the PS2 memory card to the hard disk drive or from the hard disk drive to the PS2 memory card.)



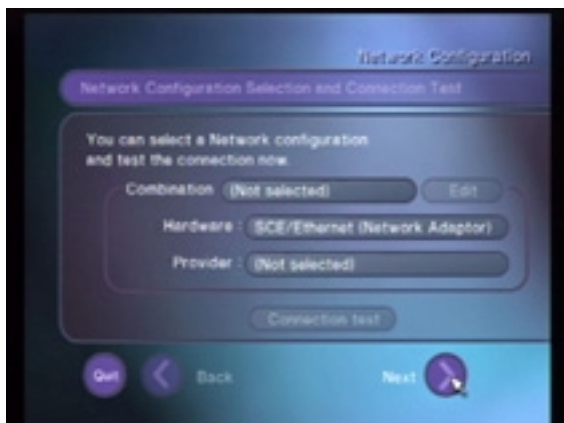
The display will change to the "Copy Hardware Settings" screen. The hardware settings that are present on the PS2 memory card and hard disk drive will be listed on this screen. Select the hardware settings that you wish to copy, then click "Copy."



When a dialog box for confirming the copy appears, click "Copy." After copying has completed, the display will change back to the "Hardware Settings" screen.

Adding Network Service Provider Settings

The following procedure is used to add a network service provider setting.



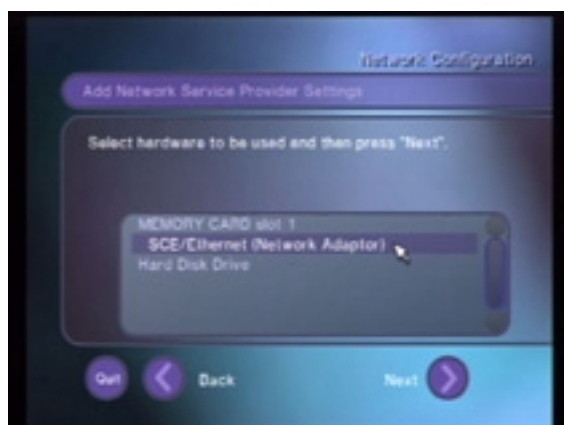
After starting up the "Network Configuration Selection and Connection Test" screen, click "Next." Depending on the setting of the flag member of the `SceNetGuiCnf_Arg` structure, this screen may be skipped and it may not be possible to click the "Next" button. For details, see "Configure startup options" in "Initialization Processing Needed Before `sceNetGuiCnf_Do()` is Called" of the "Library Startup and Shutdown" chapter in this document.



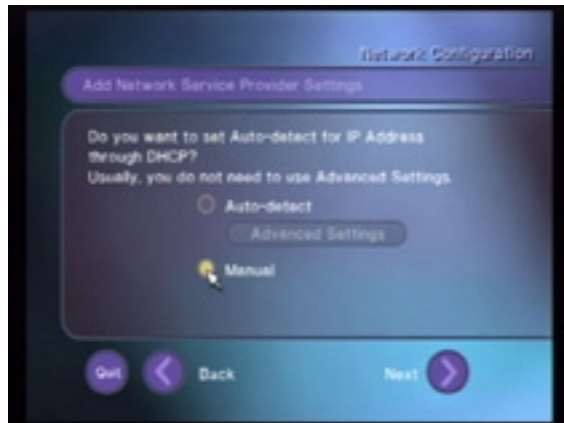
When the display changes to the "Connection Settings" screen, select "Network Service Provider Settings."



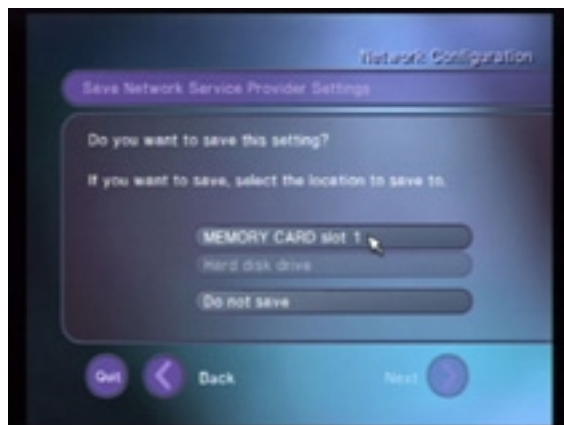
When the display changes to the "Network Service Provider Settings" screen, select "Add Network Service Provider Settings."



The display will change to the "Add Network Service Provider Settings" screen. The hardware settings that are present on the PS2 memory card and hard disk drive will be listed on this screen. (If no hardware settings have been added, processing cannot advance beyond this screen.) Select the hardware setting that will be used by the network service provider setting that you wish to add, then click "Next."



The display will change to the setting screen corresponding to the selected hardware. (The above figure shows the screen for selecting whether or not the IP address is to be automatically detected by DHCP when USB Ethernet hardware or a network adaptor is selected.) After setting the required item(s), click "Next." Items that are set can span multiple screens. The steps described above are repeated until the display changes to the "Save Network Service Provider Settings" screen.



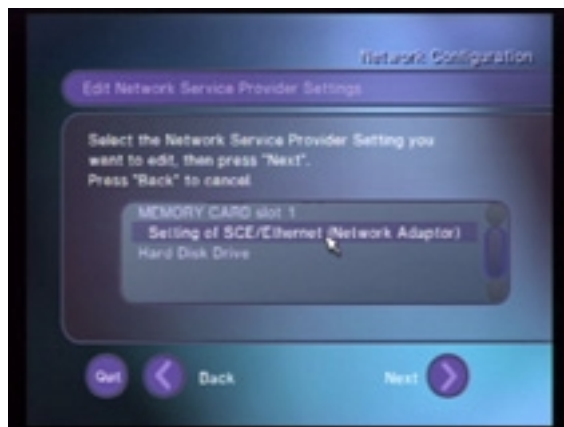
On the "Save Network Service Provider Settings" screen, select the destination for saving the network service provider settings. After the settings are saved, the display will change back to the "Network Configuration Selection and Connection Test" screen. The network service provider setting that was added will be shown selected on the "Network Configuration Selection and Connection Test" screen.

Editing Network Service Provider Settings

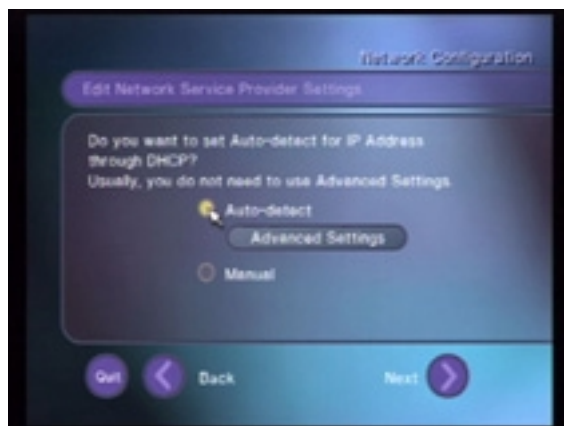
The following procedure is used to edit network service provider settings. The steps for changing to the "Network Service Provider Settings" screen are the same as those described for adding a network service provider setting.



After the display changes to the "Network Service Provider Settings" screen, select "Edit Network Service Provider Settings."



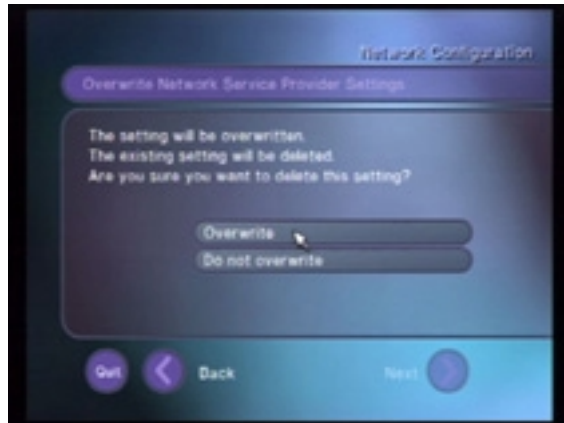
The display will then change to the "Edit Network Service Provider Settings" screen. The network service provider settings that are present on the PS2 memory card and hard disk drive are listed on this screen. Select the network service provider setting that you wish to edit, then click "Next."



The display will change to the setting screen that corresponds to the selected network service provider. (The above figure shows the screen for selecting whether or not the IP address is to be automatically

detected by DHCP when USB Ethernet hardware or a network adaptor is selected, and when the selected network service provider setting is being edited.)

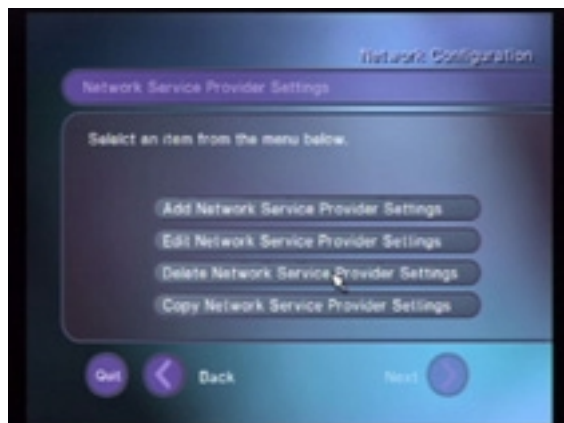
After setting the required item(s), click "Next." Items that are set can span multiple screens. The steps described above are repeated until the display changes to the "Overwrite Network Service Provider Settings" screen.



The hardware settings will be saved to the device where the selected network service provider settings reside. (If the connection state of the device changes during the procedure described above, the display is forcibly returned to the screen for selecting network service provider settings.) After the settings are saved, the display will change to the "Network Configuration Selection and Connection Test" screen. The hardware settings that were edited will be selected on this screen.

Deleting Network Service Provider Settings

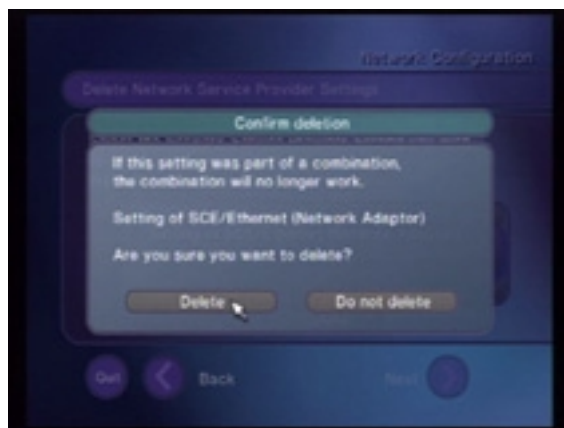
The following procedure is used to delete a network service provider setting. The steps for changing to the "Network Service Provider Settings" screen are the same as those described for adding a network service provider setting.



After the display changes to the "Network Service Provider Settings" screen, select "Delete Network Service Provider Settings."



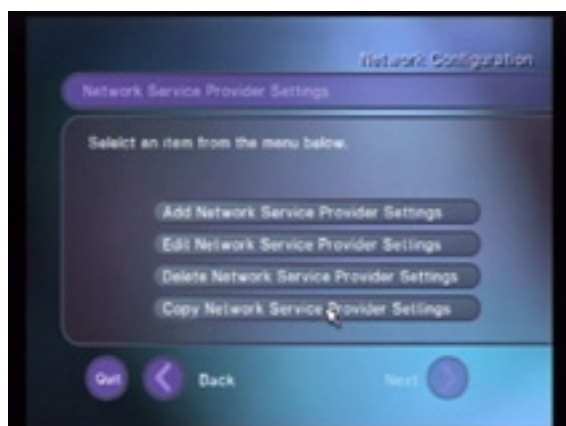
The display will change to the "Delete Network Service Provider Settings" screen. The network service provider settings that are present on the PS2 memory card and hard disk drive will be listed on this screen. Select the network service provider setting that you wish to delete, then click "Delete."



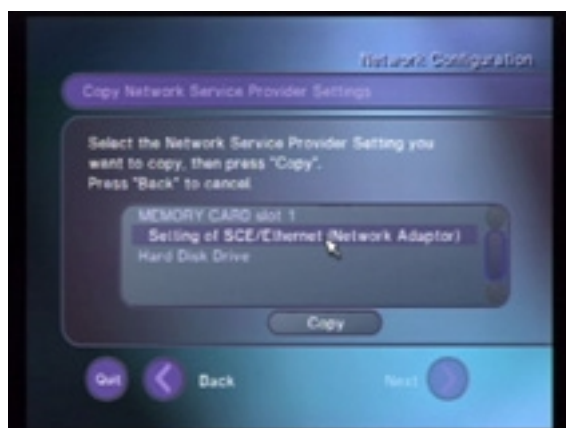
When a dialog box for confirming the deletion appears, click "Delete." After the deletion has completed, the display will change back to the "Network Service Provider Settings" screen.

Copying Network Service Provider Settings

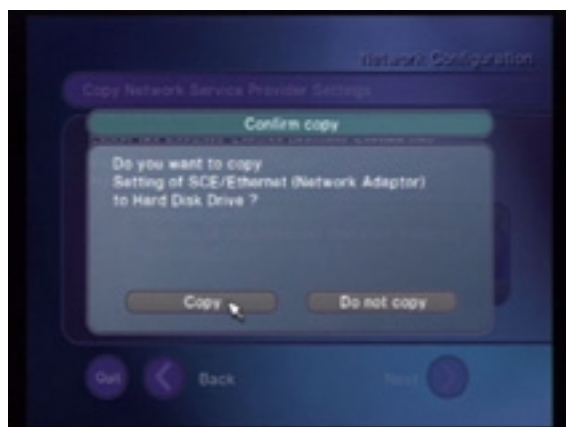
The following procedure is used to copy a network service provider setting. The steps for displaying the "Network Service Provider Settings" screen are the same as those described for adding a network service provider setting.



After the display changes to the "Network Service Provider Settings" screen, select "Copy Network Service Provider Settings." (This item cannot be selected unless the PS2 memory card and hard disk drive are connected. Copying can be executed only from the PS2 memory card to the hard disk drive or from the hard disk drive to the PS2 memory card.)



The display will change to the "Copy Network Service Provider Settings" screen. The network service provider settings that are present on the PS2 memory card and hard disk drive will be listed on this screen. Select the network service provider setting that you want to copy, then click "Copy."



When a dialog box for confirming the copy appears, click "Copy." After copying has completed, the display will change back to the "Network Service Provider Settings" screen.

Creating New Settings

The following procedure is used to create new settings.



After starting up the "Network Configuration Selection and Connection Test" screen, click "Next." Depending on the setting of the *flag* member of the `SceNetGuiCnf_Arg` structure, this screen may be skipped and it may not be possible to click the "Next" button. For details, see "Configure startup options" in "Initialization Processing Needed Before `sceNetGuiCnf_Do()` is Called" of the "Library Startup and Shutdown" chapter in this document.



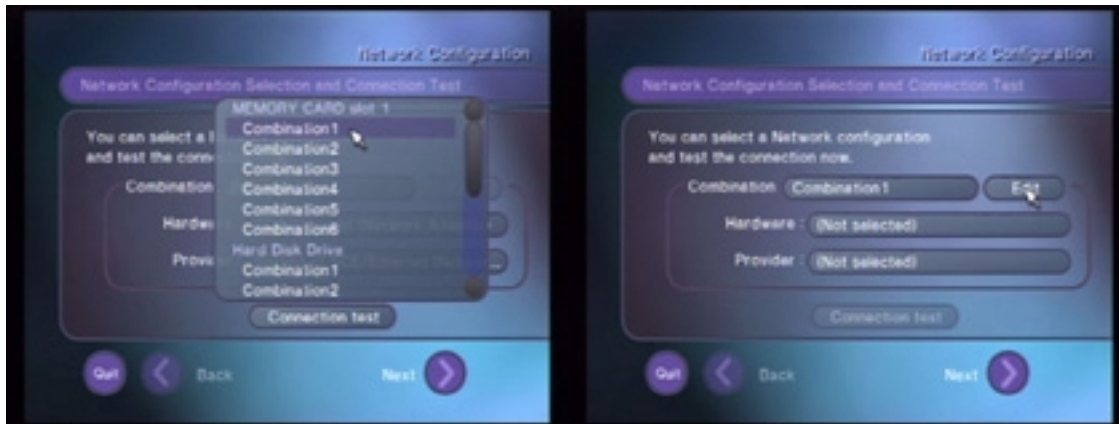
When the display changes to the "Connection Settings" screen, click "New." Selecting "New Settings" will add hardware and network service provider settings at the same time.

The hardware setting is made first, followed by the network service provider setting. Hardware settings that have been previously saved can also be used. In this case, processing shifts immediately to setting the network service provider, skipping the hardware setting. The settings are saved in the order of network service provider first, followed by the hardware setting. When a hardware setting that was previously saved is used, only the network service provider setting is saved.

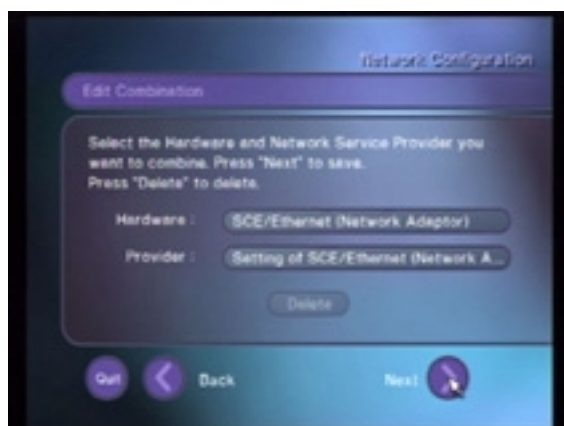
After the settings have been saved, the display will change back to the "Network Configuration Selection and Connection Test" screen. The hardware and network service provider settings that were saved will be shown selected on this screen.

Adding or Editing Combinations

The following procedure is used to add or edit a combination.



After starting up the "Network Configuration Selection and Connection Test" screen, select the combination that you wish to add or edit, then click "Edit."

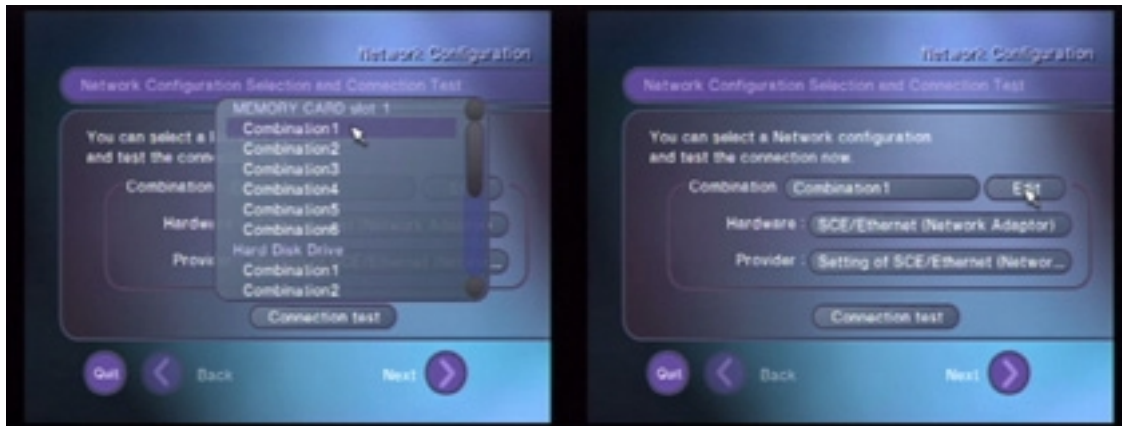


When the display changes to the "Edit Combination" screen, select the hardware and network service provider settings that you wish to combine, then click "Next." (Only hardware and network service provider settings that both reside on a PS2 memory card or that both reside on the hard disk drive can be combined. If any other combination is selected, "Next" cannot be selected.)

After the combination has been saved, the display will change back to the "Network Configuration Selection and Connection Test" screen. The combination that was added or edited will be shown selected on the "Network Configuration Selection and Connection Test" screen.

Deleting Combinations

The following procedure is used to delete a combination.



After starting up the "Network Configuration Selection and Connection Test" screen, select the combination that you wish to delete, then click "Edit."



When the display changes to the "Edit Combination" screen, select "Delete." "Delete" can be selected only in the following situations.

- If both the hardware and network service provider settings exist for the selected combination.
- If only the hardware settings have been deleted for the selected combination.
- If only the network service provider settings have been deleted for the selected combination.
- If both the hardware and network service provider settings have been deleted for the selected combination.

Even if the combination is deleted, the hardware and network service provider settings are not deleted.



When a dialog box for confirming the deletion appears, click "Delete." After deletion has completed, the display will change back to the "Network Configuration Selection and Connection Test" screen.

Selecting a Network Configuration

By combination

The following procedure is used to select hardware and network service provider settings with a combination.



After starting up the "Network Configuration Selection and Connection Test" screen, select "Combination."



After the combination is selected, the hardware and network service provider settings that were set in the combination will be shown selected.

By individual hardware and network service provider settings

The following procedure is used to individually select hardware and network service provider settings.



After starting up the "Network Configuration Selection and Connection Test" screen, choose "Hardware."



Next, choose "Provider." (You can also choose "Provider" first, then "Hardware".)



After these selections are made, the hardware and network service provider settings will be shown selected.

Connection Test

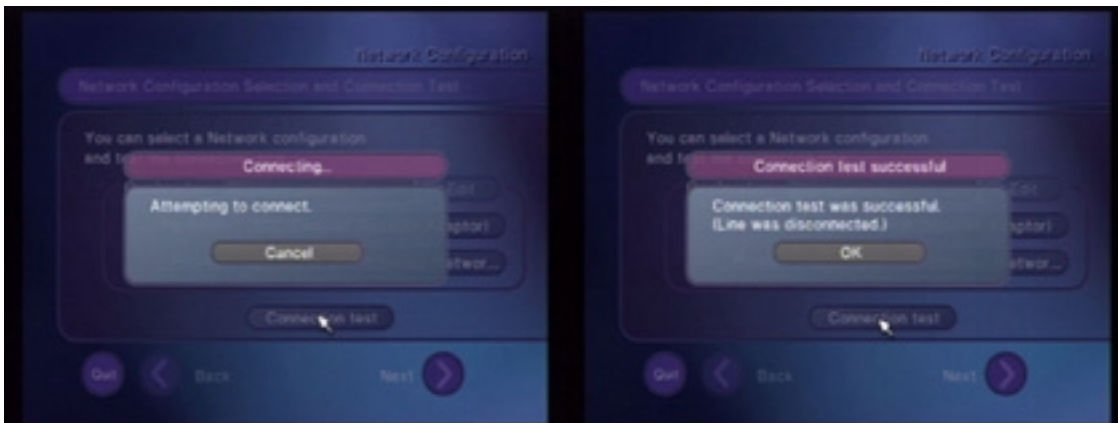
The following procedure is used to perform a connection test.



After starting up the "Network Configuration Selection and Connection Test" screen, if "Connection Test" is lit when hardware and network service provider settings are selected, click "Connection Test."

"Connection Test" will be lit when hardware and network service provider settings are selected, but only if the following condition is satisfied.

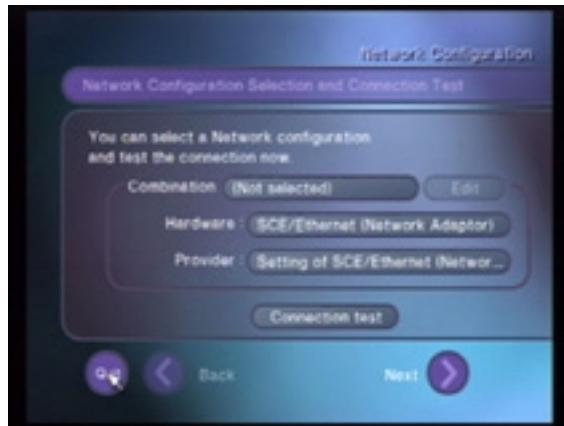
- The hardware that is set in the selected hardware setting is properly connected to the PlayStation 2 console.



After clicking on "Connection Test," the connection test will be performed. If the test completed normally, a dialog box indicating that the test was successful will appear.

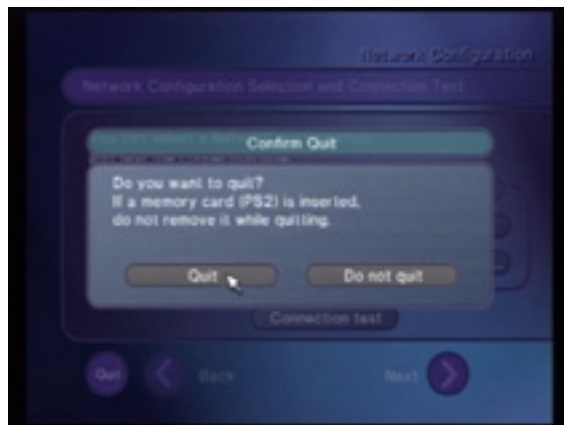
Quitting Network Configuration Processing

The following procedure is used to quit network configuration processing.



Click "Quit," which is always displayed at the lower left portion of the screen and can be selected anytime. Note that the contents of the selected hardware and network service provider settings are returned to the title application only when "Quit" is selected on the "Network Configuration Selection and Connection Test" screen.

When "Quit" is selected, `sceNetGuiCnf_Do()` returns the currently selected contents of the hardware and network service provider settings to a specified area. If no selections were made, `sceNetGuiCnf_Do()` returns the initial values (these are not the same as the default values when the settings were added, which can be specified by the title application).



When a dialog box for confirming the Quit appears, click "Quit."

Library Restrictions

TCP/IP Stack

The network configuration GUI library uses SCEI's Libinet TCP/IP stack. After `sceNetGuiCnf_Do()` terminates, connections are not limited to only those using the selected network configuration.

Specifying the Modem Driver

`sceNetGuiCnf_Do()` returns the vendor name and product name of the hardware settings as part of the selected network configuration. If you want the title application to specify the modem driver, a correspondence table that associates the vendor name and product name with a modem driver is required. To obtain the vendor name and product name that corresponds to a modem driver, you must ask the company that provides the modem driver.

Graphic System

Once `sceNetGuiCnf_Do()` starts, it initializes the graphic system. The state of the graphic system cannot be guaranteed after `sceNetGuiCnf_Do()` returns. The title application must initialize or reinitialize the graphic system before and after using `sceNetGuiCnf_Do()`.

Sound System

`sceNetGuiCnf_Do()` provides no support for the sound system.

IOP Modules

`sceNetGuiCnf_Do()` does not perform any pre-processing or post-processing for IOP modules. The title application must load the IOP modules that are required by `sceNetGuiCnf_Do()` before calling `sceNetGuiCnf_Do()` and must reset the IOP if any unnecessary IOP modules remain after `sceNetGuiCnf_Do()` terminates.

User Interface

Item selection method

`sceNetGuiCnf_Do()` uses the mouse pointer as a pointing method for item selection. It also supports only the direction keys of supported controllers for moving the mouse pointer, as well as USB mice.

Software keyboard

Software keyboard support for `sceNetGuiCnf_Do()` depends on the software keyboard defined by the title application. This is defined with the software keyboard-related callback function of the `sceNetGuiCnf_Arg` structure. The mouse pointer is also used as the pointing method for key selection, in a similar fashion as with item selection.

Character Codes

`sceNetGuiCnf_Do()` supports the following character codes.

- Shift JIS is supported for the displayed character code.
- Shift JIS is supported for communication with the software keyboard.
- UTF-8 is supported for characters that are saved in a network configuration file.

Future Version Upgrade Plans

- Making the specification of the title bar color of a one-choice dialog box separate from the specification of the title bar color of an error dialog box.

