



Sun Enterprise™ 10000 InterDomain Networks User Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A. 650-960-1300

Part No. 806-4131-10
February 2000, Revision A

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

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. For Netscape Communicator™, the following notice applies: (c) Copyright 1995 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, Sun Enterprise, OpenBoot, 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, Californie 94303 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 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. La notice suivante est applicable à Netscape Communicator™: (c) Copyright 1995 Netscape Communications Corporation. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Sun Enterprise, OpenBoot, 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.



Adobe PostScript

Sun Enterprise 10000 SSP Attributions:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

RESTRICTED RIGHTS: Use, duplication or disclosure by the government is subject to the restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause as DFARS 252.227-7013 and FAR 52.227-19.

This is scotty, a simple tcl interpreter with some special commands to get information about TCP/IP networks. Copyright (c) 1993, 1994, 1995, J. Schoenwaelder, TU Braunschweig, Germany, Institute for Operating Systems and Computer Networks. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that this copyright notice appears in all copies. The University of Braunschweig makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Contents

Preface xiii

Before You Read This Book xiii

How This Book Is Organized xiii

Using UNIX Commands xiv

Typographic Conventions xiv

Shell Prompts xv

Related Documentation xv

Ordering Sun Documentation xv

Accessing Sun Documentation Online xvi

Sun Welcomes Your Comments xvi

1. Introduction to InterDomain Networks 1

Overview of IDN 1

Linking Domains 2

Master Domain 2

Unlinking Domains 3

Force Options 3

Dismantling an IDN 4

Automated IDN Handling 4

Dynamic Reconfiguration and IDNs	4
Attach Operation	5
Detach Operation	5
Network-Wide Arbstops	6
SSP Commands	9
2. Using InterDomain Networks	11
IDN Requirements	11
Domain and SSP	11
OpenBoot PROM Variable	12
Using IDN Commands	12
Viewing IDN Status	13
Using the domain_link(1M) Command	13
▼ To Use the domain_link(1M) Command With Inactive Domains	13
▼ To Use the domain_link(1M) Command With Active Domains for TCP/IP	15
▼ To Create a Basic IDN	16
▼ To Merge IDNs	16
Using the domain_unlink(1M) Command	17
▼ To Unlink a Domain From an IDN	17
A. IDN Error Messages, Notifications, and Panics on the SSP	19
Searching this Appendix	19
Online Searching	20
Special Typographical Conventions	20
Error Type Links	20
Hard-Copy Searching	20
IDN Environment Errors	21
Host Environment Errors	22

General Host Errors	23
IDN-Related Command Errors	26
IDN Glossary	29

Figures

- FIGURE 1-1 Domain Communication Using the SMR in the Master Domain 2
- FIGURE 1-2 Three Isolated Domains 7
- FIGURE 1-3 IDN With Three Domains 8

Tables

TABLE P-1	Typographic Conventions	xiv
TABLE 1-1	SSP Commands Affected by IDNs	9
TABLE A-1	IDN Environment Errors Recorded on the SSP	21
TABLE A-2	Host Errors Recorded on the SSP	22
TABLE A-3	General Host Errors Recorded on the SSP	23
TABLE A-4	IDN-Related Command Errors Recorded on the SSP	26

Preface

This guide describes the InterDomain Network (IDN) feature, which enables dynamic system domains on an Sun Enterprise™ 10000 server to communicate with each other through the use of an internal, high-speed, memory-based, as if they are communicating over a standard network.

Note – This guide does not contain IDN configuration information or domain-side error messages. For information on how to configure an IDN or about domain-side error messages, refer to the appropriate document in “Related Documentation” on page xv.

Before You Read This Book

This guide is intended for the Sun Enterprise 10000 server system administrator who has a working knowledge of UNIX® systems, particularly those based on the Solaris™ operating environment. If you do not have such knowledge, first read all of the books in the Solaris System Administration collection in AnswerBook2™ format provided with your server and consider UNIX system administration training.

Also read and be familiar with the *TCP/IP and Data Communications Administration Guide* that is provided with your server in AnswerBook2 format.

How This Book Is Organized

This document contains the following chapters:

Chapter 1 introduces IDNs and explains their purpose.

Chapter 2 describes how to set up and to use IDNs.

Appendix A contains the IDN-related error messages, notifications, and panics that occur on the system service processor (SSP).

Using UNIX Commands

This document does 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 sources for this information:

- AnswerBook2™ online documentation for the Solaris 2.x software environment, particularly those dealing with Solaris system administration
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic conventions used in this book.

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	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type <code>rm filename</code> .

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

Application	Title	Part Number
User	<i>Sun Enterprise 10000 SSP 3.3 User Guide</i>	806-1500
	<i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i>	806-4122
	<i>Sun Enterprise 10000 Domain Configuration Guide</i>	806-4121
	<i>TCP/IP and Data Communications Administration Guide</i>	805-4003
Reference	<i>Sun Enterprise 10000 SSP 3.3 Reference Manual</i>	806-2888
	<i>Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual</i>	806-4123
	<i>Sun Enterprise 10000 Domain Error Messages</i>	806-4120

Ordering Sun Documentation

Fatbrain.com, an Internet professional bookstore, stocks select product

documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatrain.com at:

<http://www1.fatrain.com/documentation/sun>

Accessing Sun Documentation Online

The `docs.sun.com`SM 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>

Caution – The output of the AnswerBook2 collections depends on the font families you have chosen in your browser. Sun Microsystems suggests that you use a common san-serif font face for regular text and a common fixed-width face for screen text.

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:

`docfeedback@sun.com`

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

Introduction to InterDomain Networks

This chapter contains an overview of IDN and information about Domain IP addresses, dynamic reconfiguration (DR), memory error handling, network-wide arbstops, and system commands and daemons.

Note – For information and procedures on how to configure IDNs, refer to the *Sun Enterprise 10000 Domain Configuration Guide* in the Solaris 8 6/00 on Sun Hardware Answerbook Collection.

Overview of IDN

The InterDomain Network (IDN) feature supports high-speed networking between dynamic system domains (or simply, *domains*) within a single Sun Enterprise 10000 platform. The IDN driver is a DLPI exporting driver that allows domains to communicate with each other using standard networking interfaces, such as Transmission Control Protocol/Internet Protocol (TCP/IP). However, an IDN requires no cabling or special hardware.

IDNs take advantage of the Sun Enterprise 10000 hardware features that enable any set of resident domains to communicate among themselves over the system centerplane using shared memory. A shared memory region (SMR) is used as a conduit for network packets. The SMR is maintained in one domain in the IDN and is used by all other domains in that IDN.

There may be multiple, independent IDNs within a single Sun Enterprise 10000 platform. Each network can comprise multiple logical network interfaces or channels, with each channel representing a separate IP subnet. Configure the number of networks, and the domains that make up a particular network, based on

the performance considerations of your applications. For example, consider which domains require high-speed connectivity and also have sufficient processing power to effectively take advantage of the InterDomain Networks feature.

IDNs can be used for many purposes. For example, IDNs can be used for the following reasons:

- Batch data transfers
- Consolidation of domains

Linking Domains

To link domains to an IDN or to create an IDN, use the `domain_link(1M)` command. The order in which you specify the domain names is not significant. For instructions on how to use the `domain_link(1M)` command, see “To Use the `domain_link(1M)` Command With Inactive Domains” on page 13.

Whenever an argument to `domain_link(1M)` specifies a domain that is already part of an IDN, all other domains in that IDN are also linked by the `domain_link(1M)` command.

Note that when you link domains together in an IDN, each domain can communicate directly with the other domains in the network by using the shared memory region (SMR). There is no priority given to the domains based on the order in which they were added to an IDN.

Master Domain

Only one domain in an IDN is denoted as the master domain. The master domain maintains the SMR, which is used as a conduit for network traffic. For example, if *domain_a* is the master domain, *domain_b* and *domain_c* communicate with each other using the SMR maintained on *domain_a*.

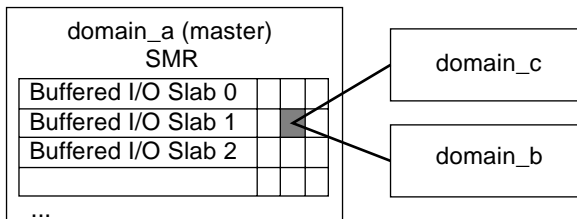


FIGURE 1-1 Domain Communication Using the SMR in the Master Domain

When you create a new IDN out of two domains that do not belong to an existing IDN, the master domain is automatically chosen by the system. After this decision is made, the master domain cannot be changed unless you unlink the master domain or unless the master domain hangs and the network is automatically reconfigured to use an alternate master. An exception to this rule occurs when two existing IDNs are merged by using a single `domain_link(1M)` command. In this case, the system determines which domain from among the two current master domains will become the master domain for the new IDN.

The system chooses the master domain by determining which domain has the greatest processing power and the widest memory bandwidth, which is a function of how many system boards with memory are contained within a domain. The domain with the greatest overall capacity is used as the master domain because it has the responsibility of servicing IDN buffer requests on behalf of other domains.

Unlinking Domains

To unlink a domain from an IDN, use the `domain_unlink(1M)` command, which accepts one or more domains as a parameter. When you unlink a domain, the system broadcasts a message to the remaining domains in the IDN to inform them that they should no longer attempt to communicate with the outgoing domain. Other domains in the network continue to communicate with each other without interruption, both during and after the unlink operation.

Although there is no particular order in which you must deconfigure an IDN link and its associated network interface(s), Sun suggests that you deconfigure the network interface by using the `ifconfig(1M)` command before you unlink the domain to prevent users from unnecessarily using the disconnected link.

By default, the system will not perform an unlink operation on an active domain if *any* domain within the same IDN is in an unknown (AWOL) state, such as halted or hung. The state of the domain is detected and reported when you perform the unlink operation.

Force Options

You can use one of two force options, `-f` or `-F`, to bypass the check for domains in an unknown state and to force the unlink operation to proceed. With the soft force option, `-f`, the `domain_unlink(1M)` command attempts to unlink all of the specified domains in the standard manner; however, if a time-out condition occurs due to the presence of an AWOL domain within the IDN, the `domain_unlink(1M)` command uses the `-F` option to remove the link, forcing the domain to be unlinked.

With the hard force option, `-F`, the `domain_unlink(1M)` command disconnects the specified domain from all of the other domains in the IDN and does so without synchronizing the disconnections. Use this option only when the specified domain is completely nonresponsive (that is, not responding to log in requests) or when it must be isolated from the IDN as part of AWOL recovery.

Caution – The force option must be used only as a recovery mechanism when a domain is known to be in an unknown state (AWOL). It must not be used under standard conditions. It could result in an arbstop if the hardware is reprogrammed while the IDN is active.

Dismantling an IDN

You can dismantle an entire IDN in a single operation, which isolates each domain that is a member of the IDN. Execute the `domain_unlink(1M)` command with at least $n-1$ names of the domains in the IDN, where n is the total number of domains within the IDN.

Automated IDN Handling

The IDN subsystem, in conjunction with support from the SSP, can automatically link and unlink domains. Automatic linking occurs at boot time if the domain has been configured as part of the IDN. Automatic unlinking occurs when one or more IDN members detect and report that another IDN member is not responding to IDN requests. If the master domain is nonresponsive, a new master domain will be elected from the available domains after the master is unlinked. Although the domain is automatically unlinked, the `domain_status(1M)` command still reports the domain as being linked.

Dynamic Reconfiguration and IDNs

DR operations work on individual domains within an IDN. The IDN traffic to and/or from the target domain is paused for only a brief period of time while DR operations are executed on the domain.

Attach Operation

When you attach a board to a domain that is part of an IDN, the following sequence of actions occur:

1. You perform the Init Attach operation.
2. You perform the Complete Attach operation, at which point DR unlinks the domain in which the board resides from the IDN. DR saves the IDN configuration information internally so that DR can relink the domain after the Complete Attach operation.
3. DR then performs the Complete Attach operation.
4. After the Complete Attach operation completes successfully, DR relinks the domain to the IDN.

Detach Operation

During a Detach operation, the following sequence of actions occur:

1. You perform the Drain operation.
2. After the Drain operation has completed and you have selected the Complete Detach operation, DR unlinks the domain from the IDN. DR saves the IDN configuration information internally so that it can automatically relink the domain after the Drain operation.
3. DR then performs the Complete Detach.
4. After the Complete Detach operation, DR relinks the domain to the IDN.

Note – The DR Complete Attach and Complete Detach operations must finish in a timely manner to prevent TCP/IP connections across the IDN from timing out. Typically, the timeout value is two minutes.

If the unlink operation that is performed by the DR subsystem fails for some reason (for example, an AWOL domain exists within the IDN), you must resolve the problem manually (that is, you must manually unlink the AWOL domain) before you attempt the DR operation again.

Network-Wide Arbstops

An arbitration stop, or arbstop, of the domain causes the domain to freeze and all hardware-level transactions to cease. When an arbstop occurs within a domain that is part of an IDN, subsequent arbstops occur in all of the other domains in that same IDN.

Note – Domains that are not members of an arbstopped IDN are *not* affected by the arbstop.

Normally, this is not a problem because arbstops rarely occur. However, if another domain in that IDN is in an unknown state and possibly attempting to communicate with the domain being unlinked, the unlink command can cause arbstops to occur, especially when it is used with the force option.

If a domain or cluster arbstop occurs, the current BBSRAM and arbstop information is dumped to the following files:

- Domain arbstop:

- \$SSPLOGGER/*domain_name*/Edd-Arbstop-bbsram-*time_stamp*

- \$SSPLOGGER/*domain_name*/Edd-Arbstop-Dump-*time_stamp*

- Cluster arbstop:

- \$SSPLOGGER/*domain_name*/Edd-MD-Arbstop-bbsram-*time_stamp*

- \$SSPLOGGER/*domain_name*/Edd-MD-Arbstop-Dump-*time_stamp*

To understand the conditions under which arbstops can occur, consider the hardware architecture that allows system boards to communicate with each other. In the following illustration, three domains exist, none of which are members of an IDN.

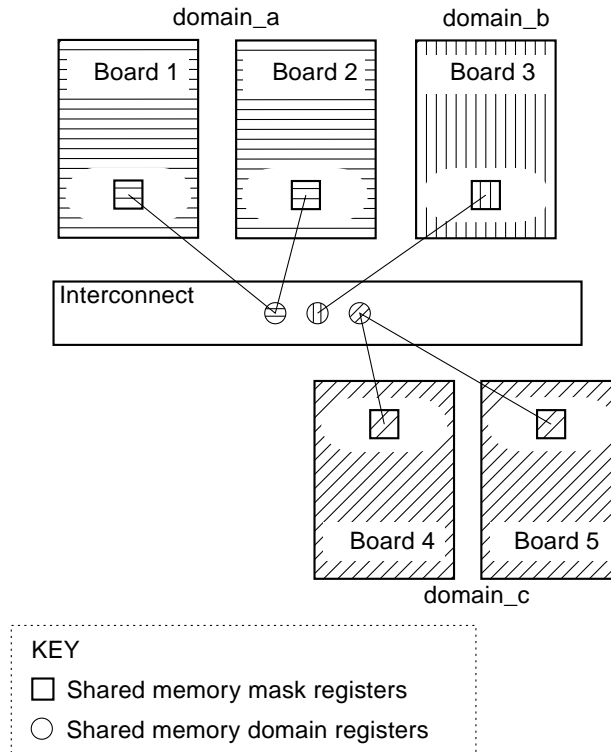


FIGURE 1-2 Three Isolated Domains

The interconnect (that is, the backplane of the system) contains shared memory domain registers, and each board contains shared memory mask registers. These registers work to support interboard communication within a domain, and they facilitate interdomain communication in the IDN environment. The shared memory domain registers on the interconnect allow a message to be forwarded to a particular destination board only if the registers are programmed to allow the originating board to send a message to the specified destination board. Correspondingly, the shared memory mask registers on a destination board allow an incoming message to be accepted only if the registers are programmed to allow that destination board to receive a message from the originating board.

In FIGURE 1-3, the shared memory domain registers on the interconnect for the three domains have been grouped to allow forwarding of messages between the domains. In addition, the boards in each domain now have shared memory mask registers programmed to accept messages from the other domains.

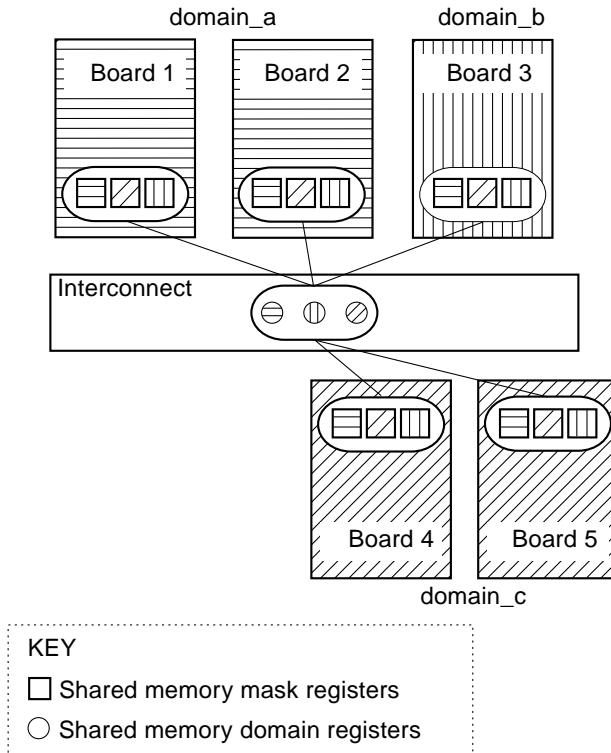


FIGURE 1-3 IDN With Three Domains

In addition to certain hardware failures, an arbstop can occur if any board attempts to send a message to another board and if *either* the shared memory domain registers, or the shared memory mask registers, do not allow communication between the two boards. During the linking or unlinking process, the `domain_link(1M)` and `domain_unlink(1M)` commands reprogram these registers to enable or disable cross-domain transactions.

If one domain is in an unknown state (for example, partially hung) and if you attempt to unlink another domain in the IDN, the `domain_unlink(1M)` command will fail unless you use one of the force options, `-f` or `-F`, because the `domain_unlink(1M)` command needs to communicate to the hung domain to ensure that all interdomain transactions have ceased before it reprograms the registers that are related to the shared memory. The `domain_unlink(1M)` command does not reprogram the shared memory domain registers if the domain is not responding. If you force the unlink to proceed, the shared memory domain registers are reprogrammed; however, the shared memory mask registers on the hung domain are not reprogrammed, so the IDN software on that domain is not aware that the disconnect has taken place.

After all of the IDN-specific registers have been reprogrammed, if the hung domain attempts to communicate with another domain, the hung domain, and any other domains in that IDN can arbstop. Thus, use the force option only if you are certain that the hung domain will not attempt any further communication with the other domains in the same network. To reduce the potential for such an arbstop, either reboot the hung domain or unlink it from the IDN before you unlink any other domain.



Caution – Do not use the force option on a known active domain unless it is absolutely necessary.

SSP Commands

The InterDomain Networks feature affects the behavior of several SSP commands. This section contains an explanation of the behavior of the commands that are affected by IDNs.

The following table contains a list of the SSP commands affected by IDNs.

TABLE 1-1 SSP Commands Affected by IDNs

Command	Affect
<code>bringup(1M)</code>	You must unlink any domain that is in an unknown state (AWOL) before you use the <code>bringup(1M)</code> command to reboot any other domain in the IDN. Note that if multiple domains within the same IDN are hung, you must unlink all of the hung domains simultaneously. In addition, you cannot unlink any nonresponsive domain when other nonresponsive domains are present in the same IDN. Finally, the <code>bringup(1M)</code> , <code>domain_link(1M)</code> , and <code>domain_unlink(1M)</code> commands cannot run concurrently.
<code>domain_remove(1M)</code>	You cannot remove a domain that is currently a member of an IDN. The domain must first be unlinked; then, you can remove it.
<code>dr(1M)</code>	DR commands and IDN commands cannot run concurrently. See “Dynamic Reconfiguration and IDNs” on page 4 for more information about the <code>dr(1M)</code> command.

TABLE 1-1 SSP Commands Affected by IDNs

Command	Affect
<code>edd(1M)</code>	When <code>edd(1M)</code> produces a dump file as a result of an <code>arbstop</code> or <code>recordstop</code> in a domain that is part of an IDN, the dump file is based on the entire set of boards comprising <i>all</i> of the domains that are members of the IDN. The <code>edd(1M)</code> daemon is also required to enable automatic linking of domains and AWOL recovery.
<code>hostint(1M)</code>	By default, you cannot issue a <code>hostint(1M)</code> operation to a domain that is a member of an IDN. Although you can override this restriction by using the <code>force</code> option, you should unlink the domain first.
<code>hpost(1M)</code>	When <code>hpost -wc</code> is used on a domain that is part of an IDN, it clears <code>recordstops</code> on all of the boards within all of the domains in that IDN.
<code>power(1M)</code>	You cannot power off a system board within a domain that is a member of an IDN unless you use the <code>force</code> option with the <code>power(1M)</code> command (refer to the <code>power(1M)</code> man page for more information on the use of the <code>force</code> option). The domain must first be unlinked from the IDN. Then, you can use the <code>power(1M)</code> command to power off the board.
<code>sigbcmd(1M)</code>	By default, you cannot issue a <code>sigbcmd</code> <i>obp</i> or <i>panic</i> operation to a domain that is currently a member of an IDN. Although you can override this restriction by using the <code>force</code> option, you should unlink the domain first.

Using InterDomain Networks

This chapter contains instructions on how to use the IDN commands. Make sure that you read Chapter 1 “Introduction to InterDomain Networks” before you attempt to use the commands in this chapter.

Note – For information and procedures on how to configure IDNs, refer to the *Sun Enterprise 10000 Domain Configuration Guide* in the Solaris 8 6/00 on Sun Hardware Answerbook Collection.

IDN Requirements

This section contains the general and OpenBoot PROM (OBP) requirements for IDNs.

Domain and SSP

Before you can use IDN commands, your system must have, at the minimum, the following software components:

- The host must have a version of the Solaris operating environment for the Sun Enterprise 10000 server, which contains the IDN driver packages (`SUNWidn.u`, the 32-bit binaries, and `SUNWidnx.u`, the 64-bit binaries).
- The SSP must have SSP 3.2, minimally, with all of the current patches.

OpenBoot PROM Variable

OBP has one variable, the shared memory region (SMR) size, `idn-smr-size` that must be set before you link any domains. A value of zero disables the IDN feature. A nonzero value indicates the number of megabytes of kernel space to reserve for the SMR.

To set this variable, boot, or halt, the system to the OBP prompt and set the variable by using the `setenv` command, as in the following example:

```
<#0> ok setenv idn-smr-size 32
```

The value of `idn-smr-size` can be set only at the OBP prompt. You must reboot the domain before the new value can take affect. You can, however, decrease the actual size of the SMR by using the `idn_smr_size idn.conf` variable. For more information about the default and suggested sizes for the `idn-smr-size` variable, refer to the *Sun Enterprise 10000 Domain Configuration Guide*.

Using IDN Commands

The following commands support IDN:

- `domain_link(1M)` - Links domains to form or expand an IDN
- `domain_unlink(1M)` - Unlinks one or more domains from an IDN
- `domain_status(1M)` - Displays information about the domains that make up all of the IDNs on the server

Note – You must be user `ssp` to run the `domain_link(1M)`, `domain_unlink(1M)`, and `domain_status(1M)` commands. Refer to the man pages for these commands in the *Sun Enterprise 10000 SSP 3.3 Reference Manual*.

Viewing IDN Status

The `domain_status(1M)` command returns a listing that provides general information about domains, as well as the list of any IDNs that contain those domains. Here is an example:

```
ssp% domain_status
DOMAIN          TYPE                PLATFORM  OS      SYSBDS
xf3             Ultra-Enterprise-10000  xf3      5.7    4 6 7
xf3-b8         Ultra-Enterprise-10000  xf3      5.7    8 9 13
xf3-b10-hme0   Ultra-Enterprise-10000  xf3      5.6    10 11
xf3-b2         Ultra-Enterprise-10000  xf3      5.8    2 14
xf3-b5-fddi0   Ultra-Enterprise-10000  xf3      5.7    0 1 5

IDN NETWORKS
0: xf3-b2 xf3-b8
1: xf3 xf3-b5-fddi0
```

The section at the bottom of this listing indicates that two IDNs exist on this server. Each IDN is identified by a number followed by the names of the domains that make up that network. Note that the number associated with the IDN is simply a tag used in the listing; it is not a persistent identifier for that IDN.

Using the `domain_link(1M)` Command

This section contains procedures for linking domains to create an IDN. The method for creating an IDN depends on the state of the domains that you want to link together. You can link inactive domains or active domains. For more information about the `domain_link(1M)` command, see “Linking Domains” on page 2.

▼ To Use the `domain_link(1M)` Command With Inactive Domains

The following procedure contains steps for linking two domains, `domain_a` and `domain_b`. If you are linking more than two domains, you must perform the domain-specific steps (that is, those executed at the domain prompt) for all of the domains.

None of the domains that are to be part of an IDN need be up and operational prior to defining an IDN; however, if the domains are not booted, the link operation updates only the logical IDN information maintained by the SSP. When the domain is brought up by using the `bringup(1M)` command, the information on the SSP about the IDN is used to configure the domain.

Note – Before you perform the steps in this procedure, you must ensure that each domain has an `/etc/hostname.idnX` file defined. For more information about this file, refer to the *Sun Enterprise 10000 Domain Configuration Guide*. If this file is not already defined, you must create it for each domain before you proceed with the remaining steps in this section.

1. Use the `domain_switch(1M)` command to ensure that the `SUNW_HOSTNAME` variable is set to the correct domain name.

The domain must be running a version of the Solaris operating environment that supports IDNs. Refer to the IDN release notes for version support information. The `domain_link(1M)` command will not succeed if the `SUNW_HOSTNAME` variable is set to a domain that is running the Solaris 2.5.1, Solaris 2.6, Solaris 7, Solaris 7 3/99, Solaris 7 5/99, or Solaris 7 8/99 operating environment.

2. On the SSP, execute the `domain_link(1M)` command to define an IDN.

```
ssp% domain_link domain_a domain_b
```

Note – Because domains can be linked when they are not booted, you cannot verify that a given domain supports IDN. If the domain does not support IDN, then upon boot, the domain will not be automatically linked.

3. Bring up the domains to the OpenBoot PROM (OBP) prompt.
4. At the OBP prompt, ensure that the IDN driver is enabled.

The `idn-smr-size` variable must be set to a valid nonzero value to enable the IDN driver.

```
<#0> ok printenv
```

If the `idn-smr-size` variable is not set properly, see “OpenBoot PROM Variable” on page 12 for instructions on how to set this variable.

5. Execute the `bringup(1M)` command for each domain.

6. Boot the domains.

After all domains are booted, the IDN between them is automatically enabled by using the SSP services that detect the booted domains.

▼ To Use the `domain_link(1M)` Command With Active Domains for TCP/IP

The following procedure contains steps for linking two domains, `domain_a` and `domain_b`. If you are linking more than two domains, you must perform the domain-specific steps (that is, those executed at the domain prompt) for all of the domains. In the following procedure, both domains are booted.

1. Ensure that each domain has an `/etc/hostname.idnX` file defined.

For more information about this file, refer to the *Sun Enterprise 10000 Domain Configuration Guide*. If this file is not already defined, you must create it for each domain before you proceed with the remaining steps in this section.

2. Use the `eeeprom(1M)` command to ensure that the IDN driver is enabled.

Note – The OBP variable `idn-smr-size` must be set prior to boot so that the operating environments will reserve the appropriate amount of memory for the SMR. By default, `idn_nwr_size` is equal to `idn-smr-size`, so typically, `idn-smr-size` must be set to an equivalent value for all of the domains in the IDN.

The `idn-smr-size` variable must be set to a valid non-zero value to enable the IDN driver.

```
<#0> ok printenv
```

If the `idn-smr-size` variable is not set properly, see “OpenBoot PROM Variable” on page 12 for instructions on how to set this variable.

3. Use the `domain_link(1M)` command to link the domains.

```
ssp% domain_link domain_a domain_b
```

▼ To Create a Basic IDN

The following procedure contains steps to set up a very basic TCP/IP network. Your configuration can vary; therefore, the examples in the steps may not work for your configuration. Refer to the *TCP/IP and Data Communications Administration Guide* for more specific information on how to set up a TCP/IP network.

1. **Use the `domain_switch(1M)` command to ensure that the `SUNW_HOSTNAME` variable is set to the correct domain name.**

The domain must be running a version of the Solaris operating environment that supports IDNs. Refer to the IDN release notes for version support information. The `domain_link(1M)` command will not succeed if the `SUNW_HOSTNAME` variable is set to a domain that is running Solaris 2.5.1 or Solaris 2.6.

2. **Use the `eeprom(1M)` command to ensure that the IDN driver is enabled in each domain.**

The `idn-smr-size` variable must be set to a valid nonzero value to enable the IDN driver. If the `idn-smr-size` variable is not set properly, see “OpenBoot PROM Variable” on page 12 for instructions on how to set this variable.

3. **Plumb and configure the IDN interfaces in each domain that is part of the IDN.**

Refer to the *Sun Enterprise 10000 Domain Configuration Guide* for more information about plumbing and configuring IDN interfaces. Note that you can link domains to an IDN before or after you plumb and configure the IDN interface.

4. **Use the `domain_link(1M)` command to link the domains.**

```
ssp% domain_link domain_a domain_b
```

▼ To Merge IDNs

- **Use the names of two domains in separate IDNs with the `domain_link(1M)` command.**

```
ssp% domain_link domain_a domain_b
```

This command merges the IDN that contains `domain_a` with the IDN that contains `domain_b`. A master domain is chosen for the new IDN from among the domains in both of the existing IDNs.

Using the `domain_unlink(1M)` Command

This section contains instructions for unlinking domains from an IDN. The method of unlinking an IDN depends of the state of the domains that you want to unlink and the state of the other domains in the IDN. For more information about the `domain_unlink(1M)` command, see “Unlinking Domains” on page 3.

▼ To Unlink a Domain From an IDN

1. Use the `domain_status(1M)` command to check the status of all of the domains in the IDN.
2. On the SSP, execute the `domain_unlink(1M)` command to disconnect IDN connections to the domain.

```
ssp% domain_unlink domain_name
```

If the IDN contains domains that are in an unknown (AWOL) state (halted or hung), you must unlink all of the AWOL domains simultaneously, or use one of the force options on the given domain. For example, if `domain_a` and `domain_c` are in unknown states, you should unlink them simultaneously with the following command:

```
ssp% domain_unlink domain_a domain_c
```

If a domain is non-responsive, you can use the force option (`-f` or `-F`) to unlink the given domain.

```
ssp% domain_unlink -f domain_b
```

Caution – Try to unlink all AWOL domains first before you attempt to unlink a domain with the force option. For more information about forcing an unlink operation, see “Force Options” on page 3.

At this point, the domain is fully unlinked from the IDN.

You can dismantle the TCP/IP stack and unlink the IDN connection in any order. Unlinking a domain from an IDN does not necessarily require that the TCP/IP stack be dismantled. In the example above, `idn0` is based on the IPv4 usage. Refer to the IPv6 documentation for the correct usage for IPv6. Note that IPv6 is not supported in the Solaris 7 operating environment. Refer to the `hosts(4)` man page for more information about configuring TCP/IP networks. Refer to the *Sun Enterprise 10000 Domain Configuration Guide* for more information on how to dismantle the IDN interfaces.

Note – If you unlink the last pair of domains in an IDN, the IDN will no longer exist, so no information will appear in the `domain_status(1M)` output.

IDN Error Messages, Notifications, and Panics on the SSP

This chapter contains the IDN error messages, notifications, and panics that occur, or are recorded, on the SSP. For information about IDN error messages, notifications, and panics that occur on the domain, refer to the *Sun Enterprise 10000 Domain Error Messages* book in the Solaris 8 6/00 on Sun Hardware AnswerBook2 collection.

IDN messages that occur on the SSP are sent to the following locations:

- `netcon(1M)` console
- `/var/adm/messages`
- `$SSPLOGGER/messages`

Note – When an IDN-related error occurs, you may see several messages that relate to the error or that give further explanation of the error. Those messages are included in the following tables.

Searching this Appendix

Locating specific error messages in this appendix depends entirely on the media type you are using. If you are using this appendix online, see “Online Searching” on page 20. If you are using this appendix in hard-copy form, search the tables alphabetically starting with the first character in the error message.

Online Searching

You can use the search engine provided in the AnswerBook2™ environment or the search engine in your browser to find a specific string of characters from an error message. Before you construct the search string, keep in mind that this appendix contains special typographical conventions. In addition, you may need to search all of the tables individually. If you know the error type (that is, where the error was encountered), use the hypertext links in “Error Type Links” to start your search.

Special Typographical Conventions

The tables in this appendix contain special typographical conventions for the names of words and values that change, depending on the type of error. When you search for an error message, keep in mind that these names appear as generic representations in italic font. The following list contains the commonly used representations used in this appendix.

- *domain_ID* for the value of the domain ID
- *domain_name* for the names of all domains
- *domain_name_a*, *domain_name_b*, *domain_name_c* for the names of the domains used with the IDN commands
- *platform_name* for the name of the Sun Enterprise™ 10000 platform
- *process_id* for the value of the process ID (pid number)
- *system_board_number* for the number of a system board (that is, 1 through 15)
- *number* for numeric values

Error Type Links

This section contains links to each of the major error type tables.

- “IDN Environment Errors”
- “Host Environment Errors”
- “General Host Errors”
- “IDN-Related Command Errors”

Hard-Copy Searching

If you are searching this appendix in hard-copy form, the tables have been sorted alphabetically to help in your search. The messages in this appendix are grouped by their error type. The error types are IDN environment errors, host environment errors, general host errors, and IDN-related command errors. If you know the type of error, start your search in that section of this appendix.

IDN Environment Errors

The following table contains the IDN environment errors that occur on the SSP.

TABLE A-1 IDN Environment Errors Recorded on the SSP

Error	Probable Cause	Suggested Action
domain_link error: domain_link: File not found in /opt/SUNWssp/release/Ultra-Enterprise-10000/5/5/1/bin or hostobjs. Please check the environment variable SUNW_HOSTNAME.	The domain name in the SUNW_HOSTNAME environment variable does not support the IDN feature. The domain specified in this variable must support the IDN feature.	Set the SUNW_HOSTNAME variable to a valid domain name.
domain_link error: <i>domain_name</i> is not a valid domain. Please set SUNW_HOSTNAME to a valid domain name.	The SUNW_HOSTNAME environment variable is set to an invalid domain name. This variable must be set to the name of a domain that supports the IDN feature.	Set the SUNW_HOSTNAME variable to a valid domain name.
domain_link: a domain cannot be IDN linked to itself. Usage: domain_link domain_name1 domain_name2	The domain names specified in the command were the same.	Use the domain names of different domains.
domain_link: cannot open connection to the snmpd agent domain_link: IDN initialization failed (idnerr = ERR_SSP_SNMP(0x200))	The SSP could not open a connection to the SNMPD agent.	Use the ps(1) command to ensure that the SNMPD process is running. If not, wait until it restarts or until the SSP program starts up successfully, then retry the command.
domain_link: domain [<i>domain_name</i>] OS version (5.6) not supported	The domain, <i>domain_name</i> , does not support the IDN feature. Both of the names specified in the domain_link(1M) command must support the IDN feature.	Use valid domain names with the domain_link(1M) command.
domain_unlink: domain [<i>domain_name</i>] is in a different IDN.	The domain names specified in the command are not in the same IDN.	Use domain names that are in the same IDN.
domain_unlink: domain [<i>domain_name</i>] not found	The domain name specified in the command does not exist.	Select another domain name.

TABLE A-1 IDN Environment Errors Recorded on the SSP (*Continued*)

Error	Probable Cause	Suggested Action
RPC: Miscellaneous tli error - An event requires attention. No such file or directory domain_unlink: IDN initialization failed (idnerr = ERR_SSP_CBS(0x202))	The SSP cannot connect to the CBS.	Use the <code>ps(1)</code> command to ensure that the CBS daemon is running. If not, wait until it restarts or until the SSP program starts up successfully, then retry the command.

Host Environment Errors

The following table contains the host environment errors. These errors indicate that the host set up is invalid.

TABLE A-2 Host Errors Recorded on the SSP

Error	Probable Cause	Suggested Action
domain_link: Another IDN operation is currently IN-PROGRESS (pid <i>process_id</i>) Retry when other operation has completed.	An IDN operation (for example, the <code>domain_link(1M)</code> command, the <code>domain_unlink(1M)</code> command, or an IDN event) was in progress.	Wait for the IDN process to complete successfully, then retry the <code>domain_link(1M)</code> command.
domain_link: ...Checking IDN state of [<i>domain_name</i>]: NOTSUPPORTED domain_link: [<i>domain_name</i>] does not support IDN	The IDN driver was not initialized on the domain, <i>domain_name</i> because the value of the OpenBoot™ PROM (OBP) variable <code>idn-smr-size</code> is set to zero.	Set the value of <code>idn-smr-size</code> to a size other than zero, and bring up the domain. Then, retry the <code>domain_link(1M)</code> command.
domain_link: Failed to acquire IDN specific lock.	Another IDN operation is in progress.	Wait for the current IDN operation to complete successfully. Then, retry the <code>link(1M)</code> operation.

TABLE A-2 Host Errors Recorded on the SSP (Continued)

Error	Probable Cause	Suggested Action
<pre>domain_unlink: ...Checking IDN state of [domain_name] : UNKNOWN domain_unlink: Cannot proceed without known domain state. domain_unlink: IDN UNLINK operation unsuccessful [domain_name] Retry domain_unlink(1M).</pre>	<p>The SSP cannot determine the state of the IDN domain. The IDN driver may be hung, or the CBE is not running.</p>	<p>Check the domain to ensure that the IDN driver is not hung and/or that the CBE is running. If necessary, reboot the domain to reset the IDN state.</p>
<pre>domain_unlink: ...Checking IDN state of [domain_name] : NOTSUPPORTED domain_unlink: Domain [domain_name] does NOT support IDN. Must force (-f) to unlink, if necessary.</pre>	<p>The IDN driver was not loaded on the domain.</p>	<p>Ensure that the domain name is correct, then retry the <code>domain_unlink(1M)</code> command. If necessary, use the soft force option, <code>-f</code>, with the <code>domain_unlink(1M)</code> command.</p>
<pre>domain_unlink: Failed to acquire global lock (bringup_dr.lock). Possibly critical host operation or IDN operation in-progress. Retry domain_unlink(1M).</pre>	<p>A host operation was in progress (for example, <code>bringup(1M)</code>, <code>DR</code>, <code>EDD</code>, or <code>IDN</code>).</p>	<p>Retry the <code>domain_unlink(1M)</code> command after the host operation has completed successfully.</p>

General Host Errors

The following table contains the general host errors.

TABLE A-3 General Host Errors Recorded on the SSP

Error	Probable Cause	Suggested Action
<pre>domain_link: ...Checking IDN state of [domain_name] : ARBSTOP Cannot link with domains that are arbstopped.</pre>	<p>The domain, <i>domain_name</i>, has arbitrarily stopped (<code>arbstopped</code>). If a domain arbitrarily stops, the event-detection daemon (EDD) reboots the domain and relinks the IDN member domains.</p>	<p>Wait for the EDD recovery script to complete successfully. The EDD script relinks the IDN member domains as part of its recovery routine. If the EDD is not running, reboot the domain manually.</p>

TABLE A-3 General Host Errors Recorded on the SSP (Continued)

Error	Probable Cause	Suggested Action
<p>domain_link: IDN LINK operation unsuccessful [<i>domain_name_a</i> + <i>domain_name_b</i>] domain_link: (IDNKERR 0x112) IDN_NWR_SIZE conflicts (expected <i>number</i> MB, actual <i>number</i> MB) with domain id <i>domain_ID</i> domain_link: Unlinking domains. Retry domain_link(1M).</p>	<p>The values of the IDN parameter <i>idn_nwr_size</i> for the specified domains do not match. The domains are automatically relinked to the IDN in which they were members.</p>	<p>Set the values of the <i>idn_nwr_size</i> variable on the specified domain(s) to the same value. Reboot the domain(s) so that the variable(s) take affect, then retry the domain_link(1M) command.</p>
<p>domain_link: IDN LINK operation unsuccessful (link <i>domain_name_a</i> + <i>domain_name_b</i>) domain_link: (IDNKERR 0x109) error programming hardware with respect to domain id <i>domain_ID</i>. Retry domain_link(1M).</p>	<p>The SSP encountered a kernel error IDNKERR_HW_ERROR (0x109) on the host domain.</p>	<p>Examine the domain-specific error for more information about this error.</p>
<p>domain_link: WARNING: Some domains possibly failed to LINK: domain_link: ...Failed to LINK: <i>domain_name</i> domain_link: domain [<i>domain_name</i>] being AWOL domain_link: IDN LINK operation unsuccessful ([<i>domain_name_a</i> + <i>domain_name_b</i>] Retry domain_link(1M).</p>	<p>The domain_link(1M) operation timed out waiting for one of the domains, <i>domain_name</i>, to link successfully to the IDN. The domain was in an unknown state (AWOL).</p>	<p>Check the platform log for an AWOL event. If one is present, wait for it to unlink the AWOL domains. Then, retry the domain_link(1M) command.</p>
<p>domain_unlink: domain [<i>domain_name</i>] being AWOL domain_unlink: IDN UNLINK operation unsuccessful [<i>domain_name</i>] Retry domain_unlink(1M).</p>	<p>The IDN driver failed to unlink the domain.</p>	<p>Check the platform log for an AWOL IDN event. If one is present, wait for the event to unlink the AWOL domain. Then, retry the domain_unlink(1M) command.</p>

TABLE A-3 General Host Errors Recorded on the SSP (*Continued*)

Error	Probable Cause	Suggested Action
<p>domain_unlink: domain <i>domain_name</i> boardset (0x8001) conflicts with MIB (0x1) May need to force (-f) domain_unlink: board configuration conflicts with expected value [<i>domain_name</i>]</p>	<p>The physical board set of the domain is inconsistent with the set in the MIB.</p>	<p>Ensure that the domain name is correct, then retry the domain_unlink(1M) command. If necessary, use the soft force option, -f, with the domain_unlink(1M) command.</p>
<p>domain_unlink: ...Checking IDN state of <i>domain_name_a</i> : UNKNOWN domain_link: ...Checking IDN state of <i>domain_name_b</i> : DOWN domain_unlink: ...Checking IDN state of <i>domain_name_c</i> : DOWN domain_unlink: Verifying IDN UNLINK... Error accessing sigblock fields in bbsram (unlink <i>domain_name_a</i>)</p>	<p>The SSP is unable to read the signature block of the domain.</p>	<p>Ensure that the domain is running. If not, reboot the domain. Then, retry the domain_unlink(1M) command.</p>
<p>domain_unlink: IDN UNLINK operation unsuccessful [<i>domain_name_a</i> - <i>domain_name_b</i>] Retry domain_unlink(1M). domain_unlink: Failed to resolve IDN linkage and unlink domain (<i>domain_name</i>) May need to retry or force (-f) domain_unlink: IDN UNLINK operation unsuccessful [<i>domain_name</i>] Retry domain_unlink(1M).</p>	<p>The linkage is inconsistent between the specified domain, <i>domain_name</i>, and the other IDN member domains.</p>	<p>Use the soft force option, -f, to unlink the domain.</p>

IDN-Related Command Errors

The following table contains command errors that are related to IDN operations.

TABLE A-4 IDN-Related Command Errors Recorded on the SSP

Error	Probable Cause	Suggested Action
<code>domain_remove</code> : Domain ' <i>domain_name</i> ' is linked to an IDN. Domain must be unlinked before it can be removed. See <code>domain_unlink(1M)</code> .	You cannot remove a domain that is a member of an IDN.	Unlink the domain before you use the <code>domain_remove(1M)</code> command.
<code>error</code> : domain <i>domain_name</i> is an IDN member, cannot disable system board <i>system_board_number</i> . Powering off a system board of an IDN domain may result in a cluster arbstop. Unlink the domain before power it off. See <code>domain_unlink(1M)</code> .	You cannot power off system boards within a domain that is a member of an IDN.	Unlink the domain from the IDN. If necessary, use the force option, <code>-f</code> , with the <code>power(1M)</code> command. Use caution with the force option, it could cause a cluster arbstop.
<code>error</code> : domain <i>domain_name_a</i> is an IDN member, cannot disable system board <i>system_board_number</i> . <code>error</code> : domain <i>domain_name_b</i> is an IDN member, cannot disable system board <i>system_board_number</i> . Powering off a system board of an IDN domain may result in a cluster arbstop. Unlink the domain before power it off. See <code>domain_unlink(1M)</code> .	These errors occur when you try to power off a bulk power supply on a system board in a domain that is a member of an IDN.	Unlink the domain from the IDN. If necessary, use the force option, <code>-f</code> , with the <code>power(1M)</code> command. Use caution with the force option, it could cause a cluster arbstop.

TABLE A-4 IDN-Related Command Errors Recorded on the SSP *(Continued)*

Error	Probable Cause	Suggested Action
ERROR: The domain is a member of an IDN. You must unlink the domain before executing this command. See <code>domain_unlink(1M)</code> .	Do not use the <code>sigbcmd(1M)</code> command to panic a domain that is a member of an IDN.	Unlink the domain from the IDN before you use the <code>sigbcmd(1M)</code> command or before you use the <code>sync</code> command in the OBP environment. If the domain is hung, use the force option, <code>-f</code> , to force the panic. Use caution in forcing the panic of a domain that is a member of an IDN. It could cause a cluster arbstop.

IDN Glossary

This contains definitions of abbreviations, words, and phrases that are used in the *Sun Enterprise 10000 InterDomain Networks User Guide*.

A

arbstop A condition in which all ASICs for the given domain cease arbitration for system buses, thus terminating all hardware transactions. Usually, arbstop errors occur when the ASICs detect hardware anomalies such as hardware parity errors or dropped transactions.

AWOL (absent without leave) When a domain is in an unknown state (for example, halted or hung) or when it is in a non-responsive state with respect to IDN requests, it is referred to as being AWOL.

If an IDN member domain detects that another IDN member domain is AWOL, that domain sends a warning message to its console and system log. The message indicates only that the domain failed to respond to an IDN message. It does not necessarily indicate that the domain is *hung*. Typically, an AWOL domain is non-responsive when it stops accepting remote logins or `ping(1M)` operations.

After a domain has been reported as being AWOL, a recovery event occurs on the SSP to resolve the situation, provided that the Event Detector Daemon has been enabled (see `edd(1M)`). A message is logged on the SSP in the SSP-specific system log files indicating the occurrence of the event.

B

- boardmask** 16-bit mask with each bit representing a system board in the Sun Enterprise 10000 server.
- bootbus SRAM (BBSRAM)** 256-Kbyte static RAM attached to each processor PC ASIC. The BBSRAM can be accessed through the PC for reading and writing by using JTAG or the processor. BBSRAM is downloaded when `hpost(1M)` or the OpenBoot PROM start up code is executed. It provides shared data between the downloaded code and the SSP.

C

- CBE** Control Board Executive.
- centerplane** A hardware component that controls the flow of data to and from the system boards that are connected to it.
- CFG message** A cross-domain IDN message exchanged during domain linking. The message contains IDN software and hardware configuration information.
- CIC** Coherency Interface Controller.
- CIC prep buffer** A staging area within the CIC hardware for data targeted for being written to certain CIC registers.
- cluster arbstop** An arbstop condition that involves the set of domains in an IDN.
See also *arbstop*.
- CMD message** A cross-domain IDN message used by IDN member domains to make certain IDN requests, such as SMR slab allocations and domain name, after they are connected to the IDN.
- CON message** A cross-domain IDN message that is exchanged during domain linking to synchronize the connection of an incoming domain with other existing IDN member domains.

D

DLPI (Data Link Provider Interface) A standard defined by the UNIX® International OSI Work Group. DLPI defines the format that STREAMS messages must take when interfacing to the datalink layer.

DMV (Databearing Mondo Vector) A subsystem used to exchange control messages between IDN member domains.

domain ID A unique numeric value that is chosen by the IDN driver and used to identify IDN member domains. This value is based on physical attributes of the domains and is guaranteed to be unique across the entire Sun Enterprise 10000 server. The values range from 0 to 15.

DR (Dynamic Reconfiguration) A software feature that enables you to logically attach and detach system boards to and from the operating system without causing machine downtime.

**dynamic system
domain**

System boards that have been logically grouped together into separate bootable operating environments.

E

Ethernet address The machine address used by the IDN network software to uniquely identify domains in the Sun Enterprise 10000 server.

edd (event-detector daemon) Initiates event monitoring on the Sun Enterprise 10000 server control board.

errno UNIX error number (refer to the `Intro(2)` man page).

F

FIN message A cross-domain IDN message that is exchanged during domain unlinking to synchronize the disconnect of an outgoing domain with other existing IDN member domains.

H

header cookie A unique value that is defined in each SMR mailbox header. The header cookie value uniquely identifies the mailbox header and provides a means for detecting possible data corruption within SMR mailboxes.

I

IDN InterDomain Network.

IDN info operation (SSI_INFO) An SSP based sigblock mailbox operation performed by SSP-based IDN commands to query IDN information from the host-side IDN driver.

IP addresses (Internet Protocol) Refer to the *TCP/IP and Data Communications Administration Guide* for more information.

L

logical interfaces The IDN driver is composed of multiple instances with each instance representing a separate logical network interface. Each logical interface can serve as a separate IP subnet.

lsb/msb Least Significant Bits/Most Significant Bits.

M

MAC header The machine address portion of the Ethernet header that contains the Ethernet address. For the IDN feature, this address is used to uniquely identify the target domain for an IP datagram.

mailbox (In the context of the IDN feature) Represents the point-to-point interdomain mailboxes that reside in the SMR. They are used to transmit IDN data packets between IDN member domains.

See also *SMR*.

master domain The IDN member domain that contains the physical SMR. The master domain exports the SMR to the slave domains. Slave domains have a logical SMR that maps to the physical SMR of the master domain. You can determine which IDN member domain is the master by using the `idd(1M)` parameter `idn_global` for the IDN driver.

See also *slave domain*.

MCADR Memory Controller Address Decoding Register.

MTU Maximum Transfer Unit.

N

netcon Network console (see the `netcon(1M)` man page).

NEGO message A cross-domain IDN message that is exchanged to initiate domain linking and to negotiate which domain is, or becomes, the master domain.

See also *master domain*.

NetWork Region The portion of the SMR that is actually used by the IDN driver for data packet (TCP/IP) communication between domains.

O

OBP OpenBoot™ PROM.

P

plumb In this guide, plumb means to configure the network by using the `ifconfig(1M)` command.

post power-on self-test.

POST2OBP data structure The data structure created by POST, which describes the physical components of the system. Resides in BBSRAM.

S

**signature block
(BBSRAM)**

(or sigblock) See *bootbus SRAM (BBSRAM)*.

**shared memory domain
registers**

Allow a message to be forwarded to a particular system board. The shared memory domain registers are located on the interconnect.

**shared memory mask
registers**

Allow an incoming message to be accepted from a particular system board. The shared memory mask registers are located on the system board.

slab

A unit of allocation of SMR space. Slabs represents an array of fixed-sized SMR buffers to be used for IDN data packets.

slab pool

A structure that is managed by the master domain to administer allocations of the SMR to the slave domains.

slave domain

An IDN member domain that *imports* a logical mapping of the physical SMR from the master domain.

SMR

Shared Memory Region.

snmpd

The SNMP proxy agent listens to a UDP port for incoming requests and services the group of objects specified in `Ultra-Enterprise-1000.mib`. See `snmpd(1M)`.

SSP

System Service Processor enables you to monitor and control the Sun Enterprise 10000 server.

STREAMS

A kernel mechanism that supports development of network services and data communications drivers. STREAMS defines interface standards for character input/output within the kernel, and between the kernel and user level. The STREAMS mechanism comprises integral functions, utility routines, kernel facilities, and a set of structures.

subnet

Local networks with large numbers of hosts are sometimes divided into subnets. See the *TCP/IP and Data Communications Administration Guide* for more information.

U

unplumb

In this guide, unplumb refers to the action of deconfiguring the network.

V

vote ticket A 32-bit quantity that is exchanged by IDN member domains during linking to determine which domain is or will become the master domain. The value is guaranteed to be unique across the entire Sun Enterprise 10000 server.

Index

C

- cabling, no special requirements, 1
- choosing master domain, 3
- configuring number of networks, 1
- consolidating physical networks, 2
- creating IDN network, example, 13

D

- domain_link, 13
- domain_link, 17
- domain_status, 13
- domain_unlink
 - forcing (-f), 3
- domains in IDN, status listing, 13
- DR and IDN, 4
- Dynamic Reconfiguration (DR) and IDN, 4

E

- example of creating IDN network, 13
- example of linking domains, 13
- example of unlinking domain, 3
- example of unlinking domains, 17

F

- forcing unlink, 3

H

- hardware, no special requirements, 1
- hpost and IDN, 10

I

- IDN
 - commands, overview, 12
 - example of creating IDN network, 13
 - multiple IDN networks allowed, 1
 - purpose, 1
 - status listing, 13
- idn-smr-size, 15, 16
- Interconnect (system backplane), 7
- interfaces, standard only, 1

M

- master domain
 - criteria for choosing, 3
- merging domains into IDN network, 2
- multiple IDN networks allowed, 1

N

- number of networks, configuring, 1

O

- overview of IDN commands, 12
- overview of unlinking domain, 3

P

- POST and IDN, 10
- purpose of IDN, 1

S

- Shared Memory Domain Registers, 7
- Shared Memory Mask Registers, 7
- shared memory region
 - idn-smr-size, 14
- shared memory region (SMR), 1, 2
 - idn-smr-size, 15, 16
- SMR (shared memory region), 1, 2
 - idn-smr-size, 15, 16
 - idn-smr-size, 14
- status listing, 13
- system backplane (Interconnect), 7

U

- unlink, forcing (-f), 3
- unlinking domain
 - overview and example, 3

V

- variables
 - idn-smr-size, 15, 16
 - idn-smr-size, 14