



Sun HIPPI/P 1.1 Reference Manual

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part Number 806-3674-10
February 2000, Revision A

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and 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, 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.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

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 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et 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 du système 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, 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.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Contents

Intro(1M)	5
blast(1M)	6
hipadmin(1M)	8
hippi(1M)	9
hippiarp(1M)	13
hippid(1M)	16
hippidb(1M)	17
hippidisp(1M)	19
hippidmpd(1M)	22
hippidnld(1M)	24
hippistat(1M)	26
hippitb(1M)	27
hippitune(1M)	28
sink(1M)	31

Maintenance Commands

NAME	Intro - HIPPI/P Administration																										
DESCRIPTION	This section describes the system commands and diagnostics utilities executed in the HIPPI/P environment.																										
LIST OF COMMANDS	<table border="0"> <tr> <td data-bbox="391 373 521 403">blast(1M)</td> <td data-bbox="724 373 1029 403">HIPPI/P diagnostics utility</td> </tr> <tr> <td data-bbox="391 436 574 466">hipadmin(1M)</td> <td data-bbox="724 436 1084 466">HIPPI/P configuration program</td> </tr> <tr> <td data-bbox="391 499 529 529">hippi(1M)</td> <td data-bbox="724 499 1105 529">HIPPI/P control and status utility</td> </tr> <tr> <td data-bbox="391 562 574 592">hippiarp(1M)</td> <td data-bbox="724 562 1252 625">HIPPI/P ARP (address resolution) display and control</td> </tr> <tr> <td data-bbox="391 646 545 676">hippid(1M)</td> <td data-bbox="724 646 1016 676">HIPPI/P support daemon</td> </tr> <tr> <td data-bbox="391 709 561 739">hippidb(1M)</td> <td data-bbox="724 709 1265 739">HIPPI/P driver debug trace display and control</td> </tr> <tr> <td data-bbox="391 772 591 802">hippidisp(1M)</td> <td data-bbox="724 772 1040 802">HIPPI/P NIC display utility</td> </tr> <tr> <td data-bbox="391 835 591 865">hippidmpd(1M)</td> <td data-bbox="724 835 997 865">HIPPI/P dump daemon</td> </tr> <tr> <td data-bbox="391 898 591 928">hippidnld(1M)</td> <td data-bbox="724 898 1208 928">HIPPI/P driver RunCode download utility</td> </tr> <tr> <td data-bbox="391 961 591 991">hippistat(1M)</td> <td data-bbox="724 961 1040 991">HIPPI/P hardware statistics</td> </tr> <tr> <td data-bbox="391 1024 561 1054">hippitb(1M)</td> <td data-bbox="724 1024 1127 1054">HIPPI/P driver debug trace display</td> </tr> <tr> <td data-bbox="391 1087 591 1117">hippitune(1M)</td> <td data-bbox="724 1087 1224 1150">HIPPI/P driver debugging and performance tuning utility</td> </tr> <tr> <td data-bbox="391 1171 513 1201">sink(1M)</td> <td data-bbox="724 1171 1029 1201">HIPPI/P diagnostics utility</td> </tr> </table>	blast(1M)	HIPPI/P diagnostics utility	hipadmin(1M)	HIPPI/P configuration program	hippi(1M)	HIPPI/P control and status utility	hippiarp(1M)	HIPPI/P ARP (address resolution) display and control	hippid(1M)	HIPPI/P support daemon	hippidb(1M)	HIPPI/P driver debug trace display and control	hippidisp(1M)	HIPPI/P NIC display utility	hippidmpd(1M)	HIPPI/P dump daemon	hippidnld(1M)	HIPPI/P driver RunCode download utility	hippistat(1M)	HIPPI/P hardware statistics	hippitb(1M)	HIPPI/P driver debug trace display	hippitune(1M)	HIPPI/P driver debugging and performance tuning utility	sink(1M)	HIPPI/P diagnostics utility
blast(1M)	HIPPI/P diagnostics utility																										
hipadmin(1M)	HIPPI/P configuration program																										
hippi(1M)	HIPPI/P control and status utility																										
hippiarp(1M)	HIPPI/P ARP (address resolution) display and control																										
hippid(1M)	HIPPI/P support daemon																										
hippidb(1M)	HIPPI/P driver debug trace display and control																										
hippidisp(1M)	HIPPI/P NIC display utility																										
hippidmpd(1M)	HIPPI/P dump daemon																										
hippidnld(1M)	HIPPI/P driver RunCode download utility																										
hippistat(1M)	HIPPI/P hardware statistics																										
hippitb(1M)	HIPPI/P driver debug trace display																										
hippitune(1M)	HIPPI/P driver debugging and performance tuning utility																										
sink(1M)	HIPPI/P diagnostics utility																										

NAME	blast - HIPPI/P diagnostics utility
SYNOPSIS	blast [-2PMrkCc] [-D <i>unit</i>] [-I <i>ifield</i>] [-l <i>size</i>] [-n <i>writes-per-pass</i>] [-m <i>passes</i>] [-u <i>ULP</i>] [-R <i>file</i>] [-U]
AVAILABILITY	SUNWhip
DESCRIPTION	<p>Use <code>/etc/opt/SUNWconn/bin/blast</code> to perform diagnostics on a HIPPI/P adaptor. <code>blast</code> can also be used with <code>sink(1M)</code> to analyze system performance.</p> <p><code>blast</code> performs a write operation multiple times. The size of the packet written is specified by <i>size</i>. The number of passes is specified by <i>passes</i>. The number of times the packet is written during each pass is specified by <i>writes-per-pass</i>. For example, to send 800 2Mbyte packets 10 times, you can give the following arguments:</p> <pre>-l 0x200000 -n 800 -m 10</pre> <p>You can also combine <i>writes-per-pass</i> into a single packet by using the <code>-P</code> flag. For example, to send 10 packets, each 1.6 Gbytes in size, you can use the following arguments:</p> <pre>-l 0x200000 -n 800 -m 10 -P</pre> <p>By default, the <code>blast(1M)</code> command writes packets that are four Kbytes in size.</p> <p><code>blast(1M)</code> works with the <code>sink(1M)</code> sample program, which reads packets. <code>sink(1M)</code> reads the HIPPI-FP header in addition to the packet data, so the <code>sink</code> packet size must be at least eight bytes larger than the <code>blast</code> packet size.</p>
OPTIONS	<p>All arguments are optional. Default values are as shown.</p> <p><code>-2</code> Runs <code>blast</code> in double-threaded mode. In this mode, the two threads write to the device simultaneously. This argument cannot be used with <code>-P</code> or <code>-C</code>.</p> <p><code>-P</code> Encapsulates each pass within a single packet. This argument cannot be used with <code>-2</code>.</p> <p><code>-r</code> Sends random data rather than printable ASCII characters. If you use this option, the checking option within <code>sink(1M)</code> is inoperative.</p> <p><code>-C</code> Establishes a long-term connection, allowing the transfer of multiple packets. This argument cannot be used with <code>-2</code>.</p>

- c
When used with the `-r` option, causes a new random packet to be generated for each write. This option simulates a real world application.
- D *unit*
Uses the specified HIPPI/P card. This option is used for platforms that support multiple HIPPI/P cards.
- I *Ifield*
Sets the I-field for the connection to the specified value. The default is zero. For more information, refer to the *Sun HIPPI/P Installation and User Guide*.
- l *size*
Uses the specified size, given in bytes, for the buffer to be transmitted.
- n *writes-per-pass*
Uses the specified number of writes-per-pass. The default is 500. When `-P` is specified, the end of a pass designates the end of a packet. When `-C` is specified, the end of a pass indicates when the connection is dropped.
- m *passes*
Performs the specified number of passes. The default is one.
- u *ULP*
Uses the specified upper layer protocol identifier for the framing protocol header. This identifier must match the upper layer protocol identifier specified by `sink(1M)`. The default is 0x82.
- R *file*
Records performance information in the specified file.
- U
Sends unknown-length (infinite) packets. The actual packet length is specified by `-l`. The packet length must be a multiple of 8 bytes.

SEE ALSO `hippi(1M)`, `sink(1M)`

NAME	hipadmin – HIPPI/P configuration program
SYNOPSIS	hipadmin [-u]
AVAILABILITY	SUNWhip
DESCRIPTION	<p>For each HIPPI/P card present in the system, use <code>/etc/opt/SUNWconn/bin/hipadmin</code> to interactively enter the IP address, netmask, and HIPPI/P switch address in the following format:</p> <pre>address netmask switch_address</pre> <p>The information is then stored in:</p> <pre>/etc/opt/SUNWconn/hippi/hipn.conf</pre> <p>where <i>n</i> is an integer, 0 through 3, inclusive.</p> <p>Each time you run <code>hipadmin(1M)</code>, you must then edit the <code>hippiarp.conf</code> utility to update the ARP configuration information, then run <code>/etc/init.d/hippi start</code>. This sequence causes execution of <code>hippiarp.conf</code>.</p> <p><code>hipadmin</code> also prompts for each NIC EEPROM update.</p> <p><code>hipadmin</code> must be executed after the HIPPI/P package has been installed, and you must have superuser permission to do so.</p>
OPTIONS	<p>The following options are supported:</p> <pre>-u Updates the NIC(s) EEPROM contents.</pre>
SEE ALSO	<code>hippi(1M)</code> , <code>hippitune(1M)</code> , <code>hippidnld(1M)</code> , <code>boot(1M)</code>
NOTE	<p>In the absence of HIPPI/P hardware, this utility asks for the number of interfaces to be configured. Based on the response, it creates the aforementioned <code>hipn</code> files. Then, when the hardware is installed, execute one of the following commands to create HIPPI/P <code>/device</code> nodes and <code>/dev</code> links:</p> <pre>ok boot diskname -r</pre> <pre># drvconfig, devlinks</pre>

NAME	hippi – HIPPI/P control and status utility
SYNOPSIS	<p>hippi [on][[short] [long]] [[fp] [ph]] [[network] [loopback]] [[switched] [direct]] [unit]</p> <p>hippi off [dump] [unit]</p> <p>hippi restart [dump] [unit]</p> <p>hippi status [unit]</p> <p>hippi accept [unit]</p> <p>hippi reject [unit]</p> <p>hippi version</p> <p>hippi cards</p>
AVAILABILITY	SUNWhip
DESCRIPTION	<p>Use <code>/etc/opt/SUNWconn/bin/hippi</code> to display the state of the HIPPI/P driver and hardware or to query the current status or version of the network cards.</p> <p>Any user can execute this command to obtain the status, version number, or number of cards on the network, but only superuser can execute it with its other options.</p>
OPTIONS	<p>The following options are supported:</p> <p>on [short long] [fp ph] [network loopback] [switched direct] [unit]</p> <p>Loads RunCode (firmware) into the HIPPI/P device and starts the device. If the driver is already active, the command fails. When you execute <code>hippi on</code> with any of its options (for example, <code>short</code> or <code>long</code>), the option value is remembered as long as the system remains up, and is reused on the next invocation of <code>hippi on</code>. You can use <code>hippitune(1M)</code> to permanently set default values.</p> <p>IP datagrams over HIPPI/P have a maximum MTU size of 65288 bytes. All HIPPI/P traffic should be limited to 64-Kbyte packet size when IP datagrams are sent over a HIPPI/P network. Setting the <code>short</code> option limits packets to 64 kilobytes, while setting the <code>long</code> option permits any size of packets to be sent over the network. The <code>long</code> option also enables you to use all of the connection-control and packet-control facilities.</p>

Use `fp` to set receive processing to HIPPI-FP mode, or `ph` to set it to HIPPI-PH mode. In `fp` mode, the NIC multiplexes the incoming packets based on the value in the ULP field of the FP header. In `ph` mode, all incoming packets go to the same place. The network driver cannot be used in `ph` mode.

The NIC usually passes HIPPI/P packets through the network interface and out over the network. It also accepts packets from the network and you can use the `loopback` option to place it in internal-loopback mode. In this mode, all packets that are sent out are internally passed back to the receive interface. All connection attempts from the network are rejected.

The NIC usually is connected to a HIPPI-SC switch (switched). To connect it to another NIC, use the `direct` option.

As installed, the defaults for `hippi on` are `short`, `fp`, `network`, and `switched`. You can use `hippitune(1M)` to change the defaults. But to change the operating mode (for example, to `short`, `long`, `fp`, `ph`, `network`, `loopback`, `switched`, or `direct`), you must deconfigure the NIC by using the `ifconfig down`. Refer to the `ifconfig(1M)` man page.

`off [dump]`

Immediately stops the HIPPI/P RunCode and places the system into a state in which it can neither accept nor transmit packets. All pending reads and writes are completed with `EINTR`.

The `-dump` option causes a dump file to be generated. The dump file contains the current state of the driver and RunCode. Customer support can use the dump file to diagnose a problem.

`restart [dump]`

Stops the RunCode. A read or write that is actively passing data is completed with `EINTR`, and the packet is truncated. Reads and writes that are waiting to use the HIPPI/P device are not affected. Firmware is loaded and started. Processing continues with the operation after the failed operation.

The `dump` option causes a dump file to be generated. The dump file contains the current state of the driver and RunCode.

`status`

Queries the current system status and reports whether the system is on or off.

If the system is on, other flags (for instance, `accept`, `reject`, and `long`) indicate if the system is accepting or rejecting connection requests and if the system allows transmission of long packets. `IS_LOOPBACK` is set when the NIC discovers that it is connected to a loopback cable. `IS_DIRECT` is set when the NIC discovers that it is directly connected to another NIC.

`LINK_ON` and `LINK_OFF` reflect the state of the optical link. `RUNCODE_ON` and `RUNCODE_OFF` reflect the operation of the RunCode. Other statistics are:

SRC connections	The number of connections generated
SRC packets	The number of packets sent
SRC failures	The number of errors encountered during an attempt to transmit packets
	No breakdown of errors on transmission is provided. In particular, connection timeouts, connection rejects, and sequence errors are all counted by this one multipurpose counter.
DST packets	The number of packets received
DST rcv on bad ulp	The number of received packets that are destined for a non-active ULP
DST hippi-le drop	The number of packets dropped due to lack of resources in the IP stack
DST data errors	The number of packets received with data errors (either parity or LLRC)
DST sequence err	The number of packets received with HIPPI/P sequence errors
DST sdic lost	The number of times the interconnect signal dropped

`accept`

Sets the system to a mode in which it accepts incoming connection requests. This is the default mode. Use this option to resume accepting connections after you have issued the `reject` option to reject them. The device must be in the on state for this command to work.

`reject`

Sets the system into a mode in which it rejects future incoming connection requests. This command does not affect established connections. The device must be in the on state for this command to work.

version

Reports the driver version number and RunCode version number of each NIC in the system.

cards

Reports the number of NICs in the host system.

SEE ALSO

hippid(1M), hippitune(1M), hippidisp(1M), hippistat(1M)

HIPPI/P 1.1 Installation and User Guide

NAME	hippiarp – HIPPI/P ARP (address resolution) display and control
SYNOPSIS	<p>hippiarp <i>hostname</i></p> <p>hippiarp -a [<i>unit</i>]</p> <p>hippiarp -h [<i>unit</i>]</p> <p>hippiarp -c [<i>unit</i>]</p> <p>hippiarp -s <i>hostname ULA logical-address</i> [<i>unit</i>] [<i>temp</i>] [<i>pub</i>] [<i>dnd</i>]</p> <p>hippiarp -d <i>hostname</i></p> <p>hippiarp -D <i>logical-address</i> [<i>unit</i>]</p> <p>hippiarp -l <i>logical-address</i> [<i>unit</i>]</p> <p>hippiarp -i [<i>unit</i>]</p>
AVAILABILITY	SUNWhip
DESCRIPTION	<p>When entered with only its <i>hostname</i> option, <code>/etc/opt/SUNWconn/bin/hippiarp</code> displays the Internet-to-HIPPI/P address translation table entry used by the Address Resolution Protocol for HIPPI/P (see RFC 1374) for the specified host.</p> <p><code>hippiarp</code> is an extended <code>arp(1M)</code> utility that performs the same functions as <code>arp</code> except for the <code>-f</code>, <code>-u</code>, and <code>-trail</code> options. <code>hippiarp</code> provides additional functions that are specific to HIPPI/P, and it provides ARP address translation information for hosts that do not support ARP over HIPPI/P.</p> <p>HIPPI/P logical addresses are 12-bit numbers that are used by the switch to route the packet. Addresses in the range <code>0xF90</code> through <code>0xFFF</code>, inclusive, are reserved (see HIPPI-SC) and cannot not be set by this utility. When sending an IP packet to a destination host, the driver will set the <code>CAMP-ON</code> and logical routing bits in the I-field for this packet.</p> <p>The adapter may be connected to a switch (switched mode) or directly connected to another adapter (direct mode).</p>
OPTIONS	<p>The following options are supported:</p> <p><i>hostname</i></p> <p>Specify <i>hostname</i> by name or number, using Internet dot notation.</p> <p><i>unit</i></p>

Represents the instance number (for example, `hipn`, where *n* is an integer 0 to 3, inclusive). To view a list of HIPPI/P devices installed on the system, you can execute `hippi version`. See `hippi(1M)`. In single-adapter configurations, *unit* is always optional. In multi-adapter configurations, *unit* is required with each flag used except `-h` and `-a`. The default for `-h` and `-a` is to display all adapters.

`-a [unit]`

Displays all current ARP entries in the kernel table. If you do not specify a unit, the utility displays a line for each unit in the system.

`-h [unit]`

Displays the ULA, logical address, and status information for the specified HIPPI/P unit. If you do not specify a unit, the utility displays a line for each unit in the system.

`-c [unit]`

Clears the ULA-to-logical-address-mapping table for the specified unit of nonreserved and nonpermanent entries. You can delete permanent entries by using the `-d` option. This option requires superuser privileges and returns an EBUSY error if the table is being updated.

`-s hostname ULA switch-address [unit]`

Creates an ARP entry for the specified host with the specified ULA (Universal LAN Address, also known as the IEEE Universal MAC Address), the HIPPI/P logical-switch address (*switch-address*) and, optionally, the specified network unit. You must provide the *unit* option if more than one HIPPI/P unit exists in the system.

The ULA is given as six hexadecimal characters separated by colons. The HIPPI/P switch address is given as three hexadecimal characters (for example, `0x3ef`). If an ARP entry already exists for the specified host, the existing entry is updated with the new information. The entry is permanent unless you specify the `temp` flag.

When a destination HIPPI/P host does not provide a ULA, the ULA must be entered as `0:0:0:0:0:0`. In this case, the utility creates a locally administered ULA that uses the logical address as the low-order 12 bits of the ULA. IP packets directed to the host are sent with zero as the ULA in both the source and destination fields.

Sun HIPPI/P supports only logical addressing for IP traffic. When a logical address of the form 0xXYZ is passed to `hippiarp -s`, the driver uses an I-field in the form 0x07000XYZ when sending IP packets to this remote host, which corresponds to a logical address with the `CAMP_ON` bit set, as specified by HIPPI-SC.

This command requires superuser privileges.

`-d hostname`

Deletes the ARP entry if one exists for the specified host. This command requires superuser privileges.

`-D logical-address -unit`

Deletes the ARP entry if one exists for the specified logical address, as long as no IP address is assigned. This command requires superuser privileges.

`-l logical-address [unit]`

Sets the logical address of the adapter switch. The switch address is coded as described above. If the adapter discovers itself at a different logical address, the discovered address is used. This command requires superuser privileges.

`-i [unit]`

Invalidate the logical address of the adapter. This command requires superuser privileges.

SEE ALSO

`arp(1M)`, `ifconfig(1M)`, `hippi(1M)`

NAME	hippidb – HIPPI/P driver debug trace display and control
SYNOPSIS	hippidb [-t <i>trace-level</i>] [-d <i>trace-level</i>] [-v <i>validation-level</i>]
AVAILABILITY	SUNWhip
DESCRIPTION	Use <code>/etc/opt/SUNWconn/bin/hippidb</code> to display and control the debug trace levels of the HIPPI/P driver. When executed with no options, the <code>hippidb(1M)</code> command displays the current status of type <code>t</code> tracing, type <code>d</code> tracing, and packet validation levels. You must have superuser permission to execute the <code>hippidb(1M)</code> command with any of its options.
OPTIONS	<p>The following options are supported:</p> <p><i>-t trace-level</i></p> <p>Traces the general operation of the driver at the specified level. The higher the trace level, the more noticeable the performance reduction. The following is a list of possible levels.</p> <ul style="list-style-type: none"> ■ 0 - Disable tracing (the default) ■ 1 - Enable general tracing ■ 2 - Enable extensive tracing ■ 3 - Same as 2 <p><i>-d trace-level</i></p> <p>Traces error paths at the specified level. This option does not affect performance. The following is a list of possible levels.</p> <ul style="list-style-type: none"> ■ 0 - Disable tracing ■ 1 - Enable tracing (the default) ■ 2 - Stop the RunCode when RunCode discovers an error <p>At level 2, a dump is extracted and the RunCode is not automatically restarted. The <code>hippi on</code> command restarts RunCode. Refer to <code>hippi(1M)</code>.</p> <p><i>-v validation-level</i></p> <p>Validates the structure of received HIPPI/P packets at the specified level. This option is used on control testing in the network driver. You can use the <code>hippid(1M)</code> utility to extract the trace buffer from the driver and format it into a text file. The following is a list of possible validation levels:</p> <ul style="list-style-type: none"> ■ 0 - Minimal validation (the default)

- 1 - Enable extensive validation (may not be available on all systems)

If the return status is -1, the validation code is not turned on in the driver and you cannot change this value.

SEE ALSO

hippi(1M), hippiarp(1M), hippistat(1M), hippitb(1M)

NAME	hippidisp – HIPPI/P NIC display utility
SYNOPSIS	<p>hippidisp <i>-D unit</i> [<i>general-options</i>] [<i>device-options</i>]</p> <p>hippidisp <i>-f filename</i> [<i>general-options</i>] [<i>device-options</i>]</p> <p>hippidisp <i>-f filename</i> [<i>general-options</i>] [<i>program-options</i>]</p>
AVAILABILITY	SUNWhip
DESCRIPTION	<p>Use <code>/etc/opt/SUNWconn/bin/hippidisp</code>, a diagnostic utility, to display information retrieved from a NIC, either directly by this utility or previously by the dump daemon, <code>hippidmpd(1M)</code>. The <code>hippidisp(1M)</code> command displays the internal structure of RunCode program files. Much of the information displayed relates to the internal operation of the driver and RunCode. This man page does not attempt to describe the various reports in detail.</p> <p>When executed with no options, <code>hippidisp</code> prints out a usage message.</p> <p>The <code>-D [unit]</code> form of <code>hippidisp</code> extracts and displays information about the specified HIPPI/P device. To view a list of HIPPI/P devices installed on the system, you can execute <code>hippi version</code>. See the <code>hippi(1M)</code> for more information about the <code>hippi(1M)</code> command.</p> <p>The <code>-f filename</code> form of the command processes the specified file, then displays the the desired records. Files contain RunCode images or NIC dump images.</p>
OPTIONS	<p>This command supports three types of options, which are described below. General options control the general operation of the utility. Device options, also called NIC Dump options, either directly access a NIC or display a NIC dump file. Program options display a RunCode file. When this command is used with its <code>-f</code> option, device options and program options are mutually exclusive.</p>
General Options	<p><code>-H</code> Displays record headers.</p> <p><code>-P</code> Displays the generally used <i>partial</i> information from the records.</p> <p><code>-F</code> Displays all of the information from the records.</p> <p><code>-x</code> Displays the entire record in hex.</p>
Device (NIC Dump) Options	<p><code>-a</code> Displays all NIC dump records.</p>

- d Displays driver records. These records contain data structures that are used by the driver to manage the NIC.
- r Displays all ring records. The rings are the principle interface between the driver and the NIC. You can have up to 256 receive rings, a send ring, an event ring, and a command ring. A list of descriptors, if any, is printed for each ring. The receive ring number corresponds to the 8-bit ULP number in the incoming packet.
- s Displays statistics for the driver and NIC.
- n Displays the NIC registers.
- l Displays the NIC SRAM contents.
- e *[unit]* Displays the NIC EEPROM contents. The NIC must be halted (`hippi off`) for the EEPROM to be displayed. Refer to `hippi(1M)`.
- m Displays the manufacturing information area of the EEPROM. This area shows the part number and revision for various components of the board (for example, the ULA address, board serial number, and manufacturing data). The NIC must be halted (`hippi off`) for the EEPROM information to be displayed. Refer to `hippi(1M)`.
- t Displays the driver trace buffer. This form of `hippidisp` uses the same format as `hippitd(1M)`.
- N Displays the NIC trace buffer. This form of the `hippidisp` command uses the same format as the `hippitd(1M)` with its `-n` option.

Program File Options

- A Displays all of the program file sections.
- L Displays all of the `LINE` records (that is, the objects that have several line number records). The source-level debugger uses the source code line number and corresponding SRAM address information.
- S Displays all of the symbol table records.

- T Displays all of the text sections (that is, `TXT1`, `TXT2`, and `TEXT` for Phase-1 text, Phase-2 text, and the RunCode text, respectively).
- P Displays all of the program counter records.
- V Verifies the checksum if it follows a text segment.

SEE ALSO

`hippi(1M)`, `hippidmpd(1M)`, `hippistat(1M)`, `hippitb(1M)`,
`hippitune(1M)`

NAME	hippidmpd – HIPPI/P dump daemon
SYNOPSIS	<p>hippidmpd [-a] [-d <i>dump_dir</i>]</p> <p>hippidmpd [-k]</p>
AVAILABILITY	SUNWhip
DESCRIPTION	<p><code>/etc/opt/SUNWconn/bin/hippidmpd</code> is a system daemon. The process forks and the parent dies. When the user requests generation of a dump file by issuing a <code>hippi off dump</code> or <code>hippi restart dump</code> command, and when the watchdog discovers that the NIC is not operating properly, the <code>hippidmpd</code> daemon extracts the relevant information and produces a dump file as described below. Refer to <code>hippi(1M)</code>.</p> <p>Starting and stopping the daemon requires superuser privileges</p> <p>By default, the dump files are placed in <code>/var/hippi</code>.</p> <p>When the first dump file is generated, a <i>Bounds</i> file is created. The Bounds file is an ASCII file that contains one line for each defined HIPPI/P card that has been dumped into the target directory. The fields in the lines are decimal numbers separated by a space. The line is terminated by a newline character. Each line contains the following fields:</p> <ul style="list-style-type: none"> ■ <code>card number</code> - The number of the card that is being dumped ■ <code>set number</code> - The set number of the most recently created set ■ <code>current files</code> - The number of sets for the card ■ <code>max sets</code> - The maximum number of sets allowed for the card (the default is five) <p>As dump requests are received, the daemon produces the requested file. When the maximum number of files is reached, the lowest-numbered file for the card is deleted to make room for the new file.</p> <p>By default, the maximum number of files is five. You can change that number by editing the Bounds file. You can delete any file, including the Bounds file, at any time.</p>
OPTIONS	<p>The following options are supported:</p> <p><code>-a</code> Produces an ASCII dump file instead of the default binary file. Use this option carefully, as the ASCII dump file can be quite large, and it does not contain as much information.</p>

-d *dump_directory* Places dump files in the specified directory. The directory must already exist and root must be able to create and update files there.

-k Kills the daemon process.

SEE ALSO hipp(1M), hippidisp(1M)

NAME	hippidnld – HIPPI/P driver RunCode download utility
SYNOPSIS	<p>hippidnld [-d]</p> <p>hippidnld [-c]</p> <p>hippidnld [-D <i>unit</i>]</p> <p>hippidnld [-l <i>file</i>]</p> <p>hippidnld [-e <i>file</i>]</p> <p>hippidnld [-r <i>file</i>]</p>
AVAILABILITY	SUNWhip
DESCRIPTION	<p>Use <code>/etc/opt/SUNWconn/bin/hippidnld</code> to manage the RunCode download to the NIC. The RunCode can be located in the EEPROM on the NIC or as a cached image in the memory space of the driver.</p> <p>The <code>hippidnld(1M)</code> command lets you use an alternative version of the RunCode for one session, or program it into the EEPROM for regular use. When the NIC is reset, it loads the cached image from the driver, if one is available. Otherwise, it loads an image from the EEPROM.</p> <p>If RunCode is not available, the <code>hippidnld(1M)</code> command fails.</p>
OPTIONS	<p>The following options are supported:</p> <p>-D <i>unit</i> In multcard configurations, use the specified card on which <code>hippidnld</code> is to perform download operations. <i>unit</i> is expressed as <code>hip n</code>, where <i>n</i> is an integer 0 to 3, inclusive. To view a list HIPPI/P devices installed on the system, you can execute <code>hippi version</code>. See the <code>hippi(1M)</code> man page for more information about the <code>hippi(1M)</code> command.</p> <p>-d Deletes the RunCode in the driver cache.</p> <p>-l <i>file</i> Downloads the hex-format RunCode to the driver cache. You can use this option while the NIC is operational. The new RunCode goes into effect when the NIC is restarted.</p> <p>-e <i>file</i> Loads a full RunCode image from a hex-formatted file into the EEPROM and preserves the existing serial number and ULA (Universal LAN Address) values. This option does not alter the driver cache. The target NIC must be halted prior to this operation.</p>

`-r file` Downloads only a RunCode image from a hex-formatted file into the EEPROM manufacturing and header information. It does not alter the driver cache. The target NIC must be halted prior to this operation.

`-c` Clears only the RunCode form the EEPROM. This option preserves manufacturing and tuning data.

SEE ALSO `hippi(1M)`, `hippistat(1M)`

NAME	hippistat - HIPPI/P hardware statistics
SYNOPSIS	hippistat [-D <i>unit</i>]
AVAILABILITY	SUNWhip
DESCRIPTION	Use /etc/opt/SUNWconn/bin/hippistat to display hardware statistics for a HIPPI/P device. To obtain network statistics, use the netstat(1M) command.
OPTIONS	The following options are supported: -D <i>unit</i> Displays statistics about the specified HIPPI/P card. <i>unit</i> is expressed as <i>hipn</i> , where <i>n</i> is an integer 0 to 3, inclusive. To view a list of HIPPI/P devices installed on the system, you can execute <code>hippi version</code> . See the <code>hippi(1M)</code> man page. If -D is not used, the <code>hippistat(1M)</code> command displays information about <code>hip0</code> .
SEE ALSO	hippi(1M), netstat(1M)

NAME	hippitb – HIPPI/P driver debug trace display
SYNOPSIS	hippitb [-n [-D <i>unit</i>]]
AVAILABILITY	SUNWhip
DESCRIPTION	Use <code>/etc/opt/SUNWconn/bin/hippitb</code> to display a formatted version of the driver and RunCode debug trace buffers. It supports driver maintenance. The format of the report depends on the version of the driver and is not detailed here.
OPTIONS	The following options are supported: -n Displays a formatted version of the NIC RunCode debug trace buffer. -D <i>unit</i> Displays information about the specified HIPPI/P device, expressed as <code>hip n</code> , where <i>n</i> is an interger 0 to 3, inclusive. To view a list of HIPPI/P devices installed on the system, execute <code>hippi version</code> . See the <code>hippi(1M)</code> man page. The <i>unit</i> is required only on multiple-adaptor configurations. A single driver trace buffer is shared by all NICs. -D is used only with -n.
SEE ALSO	<code>hippi(1M)</code> , <code>hippiarp(1M)</code> , <code>hippidb(1M)</code> , <code>hippistat(1M)</code>

NAME	hippitune – HIPPI/P driver debugging and performance tuning utility
SYNOPSIS	hippitune [-l] [-p] [-e] [-c <i>retry_count</i>] [-t <i>retry_timer</i>] [-o <i>campon_timeout</i>] [-s <i>stat_timer</i>] [-i <i>interrupt_timer</i>] [-x <i>tx_idle</i>] [-r <i>rx_idle</i>] [-w <i>dma_write_state</i>] [-d <i>dma_read_state</i>] [-h <i>pci_state_req</i>] [-D <i>unit</i>]
AVAILABILITY	SUNWhip
DESCRIPTION	<p>Use <code>/etc/opt/SUNWconn/bin/hippitune</code> to access the registers in a HIPPI/P device that are used for tuning performance and controlling device operation.</p> <p>Default values for the registers are cached in the driver so that they can be set each time the NIC RunCode is started. Specifically, they are stored in the EEPROM on the NIC so that the driver cache can be set at system boot.</p> <p>If you change the counters and timers, the RunCode operation changes immediately. Changes to the state registers become effective when RunCode is restarted. Updates to the EEPROM (-e) do not become effective until the next system boot. The EEPROM can be accessed only while RunCode is off.</p> <p>You can combine the options in a single command. If the system has more than one NIC, you must specify the NIC interface (<i>unit</i>). See -D, below.</p> <p>The <code>hippitune(1M)</code> command displays the current tuning values (-p) and the tuning values stored in the EEPROM (-p with -e). Only the superuser can change the current values and EEPROM values.</p> <p>Time values are specified in 0.97u-sec units.</p>
OPTIONS	<p>The following options are supported:</p> <p>-c <i>retry_count</i></p> <p>Retries a rejected connection the specified number of times before aborting when the HIPPI-SC camp-on bit is not set in the I-field. The <i>retry_count</i> flag is an integer of zero or more.</p> <p>-t <i>retry_timer</i></p> <p>When -c is specified with an integer greater than one, this option waits the specified number of seconds between each retry. <i>retry_timer</i> is an integer.</p> <p>-o <i>campon_timeout</i></p> <p>When the HIPPI-SC camp-on bit is set in the I-field, this option makes the adapter wait the specified number of seconds for the connection to be</p>

accepted. If the connection has not been accepted after this amount of time, consider the connection to be rejected. *campon_timeout* is an integer.

-s *stat_timer*

Places a new snapshot of operating statistics in host memory each *stat_timer* seconds, where *stat_timer* is an integer. If *stat_timer* is set to zero, the statistics are not automatically copied to host memory.

-i *interrupt_timer*

Separates back-to-back interrupts by the specified number of seconds. This option lets you prevent the adapter from generating interrupts faster than the host system can handle them. Use this option carefully; short times tend to flood the host with interrupts and long times tend to reduce responsiveness of the device. The *interrupt_timer* flag is expressed in integers.

-x *tx_idle*

Uses the specified timeout period for idle connections. If a transmit connection has not passed any data for a period of *tx_idle* seconds, where *tx_idle* is expressed as an integer, the connection is aborted.

-r *rx_idle*

Uses the specified receive timeout period for idle connections. If a receive connection has not passed any data for a period of *rx_idle* seconds, where *rx_idle* is expressed as an integer, the connection is aborted.

-w *dma_write_state*

Refers to the Roadrunner specification for bit settings.

-d *dma_read_state*

Refers to the Roadrunner specification for bit settings.

-h *pci_state_reg*

Refers to the Roadrunner specification for bit settings.

-l -e

When used together, these options place the current HIPPI/P flags into the EEPROM.

-p

When used without the `-e` option, the `hippitune -p` command displays the current values of the tuning parameters contained within the driver. When used with the `-e` option, it displays the current values of the tuning parameters in the EEPROM.

-e

Displays and modifies values in the EEPROM. The NIC must be turned off. This option is a modifier for the other options. See **EXAMPLES**.

-D *unit*

Accesses information about the specified HIPPI/P device, expressed as `hipn`, where `n` is an integer 0 to 3, inclusive. To view a list of HIPPI/P devices installed on the system, execute `hippi version`. See the `hippi(1M)` man page. If you have more than one NIC configured, you must specify this option.

EXAMPLES**EXAMPLE 1** Changing `retry_count` in the Driver Cache

The following command changes `retry_count` in the driver cache.

```
# hippitune -c 0x1234
```

EXAMPLE 2 Changing the EEPROM Values

The following command writes the `retry_count` into the EEPROM.

```
# hippitune -c 0x1234 -e
```

EXAMPLE 3 Displaying EEPROM Values

The following command displays the values currently written into the EEPROM.

```
# hippitune -p -e
```

SEE ALSO

`hippi(1M)`

NAME	sink - HIPPI diagnostics utility
SYNOPSIS	sink [-2] [-D <i>unit</i>] [-l <i>size</i>] [-n <i>number-of-reads</i>] [-u <i>ULP</i>] [-c <i>checking-level</i>] [-s] [-v]
AVAILABILITY	SUNWhip
DESCRIPTION	Use /etc/opt/SUNWconn/bin/sink with the blast(1M) command to analyze system performance.
OPTIONS	All of the Arguments are optional and default values are provided.
	-2 Causes the sink(1M) command to run in double-threaded mode. In this mode, two processes are bound to the same ULP at the same time.
	-s Binds sink to the ULP using a shared bind. The default is exclusive bind.
	-D <i>unit</i> Provides sample code for testing the specified HIPPI device, expressed as hip <i>n</i> , where <i>n</i> is an integer 0 to 3, inclusive. To view a list of HIPPI devices installed on the system, you can execute hipp version. See the hipp(1M) man page. If this option is not specified, the sink(1M) command acts for the first HIPPI card displayed by hipp version.
	-l <i>size</i> Uses the specified size, expressed in bytes, for the buffer to be received. The sink(1M) accepts packets of any length. <i>size</i> indicates the number of bytes received at one time, independent of packet size. The default is 4104 bytes, as determined by adding eight bytes for the HIPPI-FP header to the 4096-byte default buffer size of the blast(1M) sample program.
	-n <i>number-of-reads</i> Performs the specified number of reads before exiting. The sink(1M) command continues to execute until it successfully completes this number of reads or encounters an error while in checking mode. The default is to continue reading indefinitely.

- `-u ULP` Uses the specified upper layer protocol identifier for the framing protocol header. The value of the *ULP* flag must match the upper layer protocol identifier specified by the `blast(1M)` command. The default is 0x82.
- `-c checking-level` Uses the specified checking level for `sink`. If the level is 1, only the first data error is flagged; if 2, all data errors are flagged. The default is no checking.
- `-v` Generates verbose output.

SEE ALSO `blast(1M)`, `hippi(1M)`