



Platform Notes: The SunHSI/P™ Device Driver

Sun Microsystems, Inc.
4150 Network Circle,
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 816-2346-10
May 2002, Revision B

Send comments about this document to: docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054 Etats-Unis. Tous droits réservés.

Ce produit ou document est distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Contents

Preface v

1. SunHSI/P Device Driver Software 1

Software Overview 1

Installing the Driver Software 1

▼ To Install the Driver Software 2

▼ To Confirm the Software Installation 4

Configuring the Software 4

SunVTS Diagnostic Testing 5

A. `hsip_init` Options 7

Options for T1 Compatibility 7

Inverted Settings 8

Data Signal Inversion 8

Clock Signal Inversion 9

Options for Operating Modes 10

HDLC Mode 10

IBM (SDLC) Mode 11

IBM Full-Duplex Mode 11

IBM Half-Duplex Mode 11

Preface

Platform Notes: The SunHSI/P Device Driver provides configuration instructions for the Solaris™ supplement software used by the SunHSI/P™ adapter and patch panel. These instructions are designed for a system administrator with networking experience.

Using UNIX Commands

This document may not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- *Solaris Handbook for SMCC Peripherals*
- Online documentation for the Solaris software environment
- Other software documentation that you received with your system

Typographic Conventions

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output.	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output.	% <code>su</code> Password:
AaBbCc123	Book titles, new words or terms, words to be emphasized. Command-line variable; replace with a real name or value.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be <code>root</code> to do this. To delete a file, type <code>rm filename</code> .

Shell Prompts

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	<i>machine_name</i> %
C shell superuser	<i>machine_name</i> #
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Related Documentation

For information on how to install the SunHSI/P adapter and patch panel, refer to

the following manuals.

TABLE P-3 Related Documentation

Application	Title	Part Number
Installation & Service	<i>SunHSI/P User's Guide</i>	805-6943
Release Information	<i>SunHSI/P Release Notes</i>	805-6944

Sun Documentation on the Web

The `docs.sun.com` web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

`http://docs.sun.com`

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`smcc-docs@sun.com`

Please include the part number of your document in the subject line of your email.

SunHSI/P Device Driver Software

These Platform Notes include instructions for installing and configuring the software used by the SunHSI/P adapter and patch panel.

This chapter contains the following sections:

- “Software Overview” on page 1
- “Installing the Driver Software” on page 1
- “Configuring the Software” on page 4
- “SunVTS Diagnostic Testing” on page 5

Software Overview

The SunHSI/P device driver offers comprehensive compatibility with Solstice™ WAN software packages. The driver acts as a transparent interface that provides a compliant environment for the SunLink™ wide area network (WAN) packages that currently operate on similar communication modules.

The protocols that operate with the SunHSI/P adapter include systems network architecture (SNA) 3270, SNA Peer-To-Peer, open systems interconnection (OSI), X.25, internetwork router (IR), point-to-point protocol (PPP), and Frame Relay.

Installing the Driver Software

The Solaris CD, which accompanies these Platform Notes, contains the software required by the SunHSI/P adapter.

▼ To Install the Driver Software

1. **Become superuser.**
2. **Use the `prtconf -pv` command to determine if the system contains a SunHSI/P adapter.**

```
# prtconf -pv | grep pci1214,334
compatible: 'pci1214,334' + 'pciclass,078000'
name: 'pci1214,334'
```

Note – `pci1214,334` is the hardware identification label for the SunHSI/P adapter. If you see it listed in the `prtconf` output, your system contains a SunHSI/P adapter.

- If your system contains a SunHSI/P adapter, continue with Step 3.
 - If your system does not contain a SunHSI/P adapter, install the adapter and patch panel as described in the *SunHSI/P User's Guide*.
3. **Use the `pkginfo` command to check the system for previous versions of the Sun HSI/P software.**

Finding the SunHSI/P 1.0 package:

```
# /usr/bin/pkginfo | grep SUNWhsip
system    SUNWhsip    Sun HSI/P Driver for PCI
```

Finding the SunHSI/P 2.0 packages:

```
# /usr/bin/pkginfo | grep SUNWhsip
system    SUNWhsip    SunHSI/P Driver for PCI
system    SUNWhsipm   SunHSI/P Man Pages for PCI
system    SUNWhsipu   SunHSI/P Utilities for PCI
```

- If no SunHSI/P packages are listed, skip to Step 4 to continue with the software installation.
- If any SunHSI/P 1.0 packages are listed, you must remove them as described below.

Removing a SunHSI/P 1.0 package:

```
# /usr/sbin/pkgrm SUNWhsip
```

Removing SunHSI/P 2.0 packages:

```
# /usr/sbin/pkgrm SUNWhsip SUNWhsipu SUNWhsipm
```



Caution – Do not overwrite any existing SunHSI/P software packages. If you install the SunHSI/P software packages over existing SunHSI/P software packages, you will have two instances of the software packages. This may cause problems when installing or backing out of software patches.

4. Install the SunHSI/P software as described in the *Solaris Sun Hardware Platform Guide* that shipped with these Platform Notes.

When you have completed the software installation of your software, and the `pkgadd` utility has run the post-installation script, you will have created the devices illustrated FIGURE 1-1.

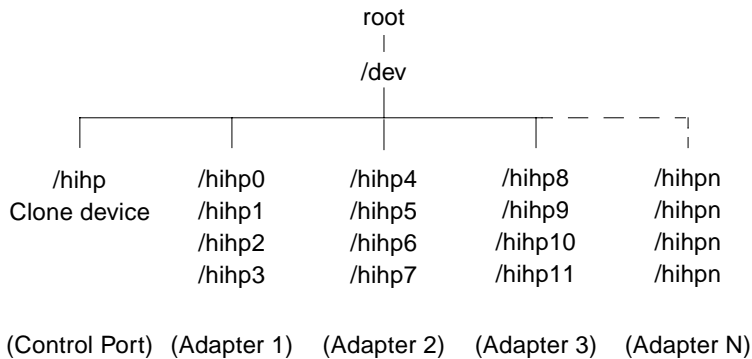


FIGURE 1-1 SunHSI/P Software Device Files

▼ To Confirm the Software Installation

1. Use the following command to test the installation of the hardware and software.

Replace *n* with the SunHSI/P port you want to test.

```
# hsip_loop -c 100 -l 2048 -s 2048000 -t 1 hihpn
```

This command will run an internal loopback test to check the SunHSI/P port. For more information, see the `hsip_loop(1m)` man page.

Configuring the Software

Use the SunHSI/P supplied system commands `hsip_init`, `hsip_loop` and `hsip_stat` to configure your system. While this section provides a short description of each command, please refer to their man pages for more information.

The following man pages are included with the SunHSI/P software:

- `hsip(7d)`
- `hsip_init(1m)`
- `hsip_loop(1m)`
- `hsip_stat(1m)`

Note – If you cannot view these man pages, you need to add the `/opt/SUNWconn/man/` directory to your `MANPATH` environment variable. Depending on the UNIX shell you are using, this variable may be defined in one of a number of startup files.

The `hsip_init` command enables you to modify some of the hardware operating modes common to synchronous serial lines. This can be useful in troubleshooting a link or when operating a communications package. See Appendix A for information about T1 and operating mode `hsip_init` options.

The `hsip_loop` command performs several loopback tests that are useful in exercising the various components of a serial communications link.

The `hsip_stat` command reports the event statistics maintained by the SunHSI/P device driver. The report may be a single snapshot of the accumulated totals or a series of samples showing incremental changes. Prior to these it prints the device name being used to query a particular device.

Note – You *must* be superuser (root user) to run the `hsip_init`, `hsip_stat` or `hsip_loop` utilities.

SunVTS Diagnostic Testing

The SunVTS™ software executes multiple diagnostic hardware tests from a single user interface. It is used to verify the configuration and functionality of most hardware controllers and devices. You operate the SunVTS software primarily from a user interface that enables you to control all aspects of the diagnostic test operation.

The `sunlink` diagnostic test, which is shipped with the SunVTS software, checks the functionality of SunHSI/P adapters. This test can be run from the SunVTS user interface or from the command line. Refer to the *SunVTS Test Reference Manual* for more information about the `sunlink` test.

Note – Some of the `sunlink` tests require a RS-449 loopback plug, which can be ordered through Sun (part number 540-1430).

Refer to the SunVTS manuals for detailed information about the SunVTS software. These documents are available on the Sun Documentation web site (<http://docs.sun.com/>).

hsip_init Options

This appendix provides more information about using the `hsip_init` utility to accommodate differing T1 or CEPT transmission equipment. It also describes the operating modes you can set using `hsip_init`. Refer to the `hsip_init` man page for more information about using this command.

Options for T1 Compatibility

The version of the `hsip_init` command shipped with the SunHSI/P software has options that enable you to invert data and clock signals to accommodate the requirements of T1 or CEPT transmission equipment.

The `hsip_init` parameters that enable for inversion are:

- `txd` - transmit data signal
- `rxid` - receive data signal
- `txc` - transmit clock signal
- `rxid` - receive clock signal

When these parameters are at their default settings, the SunHSI/P software does *not* invert the data or clock signal controlled by the parameter. To invert a signal, you specify a setting of the form *param_name=-paramname*, for example, `txc=-txc`.

As an example, suppose you want to invert the transmit and receive data signals on the first SunHSI/P port (port 0) on the second SunHSI/P adapter in your system. To do so, enter the following command:

```
# hsip_init hihp4 txd=-txd rxd=-rxd
```

To invert both clock and data signals, you would type:

```
# hsip_init hihp4 txd=-txd rxd=-rxd txc=-txc rxc=-rxc
```

Inverted Settings

The following section discusses the background and requirements for these inverted settings. The reason for inverting data signals is distinct from the reason for inverting clock signals.

Data Signal Inversion

The requirement for inverting data signals arises from the “ones density” problem you encounter with most T1 transmission lines in North America. The T1 transmission scheme uses a signaling mechanism known as Alternate Mark Inversion (AMI), in which one bits are represented by a positive or negative pulse, while zero bits are represented by the absence of a pulse. In this scheme, the polarity of each pulse must be the opposite of the polarity of the pulse that immediately preceded it. This signaling scheme makes it possible to embed a reference clock for the data into the data stream itself.

Various types of T1 transmission equipment, such as Data Service Units (DSU), Channel Service Units (CSU), repeaters, and various telephone central office equipment, must be able to keep a phase locked loop (PLL) circuit locked on to this reference clock. This PLL circuit uses the pulses generated when one bits are transmitted to lock the embedded clock to a local reference oscillator. To keep the PLL circuit locked on the extracted clock, a certain density of pulses (one bits) must be guaranteed. For North American T1 lines, the density requirement dictates that at least one out of every 16 bits must be a one (see *AT&T Technical Publication 62411*). In other words, no more than 15 consecutive zero bits can occur anywhere in the data stream.

T1 lines were originally intended to carry voice traffic, wherein the digitized voice signals could be altered to meet the ones-density requirement by forcing every eighth bit of a voice channel to be a one. This practice introduces a small—but virtually inaudible—amount of distortion in the voice signal. Digital data streams between two computers are another matter, since the corruption of even one data bit causes a packet to be rejected. Note that in a typical data packet, it is quite easy to produce bit patterns that violate the ones-density requirement. A random file could easily contain a sequence of bytes that would produce 16 or more consecutive zero bits if transmitted serially.

There are many different schemes for circumventing the ones-density requirement. The most common technique simply reserves every eighth bit of the signal for a “density bit” and forces this bit to be a one. Obviously, these bits are not available for data transmission, which means that 12.5 percent of the bandwidth of the T1 line is wasted. When you consider that the lease cost for a coast-to-coast T1 line can be exceedingly expensive, this waste of bandwidth can be unacceptable.

There are other alternatives. One of them uses a special code that transmission equipment can generate when using the AMI signaling scheme. This special code depends on the fact that two successive one bits that are represented by pulses of the same polarity result in a signal known as a “Bipolar Violation.” A CSU can be designed so that it will automatically replace any string of eight consecutive zeros with a special code pattern that contains two of Bipolar Violations. A compatible, receiving CSU recognizes this special code and converts it back to a pattern of eight zeros. This technique is known by the acronym B8ZS, which stands for Bipolar with 8-Zero Substitution.

All CEPT lines (the European equivalent of T1) mandate the use of a variant of B8ZS that holds the density requirement down to no more than three consecutive zeros. However, telephone companies in North America have been slow to adopt B8ZS, because it would entail a significant capital investment. Therefore, the B8ZS solution will not solve the ones-density problem in the short term.

An alternative to B8ZS—an alternative used by the SunHSI/P product—makes use of the fact that the HDLC framing rules, which specify that any data stream that contains five or more consecutive one bits requires that the transmitting end insert a zero bit after the fifth one bit. This guarantees that the HDLC flag pattern, 01111110 (hex 7E), does not occur randomly inside a frame. The receiving end must automatically discard the zero bit that follows a pattern of five consecutive ones. So, the HDLC framing used by the SunHSI/P software guarantees that in any set of six bits, at least one bit will be a zero (except for the flag pattern). If you include the flag pattern, you can say that in any set of seven bits, at least one bit will be a zero.

By inverting the data signal with HDLC framing on both ends of a link, the HDLC zero insertion algorithm becomes a ones insertion algorithm. This guarantees that in any set of seven bits, at least one bit will be a one. Thus, the HDLC data stream meets the density requirements of North American T1 lines without sacrificing any bandwidth.

Clock Signal Inversion

The need to invert clock lines is separate from the need to invert data lines. Most computer, modem, and terminal vendors adhere to an industry standard specification known as RS-334. This specification defines the relationship between a data bit and a reference clock on a synchronous serial link. The specification also

says that a device should transmit data with reference to the rising edge of the clock signal and that data should be received with reference to the falling edge of the clock signal.

When using long cables or cables not carrying a clock signal, a phase shift may occur, causing a high number of errors. In such cases, inverting the clock signal may correct the phase shift. You may also need to invert the clock signal when connecting a SunHSI/P port to equipment not adhering to the RS-334 standard.

Options for Operating Modes

The SunHSI/P driver operates in two main operating modes: the high-level data link control (HDLC) mode and the IBM (SDLC) mode. The HDLC mode always operates in a full-duplex, point-to-point fashion. While the IBM mode defaults to a full-duplex, point-to-point, operation, you can also set this mode to be either a half-duplex or a multi-point operation.

HDLC Mode

The default operating mode used by the SunHSI/P driver is the HDLC full-duplex protocol (`mode=fdx`). In this mode, the transmitter is always enabled, and it sends flag bytes continuously when it is not sending a data frame.

If no message is currently being transmitted, the driver attempts to start sending its next message. At this point, the driver indicates that it is busy transmitting to prevent the transmission of another message concurrently. The driver also activates a mechanism that ensures that the transmit operation will not hang if the hardware is not responding.

When the transmission is completed, the busy mechanism previously set is cleared and the next message can be transmitted. If the transmission is hung, an abort sequence is sent instead of the cyclic redundancy check (CRC) so that the receiver will not interpret the frame as valid data. The message is discarded, and the output error statistic is incremented, which allows for a proper recovery by higher level protocols.

The received data is buffered until a complete frame has been received. If any error occurs during the reception of a frame, the appropriate statistic is incremented and the frame is discarded.

IBM (SDLC) Mode

This mode is designed to support IBM system network architecture (SNA) communications. It uses most of the same protocols used in HDLC mode, with two major exceptions:

- When the line is idle, instead of sending flag bytes, the transmitter is disabled.
- The request-to-send (RTS) and clear-to-send (CTS) signals are used to gate transmission.

IBM Full-Duplex Mode

When the SunHSI/P software is set to this mode (`mode=ibm-fdx`), the software uses a full-duplex point-to-point communication protocol. Both ends of the link are expected to have RTS and CTS signals asserted at all times when data is being exchanged. When starting a message transmission, the interface raises the RTS signal and expects the CTS signal to be asserted immediately. If this is not done, all messages currently queued for transmission are discarded, and the write operation returns an error.

If the CTS signal drops before the frame transmission is complete, the frame is discarded and the abort error statistic is incremented. If the transmission underruns, an abort sequence is *not* sent and the frame is silently discarded. The RTS signal remains asserted until the data transmission is complete.

IBM Half-Duplex Mode

Half-duplex is a sub-mode of the IBM mode (`mode=ibm-hdx`). Half-duplex mode operates in the same manner as full-duplex mode, except that transmission cannot occur while receiving, and vice versa. When a transmission is completed, the RTS signal is dropped. Dropping the RTS signal tells the remote station to begin transmitting if needed.

IBM Multi Point Mode

In a multi point configuration (`mode=ibm-mpt`), more than two stations “share” a link. This configuration designates one station as a primary station and the rest as secondary stations. In this mode, the port acts as a secondary station. The primary station arbitrates traffic on the link by polling the secondary stations, asking them all if they are ready to transmit.

If a secondary station has data to transmit, it will raise its RTS signal and check for CTS signals. When a CTS signal comes up, the station may begin transmitting, following the same rules for RTS and CTS signals used in half-duplex mode. When

the transmission is complete, the secondary station drops the RTS signal, which enables another station to respond to a poll and begin transmitting. The RTS signal cannot be dropped until the transmission is complete.

Index

A

adapter
 overview, 1
alternate mark inversion (AMI), 8

B

bipolar
 8-zero substitution (B8ZS), 9
 violation, 9

C

CEPT lines, 9
channel service unit (CSU), 8
clock signal inversion, 7, 9

D

data signal inversion, 7
density bit, 9
device driver
 see software, 1
diagnostics
 hsip_loop, 4
 hsip_stat, 4
 SunVTS, 5
 testing software install, 4
directory structure, software, 3

documentation

 sunlink diagnostic, 5
 SunVTS, 5

H

HDLC, 9, 10
hsip_init utility, 4
 inverting
 clock signals, 9
 data signals, 8
 options for T1 compatibility, 7
hsip_loop utility, 4
hsip_stat utility, 4

I

installation
 software, ?? to 4

M

man pages, listed, 4

O

ones density, 8
operating modes
 HDLC, 10

- IBM full-duplex, 11
- IBM half-duplex, 11
- IBM multi-point mode, 11

P

- phased lock loop (PLL), 8
- protocols, supported, 1

R

- RS-334 EIA specification, 9

S

- SDLC mode, 10
- software
 - hsip_init, 4
 - hsip_loop, 4
 - hsip_stat, 4
 - installation, ?? to 4
 - operating modes, 10 to 12
 - protocols, supported, 1
- sunlink diagnostic, 5
- SunVTS
 - sunlink diagnostic, 5

T

- T1 requirements, 7