

May 1995

Order Number: 313168-001

---

**Paragon™ System Software**

**Release 1.3**

**Installation Guide**

---

**Intel® Corporation**

Copyright ©1995 by Intel Scalable Systems Division, Beaverton, Oregon. All rights reserved. No part of this work may be reproduced or copied in any form or by any means...graphic, electronic, or mechanical including photocopying, taping, or information storage and retrieval systems...without the express written consent of Intel Corporation. The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update or to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication, or disclosure is subject to restrictions stated in Intel's software license agreement. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraphs (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052-8119. For all Federal use or contracts other than DoD, Restricted Rights under FAR 52.227-14, ALT. III shall apply.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

286	i386	Intel	iPSC
287	i387	Intel386	Paragon
i	i486	Intel387	
	i487	Intel486	
	i860	Intel487	

APSO is a service mark of Verdex Corporation

DGL is a trademark of Silicon Graphics, Inc.

Ethernet is a registered trademark of XEROX Corporation

EXABYTE is a registered trademark of EXABYTE Corporation

Excelan is a trademark of Excelan Corporation

EXOS is a trademark or equipment designator of Excelan Corporation

FORGE is a trademark of Applied Parallel Research, Inc.

Green Hills Software, C-386, and FORTRAN-386 are trademarks of Green Hills Software, Inc.

GVAS is a trademark of Verdex Corporation

IBM and IBM/VS are registered trademarks of International Business Machines

Lucid and Lucid Common Lisp are trademarks of Lucid, Inc.

NFS is a trademark of Sun Microsystems

OpenGL is a trademark of Silicon Graphics, Inc.

OSF, OSF/1, OSF/Motif, and Motif are trademarks of Open Software Foundation, Inc.

PGI and PGF77 are trademarks of The Portland Group, Inc.

PostScript is a trademark of Adobe Systems Incorporated

ParaSoft is a trademark of ParaSoft Corporation

SCO and OPEN DESKTOP are registered trademarks of The Santa Cruz Operation, Inc.

Seagate, Seagate Technology, and the Seagate logo are registered trademarks of Seagate Technology, Inc.

SGI and SiliconGraphics are registered trademarks of Silicon Graphics, Inc.

Sun Microsystems and the combination of Sun and a numeric suffix are trademarks of Sun Microsystems

The X Window System is a trademark of Massachusetts Institute of Technology

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

VADS and Verdex are registered trademarks of Verdex Corporation

VAST2 is a registered trademark of Pacific-Sierra Research Corporation

VMS and VAX are trademarks of Digital Equipment Corporation

VP/ix is a trademark of INTERACTIVE Systems Corporation and Phoenix Technologies, Ltd.

XENIX is a trademark of Microsoft Corporation

## **WARNING**

Some of the circuitry inside this system operates at hazardous energy and electric shock voltage levels. To avoid the risk of personal injury due to contact with an energy hazard, or risk of electric shock, do not enter any portion of this system unless it is intended to be accessible without the use of a tool. The areas that are considered accessible are the outer enclosure and the area just inside the front door when all of the front panels are installed, and the front of the diagnostic station. There are no user serviceable areas inside the system. Refer any need for such access only to technical personnel that have been qualified by Intel Corporation.

## **CAUTION**

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

## **LIMITED RIGHTS**

The information contained in this document is copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure by the U.S. Government is subject to Limited Rights as set forth in subparagraphs (a)(15) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052. For all Federal use or contracts other than DoD Limited Rights under FAR 52.2272-14, ALT. III shall apply. Unpublished—rights reserved under the copyright laws of the United States.



# Preface

---

This installation guide contains instructions on how to install Release 1.3 of the operating system. It assumes that you are familiar with the Paragon™ system and system administration.

If you have just received your Paragon system, the operating system, tools, and SATs are already installed. You may still, however, need to configure the Paragon supercomputer for your network and to configure a paging tree. This guide explains how to perform those tasks.

For more detailed information about system configuration, refer to the *Paragon™ System Administrator's Guide*. For more information about the Paragon system, refer to the Paragon system documentation set and the online manual pages.

## Organization

- |           |   |
|-----------|---|
| Chapter 1 | Provides background information that you should know before beginning the system software installation. |
| Chapter 2 | Provides instructions on how to install the operating system.   |

## Notational Conventions

This manual uses the following notational conventions:

- |               |   |
|---------------|---|
| <b>Bold</b>   | Identifies command names and switches, system call names, reserved words, and other items that must be used exactly as shown.   |
| <i>Italic</i> | Identifies variables, filenames, directories, processes, user names, and writer annotations in examples. Italic type style is also occasionally used to emphasize a word or phrase. |
-

**Plain-Monospace**

Identifies computer output (prompts and messages), examples, and values of variables. Some examples contain annotations that describe specific parts of the example. These annotations (which are not part of the example code or session) appear in *italic* type style and flush with the right margin.

**Bold-Italic-Monospace**

Identifies user input (what you enter in response to some prompt).

**Bold-Monospace**

Identifies the names of keyboard keys (which are also enclosed in angle brackets). A dash indicates that the key preceding the dash is to be held down *while* the key following the dash is pressed. For example:

**<Break>**      **<s>**      **<Ctrl-Alt-Del>**

- [ ]      (Brackets) Surround optional items.
- ...      (Ellipses) Indicate that the preceding item may be repeated.
- |      (Bar) Separates two or more items of which you may select only one.
- { }      (Braces) Surround two or more items of which you must select one.

## Applicable Documentation

For information about the manuals shipped with the Paragon system, see the *Paragon™ System Technical Documentation Guide*.

## Comments and Assistance

Intel Scalable Systems Division is eager to hear of your experiences with our products. Please call us if you need assistance, have questions, or otherwise want to comment on your Paragon system.

**U.S.A./Canada Intel Corporation**  
**Phone: 800-421-2823**  
**Internet: [support@ssd.intel.com](mailto:support@ssd.intel.com)**

---

**Intel Corporation Italia s.p.a.**

Milanofiori Palazzo  
20090 Assago  
Milano  
Italy  
1678 77203 (toll free)

**France Intel Corporation**

1 Rue Edison-BP303  
78054 St. Quentin-en-Yvelines Cedex  
France  
0590 8602 (toll free)

**Intel Japan K.K.**

**Scalable Systems Division**  
5-6 Tokodai, Tsukuba City  
Ibaraki-Ken 300-26  
Japan  
0298-47-8904

**United Kingdom Intel Corporation (UK) Ltd.  
Scalable Systems Division**

Pipers Way  
Swindon SN3 IRJ  
England  
0800 212665 (toll free)  
(44) 793 491056 (*answered in French*)  
(44) 793 431062 (*answered in Italian*)  
(44) 793 480874 (*answered in German*)  
(44) 793 495108 (*answered in English*)

**Germany Intel Semiconductor GmbH**

Dornacher Strasse 1  
85622 Feldkirchen bei Muenchen  
Germany  
0130 813741 (toll free)

---

**World Headquarters****Intel Corporation****Scalable Systems Division**

15201 N.W. Greenbrier Parkway  
Beaverton, Oregon 97006  
U.S.A.

(503) 677-7600 (Monday through Friday, 8 AM to 5 PM Pacific Time)

Fax: (503) 677-9147

If you have comments about our manuals, please fill out and mail the enclosed Comment Card. You can also send your comments electronically to the following address:

**[techpubs@ssd.intel.com](mailto:techpubs@ssd.intel.com)**  
(Internet)



# Table of Contents

---

## Chapter 1 Pre-Installation Information

<b>Introduction</b> .....	1-1
<b>Typesetting and Other Conventions</b> .....	1-1
<b>Disk Space Requirements</b> .....	1-2
<b>Information Needed for Installation</b> .....	1-2
<b>Verifying Required Hardware</b> .....	1-3
<b>Powering Up the Paragon™ System Hardware</b> .....	1-3
<b>Logging into the Diagnostic Station</b> .....	1-4
<b>Verifying Configuration Files on the Diagnostic Station</b> .....	1-4
<i>DEVCONF.TXT</i> .....	1-4
<i>MAGIC.MASTER</i> .....	1-5
<b>Backing Up</b> .....	1-6
<b>Understanding Node Numbering</b> .....	1-7
CBS Numbering .....	1-8
Cabinets .....	1-8
Backplanes .....	1-8
Slots .....	1-9
OS Numbering .....	1-10

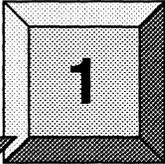
---

## Chapter 2

# Software Installation

<b>Introduction</b> .....	2-1
<b>Installing the Software</b> .....	2-2
Obtaining the Installation Files from Tape .....	2-3
Installing Boot Files on the Diagnostic Station .....	2-4
Installing the Operating System Software .....	2-7
Executing the <i>postboot</i> Installation Script .....	2-15
The Allocator Configuration File .....	2-20
Creating the .service and .compute Partitions .....	2-21
Turning off Autoreboot .....	2-22
Specifying Your Time Zone .....	2-22
Configuring the Paragon™ System for the Network .....	2-22
Resetting the System After Installation .....	2-23
User Notification .....	2-24
<b>Configuring an Additional Ethernet Board</b> .....	2-25
<b>Configuring the System for HIPPI</b> .....	2-28
<b>Installing and Configuring the PFS</b> .....	2-28
The Default /etc/fstab File .....	2-28
The Default /etc/pfstab File .....	2-29
Additional I/O Nodes .....	2-30
<b>Installing the Paragon™ System Acceptance Tests</b> .....	2-33
<b>Updating the MACS Database</b> .....	2-34
<b>Configuring MACS</b> .....	2-34
<b>Configuring NQS</b> .....	2-35

# Pre-Installation Information



1

## Introduction

This chapter provides some background information necessary for installing the Release 1.3 operating system software on the Paragon™ System hardware.

This chapter describes typesetting conventions in addition to those in the Preface, lists the disk space requirements for the system software, and lists information about your system that you need to gather before beginning installation.

The chapter also gives instructions on how to verify the required hardware and software configuration files. To verify the software configuration files, you must power up the Paragon system and log into the diagnostic station.

The chapter lists the directories and files recommended for backup before you begin the installation. A typical backup procedure is described. Finally, the chapter describes node numbering schemes.

## Typesetting and Other Conventions

The procedures in this chapter use the conventions described in the Preface. You should also be aware of the following conventions:

- The instruction “Enter *character(s)*” means type the indicated character(s), and then press the **<Enter>** key. For example, “Enter *y*” means type the letter *y*, and then press the **<Enter>** key.
- In prompts, square brackets surround a default value. Pressing **<Enter>** selects the indicated default value.
- Some steps in these procedures cause a great deal of information to be displayed. However, the steps as described here may show only the last message displayed. Also, do not be concerned if the indicated message does not appear immediately. Some steps take several minutes to complete.

## Disk Space Requirements

Software: Paragon Software Release 1.3  
Installed on Paragon System hardware  
Size: 95M bytes

Software: Paragon Boot Software Release 1.3  
Installed on the diagnostic station  
Size: 8M bytes

The boot software on the diagnostic station requires 8.0M bytes. However, using the diagnostic station to store **tar** files during installation requires additional temporary space of about 150M bytes.

## Information Needed for Installation

To perform the installation, you need to log in as *root* to the diagnostic station, so you will need the root password on the diagnostic station.

During the installation, you will be prompted for the following information:

Paragon host name (also called the network name):

Paragon Internet address:

Paragon disk type:

This is one of the following: RAID3, Maxtor76, Maxtor1240, 1gb, 4gbraid3.

Netmask address for your installation:

Broadcast address for your installation:

Gateway address for your installation:

Network time server for your installation:

You will also need network information about the system where you store the distribution files (the **tar** files on the distribution tapes). This system may be the diagnostic station.

Host name of the server with distribution files:

Internet address of the server with distribution files:

Path of the directory containing the distribution files:

Valid login name and password on the distribution server:

## Verifying Required Hardware

Before performing any of the installation steps, verify the following information.

- Verify that there is an Ethernet connection from both the diagnostic station and from the Paragon boot node to your local network. This is needed to transfer the installation files.
- Verify that there is a serial line from the boot node to the diagnostic station. This line is needed for the Paragon system console interface.

## Powering Up the Paragon™ System Hardware

The Paragon system hardware must be powered up before you can install the operating system. Use the following procedure if the system is not already powered up.

1. Locate the circuit breakers at the bottom on the back of the each cabinet.

If the circuit breakers are already up, there's no need to cycle them. Once the breakers are on, they should not be turned off.

If the breakers are down (off), flip them up starting with cabinet 0 and proceeding in numerical order through the rest of the cabinets. Cabinet 0 is the cabinet containing the diagnostic station, and is the leftmost cabinet when viewed from the back or the rightmost when viewed from the front.

2. Open the back door of any cabinet. Notice the green power control board with red LEDs. This board has three white buttons on its upper-right corner.
  - A. Press the bottom button to shut down power to the nodes.
  - B. Wait about ten seconds for the disks to spin down and then press the top button to power up the nodes.
3. Power on the diagnostic station.
  - A. The diagnostic station is located in cabinet 0, which is the rightmost cabinet when facing the cabinets from the front. Open the front door of this cabinet and swing out the diagnostic station.
  - B. Turn on the power switch located on the back of the diagnostic station.

## Logging into the Diagnostic Station

When the diagnostic station is powered up and in multiuser mode, log in as *root*. You can log in directly to the diagnostic station, or you can **rlogin** to the diagnostic station from another system on the network.

When you log directly into the diagnostic station, you have a choice to use either the Open Desk Top or the virtual terminal facility.

- **Using the Open Desk Top (ODT).** It is not necessary to run the Open Desk Top (ODT) on the diagnostic station. If you choose to do so, enter the following command:

```
DS# startx
```

To create another window with the ODT, use the left mouse button to click on the root window and select "New Window" from the menu.

The mouse has a trackball for cursor movement. The top button is considered to be the left button; the bottom button is the right. Use the mouse to switch between windows.

- **Using the virtual terminal facility.** If you choose to use the virtual terminal facility, use **<Alt-F1>** and **<Alt-F2>** to switch between windows.

**<Alt-F1>** Puts you in the default window. This is the original window and is referred to as the diagnostic window.

**<Alt-F2>** Switches to what will be referred to as the console window.

## Verifying Configuration Files on the Diagnostic Station

The *MAGIC.MASTER*, *DEVCONF.TXT*, and *MAGIC.md* files must be in the boot directory (normally */usr/paragon/boot*) on the diagnostic station.

The *SYSCONFIG.TXT* file is automatically created during installation. An existing *SYSCONFIG.TXT* is saved as *SYSCONFIG.OLD*. Also, a *SYSCONFIG.BIN* file is created or overwritten in the diagnostic directory, */u/paragon/diag*. This is a hardcoded path inside *initutil*, and so *SYSCONFIG.BIN* should not be moved.

### ***DEVCONF.TXT***

The *DEVCONF.TXT* file that describes the configuration of the devices on the Paragon system. It contains one entry for each device in the system. This file must be in the boot directory (normally */usr/paragon/boot*).

The following is an example of a typical *DEVCONF.TXT* file:

```
DEVICES
ENET 00D03
TAPE 00D03 ID 5 DAT
TAPE 00D03 ID 6 DAT
TAPE 00C01 ID 6 DAT
RAID 00A01 ID 0 SW 3.06 LV 3 DC 5 SID 0 RAID3
RAID 00B01 ID 0 SW 3.06 LV 3 DC 5 SID 0 RAID3
RAID 00C01 ID 0 SW 3.06 LV 3 DC 5 SID 0 RAID3
RAID 00D03 ID 0 SW 3.06 LV 3 DC 5 SID 0 RAID3
RAID 01A12 ID 0 SW 3.06 LV 3 DC 5 SID 0 RAID3
RAID 01B12 ID 0 SW 3.06 LV 3 DC 5 SID 0 RAID3
MIO 00A01 H04
MIO 00B01 H04
MIO 00C01 H04
MIO 00D03 H04
MIO 01A12 H04
MIO 01B12 H04
HIPPI 00A05 H04
END_DEVICES
```

This file should already exist on your diagnostic station. Refer to the *Paragon™ XP/S Diagnostic Reference Manual* for more information about the format of this file.

## **MAGIC.MASTER**

The **reset** script uses the *MAGIC.MASTER* file, which must contain the following lines:

```
BOOT_FIRST_NODE=boot_node
BOOT_CONSOLE=cm
BOOT_GREEN_LED=Dci
BOOT_RED_LED=Dcgl
```

In this listing, *boot\_node* is an integer that uses the OS node numbering method to identify the boot node. The *BOOT\_CONSOLE* string can be set to **f** to specify the **fscan** console interface or to **cm** to specify the **async** console interface using the serial port. **cm** is the default.

You can control the operation of the **reset** script by defining strings in *MAGIC.MASTER*. This facility exists to minimize the editing of the **reset** script itself. A variable in the **reset** script effectively gets a new value if you define a string in *MAGIC.MASTER* with **RST\_** prepended to the variable name. For example, to change the value of the variable **SERIAL\_DEVICE** in the **reset** script, you can define **RST\_SERIAL\_DEVICE** in *MAGIC.MASTER*.

The strings `RST_CONFIGURATION` and `RST_TOP_BACKPLANE` follow slightly different rules. The corresponding variables `CONFIGURATION` and `TOP_BACKPLANE` are not defined in the `reset` script. If `RST_CONFIGURATION` and `RST_TOP_BACKPLANE` are not defined in `MAGIC.MASTER`, `reset` calls another utility that calculates the appropriate values from the hardware configuration. If the calculated values differ from the defaults (`CONFIGURATION = full` and `TOP_BACKPLANE = D`), a message prints on the screen that displays the new values. If `CONFIGURATION` and `TOP_BACKPLANE` are defined in the `reset` script, their values are ignored.

Sometimes for debugging purposes, you may want to define a configuration that is different from your actual hardware. You would do this by editing `SYSCONFIG.TXT`. You can remove an entire backplane by deleting the lines specifying its configuration or identifying its slots as `EMPTY`. If you do either, you must specify the new configuration with `RST_CONFIGURATION` and `RST_TOP_BACKPLANE`. If `RST_CONFIGURATION` and `RST_TOP_BACKPLANE` are not present, the `reset` script will determine the real hardware configuration, not the made-up one in `SYSCONFIG.TXT`.

When you execute the `reset` script with the `ramdisk` option, it uses the `MAGIC.md` file instead of `MAGIC.MASTER`. `MAGIC.md` must contain at least the following lines:

```

BOOT_FIRST_NODE=0
REAL_FIRST_NODE=boot_node
BOOT_ROOT_DEV=md0a
NUM_BUFFERS=10
BOOT_VM_PAGESIZE=4096

```

Ensure that in this file, `REAL_FIRST_NODE` is equal to the `BOOT_FIRST_NODE` value in `MAGIC.MASTER`. As in the `MAGIC.MASTER` file, `MAGIC.md` uses the OS node numbering method. The exception is `BOOT_FIRST_NODE` in `MAGIC.md`, which does not follow a numbering scheme and must be 0.

`NUM_BUFFERS=10` and `BOOT_VM_PAGESIZE=4096` are the defaults and need not be explicitly set.

## Backing Up

Before you begin the installation, you should back up any directories and files on your Paragon system that you wish to keep.

Normally you would back up files by copying them to a tape in the DAT drive. For example, to back up `/home` on a DAT tape, log into the Paragon system, insert a DAT tape into the drive, and issue the following command:

```
# dump -0 -f /dev/io0/rmt6 -c /home
```

The DAT device name is `rmt6` on the Paragon system.

Consider backing up the following files and directories on the Paragon system:

```

/etc/TIMEZONE
/etc/bootmesh.log (for your records--do NOT restore on the new system)
/etc/devtab
/etc/fstab (for comparison--do NOT restore on the new system)
/etc/pfstab (for comparison--do NOT restore on the new system)
/etc/group
/etc/hippi.conf
/etc/hosts
/etc/hosts.equiv
/etc/networks
/etc/passwd
/etc/phones-file
/etc/printcap
/etc/profile
/etc/remote-file
/etc/resolv.conf
/sbin/init/init.d/inet
/etc/services
/etc/shells
/etc/uucp
/home (user accounts)
/usr/local (or whatever local directories you have)
/usr/spool/macs (for comparison--do NOT restore on the new system)
/usr/spool/nqs (for comparison--do NOT restore on the new system)
/var/adm/cron/at.allow
/var/adm/cron/at.deny
/var/adm/cron/cron.allow
/var/adm/cron/cron.deny
/var/adm/sendmail/aliases
/var/adm/sendmail/sendmail.cf
/var/adm/wtmp (for your records--do NOT restore on the new system)
/var/spool/mail

```

Also, you might want to save the output from a **last** command for your history records.

## Understanding Node Numbering

There are two node numbering schemes on the Paragon system: *CBS* (Cabinet:Backplane:Slot) numbering, and *OS* numbering (also called *root partition* numbering).

You should become familiar with both schemes, because you will encounter them both. For example, the *SYSCONFIG.TXT* file uses *CBS* numbering because it is primarily a hardware configuration file, and the *MAGIC.MASTER* file uses *OS* numbering because it is primarily a software configuration file.

## CBS Numbering

To identify a node with CBS numbering, you specify its cabinet, backplane, and slot. The following display shows the first few lines of an example *SYSCONFIG.TXT* file, which illustrates how the CBS numbers are used: the first entry indicates cabinet 0, the BP indicates backplane D, and the multiple S's indicate Slots 0 through 15.

```
CABINET 0
PC AP04
LED AB14

BP A AD00
S 0 EMPTY
S 1 MPNODE AG00 64 MRC 04 NIC B02 MIO H04 CTRL0 RAID3
S 2 EMPTY
S 3 EMPTY
S 4 MPNODE AG00 64 MRC 04 NIC B02
S 5 MPNODE AG00 64 MRC 04 NIC B02
S 6 MPNODE AG00 64 MRC 04 NIC B02
S 7 MPNODE AG00 64 MRC 04 NIC B02
S 8 MPNODE AG00 64 MRC 04 NIC B02
S 9 MPNODE AG00 64 MRC 04 NIC B02
S 10 MPNODE AG00 64 MRC 04 NIC B02
S 11 MPNODE AG00 64 MRC 04 NIC B02
S 12 MPNODE AG00 64 MRC 04 NIC B02
S 13 MPNODE AG00 64 MRC 04 NIC B02
S 14 MPNODE AG00 64 MRC 04 NIC B02
S 15 MPNODE AG00 64 MRC 04 NIC B02
```

## Cabinets

Cabinet numbering begins with Cabinet 0 which is the right-most cabinet as viewed from the front.

## Backplanes

There are up to four backplanes in a cabinet. The backplanes are identified alphabetically, starting with A at the bottom. Each backplane consists of up to 16 nodes, which are arranged linearly in slots.

## Slots

When the nodes are viewed from the front, the slot numbering starts at the right of the card cage and moves left. To find the LED associated with a slot, you begin counting at the lower-right corner of the backplane's block of LEDs, count up the right column, and then over to the next column to the left, and then back down in a serpentine fashion; that is, Slot 0 is the lower-right LED, and Slot 15 is the lower-left LED.

The following shows how the slots in a single backplane map to the LEDs in the cabinet door.

```
15  14  13  12  11  10  9  8 <perf> 7  6  5  4  3  2  1  0
```

Physical layout of slots in card cage (front view)

The <perf> indicates the empty performance monitor slot.

```
12      11      4      3
13      10      5      2
14       9      6      1
15       8      7      0
```

Mesh arrangement of slots as shown on LED panel (front view)

## OS Numbering

OS numbering starts at the top left of the left-most cabinet and spans all of the cabinets. When you reach the end of the right-most cabinet, return to the left-most cabinet, drop down a row, and continue counting. OS node numbering is also called root partition node numbering.

The following shows the OS numbers and the CBS numbers for all the nodes in a two-cabinet system. A row of LEDs is represented by two rows of numbers. The top row of numbers contains the OS numbers and the bottom row contains the CBS numbers. For example the node with an OS number of 21 has a CBS number of 0D09.

Cabinet 1				Cabinet 0				
0	1	2	3	4	5	6	7	OS number
1D12	1D11	1D04	1D03	0D12	0D11	0D04	0D03	CBS number
8	9	10	11	12	13	14	15	
1D13	1D10	1D05	1D02	0D13	0D10	0D05	0D02	
16	17	18	19	20	21	22	23	Backplane D
1D14	1D09	1D06	1D01	0D14	0D09	0D06	0D01	
24	25	26	27	28	29	30	31	
1D15	1D08	1D07	1D00	0D15	0D08	0D07	0D00	
=====								
32	33	34	35	36	37	38	39	
1C12	1C11	1C04	1C03	0C12	0C11	0C04	0C03	
40	41	42	43	44	45	46	47	
1C13	1C10	1C05	1C02	0C13	0C10	0C05	0C02	
48	49	50	51	52	53	54	55	Backplane C
1C14	1C09	1C06	1C01	0C14	0C09	0C06	0C01	
56	57	58	59	60	61	62	63	
1C15	1C08	1C07	1C00	0C15	0C08	0C07	0C00	
=====								
64	65	66	67	68	69	70	71	
1B12	1B11	1B04	1B03	0B12	0B11	0B04	0B03	
72	73	74	75	76	77	78	79	
1B13	1B10	1B05	1B02	0B13	0B10	0B05	0B02	
80	81	82	83	84	85	86	87	Backplane B
1B14	1B09	1B06	1B01	0B14	0B09	0B06	0B01	
88	89	90	91	92	93	94	95	
1B15	1B08	1B07	1B00	0B15	0B08	0B07	0B00	
=====								
96	97	98	99	100	101	102	103	
1A12	1A11	1A04	1A03	0A12	0A11	0A04	0A03	
104	105	106	107	108	109	110	111	
1A13	1A10	1A05	1A02	0A13	0A10	0A05	0A02	
112	113	114	115	116	117	118	119	Backplane A
1A14	1A09	1A06	1A01	0A14	0A09	0A06	0A01	
120	121	122	123	124	125	126	127	
1A15	1A08	1A07	1A00	0A15	0A08	0A07	0A00	
=====								

# Software Installation

---

2

## Introduction

This chapter describes how to install Release 1.3 operating system software on the Paragon system hardware.

Upgrading from a previous release is similar to the complete installation, except that you may be able to skip some of the configuration steps. For example, the configuration of additional I/O nodes is preserved from the previous installation unless you are adding new hardware or choose to rebuild the root file system.

Before performing any installation, please contact your on-site Intel Parallel Systems Engineer. See the Preface section “Comments and Assistance” for information about contacting SSD Customer Support.

## Installing the Software

Before installing the operating system software, install the diagnostic software. Refer to the *Paragon™ System Diagnostic DIAG2.0 Release Notes* for information about how to install diagnostic software.

- **Installation Time.** Approximately 2 hours for the complete update from Release 1.2 to Release 1.3 .
- **Installation Medium.** The system software is provided on three 0.25-inch QIC 150 cartridge tapes. In some special cases, installation **tar** files are transferred from SSD via **ftp**.

- Paragon™ System Software Release 1.3

*boot.tar*            Contains all the utilities for booting operating system from the diagnostic station as well as operating system diagnostic binaries and debugging tools.

*root.tar*            Contains all the commands that are installed on the root partition.

*usr.tar*             Contains all the commands that are installed on the */usr* partition.

*mach\_svr.tar*       Contains the microkernel, initialization, and server files.

*doc.tar.Z*           Contains online manual pages and PostScript copies of manuals for the operating system.

- Paragon™ System ParAide Software Development Environment Tools Release 1.3

*tools.tar.Z*         Contains the development tools for the Paragon™ System.

*crstools.tar*       Contains the cross-development tools for the Paragon™ System.

*tools.doc.tar*      Contains a PostScript copies of tools manuals.

- Paragon™ System SAT Software Release 1.3

*sat.tar.Z*            Contains the System Acceptance Tests source and binaries.

*sat.doc.tar*         Contains a PostScript copy of the *Paragon™ System Acceptance Test User's Guide*.

## NOTE

These instructions assume that SCO UNIX and the SCO Open Desktop are installed on the diagnostic station. If this software is not installed, contact Intel SSD Customer Support.

## Obtaining the Installation Files from Tape

If you have the installation files on one or more tapes, copy the files on each tape to the diagnostic station. These files go into the `/u/tmp` directory.

Do not copy the files into `/tmp`. If you put the tar files into `/tmp`, you risk filling up your root partition. You may choose instead to copy the **tar** files to one of your local servers.

1. Change to the directory where the distribution files will be stored:

```
DS# cd /u/tmp
```

2. Insert the distribution tape into the cartridge tape drive. Several installation tapes are provided. These release notes describe the installation of software on the following tapes:

- *Paragon™ System Software Release 1.3* which contains the Paragon operating system.
- *Paragon™ ParAide Development Environment Tools Release 1.3* which contains the System Acceptance Tests.
- *Paragon™ SAT Software Release 1.3* which contains the System Acceptance Tests.

Compilers appear on additional tapes and their installation is described in their accompanying release notes. If you are also planning to install compilers at this time, you may want to read in those tapes as well. The **postboot** script described later in this chapter allows you to choose the installation of the native compilers; but you should consult the compiler release notes for complete information. The compiler release notes contain instructions on how to install the cross compilers.

3. Use the **tar** command to copy the files from the distribution tape. The device name of the cartridge tape drive on the diagnostic station is **rStp0**.

```
✓ DS# tar xvpf /dev/rStp0
```

After the files have been copied, remove the tape from the cartridge tape drive.

4. Remove the distribution tape from the cartridge tape drive.
5. Repeat steps 2 through 4 for as many distribution tapes as you want to read in.

## Installing Boot Files on the Diagnostic Station

The following instructions assume that you are logged in as *root* to the diagnostic station.

1. If you copied the distribution files from a tape into the */u/tmp* directory on the diagnostic station, proceed to the next step.

If the distribution files are stored on some other server, you need to use **ftp** to copy the *boot.tar* file to */u/tmp*. This file requires about 8M bytes of storage. When you invoke **ftp**, it will prompt you for a login name and password on the distribution server.

```
DS# cd /u/tmp
DS# ftp IP Address of server with distribution files
Name: login_name
Password: password
ftp> cd path to distribution files
ftp> bin
ftp> get boot.tar
ftp> bye
```

2. Change to the */* directory, and use **tar** to extract the files from *boot.tar*:

### CAUTION

Ensure that no one is using any of the utilities from */usr/local/bin* when you untar *boot.tar*. If someone is using a utility when you untar *boot.tar*, that utility will not be updated.

### CAUTION

Many sites backup boot software by moving directories to another name. If you do this, untarring *boot.tar* will create directories. Unless you set your *umask* appropriately, the permissions on these directories will be wrong, and the Paragon system will not boot.

```
DS# cd /
DS# umask 022
✓ DS# tar xvfp /u/tmp/boot.tar
```

## CAUTION

Before running **buildscan**, install the diagnostic software. Refer to the *Paragon™ System Diagnostic DIAG2.0 Release Notes* for information about how to install diagnostic software. This software contains the scan driver that **buildscan** will build. Note that if the software is distributed via **ftp** rather than tape, the diagnostic software is in the file *diag.tar*.

3. If you are upgrading from Release 1.2, you must relink the kernel on the diagnostic station. To relink the kernel, issue the following commands:

```
✓ DS# cd /etc/conf/pack.d/scan
  DS# ./buildscan
```

This command takes a few minutes to rebuild the UNIX Operating System. Answer **y** to the questions. The first question asks if you want the new kernel to boot by default. The second question asks if you want the kernel environment rebuilt. Then, when the DS# prompt returns, reboot.

```
✓ DS# init 6
```

- If you are working directly at the diagnostic station, wait for the : (colon) prompt, and then press the **<Enter>** key.
- If you are remotely logged into the diagnostic station, the **init 6** closes the connection and you will need to **rlogin** again after waiting a few minutes. You can use **ping** to detect if the diagnostic station is ready to be logged into.

When the login prompt appears, log in as *root*.

4. Ensure that the following variables are defined correctly in */etc/default/tcp*:

```
✓ DOMAIN=
  NETMASK=
  BROADCAST=
```

5. Verify that the */etc/hosts* file contains the two aliases: *PARA\_ALIAS* and *DIAG\_ALIAS*. This modification needs to be performed only once during the first installation.
  - The *PARA\_ALIAS* alias identifies the IP address of the Paragon system.
  - The *DIAG\_ALIAS* alias identifies the IP address of the machine where the kernel, ramdisk, and *bootmagic* files exist. This machine is usually the diagnostic station.

Here is an example of the two lines as they might appear in */etc/hosts*.

```

xxx.yy.zzz.a ds ds.abc.def.com DIAG_ALIAS #diagnostic station
xxx.yy.zzz.ab paragon          PARA_ALIAS #Paragon system

```

6. If the diagnostic station's and the Paragon system's IP address and system network name are not what you want, change the diagnostic station's IP address and system network name in the file */etc/hosts* to the values for your installation. You should have values for the diagnostic station, the Paragon system, and loopback.

- A. Issue the **uname** command, specifying the new system name.

```
DS# uname -S <new system name>
```

- B. Change to the directory */etc/conf/cf.d* and run **configure**.

```
DS# cd /etc/conf/cf.d
DS# ./configure
```

- C. A menu appears. Choose System Name and type in the new system name.

- D. Link in a new kernel. This step is only necessary if you changed the system name. Your current directory is still */etc/conf/cf.d*.

```
DS# ./link_unix
```

- E. Reboot the diagnostic station. Answer **y** to all questions. There's a silent wait here of a few minutes. Press any key when requested.

```
DS# init 6
```

There still remains some network configuration to be performed on the Paragon system after the operating system software is installed. Refer to "Configuring the Paragon™ System for the Network" on page 2-22 for this information.

## Installing the Operating System Software

1. Ensure that the DEVCONF.TXT file in */usr/paragon/boot* is present and correct. Refer to the *Paragon™ System Commands Reference Manual* for information about how to set up this file.
2. Reset the Paragon using the **reset** command with the **autocfg** option. The **reset** script is located in */usr/paragon/boot*. For information about the **reset** command, refer to the *Paragon™ System Commands Reference Manual*.

When executed with the **autocfg** option, the **reset** script creates the file *SYSCONFIG.TXT* in the current directory and *SYSCONFIG.BIN* in the diagnostic directory */u/paragon/diag*. A previous version of *SYSCONFIG.TXT* is saved as *SYSCONFIG.OLD*.

```

✓ DS# cd /usr/paragon/boot
✓ DS# ./reset autocfg
.
.
.
---> Automatic SYSCONFIG.TXT generation.
How many cabinets are there? [1] number of cabinets
Generating /u/paragon/diag/hwconfig.txt...
Generating SYSCONFIG.TXT...
SYSCONFIG.TXT has been created.
Generating SYSCONFIG.BIN...
Using MAGIC.MASTER as the Magic file.
Creating bootmagic file...
Using 'fscan' as the console interface tool...
DS#

```

3. Reset the Paragon system using the **reset** script with the **ramdisk** option. This script resets the nodes, resets the mesh, waits for the Node Confidence Tests (NCT) to complete, downloads bootmagic strings, and initiates booting. By default, it uses the serial line with the **async** utility. To exit **async** and return to the DS# prompt, enter the key sequence ~. (or ~~, if you used **rlogin** to log into the diagnostic station). The key sequence ~q also works and does not require that you keep track of the number of remote logins.

The <ramdisk> prompt appears when the Paragon RAM disk has been loaded.

```

✓ DS# ./reset ramdisk
.
.
.
INT: SINGLE-USER MODE
<ramdisk>

```

4. Execute the **install** script.

```
<ramdisk> install
```

This script asks you to verify or provide the information necessary to complete the installation. The script gets default values from existing configuration files, as well as from information gathered during previous installations. You should be prepared to provide the following information:

Host name of the Paragon system  
IP address of the Paragon system  
Disk type of the Paragon system

Netmask address of the Paragon system  
Broadcast address of the Paragon system  
Gateway address of the Paragon system

Host name of the distribution system  
IP address of the distribution system  
Pathname to the directory containing the installation **tar** files on the distribution system  
Login name and password on the distribution system

5. The **install** script asks for the type and SCSI ID of the disk connected to the boot node on the Paragon system.

The SCSI ID is a number between 0 and 6 and identifies the address of the disk on the SCSI controller. This number is usually '0' for a Paragon system with more than two backplanes.

The script then lists the supported disk types. Press the **<Enter>** key to confirm the default value shown in square brackets (for example, [raid3]), or enter the correct value. For the disk type, you can enter the disk number (for example, '0' for raid3), or you can spell out the name of the disk type (for example, "raid3"). If you specify an invalid number or name, the script detects the inconsistency with the disklabel and prompts you again for a choice, listing acceptable choices.

Note that the script detects no difference between RAID3 and RAID5. If you have a RAID on your system, it will be set up as RAID3. If the drive is currently set up for RAID5, specify "raid3" when prompted for disk type, and it will be automatically converted from RAID5 to RAID3.

```

✓ SCSI ID for Boot Disk: [0]
=====
Paragon Operating System Installation
O.S. Install
=====

-----
✓ 0) raid3                4.70 Gigabyte Raid
1) maxtor76              0.76 Gigabyte Maxtor
2) maxtor1240 (non-raid) 1.24 Gigabyte Maxtor
3) 1gb (non-raid)       1    Gigabyte Drive
4) 4gbraid3             4    Gigabyte Raid Drive
5) 16gbraid3pfspg (raid) 16   Gigabyte Raid Drive
      -- OR --
      Type in the disktab entry name for your
      boot device. (i.e. maxtor1240)
-----
Disk Type: [raid3]

```

The script then displays the information you have entered, and asks you to confirm it. If the information is correct, press the <Enter> key, or enter 'Y' or 'y'. These disk information prompts repeat until you confirm that the correct information has been entered.

```

===== DISK INFORMATION SUMMARY =====
Disk Type is "raid3" (Disk Label will be "raid3")
SCSI ID 0

Is this correct? (y/n) [y]

```

If the disk is a RAID, the **install** script checks the configuration and performs the following actions:

- If the RAID controller is not an NCR, the script generates an error message and exits.
- If the status of the RAID controller is anything other than "Optimal" or "Degraded", or if the RAID level is anything other than '3', the script displays the following messages:

```

WARNING: The RAID drives are either configured as RAID5,
or are otherwise in need of initialization. This procedure
will destroy any data currently on the drive. Do you want
to continue? (y/n) [n]

```

To proceed with the installation, you must enter 'Y' or 'y' to both questions, and the drive is initialized. If you enter anything else, the **install** script exits.

6. The **install** script attempts to mount the root file system.

- A. If the root file system exists, the **install** script executes the **fsck** utility to clean it (if necessary). The file `/root/etc/defaults/install` is read to retrieve responses from a previous installation. If the file does not exist, reasonable defaults values are established. When the file is being read, the following message(s) is displayed:

```
Mounting /dev/rz0a...
Retrieving Previous Installation Responses...Done.
```

Even though the root file system exists, you may want to rebuild it to create a clean file system with no old files on it. The script displays the following prompt to ask if you want to rebuild the file system:

```
A root file system has been detected on /dev/rz0a.
It may not be necessary to rebuild it. If you do rebuild it,
all files in the existing root file system will be
destroyed
Do you want to rebuild it? (y/n) [n]
```

If you press the **<Enter>** key, or enter **'N'** or **'n'**, the installation proceeds without rebuilding the file system. If you enter **'Y'** or **'y'**, the script asks for confirmation:

```
Creating New Root File System. Okay? (y/n) [n]
```

At this point, you must enter **'Y'** or **'y'** to rebuild the file system and proceed with the installation. Any other response terminates the installation.

- B. If the root file system does not exist, it will be created automatically.

7. After the root file system has been mounted, the **install** script verifies the existence of the `/usr`, `/etc`, `/etc/defaults`, and `/pfs` directories, and creates them if they do not exist.

If the `/usr` and `/pfs` file systems exist, you may want to rebuild either or both of them to create a clean file system with no old files on it. For each of these file systems, the **install** script displays a message asking if you want to rebuild an existing file system. A typical message is as follows:

```
A pfs file system has been detected on /dev/rz0d.
It may not be necessary to rebuild it. If you do rebuild it,
all files in the existing pfs file system will be destroyed
Do you want to rebuild it? (y/n) [n]
```

If you press the **<Enter>** key, or enter **'N'** or **'n'**, the installation proceeds without rebuilding the file system. If you enter **'Y'** or **'y'**, the script rebuilds the file system and proceeds with the installation.

8. Next, the **install** script asks you to verify the information needed to get the distribution files. If a */etc/defaults/install* file was saved from a previous installation, the **install** script summarizes the file's information and asks if it is correct. If */etc/defaults/install* does not exist or has information missing, the script asks the needed questions. The following listing shows an example of the summary:

```

===== RESPONSE SUMMARY =====
Node Name:          system
IP Address:         00.00.00.00
Gateway:           133.26.1.1
Netmask:           255.255.255.00
Broadcast:         133.26.1.255
Distrib Node:      para_ds
Distrib IP Addr:   133.26.15.5
Distrib Path:      /u/tmp
Create /home:     Y
-----
Is the above information correct? (y/n)[n]

```

If you enter 'Y' or 'y', the installation proceeds using the default values. If you enter anything else, or if the */etc/defaults/install* file was not found, the script displays a series of prompts for you to enter the information. The default value for each item is displayed in square brackets. Press the <Enter> key to select the default, or enter a new value. In the following example, the installer entered new values for "System Name" and "System IP Address", and accepted the default values for the other items.

```

System Name [system]: paragon512
System IP Address [00.00.00.00]: 133.26.101.31
Gateway [133.26.1.1]:
Netmask [255.255.255.00]:
Broadcast Address [133.26.1.255]:
Distribution Node Name [para_ds]:
Distribution IP Address [133.26.15.5]:
Distribution Path [/u/tmp]:
Create /home File System [Y]: N

```

At the end of the questions, the script displays the new information in the response summary and asks again for confirmation. This sequence is repeated until you indicate that the response summary is correct.

9. If you chose the create a new */home* partition, the following question appears:

```
All files in /home will be destroyed. Continue?[n]
```

If you type *n*, the new */home* partition is not recreated and the installation continues. Type *y* if you want to create a new */home* partition. After typing *y*, the following message appears:

```
Creating a new home file system...
```

If the current */home* partition is mounted, it is unmounted, and the following message appears:

```
Unmounting /dev/rz0f...
newfs -d 0 -b 8192 -f 2048 /dev/rrz0f
```

10. The **install** script now sets up the network using the values you just provided. It then asks for a login name for **ftp** to use when getting files from the distribution system. The distribution system is the system from which you obtain the installation **tar** files. This may be the diagnostic station or one of your local servers.

Username for FTP'ing files from *Distribution Node Name*:

✓ Enter a valid user name. The script then invokes **ftp** and attempts to use the name to log into the distribution node. When **ftp** prompts you, enter the user name's password. The following files are retrieved from the distribution system:

↓

```
root.tar
mach_svr.tar
usr.tar
hosts
```

If the login is unsuccessful, the script displays an error message along with a request to verify the distribution information.

```
ERROR: [file] was not retrieved.
```

```
===== DISTRIBUTION INFORMATION =====
```

```
Distrib Node:   para_ds
Distrib IP Addr: 133.26.15.5
Distrib Path:   /u/tmp
```

```
-----
Is the above information correct? (y/n) [y]
```

Entering '*n*' gives you the chance to change this information. If the information is correct, enter '*y*' and the script repeats the process of asking for a login name and password, and using **ftp** to get the distribution files.

11. The remainder of the Paragon system installation is automatic, requires about 30 minutes to complete, and performs the following:

- Obtains the installation **tar** files from the distribution system.

- Preserves copies of */etc/hosts*, */etc/passwd*, */etc/exports*, */etc/group*, */etc/fstab*, */etc/printcap*, */etc/pfstab*, */etc/devtab*, */etc/services*, */etc/resolv.conf*, */etc/ntp.conf*, */etc/shells*, */var/adm/sendmail/sendmail.cf*, */var/adm/sendmail/sendmail.cw*, and */var/adm/sendmail/sendmail.st* if they exist.
  - Restores the **tar** files that were retrieved from the distribution node.
  - Creates the default */etc/fstab*, */etc/devtab*, and */etc/rc.config* files.
  - Restores copies of */etc/hosts*, */etc/passwd*, and */etc/exports*, */etc/group*, */etc/fstab*, */etc/printcap*, */etc/pfstab*, and */etc/devtab* if they exist.
  - Makes PFS directories.
12. Depending on the type of your boot disk, the **install** script creates a 32M-byte or 64M-byte paging file for the default pager on the boot node. You will get better performance but use up more disk space with a larger paging file. The script now displays the defaults for your system and asks if you would like to create a larger paging file. It puts this larger file on the home partition rather than the root partition. The following would be displayed for a RAID3.

```

Default paging file size is 64M
Bigger paging file size is 512M
Would you like to make a bigger paging file in home
partition?(y/n) [y]y
Creating a paging file. It may take up to 15 minutes ...
create_pf 512M /home/paging_file
create_pf: Creating 'paging_file' of size (512 Meg) 524288
blocks.
```

If you do not increase the size of the paging file now, you can still increase it after installation. Refer to the *Paragon™ System Administrator's Guide*.

When the `<ramdisk>` prompt returns, the installation is complete.

13. When the `<ramdisk>` prompt appears, return to the diagnostic station prompt and use the **reset** command to reboot the Paragon system. Disconnect from the Paragon RAM disk by typing `~.` (tilde period) or `~~.` if you are logged in remotely to the diagnostic station. The key sequence `~q` also works and does not require that you keep track of the number of remote logins. Then, invoke the **reset** command.

```

<ramdisk> ~.
Exiting ...
DS#
DS# ./reset
.
.
.
# <Ctrl-D>
```

When the **reset** command has completed, the prompt that appears is on the Paragon system. Enter multiuser mode by pressing **<Ctrl-D>**. Note that the **reset** script automatically enters multiuser mode if the *MAGIC.MASTER* file contains `RB_MULTIUSER=1`. The root file system is checked and mounted and then the mesh is booted.

This step automatically brings up the remaining nodes on the mesh, using the */sbin/bootmesh.sh* script, which is called by */sbin/bcheckrc*. The mesh will be booted just after the root file system is checked and mounted. Any nodes which fail to boot will be reported to the console; you can also look in the file */etc/bootmesh.log* for the list of failed nodes.

If for some reason you need to boot with mesh booting disabled, add the string

```
DISABLE_BOOTMESH=1
```

to */usr/paragon/boot/MAGIC.MASTER* on the diagnostic station. If you have booted with this string, you can later boot “by hand” by creating the Paragon file */etc/forcebootmesh*, and executing */sbin/bootmesh.sh*.

When the system completes booting to multiuser mode, you can login again. If you have increased the size of the paging file and successfully booted to multiuser mode, you may delete the original paging file, */mach\_servers/paging\_file.orig*.

## Executing the *postboot* Installation Script

Run **postboot** for the final configuration. **postboot** updates the message of the day, sets the date/time, and optionally configures the network time daemon, prompts for compiler/documentation and tools installation, and creates the terminfo databases, spell dictionaries, cat directories, and lint libraries.

If you intend to use **postboot** to install the compilers, you must have already read in the installation files from the compiler tapes. Refer to "Obtaining the Installation Files from Tape" on page 2-3 for instructions on how to read the distribution tape.

To execute the **postboot** script, type the following:

```
# cd /
# ./postboot
```

The script asks you to confirm the current date and time, and then prompts you before performing each of several tasks.

```
=====
```

### Postboot Process

This script will perform final configuration of your Paragon system.

```
Is this correct date/time: Sun Apr 27 10:30:36 PST 1947 (y/n)? [y]
n
```

```
Enter date as yymmddhhmm
9404270913
Wed Apr 27 09:13:00 PDT 1994
```

The remaining tasks this script will perform are:

- Configure Network Time Daemon
- Install compilers and/or documentation
- Install basic math libraries
- Install Paragon tools
- Install Paragon system acceptance tests
- Create the terminfo databases
- Create the spell dictionaries
- Create the cat directories
- Create the lint libraries

Note: You must install the C compiler before the lint libraries can be created.

Before each task is performed you will have the chance to choose if you want the task performed.

-----  
 The first task is to specify a network time daemon. Be prepared to provide the name of the computer that maintains your network-wide clock.

Configure Network Time Daemon ...

Do you wish to proceed with configuring network time daemon (y/n)?  
 [y]

*n*

If you want to configuring the network time daemon, enter *y*. Note that the network time daemon adversely impacts performance because it sends messages to all the nodes in the system every five seconds. If you choose *y*, you get the message,

What is name of your network time server?  
*name of network time daemon*

A warning appears if you have already defined a network time daemon. If a warning appears and you want to change the definition, confirm that you want the daemon configured.

Setting up network configuration file '/etc/ntp.conf'  
 Starting the network time server ...  
 Network Time Service started

-----  
 Installing compilers and/or documentation...

Do you wish to proceed with installing compilers and/or  
 documentation (y/n)? [y]

*y*

If you want to skip installing the native compilers, enter *n*. To install the compilers, you must have previously read in the compiler installation *tar* files onto the distribution node. If you answer *y*, the following menu is displayed.

```

===== RESPONSE SUMMARY =====
Install C compiler and its docs:          Y
Install C++ compiler and its docs:       N
Install FORTRAN compiler and its docs:   Y
Install system software documentation:   Y
Distribution Node:                       Distribution Node Name
Distribution IP Addr:                    Distribution IP Addr
Distribution Path:                       Distribution Path

```

-----  
 Is the above information correct? (y/n) [n] *y*

Answer *n* to change the defaults.

Installing basic math libraries...

Do you wish to proceed with installing basic math libraries (y/n)?  
[y]**y**

If you want to skip installing the basic math libraries, enter *n*. To install the basic math libraries, you must have previously read in the *libkmath.a*, *libkmath\_r.a*, and *bml.doc.tar.Z* files onto the distribution node. If you answer *y*, the following menu is displayed.

```
===== RESPONSE SUMMARY =====
Distribution Node:                Distribution Node Name
Distribution IP Addr:            Distribution IP Addr
Distribution Path:               Distribution Path
```

-----  
Is the above information correct? (y/n) [n] **y**

Answer *n* to change the defaults.

Installing Paragon tools...

Do you wish to proceed with installing Paragon tools (y/n)? [y]**y**

If you want to skip installing the tools, enter *n*. To install the tools, you must have previously read in the tools installation *tar* files onto the distribution node. The standard Paragon system tools are on the tape labeled *Paragon™ ParAide Development Environment Tools Release 1.3*. The Motif tools are on a separate release tape; refer to the *Paragon™ System Motif Graphics Library Release 1.2.4 Release Notes*. The DGL and OpenDGL tools are also on a separate release tape; refer to the *Paragon™ System Silicon Graphics Libraries Release 2.0 Release Notes*.

If you answer *y*, the following menu is displayed

```
===== RESPONSE SUMMARY =====
Install standard Paragon tools:  Y   ✓
Install motif:                  Y   N
Install dgl:                    Y   N
Install opengl:                 Y   N
Distribution IP Addr:            Distribution IP Addr
Distribution Path:               Distribution Path
```

-----  
Is the above information correct? (y/n) [n] **y**

Answer *n* to change the defaults. If you choose to install the tools, appropriate online documentation is also installed.

Installing system acceptance tests...

Do you wish to proceed with installing system acceptance tests  
(y/n)? [y] **y**

===== RESPONSE SUMMARY =====

Distribution Node:	Distribution Node Name
Distribution IP Addr:	Distribution IP Addr
Distribution Path:	Distribution Path

-----  
Is the above information correct? (y/n/q) [n]

✓ Answer **n** to change the defaults.

Creating terminfo databases....

✓ Do you wish to proceed with creating terminfo databases (y/n)? [y] **y**  
Running terminfo translator....

Creating spell dictionaries....

✓ Do you wish to proceed with creating spell dictionaries (y/n)? [y] **y**

Creating cat directories....

Do you wish to proceed with creating the cat directories (y/n)? [y] **n**  
Cat directories will not be created.

↑ If you choose to create the *cat* directories, note that this choice also creates the *cat* files. This choice takes a long time to execute and uses a great deal of disk space.

The *cat* files are formatted versions of the online manual pages. Because it is quicker to access a formatted manual page than an unformatted one in */usr/share/man*, some users choose to store *cat* files. However, if the *cat* directories exist and are empty, the first access to an online manual page places a formatted version in the appropriate *cat* directory. Hence, making empty *cat* directories results in, after a while, a collection of formatted files for the most frequently accessed online manual pages. This is preferable to making formatted versions of all online manual pages.

If you choose to create the *cat* directories, you might see errors similar to the following, which can be ignored: unterminated list (no .LE) -- noticed by the .SH "RELATED".

✓ Creating lint libraries....

Do you wish to proceed with creating the lint libraries (y/n)? [y] **y**

**NOTE**

To install C and Fortran cross compilers, refer to the *Paragon™ C Compiler Release 5.0 Software Product Release Notes* and the *Paragon™ C Compiler Release 5.0 Software Product Release Notes*.

**NOTE**

After **postboot** has successfully run, consider deleting the installation **tar** files from the distribution system. They will only be needed if you want to reinstall the operating system.

If you have installed any online documentation, you must make the *whatis* database. This is the database used by the **man** command with the **-k** option. To make the *whatis* database, issue the following commands:

```
✓ # cd /usr/share/man
✓ # /usr/sbin/mkwhatis
```

✓ **mkwhatis** does not successfully make the complete *whatis* file. It returns an error message that says "argument too long." The same error message is returned by **catman -w**. The resulting *whatis* file is flawed but still usable.

To allow other users to log into the Paragon system, remove the file */etc/nologin*.

```
# rm /etc/nologin
```

After **postboot** has successfully run, remove any lingering patches. In the directory */etc/patches* on the Paragon system, remove any remaining operating system patches. The **?** in the following commands indicates the path level. The latest patch level is 7.

```
✓ # cd /etc/patches
# rm -r R1.2.?
# rm README_R1.2.?
# rm ParaBOM_R1.2.?
# rm remove_R1.2.?
```

Also remove the patch files from the diagnostic station.

```
✓ DS# cd /etc/patches
DS# rm -r R1.2.?
DS# rm README_R1.2.?
DS# rm DiagBOM_R1.2.?
DS# rm remove_R1.2.?
```

## The Allocator Configuration File

The allocator configuration file (called */etc/nx/a.allocator.config*) sets limitations on what partitions can be created and what characteristics they have. When you create the partitions under the limitations set by the allocator configuration file, you use the **mkpart** command with appropriate options.

**initpart** (run when the Paragon system enters multiuser mode) creates the default allocator configuration file. **initpart** also creates a new */etc/nx/partinfo* file if the current *.partinfo* has *xy* dimensions that are inconsistent with reality or the file does not exist. Finally, **initpart** makes the root partition with permissions 754 and sets the scheduling attribute to SPS, or space share. These permissions allow only *root* to create subpartitions. So when you are installing the system you need to take into account who you want to make partitions and where they can make them.

The allocator configuration file sets the user model to allow for one gang-scheduled partition with a maximum of two NX applications per node and a minimum rollin quantum of 1 hour. The default */etc/nx/allocator.config* file looks as follows:

```
NUM_GANG_PARTS=1
DEGREE_OF_OVERLAP=2
MIN_RQ_ALLOWED=1h
```

You can disable any of these values by removing the appropriate variable from */etc/nx/allocator.config*.

If you remove the file, it will be created with defaults at the next boot. If you want to remove constraints on the user model, comment out the appropriate lines. Indicate a comment by putting a **#** in the first position of the line.

### NOTE

You should not use the **-tile** and **-MACS** options on the *allocator* line in */sbin/init.d/allocator*. Use the *allocator.config* file instead. Refer to the *allocator.config* manual page in the *Paragon™ System Commands Reference Manual*.

To have any changes to the configuration file take effect you will need to stop and restart the allocator after the changes are made. To stop and start the allocator, issue the **allocator** command.

```
# /sbin/init.d/allocator stop
# /sbin/init.d/allocator start
```

## Creating the *.service* and *.compute* Partitions

The following example shows how to create the *service* and *compute* partitions. The example assumes a 64 node system with the boot node at node 7 and additional MIO nodes at nodes 15 and 23. These are OS node numbers.

### CAUTION

The *.service* partition must contain the boot node.

Create the *service* partition. Use the *-ss* switch, so the service partition has standard scheduling. The following example allocates nodes 7, 15, and 23 to the service partition.

```
# mkpart -nd 7,15,23 -ss .service
```

Create the *compute* partition using the *-sz* switch to allocate the remaining 61 nodes to the compute partition and the *-mod* switch to set permissions that enable all users to create subpartitions in the compute partition. Note that you may want to change the permissions of the compute partition when you personalize your configuration.

```
# mkpart -sz 61 -mod 777 .compute
```

After creating the compute partition, issue the **showpart** command to display the nodes that are allocated to each partition. Verify that the partitions do not overlap (that is, each node is allocated to only one partition). For example:

```
# showpart .compute  
# showpart .service
```

If the partitions do overlap, use the **rmpart** command to remove the overlapping partitions. When you use **mkpart** to create the partitions again, you may need to use the *-nd* switch to specify the node numbers of the nodes to include in the *compute* partition. For example:

```
# rmpart .compute  
# mkpart -nd 0..5,8..14,16..22,24..63 -mod 777 .compute
```

Notice that this example did not specify any scheduling characteristics. If you do not specify a scheduling characteristic when making a partition, then the characteristics are inherited from the parent. In the case of the *.compute* partition, the default is space-shared and no gang-scheduling. To explicitly set the *.compute* partition to be gang-scheduled, you must specify a rollin quantum value with the *-rq* switch. (0 is valid, and means an application will run to completion). For example,

```
# mkpart -sz 61 -rq 0 -mod 777 .compute
```

## Turning off Autoreboot

The automatic reboot feature is enabled by default. When this feature is enabled, the Paragon system reboots whenever it stops functioning normally or is intentionally shutdown.

Turn off the autoreboot feature before performing hardware maintenance. To turn off autoreboot, log into the diagnostic station. In the file */usr/paragon/boot/fscan.cfg*, change the line

```
set autoreboot on  
  
to  
  
set autoreboot off
```

## Specifying Your Time Zone

✓ If your Paragon system arrived with the software already installed, you might need to modify the */etc/TIMEZONE* and */etc/rc.config* files on the Paragon system to reflect your time zone. You may also need to modify these files if you have reinstalled the Paragon system software.

## Configuring the Paragon™ System for the Network

If your Paragon system arrived with the software already installed, you must change the network name and IP address of the diagnostic station and the Paragon system to bring them up on the network at your site. If you are reinstalling or upgrading the system software, you should only need to reconfigure the Paragon system. Refer to “Installing Boot Files on the Diagnostic Station” on page 2-4 for information about configuring the diagnostic station for the network.

If your Paragon system has any HIPPI boards installed, you must configure their network interface when you first receive the system, and also whenever the system software is installed or upgraded. HIPPI configuration includes activating the network interface and specifying the HIPPI routing tables. For a complete description of this procedure, see the *Paragon™ XP/S High-Performance Parallel Interface Manual*.

- ✓
1. Edit the */etc/rc.config* file, and change HOSTNAME and HOSTID (IP address) to the new values.
  2. Verify that the */etc/hosts* file has entries for the Paragon system network ID (IP number) and diagnostic station network ID.
  3. If you are using the domain nameserver, edit */etc/resolv.conf*. Set the value for the domain and add nameservers if needed.

## Resetting the System After Installation

Resetting the Paragon system, once it is up and running, is performed as follows:

1. Log in or **rlogin** to the diagnostic station as *root*.
2. Connect to the Paragon system with the **console** command. The **console** command is a script that the **reset** script created in */usr/paragon/boot* the last time **reset** was run. **console** uses whatever the console connection was at the time and supports the **async**, **fscan**, and the **scanio** console interfaces. You may have to press a carriage return after issuing **console** to get a prompt.

```
DS# cd /usr/paragon/boot
DS# ./console
```

```
login:
```

3. Login as *root* to the Paragon system, unmount all file systems, and use the **halt** command to shut down the Paragon system. Then, return to the diagnostic station prompt by typing **~.** (or **~~.** if you are logged in remotely to the diagnostic station). The key sequence **~q** also works and does not require that you keep track of the number of remote logins. Then, use the **reset** script to reboot the system.

```
# cd /
# sync;sync;sync
# shutdown now
# umount -A
# halt
# ~.
DS# ./reset
# <Ctrl-D>
```

When the **reset** command has completed and the prompt returns, enter multiuser mode by pressing **<Ctrl-D>**. Note that the **reset** script automatically enters multiuser mode if the *MAGIC.MASTER* file contains *RB\_MULTUSER=1*.

When you boot the system, the **fsck** command is automatically run on each file systems defined in */etc/fstab* (such as */home*, */pfs* and the PFS stripe directories typically mounted on */home/.sdirs/vol\**).

In the case of a severely damaged file system, the **fsck** will fail and a message will be displayed to the screen. The message will scroll off the screen before the reboot is complete and you may not notice it. The system will continue to boot to a multi-user state with no other indication that anything is wrong.

If the **fsck** fails, the affected file system will not be mounted. Any files written to the unmounted file system are written into the file system containing the desired mount point instead. Later, if you notice that a file system did not mount and you **fsck** and mount it, the files that were written into the root file system become hidden. To avoid this problem, after each boot you should manually compare the mounted file systems with the contents of the *fstab* file. Make sure that the PFS stripe directories are mounted or the **fsck** data will go into the */home* file system.

For example, assume that **fsck** failed when processing the PFS stripe directories and that these stripe directories are mounted on the partition */dev/io1/rrz0c*. To manually run **fsck** on this partition, enter

```
# fsck -y /dev/io1/rrz0c
```

The **-y** directs **fsck** to answer yes to all its questions.

## User Notification

✓ Consider adding the following messages to */etc/motd*:

```
Release 1.3 of the Paragon system software has
been installed on this system. All user
applications must be recompiled and relinked
before they can be run.
```

```
PostScript copies of the release notes for
Release 1.3 of the Paragon system software can
be found in the directory
/usr/share/release_notes. This directory also
contains ASCII and PostScript copies of the
current list of system software bugs.
```

## Configuring an Additional Ethernet Board

Every Paragon system has at least two Ethernet connections, one to the diagnostic station and one to the boot node. Follow these instructions if you have another Ethernet connection to an I/O node other than the boot node. This makes for a total of three Ethernet connections, but because this is a second Ethernet connection to the Paragon nodes, these instructions are often referred to as those for configuring a second Ethernet.

1. Log in or **rlogin** to the diagnostic station as *root*.
2. Change directory to */usr/paragon/boot*. Edit *DEVCONF.TXT* to contain a line identifying the position of the second Ethernet connection.

For example, consider a two-cabinet configuration with one Ethernet connection. This connection is on the boot node which is in cabinet 0, backplane D and slot 3. The CBS number of the boot node is 00D03. Now assume that you add a second Ethernet connection to a new I/O node in cabinet 1, backplane A, and slot 12. Its CBS number is 01A12.

A typical *DEVCONF.TXT* file is shown below. The lines in **bold** are those added for the new I/O node with the Ethernet connection.

```

DEVICES
ENET 00D03
ENET 01A12
RAID 00D03 ID 0 SW 3.02 LV 3 DC 5 SID 0 RAID3
TAPE 00D03 ID 6 DAT
MIO 00D03 H04
MIO 01A12 H04
END_DEVICES

```

3. Connect to the Paragon system with the **console** command.

```

DS# cd /usr/paragon/boot
DS# ./console

```

```

.
.
.
login:

```

4. Edit the */etc/hosts* file to include the new interface, using the additional IP address and additional host name.

5. Edit the file */etc/rc.config*. Define *HOSTNAME2*, *HOSTID2* and *NETDEV2* and export their values.

```

HOSTNAME=joe
HOSTNAME2=frank
HOSTID=xxx.yy.zz.aaa
HOSTID2=xxx.yy.zz.bbb
NETMASK=255.255.255.00
NETDEV="<7>em0"
NETDEV2="<96>em0"
.
.
.
export DISPLAYTYPE HOSTNAME HOSTNAME2 HOSTID HOSTID2 NETMASK
NETDEV NETDEV2 TZ

```

The NETDEV variable identifies the node number of the Ethernet connection. Each Ethernet connection has its own HOSTNAME and HOSTID. It is not necessary to define a NETMASK2 if the second Ethernet connection is on the same class as the primary and uses the same mask value.

Login as *root* to the Paragon system and edit the file */sbin/init.d/inet*. The lines in **bold** indicate the lines that must be added to configure a second Ethernet board. Both are **ifconfig** lines. Note that the **ifconfig** line for the second Ethernet board must appear above the line for loopback: */sbin/ifconfig lo0 127.0.0.1*. This is true for all network devices, including a HIPPI controller.

```

if [ "$ARGTWO" = "force" -o "$9" = "S" ]; then
echo "Configuring network"
echo "hostname: \c"
/sbin/hostname $HOSTNAME
/sbin/hostid $HOSTID
echo "hostid: $HOSTID"
/sbin/ifconfig $NETDEV $HOSTID netmask $NETMASK -trailers
# ifconfig second ethernet
/sbin/ifconfig $NETDEV2 $HOSTID2 netmask $NETMASK -trailers
/sbin/ifconfig lo0 127.0.0.1
#
# Add default routes; start routed
#
#
#           echo "Adding route for osfgw: "
#           SUBNET=`expr $HOSTID : "\.*\.*\.*<.*>.*" `
#           /sbin/route add default 137.27.$SUBNET.1
#           /sbin/route add default $GATEWAY
fi
;;
'stop')

```

```

/sbin/ifconfig $NETDEV2 down
/sbin/ifconfig $NETDEV down
/sbin/ifconfig lo0 down

```

6. Unmount all file systems listed in */etc/fstab*, and use the **halt** command to shut down the Paragon system. Then, return to the diagnostic station prompt by typing **~**. (or **~~**, if you are logged in remotely to the diagnostic station). The key sequence **~q** also works and does not require that you keep track of the number of remote logins. Then, issue **reset** with the **autocfg** option. This ensures that the new information in *DEVCONF.TXT* is added to *SYSCONFIG.TXT*.

```

# cd /
# sync; sync; sync
# shutdown now
# umount -A
# halt
# ~.
DS# ./reset autocfg
DS#

```

7. Reset the Paragon system and enter multiuser mode. When the **reset** command has completed and the prompt returns, enter multiuser mode by **<Ctrl-D>**. Note that the **reset** script automatically enters multiuser mode if the *MAGIC.MASTER* file contains **RB\_MULTIUSER=1**.

```

DS# ./reset
.
.
.
# <Ctrl-D>

```

## Configuring the System for HIPPI

Refer to the *Paragon™ XP/S High-Performance Parallel Interface Manual* for instructions on how to configure your system for a HIPPI controller.

## Installing and Configuring the PFS

This section provides instructions for configuring a Parallel File System (PFS). The operating system installation process configures and mounts a default PFS file system as specified in the default versions of the */etc/fstab* and */etc/pfstab* files.

### The Default */etc/fstab* File

The */etc/fstab* file contains entries for file systems and disk partitions to be mounted at boot time. The */etc/pfstab* file contains definitions of the stripe groups. A stripe group consists of stripe directories where the actual PFS files reside.

The default configuration may not be the best configuration for your system. The default configuration is provided as a template that you can modify for your particular configuration.

### NOTE

Running with the PFS striped across the boot node is not recommended, unless, of course, you have only one I/O node and it is the boot node.

The default configuration assumes that you have one I/O node. It mounts */*, */usr*, and */home* as UFS partitions. It mounts */pfs* as the PFS partition with stripe group *one*. The mounts are shown in */etc/fstab*, and the stripe groups are defined in */etc/pfstab*.

When looking at */etc/fstab*, note that *io0* has partitions *rz0a*, *rz0b*, *rz0c*, *rz0d*, *rz0e*, *rz0f*, and *rz0g*.

- The partitions *rz0b*, *rz0c*, and *rz0g* are not mounted. They are not mentioned or commented out in */etc/fstab*.
- The partitions *rz0a*, *rz0e*, and *rz0f* are UFS file systems. *rz0a* is the root partition */*; *rz0e* is the user partition */usr*; and *rz0f* is the home partition */home*. This information is embodied in the following lines from */etc/fstab*.

```

/dev/io0/rz0a    /                ufs rw 0 1
/dev/io0/rz0e   /usr            ufs rw 0 2
/dev/io0/rz0f   /home           ufs rw 0 3

```

- The partition *rz0d* is the mount point for the PFS file system, and its stripe group is one. This means that */etc/pfstab* must have the stripe group *one* defined. This information is embodied in the following line from */etc/pfstab*.

```
/dev/io0/rz0d /pfs pfs rw,stripegroup=one 0 3
```

In summary,

<i>rz0a</i>	UFS, mounted, root partition
<i>rz0b</i>	not mounted
<i>rz0c</i>	not mounted
<i>rz0d</i>	mount point for PFS
<i>rz0e</i>	UFS, mounted, user partition
<i>rz0f</i>	UFS, mounted, home partition

Here is the complete text of the default */etc/pfstab*. The # at the start of a line indicates a commented line.

```
# Local (required to boot mesh) filesystems
#
/dev/io0/rz0a / ufs rw 0 1
/dev/io0/rz0e /usr ufs rw 0 2
#
# Additional local filesystems
#
/dev/io0/rz0f /home ufs rw 0 3
#
# Remote filesystems
#/dev/io1/rz0c /home/.sdirs/vol0 ufs rw 0 4
#/dev/io2/rz0c /home/.sdirs/vol1 ufs rw 0 4
#/dev/io3/rz0c /home/.sdirs/vol2 ufs rw 0 4
#/dev/io4/rz0c /home/.sdirs/vol3 ufs rw 0 4
#
# Parallel filesystems
#
/dev/io0/rz0d /pfs pfs rw,stripegroup=one 0 5
#
# NFS filesystems
#
```

## The Default */etc/pfstab* File

When looking at the default */etc/pfstab*, notice there are nine stripe groups defined. The stripe groups *eight* and *all* are the same. Please note that the names *one* through *eight* are only examples. You can use any group names you like.

Note first that */home* is mounted on partition *rz0f*. The stripe group *one* consists only of the directory */home/.sdirs/vol0* because *one* is only on that line. The stripe group *two* consists of the directories (*/home/.sdirs/vol0* and */home/.sdirs/vol1*).

Here is the default */etc/pfstab*.

```

/home/.sdirs/vol0 all one two three four five six seven eight
/home/.sdirs/vol1 all      two three four five six seven eight
/home/.sdirs/vol2 all      three four five six seven eight
/home/.sdirs/vol3 all      four five six seven eight
/home/.sdirs/vol4 all      five six seven eight
/home/.sdirs/vol5 all      six seven eight
/home/.sdirs/vol6 all      seven eight
/home/.sdirs/vol7 all      eight

```

Even with one I/O node, it may make sense to have two stripe directories. This is because a stripe directory is in a UFS file system and cannot contain files larger than 2G-1 bytes. For PFS files to be larger than 2G-1 bytes, they must be striped across more than one stripe directory, and each stripe directory must be a separate file system. Then, the portion of a PFS file in a stripe directory is always less than 2G-1 bytes while the entire file may be larger.

## Additional I/O Nodes

When you configure the PFS, you may choose to have different I/O nodes used by the PFS and the paging tree. To specify that an I/O node is *not* to be used by the paging tree, add the **NOPAGER** token to the RAID line in *DEVCONF.TXT*. Then, perform a **reset autcfg**. For example, the following line shows how to identify the I/O with CBS number 00D01 as an I/O node that will not be used for paging.

```
RAID 00D01 ID 0 SW 3.02 LV 3 DC 5 SID 0 RAID3 NOPAGER
```

To configure additional I/O nodes for PFS,

1. Follow the instructions in the previous section under “Installing and Configuring the PFS” on page 2-28 to configure file systems on I/O nodes other than the boot node.

The partitioning of the RAID subsystem is determined by the disk label. Choose either the non-boot *\*pfs* label or the non-boot *\*pfspg* label.

2. Uncomment or add the appropriate entries in */etc/fstab*. For example, if four I/O nodes are added for PFS file striping, the */etc/fstab* entries might be changed to the following:

```

# Local (required to boot mesh) filesystems
#
/dev/io0/rz0a  /                ufs rw 0 1
/dev/io0/rz0e  /usr              ufs rw 0 2
#
# Additional local filesystems
#
/dev/io0/rz0f  /home            ufs rw 0 3
#
# Remote filesystems
/dev/io1/rz0c  /home/.sdirs/vol0      ufs rw 0 4
/dev/io2/rz0c  /home/.sdirs/vol1    ufs rw 0 4
/dev/io3/rz0c  /home/.sdirs/vol2    ufs rw 0 4
/dev/io4/rz0c  /home/.sdirs/vol3    ufs rw 0 4
#
# Parallel filesystems
#
/dev/io0/rz0d  /pfs                pfs rw,stripegroup=four 0 5
#
# NFS filesystems

```

The difference from the default */etc/fstab* is that the remote file systems are uncommented and the stripe group has been changed to four.

The new configuration has the following consequences:

- The stripe group of the PFS mount has been changed from *one* to *four*. Files created in */pfs* will be striped across stripe group *four*, which is defined in */etc/pfstab* to consist of the directories */home/.sdirs/vol0*, */home/.sdirs/vol1*, */home/.sdirs/vol2*, and */home/.sdirs/vol3*. Because each of the stripe directories is the mount point of a UFS file system on a different I/O node, concurrent striping to four I/O nodes is achieved.
- PFS files are not striped across the boot node, *io0*.
- 2G-1 bytes of each RAID subsystem are used. This is because the stripe directory is mounted on a UFS file system, which has a limit of 2G-1 bytes. This means that maximum use of the data storage capability of the RAID subsystems for PFS use is not attained. However, concurrency is maximized.

If you decide to have two stripe directories per RAID subsystem, the remote mounts in */etc/fstab* might look as follows:

```
#
# Remote filesystems
#
/dev/io1/rz0c /home/.sdirs/vol10 ufs rw 0 4
/dev/io2/rz0c /home/.sdirs/vol11 ufs rw 0 4
/dev/io3/rz0c /home/.sdirs/vol12 ufs rw 0 4
/dev/io4/rz0c /home/.sdirs/vol13 ufs rw 0 4
#
/dev/io1/rz0d /home/.sdirs/vol14 ufs rw 0 4
/dev/io2/rz0d /home/.sdirs/vol15 ufs rw 0 4
/dev/io3/rz0d /home/.sdirs/vol16 ufs rw 0 4
/dev/io4/rz0d /home/.sdirs/vol17 ufs rw 0 4
#
# Parallel filesystems
#
/dev/io0/rz0d /pfs pfs rw,stripegroup=eight 0 5
```

With Release 1.3, a file write randomly chooses the stripe directory it starts with and then is sequential and cyclic. For example, assume that you are doing a 128K-byte write. The stripe unit by default is 64K. Assume that the write randomly begins on */home/.sdirs/vol7*. The first 64K bytes go to *io4*, and the second 64K bytes go to *io1* (because the access is cyclic).

The stripe directories must be owned by *root* and have write/execute permissions for *root*, write/execute permissions for *group* and *other*, and have their sticky bits set. The PFS mount point must have read/write/execute permissions for everyone and have its sticky bit set.

The file systems must be mounted before you can change their ownership or permissions. To mount the file systems listed in */etc/fstab*, issue the following command:

```
# mount -a
```

The commands to set the ownership and permissions are as follows (the leading 1 sets the sticky bit):

```
# chown root stripe_directories
# chmod 1333 stripe_directories
```

The command to do this is as follows:

```
# chmod 1777 mount_point
```

## Installing the Paragon™ System Acceptance Tests

✓ These instructions assume that you have copied the compressed **tar** files from the distribution tape to the diagnostic station or to some other server.

1. Log in to the Paragon system as *root*.
2. Establish an **ftp** connection with the server containing the files, *sat.tar.Z* and *sat.doc.tar.Z*.

```
# cd /tmp
# ftp IP address of server with distribution files
Name: login_name
Password: password
ftp> cd path to distribution files
ftp> bin
ftp> get sat.tar.Z
ftp> get sat.doc.tar
ftp> bye
```

3. Use the following steps to uncompress and untar the distribution files into the / directory.

```
# uncompress sat.tar.Z
# cd /
# tar xf /tmp/sat.tar
# rm /tmp/sat.tar
# cd /usr/share
# tar xf /tmp/sat.doc.tar
# rm /tmp/sat.doc.tar
```

If you are having disk space problems, use **zcat** to uncompress and extract. This uses less space, but takes longer:

```
# cd /
# zcat /tmp/sat.tar.Z | tar xvf -
# rm /tmp/sat.tar.Z
# cd /usr/share
# tar xf /tmp/sat.doc.tar
# rm /tmp/sat.doc.tar
```

Before running the SATs, you must configure a Parallel File System (PFS). Refer to "Installing and Configuring the PFS" on page 2-28 for instructions on configuring a PFS.

## Updating the MACS Database

This version of MACS uses a database format that is different from the format used by earlier versions of MACS. If your site has been running MACS on earlier versions of the Paragon system, you will need to either delete the old database and rebuild it, or simply update the MACS database using the **dbconvert** command. The **dbconvert** command has the following syntax:

```
# dbconvert oldfile newfile
```

The **dbconvert** command reads the old MACS database specified by *oldfile* and writes the same data in the new format to *newfile*.

### NOTE

Use the **dbconvert** command only once to convert the old MACS database file to the new format. Never use **dbconvert** with a previously converted or newly-created database.

For example:

```
✓ # cd /usr/spool/macs/private  
✓ # mv macd.data macd.data.R1.2  
# dbconvert macd.data.R1.2 macd.data  
new database version=1.3  
Database conversion completed successfully
```

### NOTE

Do not use the names *macd.data.old* or *macd.data.new*, as these names are used by MACS.

Once you have converted the MACS database, you can then start MACS manually:

```
✓ # /sbin/init.d/mac start  
MACS services provided.
```

## Configuring MACS

If you will be running MACS at your site, refer to Chapter 4 of the *Paragon™ System Multi-User Accounting and Control System Manual* for setup and configuration instructions.

## Configuring NQS

If you will be running NQS at your site, refer to Chapter 5 of the *Paragon™ System Network Queueing System Manual* for setup and configuration instructions.

