Domain/OS System Software Release Document

Software Release 10.4
Manufacturing Number 018901-A00

First Printing: March 1992

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD 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. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in sub-paragraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Preface

These release notes describe standard Domain operating system (Domain/OS) software for Software Release 10.4 (SR10.4). The document includes an overview of new and changed functionality, software installation information, documentation references, and a list of fixed and existing bugs and limitations. Note that this document provides only an overview of new functionality. Throughout this release document we refer you to various technical manuals for additional information.

The normal software installation process places a version of these release notes in each node's **/install/doc/apollo** directory. For information about SR10.3, refer to the file **os.v.10.3__notes** (**os.v.10.3.p__notes** for Series 10000 nodes).

How to Print the Release Notes

You may print this document.

If your installation uses the SysV **lp** print daemon, use an **lp** command similar to the following:

```
lp -d printer_name pathname
```

where *pathname* is the pathname of the release notes, usually
/install/doc/apollo/os.v.10.4__notes (**os.v.10.4.p__notes** for Series 10000 nodes).
Note that there are *two* underscores before **notes.**

If your installation uses the Domain print system, use the following Aegis **/com/prf** command:

```
prf pathname -pr printer_name -npag
```

If your installation uses the BSD **lpd** print daemon, use an **lpr** command similar to the following:

```
lpr -P printer_name pathname
```

CONTENTS

List of Tables

Chapter 1: New Features in SR10.4

1.1 An Overview of This Release

SR10.4 is a release of Domain system software that provides new functionality, performance enhancements, support for new products, and bug fixes that have been added since SR10.3. The major changes include:

- Compliance with major standards
- Support for new hardware
- Discontinued support for some older hardware
- New operating system features
- RAI enhancements
- X11R4 Server, clients and Motif 1.1 **mwm** Window Manager
- HP VUE 2.01
- Eight-bit Native Language Support (NLS)
- HP/DDE 2.0 (formerly named Domain/DDE)
- New Basic Linear Algebra Subroutine (BLAS) Library

1.2 Optional Product Information

In the list that follows, we indicate the optional products available with SR10.4 and their latest version number. Optional products have their own release documents online in the **/install/doc/apollo** directory for your reference.

The list is *not* intended to help you determine which optional products are available for purchase. For specific information about purchasing these products, contact your local Hewlett-Packard sales representative.

Product	m68k Version	a88k Version
Domain/Ada	6.0.m and 6.0.mpx	6.0.p and 6.0.pmx
Domain/C	6.9.m and 6.9.mpx	6.9.p with 6.9.pmx
Domain/C++	2.1.0.m	2.1.0.p
Domain/CommonLISP	4.1	4.1.p
Domain/Dialogue	2.2	2.2.p
Domain FORTRAN	10.9.m and 10.9.mpx	10.9.p with 10.9.pmx
Domain Pascal	8.9.m and 8.9.mpx	8.9.p with 8.9.pmx
D3M	6.2	6.2.p
DSEE	4.0	4.0p
DPCC	3.6	N/A
DPCE	3.6	N/A
DPCI	5.1	5.1.p
DPSS/Mail	2.3	2.3.p
DTEK 4014	2.1	2.1.p
GKS	2.0	2.0.p
2DGMR	2.3	2.3.p
3DGMR	3.1	3.0.p
GSR	2.6	2.6.p
GPIO	10.3.1	10.3.1.p
HPGL	1.0	1.0.p
HP OmniBack	2.01	2.01.p
HP OSF/Motif		1.0p
HP/PAK (formerly DPAK)	5.0	5.0.p
HP VUE	1.1	1.1p
IKON92	1.6	1.6.p
ISNA_3270	1.1	1.1.p
ISNA_LU6.2	2.0	2.0.p
LSLOCK	2.0.1	2.0.1.p
NIDL	2.0	2.0.p
NFS	2.3	2.3.p
Open Dialogue	2.1	2.1.p
PHIGS	2.0	1.0.2.p
SCAT	2.1	2.1p
SoftBench	1.01	1.01.p
SPE	2.2	N/A
UEDK	1.1.3	N/A
Domain X.25	3.2	N/A
Knowledge Broker	1.2	N/A
Apollo/TECHnet	1.1	1.1.p

1.3 Standards Compliance

Domain/OS SysV at SR10.4 conforms to the following standards:

- ANSI C—ANSI C interfaces as defined in ANS x3.159-1989 Programming Language C.
- POSIX 1003.1A—POSIX kernel interfaces as defined in IEEE Std 1003.1A10990 (which includes enhancements to the IEEE Std 1003.110988 standard).
- FIPS 151-1—FIPS 151-1 as defined in NIST Federal Information Procurement Standard Publication 151-1, 1988 and 1989 revisions.
- XPG/3—XPG/3 base as defined in the *X/Open Portability Guide Issue 3*, Volume 1 *XSI Commands and Utilities*, Volume 2 *XSI System Interfaces and Headers*, and Volume 3 *XSI Supplementary Definitions*.
- OSF OSC AES—The OSF OSC AES (Application Environment Specification) as defined in The Open Software Foundation Application Environment Specification Operating System Programming Interfaces Volume, Order No. DEV-PI-00001 Revision A. (This is an OSF order number.)

The programming interfaces defined by the above specifications are documented in the *SysV Programmer's Reference* (005799-A01). Implementation-specific details of the Domain/OS SysV implementation of POSIX.1, XPG/3, and the OSF OSC AES are described in the *Domain/OS Standards Compliance Document* (019207-A00).

1.3.1 Generating Standards Compliant Applications

At SR10.4, applications may request that system calls exhibit runtime behavior in accordance with the POSIX, XPG3, and OSF AES standards. In some cases, this behavior differs from that exhibited at SR10.3. In general, an application may be compiled to request standards compliance or 10.3 compatibility, but not both. An application that is compiled for standards compliance at SR10.4 will not be able to execute on SR10.3.

The compiler symbols _POSIX_SOURCE, _XOPEN_SOURCE, and _AES_SOURCE are used to control standard compliant behavior. An application that desires behavior compliant with a particular standard(s), should define the relevant symbol(s). Applications define compiler symbols with the -D compiler flag, or by including a #define compiler directive in the source code.

1.3.1.1 Generating Applications On a SR10.4 Node That Are Compatible With SR10.3

Do not explicitly define any standards related compiler symbols. The default is for the compiler to generate an application compatible with SR10.3. The following command lines:

```
/bin/cc -o foo foo.c            (extended ANSI mode)
/bin/cc -A ansi -o foo foo.c    (strict ANSI mode)
/bin/cc -A nansi -o foo foo.c   (non-ANSI mode)
```

will all generate applications that will run on SR10.3 nodes. Note that compiling in extended ANSI mode still presents a standard compliant compile-time interface (namespace, function interfaces, type definitions) where possible.

1.3.1.2 Generating Applications Strictly Compliant With a Particular Standard

Compile in ANSI mode with the desired compiler symbols explicitly defined. For example, the command:

```
/bin/cc -A ansi -D_POSIX_SOURCE -o foo foo.c
```

generates an application strictly compliant with the POSIX.1-1990 standard definition. *This application will not execute on an SR10.3 node.*

1.3.1.3 Generating Applications Compliant With As Many Standards As Possible

In general, the XPG3 standard is a superset of the POSIX standard and the AES standard is a superset of the XPG3 standard. All three build on the ANSI C standard. A mode is provided that offers as much compliance with all of these standards as possible. This is similar in intent to the extended ANSI mode of SR10.3, except that runtime compliance is offered as well. Where conflicts occur, SR10.4 resolves them in favor of POSIX and ANSI C.

An application with general standards compliance may be generated by compiling in extended ANSI mode and explicitly defining any of the standard compliance compiler symbols (_POSIX_SOURCE, _XOPEN_SOURCE, or _AES_SOURCE). For example, the following command line:

```
/bin/cc -D_POSIX_SOURCE -o foo foo.c
```

would generate an application compliant with ANSI, POSIX, XPG3, and OSF AES, where possible. *This application will not execute on an SR10.3 node.*

1.3.1.4 Generating SVID-Compliant Applications

To generate applications strictly compliant with the System V Interface Definition (SVID), Issue 2, compile in extended ANSI mode with the symbol **_SVID2_SOURCE** explicitly defined. For example, the command:

```
/bin/cc -D_SVID2_SOURCE -o foo foo.c
```

generates an application strictly compliant with the SVID. This application will not execute on an SR10.3 node.

1.3.1.5 Determining If An Application Is Standard Compliant

The **coffdump** utility may be used to determine if an application will request standard compliant behavior or will be compatible with SR10.3.

Execution of the command line:

```
/usr/apollo/bin/coffdump -Asv <object file>
```

produces output similar to the following:

```
                        ***SRI INFORMATION***
         Info         Combining
         Code            Rule      Value
  <object filename>:
           8              4         10
           1              4          4   (MC68020 Required)
           5              8          2   (runtype sys5.3)
           3              8          2   (systype sys5.3)
           2              4      60000   (Code Assumes Registers
                                         Preserved Across External Calls)
                                         (Procedures Preserve Registers)
```

If a row exists that contains Info Code 8, then the application is standard compliant and will not execute on an SR10.3 node. If no such row exists, then SR10.3 compliance is preserved. The third column represents bit values corresponding to the individual standards. Possible values include:

10	POSIX
100	XPG3
1000	OSF AES
1150	POSIX, XPG3, AES, and POSIX Threads

Other combinations of the individual standards are possible.

1.3.1.6 Troubleshooting Problems

PROBLEM:
Compiler gives undefined pragma warning similar to the following:

```
(0003) #pragma HP_STANDARD 4

******** Line 3: [Warning #253]  Unrecognized pragma.
```

SOLUTION:
You have an old version of the C compiler that does not support generation of applications with runtime standards compliance. You should either upgrade to version 6.9 of the C compiler or refer to section 1.8.2 of this document for instructions on setting standards compliance bits with the linker.

PROBLEM:
Loader rejects object code with the following message:

```
file is not an object module or is not executable on this machine
type (process manager/loader)
```

SOLUTION:
You may be executing a standard compliant application compiled on SR10.4 on a SR10.3 node. Either run the application on a SR10.4 node or else recompile the application without explicitly defining any standard compliance symbols (ie, _POSIX_SOURCE).

PROBLEM:
Application does not give standard compliant behavior:

SOLUTION:
The following checklist outlines some potential problems:

- Application must explicitly define a standard compliance compiler symbol (ie _POSIX_SOURCE)
- The appropriate system header files must be included. These files generally include /usr/include/sys/stdsyms.h which is required to obtain standard compliant behavior.
- Must be running a sys5.3 or FIPS environment. Check the file /etc/environ.
- Application must have been compiled with a version 6.9 C compiler.
- The application must be running on a 10.4 node.

1.4 New Hardware Support

SR10.4 provides support for the Domain Series 5500 Personal workstation, the HP Apollo 9000 Series 400 workstations, and several new peripheral devices. The following subsections briefly describe the new hardware.

1.4.1 Support for the Domain Series 5500 Personal Workstation

SR10.4 provides support for the new Domain Series (DS) 5500, which is an MC68040-based CPU board upgrade to the Domain Series 3500, 3550, and 4500 personal workstations.

All existing graphics, networks, mass storage, and backup devices presently available on the DN3500, DN3550 and DN4500 workstations are supported. All memory modules shipped with the DN3500, DN3550 and DN4500 workstations are also supported. In addition, a new 16-MB memory module has been added which gives the DN5500 a total memory capacity of 64 MB.

Because the MC68040 combines the MC68030 and MC68882 chip set into one package and provides greater performance, the Floating-Point Accelerator Board presently available on the DN3500, DN3550 and DN4500 is not supported on the DN5500.

The PC AT bus and onboard I/O are retained on the DN5500. To optimize performance, a new memory controller and bus interface have been designed. A new I/O protection mechanism has been designed to support the 4-KB page I/O mapping of the MC68040. The new CPU board uses the 25 MHz version of the MC68040.

1.4.2 Support for the HP Apollo 9000 Series 400 Workstations

This release also provides support for the HP Apollo 9000 Series 400 workstations, which consist of Models 400s, 425s, 433s, 400t, 425t, 400dl, and 425e. The Models 400s, 400t and 400dl workstations are MC68030-based personal workstations. The Models 425s, 433s, 425e, and 425t workstations are MC68040-based personal workstations. They are all compatible with other Domain systems. These workstations have their own documentation. An overview of these products follows.

The Model 400t workstation is available in 8-MB to 32-MB configurations. The Model 425t workstation is available in 8-MB to 64-MB configurations. They include the following built-in I/O interfaces: EtherLAN (jumper selectable), SCSI, Centronics parallel, and RS-232 serial. They have one modified PC AT bus slot for an optional Apollo Token Ring or 802.5 Token Ring controller board, and one slot for DIO-II or SGC graphics options. Storage options include one or two internal 200-MB or 400-MB Winchester disks and various SCSI external devices.

The Model 425e workstation is available in 8-MB to 48-MB configurations. The system includes SCSI, Centronics parallel, and RS-232 built-in I/O devices. The CPU motherboard includes several graphic options. Internal storage options include either an HP qualified 200 or 400-MB SCSI Winchester disk drive with an optional CD-ROM or 3 1/2 inch flexible disk drive. Various external SCSI devices are also supported, including a new HP qualified 1.3-GB SCSI 5-1/4 inch Winchester.

The Model 400dl is a subset of the Model 400t that is available in either an 8-MB or 16-MB configuration. The Model 400dl includes an embedded EtherLAN and RS-232 interface and a DIO-II slot that is limited to the monochrome graphics controller with the 19-inch 1280 x 1024 monochrome monitor. It contains no modified PC AT bus slot or SCSI, Centronics, and RS-232 interfaces, and offers no internal or external storage devices.

The Model 400s workstation is available in 8-MB to 128-MB configurations. The Model 425s and 433s workstations are available in 8-MB to 128-MB configurations. These workstations include the same built-in I/O interfaces and graphics options as the Model 400t and 425t. Option slots include two DIO-II bus slots, which are convertible to DIO-I slots. One DIO-II slot supports the graphics option. In addition, the Model 400s workstation can include either three more DIO-II bus slots or four ISA bus slots. The Model 425s and 433s workstations can include either three more DIO-II bus slots or four EISA bus slots. Storage options include the following internal devices: 330-MB and 660-MB Winchester disks, Cartridge Tape drive, 1.3-GB Winchester drive (425s and 433s only), Magneto Optical disk, CD-ROM drive, plus various external SCSI devices such as 8-mm tape and floppy drives.

The following graphics options are supported on Model 400t, 425t, 400s, 425s, and 433s workstations:

- Monochrome VRX controller
- Color VRX controller
- Personal VRX P2 graphics subsystem
- Personal VRX P3 graphics subsystem
- GRX grayscale controller
- CRX color controller

The following new graphics options are available for the Model 425e workstation. These options are integrated onto the Model 425e CPU board.

- EVRX Greyscale 1280 x 1024
- EVRX 2D Color 1280 x 1024
- EVRX 2D Color 1024 x 768

1.4.3 CD-ROM Reader Support

Domain/OS SR10.4 provides support for the CD-ROM drive (Series 6100 Model 700/S), a random-access, read-only mass-storage device that uses removable CD-ROM disks. The CD-ROM drive contains a semiconductor laser for reading data optically, and includes an embedded controller with a SCSI interface. The CD-ROM system is compatible with the ISO and High Sierra disk formats, and with all workstations that support SCSI devices.

Although Hewlett-Packard does not supply Domain/OS software on CD-ROM, support is provided for booting from the CD-ROM drive, for the convenience of VABs and ISVs who may wish to make use of this capability. (The proper boot prom is, of course, required).

One or more CD-ROM readers may be attached to the following node types:

SAU7 - Series 3500/3550/4000/4500 when equipped w/SCSI Disk controller
SAU9 - Series 2500
SAU10 - Series 10000 when equipped w/SCSI Disk controller
SAU11 - Series 400 (425t 425s and 425e 68040 systems)
SAU12 - Series 400 (68030 systems)
SAU14 - Series 5500 (68040 systems)

NOTE: CD-ROM is not supported on diskless nodes (dl models).

All node types (SAUs 2-12 and SAU14) may access a CD-ROM file system mounted on another node over the Domain network, providing that the CD-ROM file system has been mounted on the node to which it has been attached and the CD-ROM support software installed on the two nodes satisfies the following interoperabilty constraints:.

> The CD-ROM support software in SR10.4 will interoperate with the SR10.3 PSKQ3. It will also interoperate with SR10.3 PSKQ2 as long as the node running PSKQ2 is attached to the CD-ROM reader. A node running PSKQ2 will NOT be able to access a CD-ROM reader attached to a node running either SR10.4 or SR10.3 PSKQ3.

For information about installing and using the CD-ROM drive, refer to the *HP Series 6100 Model 700/S User's Guide* (A1999-90602), and to the section "Changes to HP Series 6100 Model 700/S CD-ROM User's Guide" in Chapter 3 of these notes.

1.4.3.1 Changes in CD-ROM Drive Usage at SR10.4

The following list is an overview of the changes in CD-ROM drive usage at SR10.4:

- Device files are now created for you during the installation of SR10.4. The files are named /dev/cdrom for SCSI target 0, /dev/cdrom_1 for SCSI target 1, and so on. These files can be used to access the CD-ROM disc if your CD-ROM drive is attached to SCSI bus 0 (the primary SCSI bus).
- At SR10.4, the CD-ROM software requires that the external pager daemon (/etc/xpager) be running. You can start this daemon by creating a file called /etc/daemons/xpager and rebooting.
- For pre-SR10.4 systems, the directory at which you mount the CD-ROM file system must not exist before the mount. At SR10.4, this requirement has changed to use the UNIX semantics, where the mount directory MUST exist before you mount the CD-ROM drive.

For more detailed information about using your CD-ROM drive at SR10.4, refer to the section "Changes to HP Series 6100 Model 700/S CD-ROM User's Guide" in Chapter 3 of these notes.

1.4.4 Series 10000 1.4-GB Disk Upgrade

Domain/OS SR10.4 supports a new high capacity 1.4-GB (formatted) Winchester disk drive available with Series 10000 Super-Workstation Systems. With this software release, the Series 10000 can support (two) external Multiple Disk Expansion Module(s), increasing the Series 10000 total hard disk storage capacity to 18-GB per system.

The new drives are the same form factor (5 1/4") as the 348-MB and 700-MB disk drives and are compatible with those drives. However, cylinder and sector striping restrictions do apply when striping across unlike drive geometries.

1.4.5 Support for new 2D/3D Graphics Cards on Models 425t and 425s

Domain/OS SR10.4 provides support for the 2D/3D (GRX) Grayscale Graphics Card (HP A1924A). The grayscale 2D/3D (wireframe) graphics card provides software-assisted graphics funtionality, and includes the following features:

- 8 monoplanes; 256 levels of gray
- Accelerated raster operations and vectors
- Hardware writable cursor
- Support for the 19-inch, high resolution (1280 x 1024) 72 Hertz Monochrome Monitor (HP 98774A)

This release also supports the 2D/3D (CRX) Color Graphics Card (HP A1659A). This graphics card is an 8-bit pseudo-color, double buffered 2D/3D (wireframe) that provides software-assisted graphics functionality. The card has the following features:

- 8 color planes; 256 location color look-up table; 8/8 double buffers
- Accelerated raster operations and vectors
- Hardware writable cursor
- Support for HP A1097A (for northern hemisphere) or HP A1097B (for southern hemisphere) 19-inch high resolution (1280 x 1024) 72 Hertz color monitor

1.4.5.1 True Color Emulation for the 2D/3D Cards

The 2D/3D graphics cards support true-color emulation through GPR and GSR. This emulation can be used to display true-color bitmaps, even though there are only 8 planes of frame buffer memory. True-color pixel blts from main memory to the screen are color-packed and dithered.

The only drawing operations supporting this functionality are GPR and GSR pixel blts from a main memory bitmap to the screen, and through the **gpr_$write_pixels** call. For GPR and GSR pixel blts, the main memory bitmap must be a pixel oriented bitmap with three sections of 8 bits, or one section of 32 bits allocated size and 24 bits pixel size. Raster operations and use of a plane mask are not supported for these "dithered" transfers. That is, the raster operation for all planes should be "destination equals source" and writes to all planes should be enabled in the plane mask.

To use the emulation, the application initializes a pseudo color graphics window in a normal fashion, using either **gpr_$direct** or **gpr_$borrow**, *not* **gpr_$direct_rgb** or **gpr_$borrow_rgb**.

Turn the emulation on with the **gpr_$set_dither_mode** call with an argument of **gpr_$dither_rgb666_4x4**. Turn the emulation off with the same **gpr_$set_dither_mode** call with an argument of **gpr_$dither_off**. To inquire the current state of dithering, use the **gpr_$inq_dither_mode** call.

A color map consistent with the dithering emulation must be loaded to properly view the results. This can be accomplished by executing the **lcm -rgb666** command from a shell. Alternatively, you can find the appropriate color map in **/sys/node_data/etc/dm_display/color_map.rgb666**.

The 2D/3D graphics card uses 6 shades each of red, green and blue for its color packing, resulting in a total of 216 shades. The colormap slots from 40 to 255 are needed to view dithered bitmaps. Color-packed blts are dithered, using a 4x4 dithering cell.

1.4.5.2 Greyscale Colormap Processing for the 2D/3D Cards

To allow color GPR and GSR applications to work with the greyscale monitor with minimum modification, color values are automatically converted to grey values of approximately equal brightness. This is accomplished by placing a weighted average of input color values into the hardware colormap. The average used is approximately as follows:

ColorMapValue = 0.30 x InputRed + 0.59 x InputGreen + 0.11 x InputBlue

When reading the color values from the colormap through GPR or GSR calls, the values returned did not have the above weighting function applied to them. The same color values input to **gpr_$set_color_map** are returned by **gpr_$inq_color_map**. This processing doesn't take place on color boards.

Some properties of this implementation are:

1. Setting a color map slot to red = green = blue = 255 results in the brightest white displayed on the screen. Setting a color map slot to red = green = blue = 0 results in the blackest black displayed on the screen.

2. Setting all three color map components equal gives "intuitive" results. For example, setting red = green = blue = 127 results in, 127/255, or nearly half white.

3. Colors that differ mainly in their blue component are difficult to distinguish on the greyscale monitor.

1.4.6 New Data Types

The following subsections describe new data types added to this software.

1.4.6.1 New Data Types for the HP A1924A and HP A1659 Graphics Cards

The following data types were added to identify the new graphics cards.

- The value **gpr_$color_14_1280x1024** was added to the type, **gpr_$display_config_t**, which is returned from a call to **gpr_$inq_config()**.
- The value **gpr_$ctl_color_14** was added to the type, **gpr_$controller_type_t**, which is returned from a call **gpr_$inq_disp_characteristics()** or **gpr_$inq_display_characteristics()**.
- The value **pad_$color14_display** was added to the type, **pad_$display_type_t,** which is returned from a call to **pad_$inq_disp_type()**.
- To distinguish between greyscale and color versions of these cards, application programs can examine the n_primaries field of the **gpr_$disp_char_t** structure returned by **gpr_$inq_disp_characteristics()** or **gpr_$inq_display_characteristics()**. Greyscale cards have one (1) primary. Color cards have three (3) primaries.

1.4.6.2 New Data Types for the Model 425e

The following data types were added to identify the Model 425e workstation:

- The value **gpr_$controller15** was added to the type, **gpr_$display_config_t** which is returned from a call to **gpr_$inq_config()**.
- The value **gpr_$ctl_controller_15** was added to the type, **gpr_$controller_type_t** which is returned from a call **gpr_$inq_disp_characteristics()** or **gpr_$inq_display_characteristics()**.
- The value **pad_$controller15_display** was added to the type, **pad_$display_type_t** which is returned from a call to **pad_$inq_disp_type()**.

1.4.7 Tape Drive Support

The Model 1300S Digital Data Storage Tape Drive (Product Number C1512A) and the Autoloading 1/2-inch Magnetic Tape Unit (Product Number 7980S) have been verified and released for use with the HP Apollo 9000 Series 400 Workstations and this software. For more information about the Model 1300S, refer to the *Model 1300S User's Manual* (C1512-90901). For more information about the Model 7980S, refer to *Installing and Using the 9-Track 1/2-Inch Tape Drive with an HP9000 Series Domain/OS System* (07980-90020).

1.4.7.1 Using a DDS-Format Drive with an HP 9000 Series 400 Domain/OS System

The 4mm DDS-Format DAT drives operate in a manner similar to traditional 9-track tape drives, such as the HP 7974, HP 7978, and HP 7980. UNIX utilities **cpio**, **tar**, and **mt** can access the drives. The Aegis commands **wbak**, **rbak**, and **rwmt** are supported. These utilities allow unattended backup for most applications.

Domain OmniBack Version 1.2 does not support the DDS-Format drives. Use OmniBack Version 2.0.

1.4.7.2 Configuring the Drive for Domain/OS SysV or BSD Environments

NOTE: If you are running in the Aegis environment, you do not need to use the special device driver files described in this section.

Supplied Device Files

Domain/OS provides a limited set of device files that allow you to access the drives immediately. The following device files ship with the system software:

```
/dev/rmts8  = SCSI ID 1 Rewind on close
/dev/rmts9  = SCSI ID 2 Rewind on close
/dev/rmts10 = SCSI ID 3 Rewind on close
/dev/rmts11 = SCSI ID 4 Rewind on close
/dev/rmts12 = SCSI ID 1 No rewind on close
/dev/rmts13 = SCSI ID 2 No rewind on close
/dev/rmts14 = SCSI ID 3 No rewind on close
/dev/rmts15 = SCSI ID 4 No rewind on close
```

All of the supplied device files are default density.

Creating Special Device Files

While default device files are provided with Domain/OS, you may wish to create your own device file by using the following command syntax:

/etc/mknod *<filename> <filetype> <major number> <minor number>*

NOTE: You must be logged in as a superuser to perform these steps.

<filename>

You can use the naming convention for the character device files for magnetic tapes which is described in the mt(7) section of your appropriate Command Reference Manual. The name of the device file ends with it's minor device number. Using the convention helps you keep track of how the minor numbers for drives are set up. We recommend that you use the traditional naming schemes. While the supplied limited set of device files do not follow this convention, the examples indicated do.

<filetype>
All magnetic tape drive device files should be filetype **c**, for character type.

<major number>
The major number is 24.

<minor number>
The minor number is in the form of ddntttbb and specifies the following information:

-drive density (dd), (00 is the only option)

-rewind (0) or no rewind (1) on close (n)

-the 4mm drive's target SCSI ID that you selected with the drive's address switches (ttt), (000 - 110)

-the system's SCSI controller number (bb), (must be zero(00))

NOTE: There must be a digit in each position of ddntttbb (ddntttbb is a binary string). Hex equivalents can be determined using the table that follows.

To find the *<minor number>*, cross reference the system's SCSI controller number with the 4mm drive's target SCSI ID and the rewind/no rewind option shown the following table.

Drive Target SCSI ID	Rewind Y/N	SCSI Controller Number	Minor Number (Hexadecimal)
0	Y	0	0x0
0	N	0	0x20
1	Y	0	0x4
1	N	0	0x24
2	Y	0	0x8
2	N	0	0x28
3	Y	0	0xC
3	N	0	0x2C
4	Y	0	0x10
4	N	0	0x30
5	Y	0	0x14
5	N	0	0x34
6	Y	0	0x18
6	N	0	0x38

Where: 0x indicates that the number is in hex. The command line interpreter assumes decimal unless you prefix the number with 0x.

Example:
dd=00, n=0, ttt=011(binary, 3 decimal), bb=00
would result in ddntttbb = 00001100(base 2) = C(base 16) = 12(base 10)
and the command syntax: /etc/mknod /dev/rmts12 c 24 0xC, or
/etc/mknod /dev/rmts12 c 24 12 (decimal)

or, if n=1, ddntttbb = 00101100(base 2) = 2C(base 16) = 44(base 10)
and the command syntax: /etc/mknod /dev/rmts44 c 24 0x2C, or
/etc/mknod /dev/rmts44 c 24 44 (decimal)

1.4.7.3 Testing the Configuration

To verify that the UNIX configuration is working, enter:

mt -f *<filename>* **-scsi rewind**

To verify that the Aegis configuration is working, enter:

rbak -dev mt0 -rewind

NOTE: In this example mt0 specifies a drive target SCSI ID of 1. See the **rbak** and **wbak** Command Options section for a more complete explanation and list of **-dev mt** options.

These commands rewind the tape and should not return any error messages. The lights on the front panel of the drive should flicker momentarily.

1.4.7.4 Using the Drive with Domain/OS

If you are familiar with 9-track tape drives such as the HP7980 or the CDC6250, you will find the 4mm DDS-Format drives behave in a similar way. LBOT (Logical Beginning of Tape) on DDS-Format tapes are equivalent to BOT on 9-track tapes. "Filemark" is the same as EOF. "Record" is the same concept for both types of drive. Spacing operations are similar, and the concept of positioning the tape head after filemarks, or after a specified number of records is the same. Tape record sizes can be of variable length, from 1 byte up to 64 KB.

Refer to the appropriate Command Reference Manual for information on UNIX and Aegis commands to use the 4mm DDS-Format drive.

1.4.7.5 Domain/OS UNIX Commands

The three UNIX commands, **cpio**, **tar**, and **mt** are used for accessing tape drives. If you are familiar with 9-track drives, you should not see any difference in the use of these commands. The DDS-Format drive is compact and easy to use. The following examples use the supplied device file **/dev/rmts8**.

cpio - The command is used for archiving files and file systems. It is commonly used in a filter with the find command as follows:

find *<directory>* **-print | cpio -oB > /dev/rmts8**

If the archive is too large to fit on one cassette, **cpio** prompts you to eject the cassette manually and insert another one. This means that the possibilities for multiple tape archive are the same as for 9-track drives. To recover files from the archive, type:

cpio -iBdx < /dev/rmts8 *<filelist>*

tar - The tar command saves and restores files and directories on tape.
To archive a directory type:

tar cf /dev/rmts8 *<directory>*

The **tar** command writes the directory as a sequence of 8-KB records. To restore the directory from the archive, type:

cd *<directory>*
tar xf /dev/rmts8

mt - The **mt** command under Domain/OS supports only one SCSI operation; rewinding the tape. To rewind the tape, type:

mt -f *<filename>* **-scsi rewind**

1.4.7.6 Domain/OS AEGIS Commands

The three AEGIS commands, **rbak**, **wbak**, and **rwmt** are used for accessing tape drives. This section provides a brief summary of how to write objects to, or restore objects from, a 4mm tape by using the rbak and wbak commands.

To read or restore files from another system use the **read_write_magtape** (**rwmt**) command. The **rwmt** command can read unlabeled tapes, as well as ANSI level 1 through 4 labeled tapes. The 4mm tape must, however, be in DDS format. Refer to the appropriate Command Reference manual for further details on reading/restoring non-Domain/OS formatted files by using **rwmt**.

When performing operations using the 4mm drive you commonly use the following options with the **wbak** and **rbak** commands. The Domain help files and the appropriate Command Reference manual contain more detailed information about the commands and options described in the following text.

1.4.7.7 rbak and wbak Command Options

This section describes options used with the **rbak** and **wbak** commands.

-dev mt#

where: # is the drive's target SCSI ID minus one(# = SCSI ID - 1)

This option specifies that the device you want to access is a 4mm tape device. It also specifies which SCSI device ID is to be used.

The valid options are:

-dev mt0	(for SCSI ID 1)
-dev mt1	(for SCSI ID 2)
-dev mt2	(for SCSI ID 3)
-dev mt3	(for SCSI ID 4)
-dev mt4	(for SCSI ID 5)
-dev mt5	(for SCSI ID 6)

The **-dev mt** option is used with **rbak** and **wbak**.

-no_eot

This option prevents the write program from placing an end-of-tape (eot) indication on the tape. Use **-no_eot** when you are using multiple invocations of **wbak** to copy objects sequentially onto a tape. You must use **-no_eot** to prevent the tape from rewinding to the beginning before searching for the next specified file positions. This option is for **wbak** only.

-reo

This option reopens the tape at the tape's current position. This option is for **wbak** only.

-rewind

This option rewinds the tape to the beginning. You must use this option to rewind the tape when you perform a read with **rbak**. Otherwise, the tape remains at its current file position. You can rewind a tape by specifying **-rewind** and **-dev mt#** as the only options for **rbak**.

Tape Close

When an application closes a device, either by exiting or by explicitly closing the device, the driver's close routine is called. In order that many diverse tape applications can use the full functionality of the drive, the algorithm used in tape drivers during the close routine is as follows:

1. If the last operation was a write (outputting data to the tape), the driver automatically writes two filemarks.

   -If the "no rewind" bit is not set, the tape is then rewound to LBOT.

   -If the "no rewind" bit is set, the driver backspaces the tape one filemark and exits. This leaves the tape head positioned between the two concluding filemarks.

2. If the last operation was a read, then:

   -If the "no rewind" bit is not set, the tape rewinds to LBOT.

   -If the "no rewind" bit is set, no tape motion occurs and the driver exits.

1.4.7.8 DDS-Format Tapes Compared to 9-Track Tapes

Other important points when comparing tapes for DDS-Format drives and 9-track tapes are as follows:

Data Coding: DDS-Format tapes use a different coding format from that used by 9-track tapes.

EOD: When a DDS-Format drive writes a sequence of records to tape and then rewinds, it inserts an EOD area at the end of the records. Subsequent reads ignore any data after the EOD area. 9-track tape drives do not have any parallel functionality.

Write-protecting media: DDS-Format cassettes use a small write-protect tab to write protect the media. 9-track tapes use write-protect rings.

On-line/Off-line: The meaning of on-line and off-line is similar for DDS-Format drives and 9-track drives. If a cassette is loaded in a DDS-Format drive and the device is power-cycled, or Domain/OS is rebooted, the mechanism will cause the drive to go off-line. To bring it back on-line, eject the cassette and reload it.

Streaming and immediate report: DDS-Format drives support both streaming and immediate report modes.

The DDS-Format drive will respond to all Domain/OS commands appropriate for magnetic tape devices. For more information about magnetic tape devices refer to the appropriate Domain/OS Command Reference Manual.

1.4.8 New Series 400 Printer Support

A Timeout and handshake mode has been added to the HP Apollo 9000 Series 400 parallel port. Refer to the configuration file **‘node_data/etc/pio1.conf**.

NSTRB_ONLY: (Strobe only)

Use this handshake for devices that do NOT raise busy in response to the leading edge of the strobe. The interface emits a one microsecond strobe pulse, then waits for the acknowledge pulse from the device. This mode can be slower than BNACK or BONLY, so use it only if necessary. Devices that require this handshake mode include the QMS, Tektronix 4693DX, and Laser26 printers.

TM:*nnnn*

Where *nnnn* is specified in seconds.
A Timeout value can be set by the user to specify when the OS will time out waiting for a response from the pio device. The default timeout value is 2 minutes. The value is represented in seconds: TM:120.

1.4.9 Model 20GB/A Optical Disk Library System (Autochanger) Support

Software Release 10.4 provides support for the HP Series 6300 Model 20GB/A Optical Disk Library System (autochanger). This release supports the following software configurations:

- **Autochanger Server**
  HP Apollo 9000 Series 400 Model 425x running SR10.4
  HP Apollo 9000 Series 400 Model 433s running SR10.4
- **Autochanger Client**
  DN2500 running SR10.4
  DN3x00 running SR10.4
  DN4x00 running SR10.4
  DN5x00 running SR10.4
  DN10000 running SR10.4
  HP Apollo 9000 Series 400 Model 425x running SR10.4
  HP Apollo 9000 Series 400 Model 433s running SR10.4

1.4.9.1 An Overview of the Model 20GB/A Autochanger

The Model 20GB/A autochanger is an online storage device with a total capacity of 20 gigabytes of data. The autochanger provides reliable and economical data storage. The cabinet contains the following:

- an autochanger controller
- two 5.25-inch rewritable optical disk drives
- a mail-slot for inserting and removing disks
- storage slots for thirty-two 650 MB (1024 btye/sector format) or 594 MB (512 btye/sector format) magneto-optical (Continuous Composite (C*C) format) disks

1.4.9.1.1 Hardware Requirements

To incorporate the autochanger into the Domain system, the following hardware requirement must be met.

1. The autochanger server node must be a Model 425x or Model 433s Workstation. Nodes should be configured with a minimum of 16 MB of memory.

1.4.9.1.2 Model 20GB/A Install Information

The following are install requirements for the Model 20GB/A autochanger:

1. Always power on the Model 20GB/A before booting the node.
2. If you boot from a tape or from a CDROM release medium provided by an aftermarket supplier (HP does not supply Domain/OS on CD-ROM), DO NOT try to mount autochanger media from the boot shell. Wait until the **/etc/rc** script runs, until you run a single-user shell, or until the window system boots. You may freely mount autochanger media from the boot shell if you boot from a Winchester disk

or over the network.

The following manuals are shipped with the Model 20GB/A.

Unpacking Instructions (C1700-90073)

HP Series 6300 Model 20GB/A Optical Disk Library System
CE Installation Guide for Apollo Sites (018905-A00)

Administering the Optical Disk Library System on Domain/OS (018904-A00)

Optical Disk Library System User's Guide (C1700-90075)

1.4.10 New and Updated Online Documentation

The following man pages and help files have been updated to reflect support for the autochanger:

acadmin
chuvol
mkdev
netsvc
mtvol

Also see the new domain_examples for **/ac/rc.ac.unix** and **/ac/rc.ac.aegis**. The examples describe startup scripts that automatically set up the autochanger.

1.4.10.1 Autochanger Limitations

- The autochanger's rewritable optical disk drives have an access time that is 2-5 times slower than high performance magnetic hard disks; therefore, the Model 20GB/A should not be used as a hard disk replacement. See *Administering the Optical Disk Library System on Domain/OS* (018904-A00) for information about performance management.
- The Model 20GB/A cannot be used as a bootable device.
- SCSI cable length is limited to 6 meters.
- Each server node can support only one Model 20GB/A autochanger.
- It is recommended that the server node be used as a dedicated server for the autochanger.
- If you boot from a tape or from a CDROM release medium provided by an aftermarket supplier (HP does not supply Domain/OS on CD-ROM), DO NOT try to mount autochanger media from the boot shell. Wait until the **/etc/rc** script runs, until you run a single-user shell, or until the window system boots. You may freely mount autochanger media from the boot shell if you boot from a Winchester disk or over the network.
- Client nodes requiring access to the autochanger must be running SR10.4. Any client node running pre-SR10.4 software will receive an error when accessing surfaces within the autochanger. Access to the server's local disk is not limited.

- Caution should be used with regard to simultaneous access to the autochanger. The autochanger, and its supported software, can handle reasonable attempts at simultaneous access. For example, simultaneous access by users to one or two separate surfaces does not pose a problem. Caution should be taken, however, with multiple users attempting to access more than three surfaces simultaneously via the network; this applies especially to large files (greater than 30 megabytes). Such an attempt may result in a client node experiencing network timeouts because the server node is busy handling multiple network requests and disk access to the autochanger.

  Please refer to the manual *Using the Optical Disk Library System under Domain/OS (018904-A00)* for further system precautions.

1.5 Discontinued Support for Some Older Hardware

At SR10.4, support for the following hardware is discontinued:

TABLE 1-1. Unsupported Machine Types at SR10.4

Unsupported Machine Types		
SAU	**Node Name**	**CPU Type**
sau2	dn300	68010
	dn320	68010
	dn330	68020
sau3	dsp80	68010
	dsp90	68020
sau4	dn460	68010
	dn660	68010
sau5	dn550	68010
	dn560	68020
	dn570	68020
	dn580	68020
sau6	dn560T	68020
	dn570T	68020
	dn580T	68020
	dn590T	68020

1.6 New Software Release Bulletin on Media Lists All Fixed Known Problems

New with SR10.4, we are providing a Domain Software Release Bulletin (SRB) on the SR10.4 media. The SRB documents all customer reported known problems that have been resolved with SR10.4. It also covers some associated optional products such as the compilers. A complete list of the products described in the SRB can be found in the software release contents of the SRB. The SRB is located in the files:

/iustall/doc/apollo/os.v.10.4__software_release_bulletin (m68k version)
/install/doc/apollo/os.v.10.4.p__software_release_bulletin (a88k version)

1.7 New and Changed Operating System Features

The following sections describe new and changed operating system features in this release.

1.7.1 Overview of Domain/OS Threads Package

At SR10.4 Domain/OS provides a threads package, which includes facilities for thread management, thread priority scheduling, synchronization primitives, thread cancellation, process control and thread-specific data handling.

In addition to the standard library facilities that are based on draft 4 of the IEEE P1003.4a-1990 standard, Domain/OS provides a low level set of calls based on the Mach thread interface. Domain/OS also supplies new calls to add functionality that Mach does not provide.

1.7.1.1 Background

A thread is a single sequential flow of control within a process. Each thread has its own thread ID, scheduling priority and policy, the state of any timers, **errno** value, thread-specific key bindings, and required system resources to support a flow of control.

Prior to SR10.4 the standard Domain/OS runtime model consisted of a collection of processes running on a node, where a process was an address space containing executable code, a user stack, a set of protections, one or more data sections, and a set of registers for the CPU, including a PC.

At SR10.4, the Domain/OS thread model splits the process in two pieces. The process continues to be an address space, which contains executable code, a set of protections, and one or more data sections. A thread consists of a register set and a user stack. This split enables more than one thread to run in a single process, allowing concurrency within a single program.

1.7.1.2 Benefits of Implementing Threads

There are several areas of application programming that can benefit from a thread interface. The use of threads enable applications to perform the following tasks:

Simultaneous processing on multi-CPU machine
> For example, the compiler or the application programmer could split apart a problem into chunks that could be worked on simultaneously.

Management of multiple clients
> For example, a server process may need to watch more than one client at

a time.

True Asynchronous I/O

For example, a specific thread can be dedicated to watching one stream while the other threads in the process did other work.

1.7.1.3 Organization of Threads Interfaces

The Domain/OS threads package provides two interfaces for applications programmers. Access to thread functionality is available via:

- The standard C Library programming interface, which consists of **pthread_xxxx** calls and are based on draft 4 of the IEEE P1003.4a standard.
- The Domain/OS intermediate level interface, which contains **thread_xxx** calls. Most of these calls come directly from the Mach thread interface. Except as noted, they work exactly like the equivalent Mach call. These calls are provided solely for compatibility with the Mach operating system. New applications should use the standard C Library Pthread interface calls instead.

The kernel directly provides the lowest level threads interface. However, this level is not exported to user applications.

1.7.1.4 Intermediate Level: Domain/OS Threads Interface

The Domain/OS Threads Library consists of Mach-equivalent calls, Domain/OS added calls, and convenience routines.

1.7.1.4.1 Basic Data Structures

Data structures for intermediate level thread calls are as follows:

thread_t

The basic thread ID. This is a black box value that is not significant outside the thread system. It is returned by **thread_create**. No two active threads in the same process will have the same thread ID (currently, no two active threads on a node will have the same thread ID, but this is not guaranteed).

threadp_key_t

This is the key assigned for a data context (that will hold per-thread data) when it is created. It is returned by **threadp_init**, and must be shared between all threads using this particular context.

pid_t

The process ID, as defined by POSIX.

kern_return_t

A Mach definition that is an enumeration of the possible values a system call can return. It indicates the success or failure of the call. It will either be KERN_SUCCESS to indicate that the call succeeded, or some other value to describe the reason for the failure.

1.7.1.4.2 Mach-equivalent Calls

The following list specifies Mach equivalent calls:

thread_create
create a new thread

thread_terminate
initiate thread termination

thread_suspend
increment the thread's suspend count

thread_resume
decrement the target thread's suspend count

thread_abort
return interrupt status

thread_self
return the thread ID

thread_info
obtain information about the specified thread.

thread_state: **thread_get_state, thread_set_state**
get or set the CPU state of the thread.

1.7.1.4.3 Added Calls

These are the calls we have added to support the Mach thread interface on Domain/OS:

thread_set_priority
set values for the scheduling priority

thread_handle_signals
identify the thread context that will handle signals (only one thread can handle asynchronous signals)

thread_inhibit, thread_enable
increment and decrement the inhibit count

thread_cleanup
initiate thread termination and detect termination status

thread_yield
inform the scheduler that this thread no longer needs to run.

1.7.1.4.4 Convenience Routines

These calls are not really part of the true OS layer, but are intended to be a layer below the applications interface. They are primarily convenience routines, but some of them set up internal state that is absolutely required for a normal applications thread to run in the Domain/OS environment.

thread_startup
start a new thread.

1.7.1.4.5 Per-Thread Storage Manager Calls

One of the items deemed useful for some thread implementations is per-thread storage. This mechanism allows per-thread context to be stored and retrieved simply. The context is created using **threadp_init** and a key is associated with it. Each thread that wishes to use this context must then allocate its private data, then call **threadp_set** with a pointer to it (or the data itself if it is no larger than a pointer). After this is accomplished, all calls to **threadp_get** with this key will produce the context for the current thread.

The Per-Thread Storage Manager comprises the following intermediate level calls:

threadp_init
create a new per-thread data context.

threadp_set
set the per-thread data for current thread.

threadp_get
get the per-thread data for the current thread.

1.7.1.5 Applications Level: Standard C Library Programming Interface

Domain/OS provides a set of facilities that enable programmers to:

- Create and manage multiple threads within a single process
- Control scheduling of multiple threads within a process
- Maintain data consistency between threads
- Cancel threads
- Associate data with specific threads

This is the interface that most applications should use. It is the most portable interface. It is based on the POSIX P1003.4a/D4 interface, and closely matches the interface provided in HP/OSF1. Since this interface is not yet approved, there may be incompatible changes when the final version is approved.

1.7.1.5.1 Standard C Library Programming Calls

The following programming interfaces are provided:

pthread_attr_create
Creates a thread attributes object

pthread_attr_delete
Deletes a thread attributes object

pthread_attr_getstacksize
Returns the value of the stack size attribute of a thread attributes object

pthread_attr_setstacksize
Sets the value of the stack size attribute of a thread attributes object

pthread_cancel
Initiates termination of a thread

pthread_cleanup_pop
Removes a cleanup routine from the top of the stack of the calling thread

pthread_cleanup_push
Pushes a cleanup routine onto the stack of the calling thread

pthread_cond_broadcast
Wakes up all threads that are waiting on a condition variable

pthread_cond_destroy
Destroys a condition variable

pthread_cond_init
Creates a condition variable

pthread_cond_signal
Wakes up a thread that is waiting on a condition variable

pthread_cond_timedwait
Waits on a condition variable for a specified period of time

pthread_cond_wait
Waits on a condition variable

pthread_condattr_create
Creates a condition variable attributes object

pthread_condattr_delete
Deletes a condition variable attributes object

pthread_create
Creates a thread

pthread_detach
Detaches a thread

pthread_equal
Compares two thread identifiers for equality

pthread_exit
Terminates the calling thread

pthread_getspecific
Returns the value bound to a key

pthread_join
Waits for a thread to terminate

pthread_keycreate
Creates a key to be used with thread-specific data

pthread_mutex_destroy
Deletes a mutex

pthread_mutex_init
Creates a mutex

pthread_mutex_lock
Locks a mutex

pthread_mutex_trylock
Tries once to lock a mutex

pthread_mutex_unlock
Unlocks a mutex

pthread_mutexattr_create
Creates a mutex attributes object

pthread_mutexattr_delete
Deletes a mutex attributes object

pthread_once
Calls an initialization routine

pthread_self
Returns the ID of the calling thread

pthread_setasynccancel
Enables or disables the asynchronous cancelability of the calling thread

pthread_setcancel
Enables or disables the general cancelability of the calling thread

pthread_setspecific
Binds a thread-specific value to a key

pthread_testcancel
Creates a cancellation point in the calling thread

pthread_yield
Allows the scheduler to run another thread instead of the current one

1.7.1.6 Using The New Interfaces

The intermediate level interface is made available by including:

```
#include <apollo/thread.h>
```

This only defines the interface, it does not define the thread safe errno macro, or any of the other things needed for thread safe applications. The application must manage these things for itself, e.g. use mutex locks and make sure not to call any "unsafe" library routines.

The C library level interface is recommended for all applications. It is made available by defining a "feature test macro" at compile time and including a header file in the application (both of these are defined by POSIX):

```
for /com/cc:  -def _POSIX_THREADS_SOURCE
for /bin/cc:  -D_POSIX_THREADS_SOURCE

#include <pthread.h>
```

Using any of these interfaces means that the object will not run on systems prior to SR10.4 since these interfaces are new with SR10.4. Defining the _POSIX_THREADS_SOURCE macro ensures that an attempt to run the object on prior versions will generate a comprehensible error message. Otherwise, the program will run and then abort when it first references one of the new entries.

1.7.1.7 Threads Support in Domain/OS Commands

The following commands have been modified for threads support (the option name was chosen to match HP/OSF1):

- New option to **/com/tb** and **/usr/apollo/bin/tb**:

  -m Trace all threads in the process.

  This applies to active process and diagnostic tracebacks.

- New option to **/bin/ps** and **/com/pst**:

  -m Print thread list for each listed process.

For **/bin/ps** and **/com/pst**, the default is not to list threads information. When the **-m** option is specified, the cpu time for each thread is listed in the TIME field, while the total cpu time is reported in the TIME field of the associated process.

When _THREADSAFE_ERRNO or _POSIX_THREADS_SOURCE is defined, **errno** is defined as a macro rather than a variable, and hence, is not visible to the debugger. In DDE the **errno** value associated with the primary thread can be accessed using the following technique:

```
dde> # Get the address of the errno global variable
dde> target command esa errno
0x3b3d8404
dde> # Dereference address to get value
dde> print *(int*)0x3b3d8404
*: 2
dde> # Define a macro for convenience
dde> define errno *(int*)0x3b3d8404
dde> print errno
*:
```

1.7.1.8 Compatibility Issues And Caveats

Applications that use the new interfaces and define _POSIX_THREADS_SOURCE will get a relatively thread safe environment. That is, **errno** will be defined as a macro (see below) and most **stdio** functions will use locking to make them thread safe. However, as noted above, the object will not run on previous releases. The following sections address issues for existing applications that will not be changed to use the new interfaces.

1.7.1.8.1 CPS Tasking

The CPS tasking package, described in the "Concurrent Programming Support (CPS) Reference" manual, has been re-implemented to use the intermediate level thread interface. This means that tasks can now be truly concurrent because they are now scheduled by the kernel rather than a user-space library. For instance, when one task takes a page fault, another task in the same process could run.

This "true" concurency can create problems if an application uses **pfm_$inhibit/enable** calls to prevent task switching, since the kernel scheduler does not pay attention to this state. The "safe" way to control concurreny between tasks is with mutex locks.

There is also an issue with the global **errno** as described in the next section.

1.7.1.8.2 Errno

As part of the threads implementation, **errno** has changed from a per-process global variable to a macro that calls a function that returns a pointer to a per-thread **errno** value. The old global **errno** is still used by default, as noted below.

For backward compatibility with previous releases (prior to SR10.4), applications should observe the following rules:

- If your application does not use CPS tasking and is not callable by anything that might be using CPS tasking, then you do not need to do anything. Most user applications fall into this category. By default, you will get an object that uses the global **errno** (no **errno** macro), and the current versions of **stdio** functions (not thread safe) and runs on SR10.3 (and earlier) and SR10.4. (For those "in the know": even though NCS uses tasks, and applications may be using NCS without knowing it, they are all right as long as they are not the target of an NCS callback.)

  An application that uses CPS tasking may also fall in this category if it does not set or use **errno**. A tasking application that uses **errno** is currently "unsafe" in small windows where another task could run and change the global **errno** before the first task got a chance to look at it. Thus, most tasking applications may already be safe with respect to **errno**. In this case, no change should be needed.

- If your application explicitly uses CPS tasking or could be called by something that uses tasking, and it sets/uses **errno**, then you need to compile it with the following compile-time option:

  ```
  for /com/cc: -def _THREADSAFE_ERRNO
  for /bin/cc: -D_THREADSAFE_ERRNO
  ```

  You must also make sure that your application is linked with the sr10.4 crt0.o and that your program starts at the correct place in crt0.o. This is the standard behavior for **/bin/cc**. You should also be using compiler version 6.85 or later. Users of **/com/cc** must explictly bind in a crt0.o (just as they have always done).

  Doing this will produce an object that will run on SR10.3 (and earlier) and SR10.4. It will use the thread safe **errno** macro on sr10.4 and the global **errno** value on previous releases. It will have the current versions of the **stdio** function (not thread safe). This should not be a problem if the program's use of **stdio** is currently task safe.

  This assumes you are using SR10.4 header files and compiling on a sr10.4 node. If you compile on an SR10.3 node, you will get an unresolved reference from the linker. By default, this will abort the link and not produce an object. To turn this into a warning and avoid the failure, you need the **-A noallres** option. The message is generated because the linker will not be able to find the function called by the **errno** macro, **unix_static_$errno_ptr**, on the SR10.3 node. This is all right because the SR10.4 crt0.o will fix things at run time so that this reference will be resolved.

- If your application is a global library and it sets/uses **errno** or any **stdio** functions, you need to compile it as if it were using the new interfaces:

  ```
  for /com/cc: -def _POSIX_THREADS_SOURCE
  for /bin/cc: -D_POSIX_THREADS_SOURCE
  ```

  This is because a global library cannot tell what sort of application (e.g. a multitasked application) might call it, so it has to be safe. This means that you will need one version of the global library for SR10.4 and another one (compiled without the macro set) for previous releases. A library can't use the same "trick" as an application because it does not have a crt0.o to make it runnable on SR10.3.

The preceding rules apply to C programs. If your program is written in Pascal and falls into one of the categories that needs changes, you must add the following definitions to your application:

```
procedure unix_static_$set_errno (in errno_value : integer32); extern;
function  unix_static_$get_errno : integer32; extern;
```

and then change:

```
errno := foo;      to  unix_static_$set_errno(foo);
foo := errno;      to  foo := unix_static_$get_errno();
```

This is because Pascal does not have a macro facility to hide this. Also, after doing this, your object will not run on previous releases because these entries are not available, and there is no equivalent to crt0.o to define them at runtime.

Here are some additional points:

- Make sure your application gets **errno** by using "#include <errno.h>", not by defining it itself (i.e. by "extern int errno"). The macro is defined by the header file.
- In the sys5.3 environment, "#include <math.h>" declares **errno** itself, without including <errno.h>. This means that if your program uses **math.h**, it must be included *after* **errno.h** in order to pick up the macro.

1.7.1.8.3 stdio And POSIX Reentrant Functions

POSIX defines a number of reentrant functions and changes in the behavior of **stdio** functions as part of the pthread specification. These new functions and behaviors are needed because the corresponding existing POSIX functions could not be made thread safe without changing their interfaces (e.g. an interface that returns a pointer into per-process static storage).

All of the specified functions are available when _POSIX_THREADS_SOURCE is defined. POSIX also specifies that the availability of the reentrant functions can be controlled separately. In anticipation of the macro name that will be picked by the final standard, we have provided the _POSIX_REENTRANT_FUNCTIONS_SOURCE macro to make just the reentrant functions available, without the other changes that go along with the _POSIX_THREADS_SOURCE macro.

Further, in anticipation of XOPEN, which specifies a number of functions that are not specified by POSIX, and are also not thread safe for the same reasons, we have also provided corresponding reentrant functions. These are separately controlled by the _XOPEN_REENTRANT_FUNCTIONS_SOURCE macro (which also turns on the POSIX reentrant functions). If you have specified _POSIX_THREADS_SOURCE and _XOPEN_SOURCE, then we turn on the XOPEN reentrant functions by default.

The above description is embodied in the **<sys/stdsyms.h>** header file in a more succinct (and perhaps understandable) fashion.

To provide compatibility with HP/OSF1, we have provided a macro to "turn off" the thread safe behavior of **getc** and **putc** that are obtained by default if _POSIX_THREADS_SOURCE is defined. By defining _STDIO_UNLOCK_CHAR_IO before including **<stdio.h>** (which see), the default action is changed to unlocked **putc** and **getc**. A file lock can still be placed around a block of **putc**'s or **getc**'s regardless of the locking mode, and invoking the locked or unlocked version directly always overrides the default action. Our implementation of locked **getc** and **putc** should be fast enough that this macro will not be needed.

Finally, we believe that the current interfaces to the reentrant functions, defined in Draft 4 of the pthreads document, will be changed in the final, approved, document. Therefore, the reentrant functions are only available in a bindable library, **libc_r.a**. This library must be specified for the link step of a compilation either by using the **-lc_r** argument to **/bin/cc** or **/bin/ld**, or by specifying **/usr/lib/libc_r.a** as one of the objects to **/com/bind**. By using a bindable library, user code is insulated from changes in the interfaces since we will continue to support the current interface (which will be bound into current user code).

1.7.1.9 Support for Threads in Domain/OS Debuggers

Domain/OS debuggers, including DDE, handle threads transparently. The underlying Ptrace facility has been extended to accommodate multiple threads.

Debuggers work on one thread at a time. The first thread to hit "breakpoint" is what the debugger processes. Displaying the registers shows the current thread's state.

The "go" command resumes program execution and all threads continue processing; however, the "single-step" command continues the current thread, leaving all other threads suspended.

If you issue an interrupt (Ctrl/C) from the debugger, control is passed to the main thread (the thread that handles signals).

Current Limitation

During the debugging process, if you single-step into a kernel call which blocks, your program may deadlock. You can avoid this situation by watching for kernel calls during the "stepping" process, and using the "go" command instead.

1.7.1.10 Thread-safe Libraries

The following libraries are thread safe:

clib, libc

Provides the UNIX system interface, including all the **stdio** calls (and, by extension, any type managers called by **stdio** calls).

pmlib

Provides Domain process management.

streams

Provides IOS and a number of global type managers.

The following are potential problems with other libraries that may not be thread-safe:

errno Usage

Programs that use CPS Tasking or Pthreads may have problems with libraries that are not thread-safe. A library routine that uses the global **errno** may fail if it is called from any thread but the first one. (The first thread is called the "Distinguished Task" under CPS and the "Initial Thread" under Pthreads.)

This occurs because **errno** is now a macro that expands to code that fetches a thread-private **errno**. The global **errno** belongs only to the first thread. Libraries that are not thread-safe continue to use the global **errno**, no matter which thread calls into them.

However, the multi-threaded program must be compiled with the **errno** macro. Therefore, there is a mismatch between the program and the library as to the actual location of **errno**. One symptom of this problem is that the library routine returns -1, but errno is 0 (or some unexpected value).

The possible remedies are:

- Recompile the library with -D_THREADSAFE_ERRNO.
- Allow only the first thread to call into these libraries.

- Don't use errno.

Signals and Locks

Thread-safe code generally uses locks to protect shared data structures. There are a variety of lock packages in use. Unfortunately, only the traditional Domain **mutex_$xxx** family protects against asynchronous interrupts (usually signals).

The symptom of this problem is deadlock. The trace-back will reveal some thread waiting for a spinlock or recursive mutex lock. There are two scenarios:

- The signal handler deadlocks, because it has re-entered some library routine that has already acquired the lock. This is apparent from the trace-back.
- A lock is found to be locked for no apparent reason. This can happen when a signal handler decides to longjump to a saved stack frame instead of returning to the point of interruption. Presumably, the interrupted code had been holding the lock.

The possible remedies are:

- Use sigpoll to check for signals. This may be done in a dedicated thread.
- Protect interrupt-sensitive routines by blocking signals. This may be done using **sigblock**, or **pfm_$inhibit/pfm_$enable**.
- Use the **mutex_$lock/mutex_$unlock** calls around interrupt sensitive code (or even code that is not thread-safe).
- Eliminate most library calls from signal handlers.
- Don't use longjump from a signal handler unless you are sure no locks can be held.

Global Data

A library may not be thread-safe because it uses global, per-process, data in such a way that multiple threads will interfere with each other when accessing the data.

The symptoms of this problem will be varied and hard to pin down since they involve global data changing "asynchronously" (as different threads read/update it).

The possible remedies are:

- Change the library to use locking around accesses (reads or writes) to the global data.

- Add locking around calls to the unsafe library functions in the application. (Note the caveats about locking in the section above.)

1.7.1.11 Undocumented Calls

The following calls are not included in the current POSIX draft. We do implement these calls, but they are not documented:

getgrent_r
getpwent_r
getutent_r
getutid_r
getutline_r
pututline_r
getpass_r
localeconv_r
nl_langinfo_r
setlocale_r
rand_r

1.7.2 Disk Quota Facility

Domain/OS SR10.4 provides an optional disk quota facility consisting of:

- A new invol **option** (12) that creates a quota table capable of supporting a specified number of users. This number can be increased by a subsequent **invol**, but when this is done all table entries are lost and must be re-entered.

  This **invol** option can be performed on an existing logical volume without disturbing data.

- The following new commands, which allow a system administrator (running as ROOT) to add, change and examine quota entries as well as to enable and disable the quota sub-system:

  edquota
  quotaon
  quotaoff

Under this facility, space occupied by a file is charged to the user that "owns" the file (the file owner, as shown by **/bin/ls -l**). If, at page fault time, allocation of a disk block would push the quota for the owner of the file over the threshold, the operation fails and the program is faulted.

Compatibility

The on-disk disk quota structures are ignored if the disk is mounted under a pre-SR10.4 OS.

NOTES:
When **salvol** is run, the usage information recorded in the disk quota table for a volume is updated to reflect the actual ownership of blocks on the disk.

The contents of the disk quota table are not saved and restored by the standard backup tools (e.g., OMNIBACK). To back up a disk quota table, an administrator can save the contents of the quota table in an ordinary file (e.g., via /etc/edquota -l > <*quota-list-file*>) which *will* be backed up; then, following a restore, he can re-initialize the quota table from the contents of this file.

WARNING:
A pre-SR10.4 **salvol**, if run over a disk configured for disk quotas, will corrupt the quota structures, resulting in unspecified behavior when that disk is mounted and used again under SR10.4.

CAUTION:
When implementing disk quotas on a node, be sure to include the following persons:

admin
bin
daemon
lp
root
sys_person
user
uucp
none

1.7.3 External Pager Daemon

The XPAGER daemon, new at SR10.4, is an external pager. This daemon interacts with the kernel and allows page faults on certain types of mapped objects to be handled via a user-space type manager. Currently, this mechanism is only used for accesses to CD-ROMs.

1.7.4 Overview of NCS 2.0

Domain/OS SR10.4 supports version 2.0 of the Network Computing System (NCS), a remote procedure call (RPC) facility that supports the development and execution of distributed client-server applications in heterogeneous networks.

NCS 2.0 contains the RPC component of version 1.0 of the OSF Distributed Computing Environment (DCE) offering; it also includes compatibility features that allow programs written using the NCS 1.5.1 Application Programming Interface (API) to be built and executed on NCS 2.0 platforms.

With a few exceptions, NCS 2.0 is compatible with all previous NCS implementations released by Apollo or Hewlett-Packard. In particular, existing applications built under NCS 1.5.1 and NCS 1.1 may be run in the NCS 2.0 environment; they need not be modified in any way. (See the section "Interoperating NCS 2.0 With Earlier Releases" for more information.)

1.7.4.1 Product Structure

NCS 2.0 consists of two products: NCS/NCK 2.0 and NCS/NIDL 2.0.

NCS/NCK, the Network Computing Kernel product, provides support for the execution of NCS-based applications. NCS/NCK 2.0 is part of SR10.4 base software and includes:

- The NCS 1.5.1 runtime library (**/lib/ddslib**)
- The NCS 2.0 runtime library (**/lib/dds2lib**)
- The global location broker daemon (**glbd**), which may be used only by programs written with the NCS 1.5.1 API
- The remote procedure call daemon (**rpcd**), which manages both the DCE RPC Endpoint Map and the NCS 1.5.1 Local Location Broker database
- Administrative tools and configuration files

NCS/NIDL, the Network Interface Definition Language product, provides support for the development of NCS-based applications.

The NCS/NIDL 2.0 product is not part of SR10.4 base software; it is a layered product, priced separately. The NCS/NIDL 2.0 product is not supported on releases of Domain/OS prior to SR10.4. On such releases, the NCS/NIDL 1.5.1 product is still supported. For more information on compatibility, see the Release Document for the NCS/NIDL product.

The NCS/NIDL 2.0 product includes stub-generating compilers for versions 1.5.1 and 2.0 of the interface definition language. NCS/NIDL 2.0 is a separate layered software product and includes:

- An improved compiler for NCS/NIDL 1.5.1
- A compiler for IDL, the DCE RPC Interface Definition Language
- A translator to convert interface definitions from NCS/NIDL 1.5.1 to IDL
- Header files and NCS interface definition files

The Application Programming Interfaces (APIs) and the NCS/NIDL programming language have changed substantially in the NCS 2.0 product. For an explanation of how to convert applications written with the NCS 1.5.1 APIs to the NCS 2.0 APIs, see the NCS 2.0 Transition Guide.

1.7.4.2 Features

The key features of the Network Computing System include

- Transparent procedure call invocation, with support for multiple input and output parameters and for function return values
- A comprehensive set of scalar and aggregate data types
- Automatic generation of client and server stub modules from high-level interface definitions
- Support for various execution semantics (e.g. "at-most-once" vs. "idempotent") independent of the underlying network transport
- Transmission from server to client of exceptions that occur during the execution of a remote procedure call
- Forwarding from client to server of call-cancellation requests

The following new features are available to applications written using the NCS 2.0 API.

- Full support for pointer data types
- Improved bulk data transfer
- Unlimited RPC argument size
- The ability to maintain state across remote procedure calls

1.7.4.3 Programming Issues at NCS 2.0

This section of the Release Notes describes significant changes to the application programming environment for software developed using the NCS 2.0 API.

1.7.4.3.1 Running NCS 1.5.1 Applications Under NCS 2.0

Programs written using the NCS 1.5.1 API will continue to run under NCS 2.0, and will continue to have access to the NCS 1.5.1 location services. You do not need to modify your NCS 1.5.1 applications to make them run in the NCS 2.0 environment.

With a few exceptions, the NCS 2.0 product is compatible with all NCS implementations released by Apollo or Hewlett-Packard. In particular, existing applications built with NCS 1.1 and NCS 1.5.1 may be run in the NCS environment; they need not be modified in any way. See the section "Interoperating NCS 2.0 With Earlier Releases" for more information.

NCS 2.0 includes software and documentation to assist developers in converting applications from the NCS 1.5.1 API to the NCS 2.0 API. Programs that use the NCS 1.5.1 API and depend on the GLB for location services cannot be converted to the NCS 2.0 API, since the NCS 2.0 API does not provide global location services.

1.7.4.3.2 Naming and Authentication Issues at NCS 2.0

When integrated into a complete OSF DCE environment, NCS 2.0 offers access to the naming service (**rpc_ns_*** routines) provided by the DCE Cell Directory Service (CDS) and to the authentication service (**rpc_*auth*** routines) provided by the Security component of the DCE. On Domain/OS SR10.4, the DCE CDS and Security components are not currently available, and calls to the **rpc_ns_*** and **rpc_*auth*** routines will not work.

The NCS 2.0 product does not provide any global location or naming services for applications written using the NCS 2.0 API. Such applications can use the local location service provided by the endpoint map, but must use string bindings to specify the host on which a server is running.

1.7.4.3.3 Supported Network Protocol Families

NCS 2.0 can support remote procedure calls over UDP/IP, TCP/IP, and DDS protocols. On a particular operating system, NCS 2.0 typically supports a subset of these transport protocols. On Domain/OS SR10.4, the supported protocols are UDP/IP and DDS.

1.7.4.4 System Administration Issues at NCS 2.0

This section summarizes some significant changes to system administration at this release of NCS. For full details, see *Managing NCS Software.*

1.7.4.4.1 New Daemon: rpcd replaces llbd

At NCS 2.0, NCS location service administration has changed. Instead of running the **llbd** daemon to manage the local location broker (LLB) database, you run the **rpcd** daemon. The **rpcd** daemon provides both the LLB functionality to NCS 1.5.1 API applications and the endpoint-mapping functionality to NCS 2.0 API applications. To start **rpcd** automatically at system boot, create the file **/etc/daemons/rpcd.**

To ease the system administration transition from NCS 1.5.1 to NCS 2.0 on Domain/OS, **/etc/ncs/llbd** is installed as a link to **/etc/ncs/rpcd**, and the operating system will start **/etc/ncs/rpcd** at boot time if either **/etc/daemons/llbd** or **/etc/daemons/rpcd** exists. Therefore, unless you have greatly altered the Domain/OS startup scripts, you need do nothing to ensure that a host that ran **llbd** on SR10.3 runs **rpcd** on SR10.4.

Only the root user may start the **rpcd** daemon from a shell. In earlier releases of NCS, any user could start **llbd**; however, if started by a non-root user, **llbd** would only listen on the dds protocol. On hosts running NCS 2.0, non-root users who start **rpcd** from a shell will receive an error message, and **rpcd** will not be started.

Unlike **llbd, rpcd** has no **-listen** option; instead, you can specify the protocol family as the last argument on the command line, as in **rpcd dds**

See the revised version of *Managing NCS Software* (included with this release) for further details.

1.7.4.4.2 Changes to the glbd Database

The **glbd** database format and database filenames have changed at NCS 2.0. When the NCS 2.0 **glbd** comes up, it searches for an NCS 1.5.1 database. If one is found, **glbd** automatically reformats the database and saves it in the new location, saving the NCS 1.5.1 database files under the names **/sys/node_data.glb.e.bak** and **/sys/node_data/glb.p.bak**. You may delete these backup files after you are confident that NCS 2.0 is running successfully.

glbd filenames at NCS 1.5.1	**/sys/node_data/glb.e**	**/sys/node_data/glb.p**
glbd filenames at NCS 2.0	**/sys/node_data/glb.d**	**/sys/node_data/glb.r**

1.7.4.4.3 Changes to Default Protocol Used by glbd

The default protocol family used by the **glbd** daemon for GLB database replication has changed. Before NCS 2.0, the default protocol family was dds; at NCS 2.0, the default protocol family is ip.

As a result of this change, the default interpretation of the **glbd** switch **-from** has also changed. If you are administering a site whose hosts run several different versions of NCS, you should avoid ambiguity by explicitly specifying the protocol family when using the **-from** option, as in glbd -create -from ip:martha

Another consequence of the new default protocol family is that the **glbd** command **glbd -create -first** now puts the ip address of the **glbd** into the replica list; **glbd -create -first** will work *only* if the host on which the command is issued is running the **tcpd** daemon.

To override the default protocol family, you should use the **glbd** switch **-family**. For instance, glbd -create -first -family dds will use dds as the default protocol family for replication, and will identify the new replica in the replica list by its dds address.

1.7.4.4.4 glbd Runs in Background by Default

The **glbd** daemon now automatically starts up in background; if you prefer to run **glbd** in foreground, use the **-foreground** or **-D** switches. See the **glbd** man page for full details.

1.7.4.5 Interoperating NCS 2.0 with Earlier Releases

In general, NCS 2.0 is compatible with applications written under earlier versions of NCS; in particular, applications built under NCS 1.5.1 and NCS 1.1 can run unchanged in an NCS 2.0 environment. System administrators and software developers responsible for a network running several releases of NCS should be aware of the following NCS compatibility issues.

- Programs built with the version of NCS/NIDL 1.0 released with Domain/OS SR9.5, SR9.6, and SR9.7 may not interoperate with programs built with later versions of NCS/NIDL. If you use NCS/NIDL 1.0, you should use the patched version that was released in November 1987.

- Hosts running versions of NCS prior to 1.5.1 should not share a Location Broker cell with hosts running NCS 2.0, because pre-NCS-1.5.1 versions of the **glbd** daemon and the **drm_admin** command cannot interoperate with their NCS 2.0 counterparts. In particular, the following releases of NCS have **glbd** and **drm_admin** components that cannot interoperate with NCS 2.0:

  - NCS as shipped on any version of Domain/OS at or before SR10.1
  - NCS 1.0 or 1.1 on any platform

  All other HP or Apollo releases of NCS can share a Location Broker cell with hosts running NCS 2.0. As in previous releases, the non-replicatable GLB daemon (**nrglbd**) that is provided for some platforms cannot interoperate with any version of the replicatable GLB daemon (**glbd**). In any network or internet containing a host that can run **glbd**, we recommend that **glbd** be used to provide global location service. See *Managing NCS Software* for more information.

- Stubs built by the IDL compiler released as part of the HP OSF/1 Technology Release cannot be used on a Domain/OS SR10.4 platform, because the two releases have different versions of the DCE header files.

- Programs based on stubs built by the HP OSF/1 TR IDL compiler may not interoperate with programs built by the SR10.4 Domain/OS NCK/NIDL compiler. (Hewlett-Packard will release a patch for HP OSF/1 TR to fix both of these last two problems.)

1.7.4.6 Pathnames for NCS 2.0 Files

This section lists the SR10.4 pathnames for NCS 2.0 files.

Daemons and utilities	**/etc/ncs/glbd** **/etc/ncs/rpcd** **/etc/ncs/llbd** (link to **rpcd**) **/etc/ncs/lb_admin** **/etc/ncs/drm_admin** **/etc/ncs/rpccp** **/etc/ncs/uuid_gen** **/etc/ncs/uuidgen**
GLB configuration files	**/etc/ncs/glb_obj.txt** **/etc/ncs/glb_site.txt**
UUID-to-name mapping file	**/etc/ncs/uuidname.txt**
LLB database file	**/tmp/llbdbase.dat**

Endpoint map database file	**/tmp/rpcdep.dat**
GLB database file	**/sys/node_data/glb.d**
GLB replica list and propagation queue	**/sys/node_data/glb.r**
GLB log file	**/sys/node_data/system_logs/glb_log**
NCS 1.5.1 system IDL directory	**/usr/include/idl**
NCS 1.5.1 system IDL files	**/usr/include/idl/*.idl**
NCS 1.5.1 system header files	**/usr/include/idl/c/*.h**
NCS 2.0 system IDL directory	**/usr/include/dce**
NCS 2.0 system IDL files	**/usr/include/dce/*.idl**
NCS 2.0 system ACF files	**/usr/include/dce/*.acf**
NCS 2.0 system C header files	**/usr/include/dce/*.h**
Header files related to threads, exception handling, and fault handling	**Various *.h files in /usr/include**
NCS 1.5.1 Runtime	**/lib/ddslib**
NCS 2.0 Runtime	**/lib/dds2lib**

1.7.4.7 NCS 2.0 Documentation

NCS system administration for both NCS 1.5.1 and NCS 2.0 platforms are described in the following manual:

- *Managing NCS Software* (D-11895-E)
  describes how to administer location services that support NCS-based applications. This manual is included in the 10.4 base documentation kit.

The following manuals are shipped with NCS/NIDL 2.0, a separately-priced product:

- *NCS 2.0 Programmer's Guide* D-19572-E
- *NCS 2.0 Programmer's Reference* D-19563-E
- *NCS 2.0 Transition Guide* D-19564-E

The following manuals, which are not shipped with the NCS 2.0 product, document the Application Programming Interfaces (APIs) and other features of NCS 1.5.1:

- *Network Computing System Reference Manual* D-10200-C
  This book is a reference manual for programmers developing applications based on NCS 1.5.1.
- *Network Computing System Tutorial* D-18355-B
  This book provides a step-by-step approach to programming with Version 1.5.1 of NCS to create heterogeneous distributed computing applications.

The following manual, not shipped with NCS 2.0, gives an overview of the design philosophy of NCS and gives detailed protocol specifications for the architecture on which NCS 1.5.1 is based:

- *Network Computing Architecture* D-10201-B

1.7.4.7.1 Additions or Changes to Documentation

The new edition of *Managing NCS Software* includes information on managing both NCS 1.5.1 and NCS 2.0; it thus supersedes the previous edition.

The other NCS 2.0 manuals document only the NCS 2.0 programming interfaces and commands; they do not cover the NCS 1.5.1 programming interfaces and commands. Users who are developing and maintaining NCS 1.5.1 software should continue to use the *Network Computing System Reference Manual* and *Network Computing System Tutorial* as references.

1.7.5 TCP/IP Enhancements

Enhancements to TCP/IP at SR10.4 include the following:

- Changes to the name server, including two new tools for the network administrator — **/etc/named-xfer**, a **named** database-download utility, and **nslookup**, a **named** database-query utility distributed in the **/domain_examples** directory.
- Changes to the **/etc/rc.local** startup file
- New guidelines for using **/etc/resolv.conf**
- Tightened security and a new configuration procedure for **tftpd**
- Tightened security for **ftpd**
- New guidelines for thread-safety
- BSD4.3 TCP enhancements incorporated into Domain/OS TCP

- More rigorous parameter-checking in the setsockopt() system call
- TCP/IP-related copyright information

Each enhancement is described briefly below, along with pointers to more complete documentation. In addition, both Domain/OS TCP/IP manuals appear in new editions at SR10.4: *Configuring and Managing TCP/IP* (008543-A03) and *Using TCP/IP Network Applications* (008667-A01).

1.7.5.1 Changes to the Name Server

The BSD name server program provides a mechanism for translating host names into addresses. It is designed to handle address translation in large internets.

SR10.4 provides a new revision of the name server program (BIND 4.8.3) that features two tools for working with the name server database files and a new dependency on the **lo0** interface (localhost).

These new features are described below.

1.7.5.1.1 The Database-Download Tool /etc/named-xfer

The SR10.4 version of BIND includes the **named** database-download utility **/etc/named-xfer**. This utility allows the network administrator to manually control the transfer of naming information between primary and secondary servers in a particular zone of authority. For a complete description of **/etc/named-xfer**, see Chapter 4 of *Configuring and Managing TCP/IP* and the **named-xfer**(8C) manual page.

1.7.5.1.2 The Database-Query Tool nslookup Available in /domain_examples

SR10.4 introduces **nslookup**, a **named** database-query tool. This unsupported tool is distributed only in the **/domain_examples/tcp/nslookup** directory.

You use **nslookup** to query Internet domain name servers. **nslookup** has two modes: interactive and non-interactive. Interactive mode allows you to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode allows you only to print the name and requested information for a host or domain.

At SR10.4, we supply the sources to the **nslookup** program in the **/domain_examples/tcp/nslookup** directory. You can read the code and accompanying documentation in that directory as an example of how we ported **nslookup** to Domain/OS. Note that we are not committing to full support of **nslookup** at this time.

For more information about using the **nslookup** sources, see Appendix G of *Configuring and Managing TCP/IP*.

1.7.5.1.3 Dependency on lo0 (localhost)

The SR10.4 version of **named** has a dependency on **lo0** (localhost) and will fail to run if that interface is not available. On the same topic, see the following note about the change to the **/etc/rc.local** startup file.

1.7.5.2 Changes to the /etc/rc.local Startup File

There are a number of changes to the **/etc/rc.local** startup file at SR10.4. These include:

- New dependency on **lo0**
- New startup of NFS Services
- New default method of name-address resolution for **/etc/nmconfig**

Each change is decribed below. For more information about the **rc.local** file and how to edit it, see Chapter 3 and Appendix A of *Configuring and Managing TCP/IP.*

1.7.5.2.1 New Dependency on lo0

Because of a requirement in the new version of **named**, the following line in the **/etc/rc.local** startup file is unconditionally uncommented:

```
/etc/ifconfig lo0 127.0.0.1
```

1.7.5.2.2 New Startup of NFS Services

The **/etc/rc.local** startup file now runs **/etc/rc.nfs** (if it exists). This file performs NFS initialization and starts the various NFS daemons (for which entries must exist in the **/etc/daemons** directory). For more information, see *Using NFS on the Domain Network.*

1.7.5.2.3 New Default Method of Name-Address Resolution for /etc/nmconfig

The **/etc/nmconfig** utility is a configuration tool that allows the network administrator to specify whether to use **named** or **/etc/hosts** for Internet name-address resolution. When **/etc/nmconfig** is invoked in the **rc.local** file, the new default method of name-address resolution is **named**, but only if the file **/etc/daemons/nmconfig** exists. Alternatively, if **/etc/hosts** is the preferred method of name-address resolution, the network administrator should make sure that the file **/etc/daemons/nmconfig** does *not* exist. For more information on **/etc/nmconfig**, see Chapter 4 of *Configuring and Managing TCP/IP.*

1.7.5.3 New Guidelines for Using /etc/resolv.conf

The resolver configuration file, **/etc/resolv.conf**, is now required if your network uses **named** for name-address resolution, whether or not the name server daemon is running on the local node. The **/etc/resolv.conf** file must reside on any node running **named** locally, but only the first line — the **domain** keyword followed by the domain name — is required. If the node is not running **named** locally, the file is a link to **/etc/resolv.conf** on the TCP/IP administrative node and must contain both the **domain** keyword followed by the domain name, and the **nameserver** keyword followed by the IP address of the remote name server that will answer the local node's name queries. For more information, see Chapter 3 of *Configuring and Managing TCP/IP*.

1.7.5.4 Tightened Security and a New Configuration Procedure for tftpd

Security restrictions for reading files through the Domain/OS **tftpd** server have been tightened. See the manual pages for **tftp** and **tftpd**, and *Using TCP/IP Network Applications* for complete details.

Configuration of the **tftpd** server has also changed. See the **tftpd** manual page for complete details.

1.7.5.5 Tightened Security for ftpd

The Domain/OS **ftp** server, **ftpd**, will no longer accept logins to an account with a null password. This change is described in the **ftpd** manual page.

1.7.5.6 New Guidelines for Thread Safety

A multi-threaded application using sockets (TCP/UDP) will function correctly only if multiple threads do not simultaneously perform operations on the same socket.

The only exception to this rule is in the case of UDP sockets. Multiple threads can simultaneously perform READ operations on the same UDP socket. Also for UDP sockets, if a connect call has not been done (common practice is not to connect UDP sockets), then multiple threads can simultaneously perform WRITE operations (sendto) on the same socket. However, we recommend that multiple threads not perform simultaneous operation on the same socket.

The name resolver library, **libresolv**, is not backward-compatible, which means that the SR10.4 **libresolv** cannot be run on a SR10.3 or earlier system. **libresolv** continues to pass information using a static area. Thus, **libresolv** is not safe for concurrent use by multiple threads of the same process. The **gethostbyname** manual page gives details about parameter passing to the **libresolv** calls.

1.7.5.7 BSD4.3 TCP Enhancements Incorporated into Domain/OS TCP

A number of BSD4.3 TCP enhancements and improvements have been incorporated into Domain/OS TCP at SR10.4. These changes make TCP more robust and efficient, improving the perceived performance over wide area networks and long delay lines. A better estimate for the retransmit timer is obtained through the use of the mean and the mean deviation of observed round trip times. Congestion control and avoidance is done by changing window sizes based on the arrival of acknowledgements. Small packets are avoided by delaying acknowledgements.

Below is a list of some of these new features:

- Slow start (the number of outstanding unacknowledged packets is slowly increased)
- Dynamic transmission window resizing on congestion and reducing the window size on lost packets for congestion avoidance

- Retransmit timer estimate based on Round Trip Time (RTT) mean and mean deviation
- Exponential retransmit backoff, and Karn's clamped retransmit backoff algorithm
- Fast retransmit on receiving duplicate ACKs
- Better receiver delayed ACK policy
- Nagle's small packet avoidance
- Sender and receiver silly window avoidance

1.7.5.8 More Rigorous Parameter-Checking in the setsockopt() System Call

At SR10.3 and later, the setsockopt() system call performs more rigorous checking of parameters passed. As documented in the manual page, the "optlen" for most socket-level options needs to be at least sizeof(int).

1.7.5.9 TCP/IP-Related Copyright Information

Domain/OS TCP network applications are based on code ported from several sources.

Many of the network applications are based on the BSD4.3 release and carry the following copyright notice:

```
/*
 * Copyright (c) 1983 Regents of the University of California.
 * All rights reserved.  The Berkeley software License Agreement
 * specifies the terms and conditions for redistribution.
 */
```

Other instances of new TCP/IP-related copyright information are described below.

1.7.5.9.1 Name Server Copyright

The name server code and associated utilities are based on the BIND 4.8.3 release and carry the following copyright notice:

Copyright (c) 1985,1989 Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms are permitted provided that: (1) source distributions retain this entire copyright notice and comment, and (2) distributions including binaries display the following acknowledgement: "This product includes software developed by the University of California, Berkeley and its contributors" in the documentation or other materials provided with the distribution and in all advertising materials mentioning features or use of this software. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR

IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

1.7.5.9.2 SNMP Copyright Information

The simple network management protocol server and associated utilities are based on the CMU SNMP 1.0 release and carry the following copyright notice:

/**

Copyright 1988 by Carnegie Mellon University

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

****

1.7.5.9.3 gated Copyright Information

The **gated** server and utility code in **/domain_examples** is based on the Cornell GateD 2.0 release and carries the following copyright notice:

/**

GateD, Release 2

Copyright (c) 1990 by Cornell University
All rights reserved.

Royalty-free licenses to redistribute GateD Release 2 in whole or in part may be obtained by writing to:

Center for Theory and Simulation in Science and Engineering
Cornell University
Ithaca, NY 14853-5201.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

GateD is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing Protocol. Development of Release 2 has been supported by the National Science Foundation.

The following acknowledgements and thanks apply:

Mark Fedor (fedor@psi.com) for the development and maintenance up to release 1.3.1 and his continuing advice.

****

Portions of this software may fall under the following copyrights:

Copyright (c) 1988 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
**/

1.7.5.9.4 TFTP Copyright Information

The trivial file transfer protocol program and server are based on MIT Project Athena code and carry the following copyright notice:

/* Copyright 1984,1985 Massachusetts Institute of Technology

Permission to use, copy, modify, and distribute this program for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies and supporting documentation, the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that copying and distribution is by permission of M.I.T. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty. */

1.7.6 New Version of SysV lint

At Software Release 10.4 of Domain/OS, the SysV environment provides a new version of the **lint** utility that supports ANSI C programs, Domain/C extensions, and SR10.3 header files. The behavior of BSD **lint** remains unchanged.

The SR10.4 version of SysV **lint** supports two new options, **-s** and **-A**, and two new preprocessor symbols. The new options are as follows:

-s

Makes stricter checks about pointer and structure alignments that can prevent portability. Complains about a cast that converts a pointer from a less restrictive alignment to a more restrictive alignment. Complains about a structure member that is not naturally aligned.

-Amode

Specifies the compilation standard to be used by lint. The mode can be one of the following:

xansi Extended ANSI mode. Uses the ANSI C preprocessor; issues warnings for function calls not in the scope of a function prototype; allows Domain extensions; does not define the preprocessor symbol **lint**. This mode is the default.

ansi Strict ANSI mode. Uses the ANSI C preprocessor; issues warnings for function calls not in the scope of a function prototype; does not allow Domain extensions; does not define the preprocessor symbol **lint**.

nansi Non-ANSI mode. Uses the K&R C preprocessor; does not issue warnings for function calls not in the scope of a function prototype; allows Domain extensions; defines the preprocessor symbol **lint**.

Two new preprocessor symbols, **__lint** and **__LINT__**, are defined to allow certain questionable code to be altered or removed for lint. In addition, the pre-processor symbol lint is defined if **-A nansi** is specified.

1.7.7 Streams Type Manager

A streams type manager for compressed files is available in Domain/OS SR10.4. It allows files generated by the BSD **compress** command to be automatically uncompressed when the object is opened. To use it, execute the **obty** command to set the type of compressed files to "compress". The type manager is limited to read-only access.

1.7.8 New Registry Server

The new registry server (**rgyd** Version 1.3) supplied with SR10.4 is incompatible with older versions of the **rgyd** (Version 1.2 and earlier). The SR10.4 **rgyd** corrects a problem with previous versions of **rgyd** in which **rgyd** corrupted the registry database for an account when its group or org affiliation was changed. This corruption disabled users with corrupt accounts from changing their own passwords.

To correct this problem, we recommend that you replace all previous versions of **rgyd** with the SR10.4 **rgyd** version. To do this, replace **/etc/rgyd** at all registry sites with the SR10.4 version, then bring up the master registry, followed by all the slaves. Bringing up the new **rgyd** as the master will force-fix the registry database so that any corrupted accounts will work again.

The SR10.4 **rgyd** will work on any node running SR10.2 or greater. If by accident you mix the SR10.4 **rgyd** with older versions running in your network, account data will not be propagated between the two versions.

Step-by-step instructions:

1. Make sure you have a backup copy of your master registry and the old versions of **rgyd**.
2. Kill all **rgyd**s currently running in your network.
3. Replace **/etc/rgyd** on the SR10.2 or SR10.3 registry sites with the SR10.4 **/etc/rgyd**.
4. Bring up **/etc/rgyd** at the master registry site.
5. Bring up **/etc/rgyd** at all slave sites.

1.7.9 Hardcopy enhancements

At SR10.4, a number of enhancements have been made in Domain/OS hardcopy capabilities. These are described in the following sections.

1.7.9.1 -check option

Beginning with this release, the **prflib -check** option is now the default mode when queuing print jobs. This means that **prflib** will require that the printer name be a vaild SR10.x printer name rather than an alias or SR9.7 printer name. If you are queuing jobs to an Sr9.7 printer, you must add the **-pre10** switch to the list of **prflib** or **prf** options.

Networks consisting of only SR10.x printers may be impacted if print scripts exist which have been queueing jobs to SR10.x printer alias names and relying on the pre10q daemon to requeue these jobs. In this case you should locate and modify those scripts, either by changing the printer name or adding pre10q to the command line.

With this change, if you specify an incorrect printer name the error message

```
"?(prf) Problem with option "pr" - specified printer does not exist (US/print utility)
```

will appear, and the job will not be queued.

1.7.9.2 Print System Cache

A print cache has been added to improve communications throughput and aid in requeuing print jobs . The cache consists of 3 files :

/sys/node_data/print.g	glb entries for print managers and/or print servers
/sys/node_data/print.j	a list of jobs
/sys/node_data/tmp/print.p	a list of printer names and their print managers.

The performance gains will depend on the number of print server and print managers on the network - the larger these numbers, the greater the performance gain. However, if a printer or print manager is moved, a one time performance hit will occur. This is due to an incorrect glb entry in the cache, which the rpc code must time out on. Once this event occurs, the offending entry is removed from the cache, and a new entry retrieved from the glb.

1.7.9.3 Native Language Support

The symbol set encodings for the ascii to PostScript filter have been moved to a set of external files. The external symbol set file name is

/sys/hardcopy/filters/sym_set.ps/*SS_name*

where *SS_name* = the symbol set name set by the **prf** option **-sym[bol_set]** or by the **set_locale** variable. Predefined symbol sets are :

Latin-1

Users can modify these symbol set encodings or add new ones. The README file in the above-named directory explains how to modify or add new symbol set encodings.

There is also a new **prf** command option for the ascii to PostScript filter :

-symbol_set *symbol_set_name*

The default symbol set is **Latin-1**.

The command line setting takes precedence over the **LANG** environment variable.

1.7.9.4 X Bitmap Support

X bitmaps generated on Domain workstations can now be printed on the Tektronix 4693DX printer and PostScript printers. The supported formats include

- X window dump XYPIXMAP format
- X window dump ZPIXMAP format
- X bitmaps

  These bitmaps can be printed using any of the command line switches you would ordinarily use for bitmaps, including magnification, orientation, margins and black/white reversal.

1.7.9.5 New Spinwriter Point Sizes

The spinwriter driver now accepts the following point sizes, line spacing and paper sizes:

paper_width	5.5 to 13.2 inches
paper_length	5.5 to 13.2 inches
point size	7 point to 12 point
lpi	6 and 8 lines per inch.

Note that if these parameters are not currently specified in your print server configuration file they must be added. Select values that match the paper size and point line spacing settings on your spinwriter.

1.7.9.6 Cancel Active Print Job with prf -cancel

You can now cancel an active print job via the **prf -cancel** command. Previously you had to use the **prf -sig_printer** option to cancel an active print job.

1.7.9.7 Pitch Option Added to Print Server Configuration Driver

The pitch option has been added to the print server configuration driver. Use this option to select a character spacing other than the default character spacing for the Genicom printer or user written drivers that support pitch.

1.7.9.8 Startup Page Suppression Option Added to Print Server Configuration File

A startup page suppression option has been added to the print server configuration file. If you specify

```
startup_page off
```

the print server startup page will not be printed at the printer. Otherwise, the startup page will be printed.

1.7.9.9 Hooks added to Allow Adding an Audit Filter to the Filter Chain

Hooks have been added to the print server to allow programmers to add an audit filter into the filter chain. The audit filter is called after the filters but before the driver render routine. The audit filter format is identical to that of any other print server filter.

To invoke the audit filter, add the key word **audit_filter** followed by the name of the filter to your print server config file. Note that the audit filter must reside in **/sys/hardcopy/filters**. A sample audit filter coded in c is provided in **/domain_examples/hardcopy/filter** . This sample audit filter recovers the user name and prints the number of pages in a text file.

1.7.9.10 Print Server, Print Manager, and pre10q Daemon

The print server, print manager and pre10q daemon now name themselves as **prsvr:printer_name** , **prmgr:prmgr_name** and **pre10q**. In addition a command line option **-n** has been added that overrides the default name.

1.7.9.11 Driver Added to Requeue prf Print Jobs to a BSD Printer

A driver, which will requeue **prf** print jobs to a BSD printer, has been added to the base set of drivers. This driver exports all the standard BSD **lpr** command line options to the **prf** command with the following modifications :

BSD Option	prf Command Line Option
-p	-upr
-c	-cif
-P	-lpr lpr_printer_name
-C	-cl banner_page_class_name
-J	-j burst_page_job_name
-T	-ti pr(1) title_name

Note that the option and its value on the **prf** command command line must be separated by a space. Also, single letter options must include a value or be followed by another option .

By default the driver uses its Aegis print name as the **lpr** printer name unless **-lpr** is specified on the **prf** command line. The **-C**, **-J**, and **-T** options control the names printed on the burst page. If these are not specified on the **prf** command line the defaults are the user name and spool file leaf name.

To use this driver, set up your BSD print system. Be sure that the node running the Aegis print server includes a **printcap** file or link to one and has the BSD software installed. Set the **prsvr** configuration file **device_driver** parameter to **lpr**.

1.7.9.12 Dumb Driver Added to Driver Set

A dumb driver has been added to the base set of drivers. This driver will format text using the **prf** command line options and send it to the specified output port. There are no limitions on any of the text parameters, nor does the driver emit any escape codes to control the printer. This means that you must send text jobs using the default settings to which your printer is currently set.

The dumb driver also supports single plane gpr bitmaps. It will emit the scan lines as a byte stream with no format or control characters. This driver also accepts transparent jobs. The implications here are you have connectivity to any device by just setting the correct I/O parameters (**sio,pio** or **ikon**).

To use this driver, set the **prsvr** configuration file **device_driver** parameter to **dumb.** The source code for this driver is located in **/domain_examples/hardcopy/driver/driver_*.**

1.7.9.13 Example Driver and Filters Updated

The example driver and filters have been updated in **domain_examples/hardcopy**. The sample filter is the source code for the ascii formatting filter, which is part of the standard set of filters. In addition, the C code sample driver now uses the .h header files, including the new **prsvr.h** header file.

The placeholders for the FORTRAN examples have been removed. There are no plans to supply FORTRAN example programs. The FORTRAN header file will continue to be supported.

1.7.9.14 New Streams Interface Mode

A new interface mode, **streams**, has been added to support any device or file that can be opened via Aegis or UNIX streams calls. This interface mode informs the driver to open the port without setting any control parameters. Use this mode for the series 400 parallel port, to direct print server output to a file, or to disable the automatic setting of sio line parameters.

1.7.9.15 Option Added to Versatec Driver

An option was added to the Versatec driver at SR10.3 to enable a user to set the state of the plot/print control line. This line is used when sending files in transparent mode to a versatec plotter or rasterizer. The line's state may have to be changed from the default (**print**) for some device/file format combinations. To use this option specify

```
-vers_pp -plot
    or
-vers_pp -print
```

on the **prf** command line.

1.7.9.16 PostScript Error Messages to Print Server Transcript Pad

The code to write PostScript error messages to print server transcript pad can be accessed by entering the sio device name in the **monitor_device_name** field in the print server configuration file; for example:

```
monitor_device_name /dev/sio1
```

1.7.9.17 'Country' Option for Genicom Driver

The Genicom driver includes an extensible option named 'country', which informs the printer which character set to use for ascii text. The **prf** syntax is:

prf -pr *ge_printer_name* **-country** *i ...*

where *i* is an integer that specifies the country code as defined in the programming section of the Genicom users manual. In most manuals these are defined as National Character Set Codes. Note that these codes are different from the country codes as they appear on the printer configuration menu.

The code '99' tells the driver to not change the default front panel country code setting .

You can also specify the country code in the print server configuration file as

ext_opt country *i*

where *i* is defined as above.

1.7.9.18 New Filter to Convert Pixel Bitmaps to Single Grey Plane Image

A new filter has been added that converts pixel bitmaps of 1 to 24 bits per pixel to a single plane grey scale image. This filter is now the default filter used by the **postsc** and **imagen** drivers to process multiplane bitmaps. To use this filter with other drivers, or to specify a different filter chain for the **postsc** or **imagen** driver, use the **-filter** option on the **prf** command line.

1.7.9.19 prflib Creates Case Correct Names

prflib now creates case correct names in the spool directories.

If your spool directory is on an SR9.7 node, the printer name must contain all lower case letters.

1.7.10 Changes to Commands

1.7.10.1 New Sendmail

The new Sendmail shipped with SR10.4 is sendmail-5.65c+IDA-1.4.4.

In addition to numerous bug fixes, the original IDA enhancements, plus the UIUC/NIU and contributed changes, provide the following:

- support for Dbm(3) files - **dbm**, **ndbm**, **sdbm**, **mdbm**, and **gdbm**
  - allows pathalias database to be directly used
  - allows choice of mailer to be table driven
  - allows UUCP and domain name aliasing
- improved support of MX records
- split header rewriting between envelope and headers
- improved test mode
- support for multi-token matches in .cf macros and classes
- batched SMTP support
- allow set (class) declarations to use programs as well as files to define a set
- delayed macro evaluation using $&x syntax
- RFC822 quoted macro expansion using $!x syntax
- an excellent general purpose m4 template (in **/domain_examples**) for .cf file generation
  - supports pure UUCP site requirements
  - supports pure Internet site requirements
  - supports a hybrid of UUCP site and Internet site
  - supports hidden and isolated local area networks connected via a gateway (either UUCP or IP) to the Internet
  - supports genericfrom database mapping actual user names to generic user names.
  - supports pathalias database
  - support UUCP and domain name aliasing

For nodes configured with Internet Domain Name Server (See *Configuring and Managing TCP/IP* (008543-A00) for information on the Internet Domain Name Server) some minor changes to your configuration file may be required. Specifically, single domain names may need to be changed to fully qualified domain names due to the way the new

Sendmail handles the resolver's **RES_DEFNAMES**. For example, in some of our internal configuration files, we replaced *<nodename>* (our mail relay host) with *<nodename>***.ch.apollo.hp.com.**

If the Internet Domain Name Server is not used, configuration files do not require any modifications.

Once the new Sendmail is installed, the configuration file may be "frozen". If the file **/usr/lib/sendmail.fc** exists on your node, either it must be deleted or the configuration file must be "frozen", regardless of whether or not the file has been modified. If **/usr/lib/sendmail.fc** doesn't exist on your node, freezing the configuration file is optional. To freeze the configuration file use the following command:

```
/usr/lib/sendmail -bz
```

NOTE: You cannot do this on a DN10000. the SR10.4.p version of **sendmail** does not support freezing the configuration file.

1.7.10.1.1 Sendmail -N option obsolete

The **sendmail** option **-N**, which was used to supply a "home network name", is now obsolete. Remove it from any command-line options and/or from configuration file(s) or it will prevent sendmail from operating properly.

1.7.10.1.2 sendmail.cf Files No Longer Supplied

At SR10.4, the following files are no longer supplied:

```
/sys5.3/usr/lib/sendmail.cf
/bsd4.3/usr/lib/sendmail.cf
```

1.7.10.1.3 Important Installation Note

Before you install 10.4, preserve any **/usr/lib/sendmail.cf** file that you might already have by adding it to your preserve.list.

1.7.10.2 Change to /bsd4.3/bin/date

Prior to SR10.4, if **/bsd4.3/bin/date** was used to change the clock on a node running **timed**, the time would also be changed on all other machines running **timed** on that local area network. The **-n** option overrode this behavior, changing the time only on the local node.

At SR10.4, you must specify a new option, **-N**, to change the time over a local area network. Behavior of the **-n** option is unchanged, and is now the default.

1.7.10.3 Option Added to /com/pst

The following option has been added to **pst** for SR10.4:

-nz When used with **-r**, displays only processes that have used a non-zero amount of CPU time during the sample period.

1.7.10.4 Change to /com/tb

In previous releases, the **tb** command was owned by root and was **setuid**. This allowed users to get tracebacks of any active processes on the node. For security reasons, we have changed **tb** so that only node owners (users with "w" access to /sys/node_data/etc/node_owners) can do this. All users can still traceback their own processes.

1.7.10.5 Changes to /com/crp

A new option has been added to the **crp** command:

-full Enable local support of the pad_$def_pfk and pad_$dm_cmd calls when issued by a remotely executing program or by the local user. Default is to remove support of these commands.

NOTE: Use of this option should be restricted to those times when the remotely executing program(s) require(s) access to these functions, as it poses a security risk to the user.

A warning message is displayed to local users if one of the calls is specified without using the **-full** option for the initial crp:

```
Unable to execute command, crp is not invoked with -full
option
```

In addition, the behavior of the **crp** command has been changed as follows: If a user other than root attempts to access a crp device (/dev/crpxx) which the user does not own, the user will be denied access to the device.

1.7.10.6 Change to ps Command

At SR10.4, the **ps** command shows command line arguments associated with a process only if the displaying process (the one running **ps**) can map the other processes address space. On an SR10.3 node, for example, you might see

```
 4125 ?       S <   6:36 /etc/Xapollo -K /usr/X11/lib/keyboard/keyboard.config -D1 s+r-
```

On an SR10.4 node, this will be displayed as

```
 4125 ?        S <   6:36 [ Xapollo ]
```

This is a security feature. If you do not want this behavior, but prefer **ps** to behave as it did at SR10.3, make **/bin/ps** setuid root.

1.7.10.7 dmlock Command to Lock Screen

Domain/OS SR10.4 provides a new command — **/usr/apollo/bin/dmlock** — that you can use to lock and cover your workstation screen. To unlock and uncover the screen, enter your password.

1.7.10.8 New spm Command Options

The following new /sys/spm/spm command options are supported when specified in the /sys/node_data/startup.spm file:

max_conc *x* Specifies the maximum number of concurrent, interactive (**-cp**) background processes that may run on the node at once. *x* may range from 0 to any value that is reasonable with respect to machine, os, and available disk space. This option does not limit SR9.7 **-cp** processes since there was no concept of "interactive"ness for 9.7. The default is to allow as many processes as are requested. Use of this option does not interfere with the operation of OmniBack, as it does not create **-cp** background processes. One suggested use is to limit the number of DSEE builds that are supported on the node concurrently.

no_crp_me If specified, this option disallows any **crp** operation where the **-me** option is specified, thereby forcing the user attempting to **crp** on to the node to log in with an explicit password. This provides further security from root users. The default is to allow the **-me** option to be specified.

1.7.10.9 Change to Unsupported Boot Shell GO Command

Although the capability to type **EX** (to the DM) to return to the boot shell, and then type **GO** to reload user-space software, is not supported, you should nevertheless note that this functionality has changed slightly at SR10.4. At SR10.4, when you enter GO at the boot shell all mounted disks except for the boot volume are dismounted automatically.

1.7.10.10 Changes to /bin/df Command

At SR10.4, the **/bin/df** command lists the free space on all volumes shown in **/etc/mnttab** (alternatively, **/etc/mtab**). Specifically:

- NFS and CDROM volumes are listed.
- If, for some reason, a volume is listed twice in the mount table, **/bin/df** lists it twice as well.

- Volumes mounted during program execution via calls to the **mount()** system call do not appear, since such mounts do not create entries in **/etc/mnttab** (**/etc/mtab**).

Note that, when a node is booted diskless, its */etc/mnttab* (**/etc/mtab**) file initially contains a single entry for the boot volume of the node that it is booted from. For example, if node //chess is booted diskless off node //atari, the **/etc/mnttab** file for //chess will contain an entry like

```
//atari  /  5.3  rw  680816728 0 0
```

This volume will be listed by a **/bin/df** command executed on the diskless node.

1.7.11 Enhanced Network Protocol to support Autochanger

SR10.4 provides an enhanced network protocol that supports remote access of the Optical Disk Library System (autochanger) despite the long access delays the Autochanger may experience when swapping media. SR10.3 does not support these network protocols, and therefore SR10.3 nodes cannot reliably access Autochanger data over the network.

1.8 ANSI C Support

SR10.4, as did SR10.3, supports full header file and library support for ANSI C. For information about ANSI C support and how to install it, see the SR10.3 Release Notes.

1.9 Changes to Domain/OS C at SR10.4

The following sections describe changes to Domain/OS C at SR10.4.

1.9.1 C Header File Changes

At SR10.4 many of the C language header files (those in /bsd4.3/usr/include and /sys5.3/usr/include) have been changed to add function prototype declarations for library functions not previously declared and not defined by the ANSI C or POSIX or XOPEN standards. These declarations are only enabled when the _APOLLO_SOURCE macro is defined, and that macro is defined by default when compiling in the non-ANSI or extended ANSI modes.

The new function prototypes may cause problems when compiling programs that contain declarations for these functions that conflict with the header files. If this occurs, you have two ways to remove the conflicts:

- Modify the program source to remove the conflicting function declarations.
- Disable the new function prototypes by causing the _APOLLO_SOURCE macro to be undefined. However, there are other declarations that will be disabled as well, so this may cause other problems.

The _APOLLO_SOURCE macro can be undefined by compiling in strict ANSI mode (-A ansi with /bin/cc or -ansi with /com/cc), or by explicitly undefining the macro in non-ANSI or extended ANSI modes using -U_APOLLO_SOURCE with /bin/cc or -undef _APOLLO_SOURCE with /com/cc. In strict or extended ANSI mode, additional sets of standard declarations needed by the program can be enabled by defining additional macros (e.g., _POSIX_SOURCE, _XOPEN_SOURCE, _SYS5_SOURCE). In non-ANSI mode at least one of these macros must be defined, otherwise the _APOLLO_SOURCE macro becomes defined by default.

1.9.2 Enforcing C Standards Conformance: -A standard and -standard Linker Option

At SR10.4, the bind and ld linkers support a new option, -A standard (ld) and -standard (bind). This option sets a bit in the .sri record of the object file so that the loader will enforce the appropriate C standard(s) (ANSI, POSIX, XPG3, OSF AES, and so on).

You need to use this option only if the following two conditions are both true:

- You are using Domain/C Version 6.8 at SR10.4, and you define one of the standards-conformance macros, such as _POSIX_SOURCE or _XOPEN_SOURCE.
- You receive "Unrecognized pragma" warnings from the compiler concerning the HP_STANDARD pragma. At SR10.4, some Domain/OS header files contain this pragma, and Domain/C Version 6.9 recognizes it; Version 6.8 does not.

If you receive one of these warnings, invoke bind or ld with the following syntax:

-A standard,*num* (ld)
-standard *num* (bind)

where *num* is the number specified as an argument to the HP_STANDARD pragma. For example, suppose you receive the following warning message:

```
     (0003) #pragma HP_STANDARD 5

    ******** Line 3: [Warning #253]  Unrecognized pragma.
```

In this case, specify either

/com/bind *program_name*.**bin -standard 5 -b** *program_name*

or

/bin/ld *program_name*.o **-A standard,5**

To avoid having to use the **-A standard** or **-standard** option, do either of the following:

- Upgrade to Domain/C Version 6.9. We recommend that you do this.
- Do not define a standards-conformance macro.

NOTE: Do not use the HP_STANDARD pragma in your own source code. This pragma is for HP internal use only.

1.10 Changes to the Domain/C++ Header Files

A script that modifies the Domain/C++ Version 2.1 layered product has been added to the SR10.4 install mechanism. This script deletes header files which were delivered as part of the C++ product but are no longer required.

For Domain/OS releases previous to SR10.4, C++ supplied its own set of header files, because the "$SYSTYPE/usr/include/..." header files included with Domain/OS were incomplete. The SR10.4 header files have been updated to include the information which was formerly in the corresponding C++ header files, so some C++ header files are no longer needed.

This script is run automatically when you install SR10.4 on any machine that has C++ installed on it. If you install C++ after SR10.4, you must run this script by hand before you can use C++. If you don't run this script, problems may result when users compile their code. You must be root to run the script.

Use one of the following commands to execute this script by hand:

For M68K machines:

```
//<authorized area>/install/ri.apollo.os.v.10.4/install_utils/c++_include.sh \
//<authorized area>/install/ri.apollo.os.v.10.4 //<target node>
```

For A88K machines:

```
//<authorized area>/install/ri.apollo.os.v.10.4.p/install_utils/c++_include.sh \
//<authorized area>/install/ri.apollo.os.v.10.4.p //<target node>
```

The install script will delete 75 files and links. After the install script is run, the following C++ header files remain:

/usr/apollo/include/CC/bsd4.3/Ostream.h (link)
/usr/apollo/include/CC/bsd4.3/assert.h (link)
/usr/apollo/include/CC/bsd4.3/complex.h (link)
/usr/apollo/include/CC/bsd4.3/complex.hxx (link)
/usr/apollo/include/CC/bsd4.3/demangle.h (link)
/usr/apollo/include/CC/bsd4.3/fstream.h (link)
/usr/apollo/include/CC/bsd4.3/generic.h (link)
/usr/apollo/include/CC/bsd4.3/generic.hxx (link)
/usr/apollo/include/CC/bsd4.3/iomanip.h (link)
/usr/apollo/include/CC/bsd4.3/iostream.h (link)
/usr/apollo/include/CC/bsd4.3/libc.h (link)
/usr/apollo/include/CC/bsd4.3/libc.hxx (link)
/usr/apollo/include/CC/bsd4.3/malloc.h
/usr/apollo/include/CC/bsd4.3/math.h
/usr/apollo/include/CC/bsd4.3/new.h (link)
/usr/apollo/include/CC/bsd4.3/osfcn.h (link)
/usr/apollo/include/CC/bsd4.3/osfcn.hxx (link)
/usr/apollo/include/CC/bsd4.3/stdarg.h (link)
/usr/apollo/include/CC/bsd4.3/stdarg.hxx (link)
/usr/apollo/include/CC/bsd4.3/stddef.h (link)
/usr/apollo/include/CC/bsd4.3/stdiostream.h (link)
/usr/apollo/include/CC/bsd4.3/stream.h (link)
/usr/apollo/include/CC/bsd4.3/stream.hxx (link)
/usr/apollo/include/CC/bsd4.3/string.h
/usr/apollo/include/CC/bsd4.3/strstream.h (link)
/usr/apollo/include/CC/bsd4.3/sys/fcntl.h
/usr/apollo/include/CC/bsd4.3/sysent.h
/usr/apollo/include/CC/bsd4.3/vector.h (link)
/usr/apollo/include/CC/bsd4.3/vector.hxx (link)
/usr/apollo/include/CC/common/Ostream.h
/usr/apollo/include/CC/common/assert.h
/usr/apollo/include/CC/common/complex.h
/usr/apollo/include/CC/common/demangle.h
/usr/apollo/include/CC/common/fstream.h
/usr/apollo/include/CC/common/generic.h
/usr/apollo/include/CC/common/iomanip.h
/usr/apollo/include/CC/common/iostream.h
/usr/apollo/include/CC/common/libc.h
/usr/apollo/include/CC/common/new.h
/usr/apollo/include/CC/common/osfcn.h
/usr/apollo/include/CC/common/stdarg.h
/usr/apollo/include/CC/common/stddef.h
/usr/apollo/include/CC/common/stdiostream.h
/usr/apollo/include/CC/common/stream.h

/usr/apollo/include/CC/common/strstream.h
/usr/apollo/include/CC/common/vector.h
/usr/apollo/include/CC/sys5.3/Ostream.h (link)
/usr/apollo/include/CC/sys5.3/assert.h (link)
/usr/apollo/include/CC/sys5.3/complex.h (link)
/usr/apollo/include/CC/sys5.3/complex.hxx (link)
/usr/apollo/include/CC/sys5.3/demangle.h (link)
/usr/apollo/include/CC/sys5.3/fstream.h (link)
/usr/apollo/include/CC/sys5.3/generic.h (link)
/usr/apollo/include/CC/sys5.3/generic.hxx (link)
/usr/apollo/include/CC/sys5.3/iomanip.h (link)
/usr/apollo/include/CC/sys5.3/iostream.h (link)
/usr/apollo/include/CC/sys5.3/libc.h (link)
/usr/apollo/include/CC/sys5.3/libc.hxx (link)
/usr/apollo/include/CC/sys5.3/math.h
/usr/apollo/include/CC/sys5.3/new.h (link)
/usr/apollo/include/CC/sys5.3/osfcn.h (link)
/usr/apollo/include/CC/sys5.3/osfcn.hxx (link)
/usr/apollo/include/CC/sys5.3/rand48.h
/usr/apollo/include/CC/sys5.3/regcmp.h
/usr/apollo/include/CC/sys5.3/stdarg.h (link)
/usr/apollo/include/CC/sys5.3/stdarg.hxx (link)
/usr/apollo/include/CC/sys5.3/stdiostream.h (link)
/usr/apollo/include/CC/sys5.3/stream.h (link)
/usr/apollo/include/CC/sys5.3/stream.hxx (link)
/usr/apollo/include/CC/sys5.3/string.h
/usr/apollo/include/CC/sys5.3/strstream.h (link)
/usr/apollo/include/CC/sys5.3/sysent.h
/usr/apollo/include/CC/sys5.3/vector.h (link)
/usr/apollo/include/CC/sys5.3/vector.hxx

1.11 Change to the Domain/Ada Interface Files

The optional product Domain/Ada makes use of interface files that are optionally installed as part of Domain/OS. For SR10.4, these interface files, which reside in

/usr/apollo/ada/isp_m68k/apollolib
/usr/apollo/ada/isp_a88k/apollolib

have been changed to comment out certain function specifications considered both unnecessary and likely to cause programming errors. The commented sections begin with the following text:

```
-- This function can only be translated as a procedure:
```

1.12 Compressed Debug Information with C 6.9 and Pascal 8.9 Compilers

The Domain C Version 6.9 and Domain Pascal Version 8.9 compilers can produce output object files with compressed debugging information. Use the (-cd) compiler option to request that the debugging information be compressed; the default option (-ncd) is for no debugging compression. (See the release document for the language you are interested in for information on the -cd and -ncd compiler options).

Do not confuse debugging information compression with data compression in the object file. Data compression is an existing feature of Domain object files and is controlled by the -compress and -ncompress compiler options. For more information on data compression, see the appropriate Domain language reference manual.

You must use Version 2.0 or later of the HP Distributed Debugging Environment with object files that have compressed debugging information. Earlier versions of DDE (such as Domain/DDE Version 1.0) will not recognize the new object file format.

As part of a new approach to handling debugging information, the Domain C Version 6.9 and Domain Pascal Version 8.9 compilers now place include file information in a new .s section in the COFF object file. This .s section is part of the debugging information in the object file.

The compression of debugging information occurs when several modules include the same file and thus have identical .s sections. The binder and the link editor overlay these .s sections when combining modules to create the final object file; this overlaying can reduce the overall size of the file significantly, especially for strongly typed languages such as Pascal.

NOTE:
Compression of debugging information is not supported by C++. If you use the -cd option when compiling a C++ program, the compilation will fail, and you will receive an error message.

An object file can contain several, one, or no .s sections, depending on the number of include files associated with the object file. A .s section does not necessarily have a one-to-one correspondence to an include file. If two include files have circular dependencies on each other, they may be merged into one .s file.

The name of a .s section is a ".s" followed by a hexadecimal number, for example:

.s.ABCDABCDABCDABCD

Each .s section has a unique number.

The include table record in a section lists all of the .s sections required by that section. The include table record appears only in .symbols or .s sections. A .s section may need an include table record if it has dependencies on other .s sections.

An include reference in a .symbols or .s section is actually a user-defined type descriptor that refers to the include table record for that .symbols or .s section. This type descriptor provides an index into the list of .s sections in the include table record. The formats for the include table record and include reference type descriptor are in the /usr/include/apollo/dst.h header file.

1.13 RAI Enhancements

Domain/OS SR10.4 includes a number of technical changes to the installation tools. The new features of the installation tools enable you to:

- Deinstall an entire product, using the **-D** option of the **install** or **install++** tool.
- Deinstall selected product subcomponents, using the **-d** option of the **install** or **install++** tool. The former **install++ -d** option (answer unanswered configuration questions with their default values) is now the **-j** option.
- Reset an installed product's Access Control Lists (ACLs) to their original settings (as defined in the product's release index), using the **-A** option of the **install** or **install++** tool.
- Reanswer a single, specified configuration question for a product, using the **reanswer** command of the **config** tool, rather than have to reconfigure the entire product.
- Specify default link-to text when configuring products, using the **set linkprompt** command of the **config** tool, rather than have to repeatedly reenter the same link-to destination.
- Merge PSKs with Domain/OS in an Authorized Area, using the **mrgri** tool. To accommodate this extension, **mrgri** has two new command-line options: **-merge** and **-cmpexe**.

Also, the functionality of **-m** option (do not respect product customization) of the **install** program has been modified. It now changes directories to links (or links to directories) when called for by the product configuration, if the directory in the installed product had been manually changed to a link (or vice versa). Formerly, **install** did this for files, but not for directories.

Finally, the **inprot** tool, which enabled you to modify the Access Control Lists (ACLs) of installed objects, is no longer provided. **inprot** has been superseded by the **setprot** tool, which enables you to modify ACLs in a much easier way. **setprot** is *not* part of the installation tool set, but provided as part of the Domain/OS command set. It is documented in the *Domain System Administration Reference*.

1.14 User Interface Technologies on Domain/OS

In addition to the continuation of X11 R3 support as provided in SR10.3, Domain/OS SR10.4 provides support for X11 R4 user interface technologies on HP Apollo 9000 Series 400 workstations and the Domain Series 2500, 3000, 3500, 4000, 4500 and 5500 workstations running SR10.4 (SAU types 7, 8, 9, 11, 12, and 14 only). X11 R4 user interface technology support described in this section is not supported on the DN10000; User interface technology support on the DN10000 at SR10.4 is identical to that provided at SR10.3.

Use of any X-based user interface software on SR10.4 requires that either BSD4.3 or SYS5.3 or both be installed on the system, and that all X-based software be launched from a BSD4.3 or SYS5.3 shell such as **sh**, **csh**, or **ksh**. Pure AEGIS systems and launching of X-based software from **/com/sh** are not supported.

The X11 R4 user interface technology support consists of:

- HP VUE 2.01
- Motif 1.1 Window Manager (**mwm**)
- R4 Borrow mode server (**Xdomain**)
- R2/R3 Share mode server (**Xapollo**)
- Key Mappings
- Scalable typefaces
- Font Administration
- Shared X11 R4 libraries
- Shared Motif 1.1 libraries
- Shared X11 R3 libraries (preserves functionality of R3 clients)
- R4 MIT clients

Currently available as a layered product is the User Environment Developer's Kit Version 1.1.3. If ordered and installed in association with this release, it will provide:

- X11 R4 archived libraries (Xlib, Xt Intrinsics, and associated include files)
- OSF/Motif 1.1 archived libraries (Xm, Mrm, UIL, and associated include files)
- Man pages

NOTE: As of May 2, 1991, the X/Motif development environment is shipped as a layered product. It is no longer included as part of the standard base software releases, as the X11Release 3 version was in SR10.3.

The following sections describe the user interface technology support included in this release.

1.14.1 mwm Window Manager

The OSF/Motif Window manager (**mwm**) provides a method of controlling the size, shape, state (icon or normal), and location of windows on your display. Refer to *Using the X Window System* for details about **mwm**. Information is also available in the man page for **mwm (/usr/man/cat1/mwm.1)**.

The **mwm** client carries a license for the Motif environment on any authorized platform that is covered under a maintenance agreement. Hewlett-Packard Company grants a license for the Motif environment to all systems covered under maintenance agreements.

1.14.2 Xdomain Server

You can run either the share mode server (**Xapollo**) or the X Window System borrow mode server (**Xdomain**).

The share mode server (**Xapollo**) is described in *Using the X Window System on Apollo Workstations* (015213-A02). Note that **Xapollo** now names its process **xapollo**.

The borrow mode server (**Xdomain**) is described in these release notes.

The **Xdomain** server is **/etc/Xdomain**.

For best performance use UDS (Unix Domain Sockets) for local clients, and TCP internet sockets for remote clients.

Keycode values are different between the **Xdomain** and **Xapollo** servers.

Type **xmodmap -pk** if you need to see the actual keycode values being used.

1.14.2.1 Switching Between Borrow Mode Server and the Display Manager

To switch between the borrow mode server (Xdomain) and the Display Manager, Press [Left Shift], [Ctl], and [F9] simultaneously.

Another way to switch from the Display Manager to the borrow mode server (Xdomain) is to execute the new client **/user/X11/bin/dmtox**. When this client is run iconized in the Display Manager environment, an icon marked "X" is displayed. Move the cursor over the **dmtox** icon and press Shift-Pop to de-iconize the **dmtox** icon. Or assign the iconize/de-iconize function to a mouse button; for example, the following command entered in the DM command window assigns the function to the middle mouse button:

```
kd m2 icon ke
```

Xdomain responds to UNIX signals for switching. The UNIX signal USR1 tells Xdomain to return the display and allow the DM to run. The UNIX signal USR2 tells Xdomain to reborrow the display and become the active window system.

If Xdomain is running and the DM has current control of the display, the server recycles as expected, but takes back control of the display from the DM after the last X client disconnects.

To disable the toggling mechanism in the server, invoke the server with the no borrow option:

```
/etc/Xdomain noborrow
```

This prevents the borrow mode server from responding to the USR1 and USR2 UNIX signals, and the [Left Shift], [Ctl], and [F9] key sequence.

1.14.2.2 Starting an X Server From a Display Manager Shell

Run only one X server at a time. If Xdomain or Xapollo is already running when the new server is started, they conflict over hardware and software resources, with unpredictable results.

You can start the Xdomain server from a Display Manager Shell in any of five ways:

- At the command line, enter:

  ```
  /etc/Xdomain bc
  ```

  Refer to the Xdomain man page for all the possible options.

  When starting the server in this way, no window manager or clients will start automatically. You might then start mwm by typing

  ```
  mwm &> /dev/null &
  ```

  This mode is most useful for observing error messages generated by the server. Most other startup mechanisms redirect messages generated by the server.

- To use the **xinit** client, type

  ```
  xinit -- /etc/Xdomain :0
  ```

  This client starts the Xdomain server. Refer to the **xinit** man page for more options.

  xinit is the most flexible/standard low-level startup mechanism available. Both

x11start (borrow mode) and **startx** (share mode) are wrappers around **xinit**. Knowing this, you may choose to invoke **xinit** directly using your server of choice as a parameter.

- To use the **x11start** client, type:

  x11start

  This client starts the Xdomain server and any clients it finds listed in the **.x11start** file.

  The system-wide file, **/usr/X11/lib/sys.x11start**, starts the **xterm** and **mwm** clients. If you want other clients to run as part of starting Xdomain, copy the file as **.x11start** into your home directory and make the appropriate changes. The system-wide file is used only if there is no file in your home directory.

  Entries in the **.x11start** file use this syntax:

  ```
  client [-options] [&]
  ```

  The client name and options are the same ones that you would use to start the client from the command line. The "&" tells Xdomain to run the client as a background process, as shown in this exam- ple:

  ```
  xclock -digital &
  ```

- To start the Xdomain server and **/usr/X11/lib/xdm/Xsession** from a Display Manager, run **xsession** from a pad. (If you want to be logged out of the DM when you end your xdm session, run **xsession -lo** .)

  The **xsession** script starts Xdomain and **/usr/X11/lib/xdm/Xsession**, and runs **dmtox** in your pad, which becomes iconized, with an "X" displayed. From the DM, move the cursor over this icon and press Shift-Pop to switch to the Xdomain environment.

  You can invoke **xsession** from a DM login script. For details, see the instructions at the beginning of **/usr/X11/bin/xsession**.

- The fifth way of starting the Xdomain server also starts up HP VUE 2.01. See Section 1.14.3 for information on this method.

1.14.2.3 Starting an X Server at Boot Time

- To start the Xdomain server from xdm when your system boots:

  1. Touch the **/etc/daemons/xdm** file.
  2. Remove the file **/etc/daemons/vue** if it exists.
  3. Reboot your system.
  4. If it does not start correctly, remove **~/.Xsession** and try again. Then modify **~/.Xsession** until it works as you want.

  This will actually run **vuelogin** instead of **xdm** on m68k machines, but using the **xdm** scripts. (At SR10.4, **vuelogin** replaces **xdm** as the session manager on m68k systems; **xdm** continues to be the session manager on a88k systems.)

  When booting your node in this way, the DM won't start. Hence, you should not attempt to toggle from the Xdomain server.

  NOTE:
  If you bring the Xdomain server up from bootup using **xdm** (as described in this section), HP VUE 2.01 will not be started. The **Xsession** script in **/sys/node_data/etc/xdm** is designed to start HP VUE 1.1 if it is installed. To start HP VUE 2.01, also touch **/etc/daemons/vue** (as described just above).

- To start the Xdomain server and HP VUE 2.01 from vuelogin when your system boots:

  1. Touch the **/etc/daemons/vue** file.
  2. Touch the **/etc/daemons/xdm** file.
  3. Reboot your system.
  4. If it does not start correctly, remove both **~/.xsession** and **~/.Xsession** and try again. Then modify **~/.xsession** (or **~/.Xsession** if you want the same behavior for both vue and xdm mode startups) until it works as you want.

  When booting your node in this way, the DM won't start. Hence, you should not attempt to toggle from the Xdomain server (If you do, you will have no user interface to work with, and no way to get back to Xdomain). You may want to change the file **/usr/lib/X11/vue/Vuelogin/Xservers** to add the **noborrow** option to the line invoking Xdomain, so that it reads

```
:0 local /etc/Xdomain noborrow bc
```

1.14.2.4 Compatibility with Clients

Refer to *Using the X Window System* for a list and description of X11 R4 clients. R3 clients should run correctly with Xdomain, but are not guaranteed to do so.

When running older X11R3 clients such as xterm against an R4 server, start the server with the **bc** (backwards or bug compatibility) option.

NOTE:
Certain R3 clients such as **xterm** cannot run against the X11R4 borrow mode server unless the server is invoked with the **bc** bug compatibility option. Many X11R3 clients have invalid parameters in some of the calls that weren't trapped by X11R3 servers. The X11R4 server traps these invalid parameters unless the **bc** flag is used. As a general guideline, you should use the **bc** flag when running with R3 clients, but run in normal X11R4 mode when developing or running X11R4 clients. If you see errors related to XPointerGrabs and Events, you may be seeing one of these R3 to R4 bugs.

1.14.2.5 Compatibility with Window Managers

The Xdomain server runs with **mwm** or **twm**. Both of these window managers are described in the *X Window System User's Guide* for X11 R3 and R4 by O'Reilly and Associates (015534-A00). This manual can be ordered from Apollo Direct at 1-800-225-5290. Customers in Europe and Intercon can contact their local HP office.

The Xdomain server also runs with **vuewm**, which is part of HP VUE 2.01 described below.

1.14.2.6 Cursor Shapes

A variety of cursor shapes are found in the **/usr/X11/include/bitmaps** directory. These cursors can be used with **Xdomain.**

1.14.2.7 New Support for DNx500 DVS Displays

This release provides the capability to run Xdomain on DN3500, DN4500, and DN5500 DVS displays.

DVS systems with 40 plane graphics have added support for 24-bit TrueColor and 24-bit DirectColor X visuals. A simple true color
example can be found in

/domain_examples/gpr-to-x_examples/Xlines_in_true_color.c.

1.14.3 Key Mappings

Key mappings are in the **/usr/X11/lib/XKeysymDB** file.

Use the following key bindings for the OSF virtual keys:

osfCancel	Escape
osfLeft	Left
osfUp	Up
osfRight	Right
osfDown	Down
osfEndLine	apRightBar
osfBeginLine	apLeftBar
osfPageLeft	apLeftBox
osfPageRight	apRightBox
osfPageUp	apUpBox
osfPageDown	apDownBox
osfBackSpace	BackSpace
osfDelete	apCharDel
osfSelect	Select
osfAddMode	Shift F8
osfHelp	Shift Help
osfMenu	F4
osfMenuBar	F10
osfCopy	apCopy
osfCut	Shift apCut
osfPaste	apPaste
osfQuickPaste	apLineDel

(The Select key is the Mark key on the keyboard.)

1.15 HP VUE 2.01

Domain/OS SR10.4 provides support for HP VUE 2.01 on HP Apollo 9000 Series 400 workstations and the Domain Series 2500, 3000, 3500, 4000, 4500 and 5500 workstations running SR10.4 (SAU types 7, 8, 9, 11, 12, and 14 only). HP VUE 2.01 is not supported on the DN10000.

HP VUE (HP Visual User Environment) is a graphical user interface for Domain/OS. It includes:

- The window manager (**vuewm**), which provides multiple workspaces and the workspace manager (also called the front panel).
- The session manager (**vuesession**), which provides the ability to save and restore user sessions.
- The file manager (**vuefile**), which provides file browsing and file management capabilities.
- The help manager (**vuehelp**), which provides application and context-sensitive help.
- The style manager (**vuestyle**), which provides users with an easy way to customize certain aspects of the appearance and behavior of their workstation such as colors,

fonts, xhost permissions, session restoration, and mouse sensitivity.

- A Broadcast Message Server, which allows the HP VUE managers to interact with one another.

See the *HP VUE User's Guide* (B1171-90042), the *HP VUE System Administration Manual* (B1171-90044), and the *HP VUE Configuration Guide for Domain/OS Systems* (B1171-90046) for information on using, administering, and configuring HP VUE 2.01.

1.15.1 HP VUE 2.01 Requirements

You must be running TCP/IP in order to run HP VUE 2.01.

You must use the Xdomain server with HP VUE 2.01. Unlike HP VUE 1.1, which will work with either the share-mode server (Xapollo) or borrow- mode server (Xdomain), HP VUE 2.01 requires the borrow mode server (Xdomain). (HP VUE 1.1 is a layered product.)

You must have either BSD4.3 or SYS5.3 or both install on the system, and all users logins through vuelogin must use a BSD4.3 or SYS5.3 shell such as **sh**, **csh**, or **ksh**. The AEGIS **/com/sh** is not supported through **vuelogin**.

1.15.2 File Locations on Domain/OS Systems

There are several fundamental differences in file locations (with respect to standard X Windows file locations) on Domain/OS systems that affect HP VUE.

For example, the following X11 system directories have different locations on Domain/OS systems than on HP-UX systems.

HP-UX Location	Domain/OS Location
/usr/lib/X11	/usr/X11/lib
/usr/bin/X11	/usr/X11/bin

The HP VUE documentation refers in all cases to the standard X Windows paths.

On Domain/OS systems, HP VUE makes use of links between the standard directory locations and the corresponding Domain/OS directories. The actual files reside in their Domain/OS locations.

For example, executing:

```
cd /usr/lib/X11
pwd
```

displays

```
        /usr/X11/lib
```

The links set up for the X11 system directories use the absolute paths to the local system files. For example, executing:

```
cd /usr/lib/X11
pwd
```

on local system **//sysa** while browsing the remote system **//sysb** displays

```
        //sysa/usr/X11/lib
```

1.15.2 Starting Xdomain and HP VUE 2.01 From a Display Manager Shell

To start the Xdomain server and HP VUE 2.01 from a Display Manager:

1. Touch the **/etc/daemons/vue** file.
2. Run **hpvue2** from a pad. (First make sure no X server is running.) If you want to be logged out of the DM when you log out of HP VUE 2.01, run **hpvue2 -lo** .

The **hpvue2** script starts Xdomain and HP VUE 2.01, and runs **dmtox** in your pad, which becomes iconized, with an "X" displayed. From the DM, move the cursor over this icon and press Shift-Pop to switch to the HP VUE 2.01 environment.

1.15.3 Starting Xdomain and HP VUE 2.01 From a Display Manager Startup File

To start the Xdomain server and HP VUE 2.01 from your login script, place

```
cp /bin/ksh /usr/X11/bin/hpvue2

        or

cp /bin/ksh /usr/X11/bin/hpvue2 -lo
```

in an appropriate DM startup file, such as **˜/user_data/startup_dm.xxx.**

1.16 Eight-Bit Native Language Support (NLS)

At SR10.4 Domain/OS provides eight-bit Native Language Support (NLS), which includes standard ASCII characters and Western European character sets. Domain/OS NLS consists of a set of new and updated C library functions and related commands. See Chapter 6 of these Release Notes for an introduction to and description of Domain/OS

NLS.

1.17 Network License System (NetLS)

The Network License System (NetLS) provides electronically licensed access to software products in a heterogeneous network environment. The Network License System consists of two products:

- The License Server Lock (LSLOCK), which enables software vendors to license their software products.
- The Network License Server (LSSERVER), which provides the runtime environment for software products licensed with the License Server Lock.

The License Server Lock provides vendors of software products with

- Libraries of calls to the license server that enable software developers to license their software products
- A tool, **nls_pass**, that creates licenses for customers of the licensed software products

The Network License Server (LSSERVER) provides customers of such licensed software products with

- The network license server daemon (**netlsd**), which administers licenses
- Administrative tools for managing license servers: **ls_admin**, **ls_rpt**, and **ls_stat**

SR10.4 includes the Network Licence Server (LSSERVER) component of NetLS. LSLOCK is a layered product which is currently available on HP/UX and Sun platforms as well as on Domain/OS. Any licensed applications running in the SR10.4 environment will work as the LSSERVER is contained in the SR10.4 base software.

1.18 Version 2.0 of HP/Distributed Debugging Environment

HP/Distributed Debugging Environment Version 2.0 replaces Domain/DDE Version 1.6 at SR10.4. HP/DDE Version 2.0 provides the following new features:

- Graphical user interface based on OSF/Motif standards and the X Windows System. Motif is the debugger's default user interface; the Display Manager user interface is available using the **-ui** option to the **dde** command as follows:

  dde -ui apollo

- **Restart** command. Terminates the target program and restarts it. The debugger restores its working directory and streams to their original states. Whenever possible, breakpoints and watchpoints are preserved when the program is restarted.

- **Property flags** command. HP/DDE uses options to the **property flags** command to control the behavior of debugger managers.
- **Call** command. Evaluates name or expression and prints value.
- **Delete declarations** command. Deletes one or more declarations.
- **List declarations** command. Lists all declared items in the program.
- **List images** command. Displays the list of executables loaded in the address space of the current target program.

1.19 BLAS Routines Integrated in /lib/syslib.*

BLAS stands for Basic Linear Algebra Subroutines. The BLAS set comprises the low-level vector and matrix routines upon which Linpack (the linear algebra package), eispak (the eigenvector package), etc. are based. At SR10.4, the BLAS routines have been integrated in /lib/syslib.*.

For more information about the BLAS routines, see Chapter 7 of this Release Document.

1.20 Additional Fault Recovery Functionality

NOTE:
(This note applies to BSD4.3 and Sys5.3 /usr/include/sys/signal.h files *only*.)

For SR10.4, additional fault recovery functionality has been added to allow the UNIX user to continue after memory reference related faults(i.e., UNIX_SIGSEGV). This type of fault can be recovered from if:

1. A user signal handler has been established to handle UNIX_SIGSEGV signals and
2. That handler resolves the faulting condition

UNIX_SIGSEGV faults that occur as a result of a memory fault, and not from a cross-process signal with a UNIX_SIGSEGV signo, CANNOT be ignored. Also, only one UNIX_SIGSEGV fault can be handled at a time. If multiple faults occur, only the last fault can be recovered from.

The changes to the user interface are limited to changes in the sigcontext_t structure that the OS passes to the signal handler.

From **/usr/include/sys/signal.h** :

```
struct          sigcontext {
     int     sc_onstack;                 /* sigstack state to restore */
     int     sc_mask;                    /* signal mask to restore */
     int     sc_sp;                      /* sp to restore */
     int     sc_fp;                      /* fp to restore */
     int     sc_ap;                      /* ap to restore */
     int     sc_pc;                      /* pc to restore */
     int     sc_ps;                      /* psl to restore */
     int     sc_fa;                      /* fault address for sigsegv only */
     char    sc_df;                      /* sigsegv data_fault */
     char    sc_read;                    /* sigsegv read/write flag */
     char    sc_return_from_segv;        /* acknowledge sigsegv signal, set by hand
};
```

Four new fields have been added:

1. SC_FA is the address of the faulting reference.
2. SC_DF is a boolean flag that indicates whether this fault was due to an instruction or data reference.
3. SC_READ is a boolean flag that indicates the type of memory access, read(true) or write(false). An instruction fetch fault always sets SC_READ true.
4. SC_RETURN_FROM_SEGV is a boolean flag that must be set by the signal handler in order to return to the faulting code. This field is significant ONLY for UNIX_SIGSEGV signals. In the event an application exists that establishes a UNIX_SIGSEGV signal handler, regardless of any attempt by the user to recover from the fault, pre-SR10.4 behavior will be maintained(i.e., no return permitted) until the application explicitly sets this flag before returning from the signal handler.

1.21 Enhanced Protection Features

Prior to SR10.3, **rbak -sacl** could be used to restore protected subsystems from media, independent of user privileges. At SR10.3 and earlier releases, **cpf -sacl** and **cpt -sacl** could be used to move protected subsystems from one storage media to another, independent of user privileges.

PSK Q2-91 introduced a fix to the above protection problems. The fix can be enabled (and disabled) by using the **/etc/lprotect** command, which has been modified as follows:

/etc/lprotect *[-protect [unix | owners | aegis]]*
[-rmtroot [all | none | readonly]]

The original behavior of SR10.3 is preserved in this release as the default. The initial protection mode of a node is now the **owners** mode, in which every user with write access to the file **‘node_data/node_owners** is privileged to perform these restricted operations. Because the **node_owners** file is installed with ‘other’ access including 'write', all users have the privilege to use the **-sacl** options and to mount and dismount file systems. System administrators may either alter the protection of **'node_data/node_owners** or use the **lprotect** command to alter this default behavior if desired. This default mode was selected to permit system administrators the ability to restrict the actions of users in the level II boot shell.

To provide UNIX-like restrictions of these privileges, type the following at a shell prompt:

```
/etc/lprotect -protect unix
```

To disable the fix and return the protection mode back to the default behavior, type:

```
/etc/lprotect -protect aegis
```

With UNIX protection enabled, the following changes in system behavior occur:

- **mtvol** and **dmtvol** are restricted to **root. mount** and **umount** continue to require that the user be **root**.
- Attempts to copy files with their original ACLs are restricted. If the file being copied is a protected subsystem, and the user attempting to copy the file is not **root** (or does not possess appropriate privilege or neither), the attempt to assign the ACL to the new file will fail. When a **cp** command fails for this reason, the new file still exists, but is left with the ACL it would have had if the **-P** option had not been specified (the default ACL). If a **cpf** or a **cpt** command fails for this reason, there is no target object.
- Attempts to restore files from tape media with their original ACLs (**rbak -sacl**) are restricted. If the file being copied is a protected subsystem, and the user attempting to copy the file is not **root** (or does not possess appropriate privilege or neither), the attempt to assign the ACL to the new file will fail. When **rbak** fails for this reason, the new file still exists, but is left with the ACL it would have had if the **-sacl** option had not been specified (the default ACL).
- Versions of DSEE prior to Version 4 require root privileges to use the **create library** and **create element** commands. Note that this is the case only if the PSK is installed on the node where the DSEE source libraries actually reside.

To enable "owners" protection, type:

```
/etc/lprotect -protect owners
```

With "owners" protection enabled, only those users who have write access to the **'node_data/node_owners** file at the time this command is issued are able to perform the restricted operations. Therefore, **w** rights to the **'node_data/node_owners** file identifies a node owner.

It is usually the case that system administrators will either modify the node owners file or add the appropriate **lprotect** command to the **/etc/rc** file in order to satisfy their own security policy requirements.

NOTE: The protection enabled by the **lprotect** command affects only the node it is invoked on. For example, as noted in behavior number 4 above, DSEE behavior only changes if the PSK is installed on the node where the DSEE source libraries actually reside.

We have rewritten various **man** pages and help files associated with the protection changes. New files include:

/sys/help/cpf.hlp
/sys/help/cpt.hlp
/sys/help/dmtvol.hlp
/sys/help/lprotect.hlp
/sys/help/mtvol.hlp
/sys/help/rbak.hlp
/sys/help/sigp.hlp
/bsd4.3/usr/man/cat1/rbak.1
/bsd4.3/usr/man/cat8/lprotect.8
/sys5.3/usr/catman/a_man/man1/lprotect.1m
/sys5.3/usr/catman/u_man/man1/rbak.1

Chapter 2: Installing SR10.4

Consult the manual *Installing Domain Software* (008860-A03) for procedures for installing SR10.4. The A03 version documents changes made to the installation tools since SR10.2, and is a complete rewrite of the previous (A02) version of the manual, released at SR10.2. The preface of the manual summarizes the changes made to the installation tools. The new tools are provided with the SR10.4 distribution media. The manual describes how to install SR10.4 on a new or **involed** node from distribution media, how to install SR10.4 on a new or **involed** node across the network from an Authorized Area, how to load SR10.4 into an existing Authorized Area, and how to install SR10.4 as an update to an earlier version of Domain/OS.

This chapter includes installation information that is specific to revision 10.4 of Domain/OS and that supplements the generic procedures in the installation manual. Be sure to read the information in this chapter that describes the different install configurations of SR10.4 and their sizes. Note that systems that have smaller disk drives may be unable to accommodate the larger configurations.

Tables 2-1 and 2-2 include the sizes for nodes that run SAUs 7, 8, 9, 11, 12, or 14. Tables 2-3 and 2-4 include the sizes for SAU10 nodes (a88k machines).

2.1 Installation Tool Compatibility

New versions of the installation tools are provided with SR10.4. These tools are backwards compatible. That is, you can use them to install any RAI-installable product, including previously-released optional products, patches, and PSKs, and earlier versions of Domain/OS. Conversely, you can use earlier versions of the installation tools to install SR10.4; however, we recommend that you use the new tools to install SR10.4 and that you replace all older versions of the tools with the new tools.

2.2 Strategies for Installing SR10.4 and Restoring the Installation Tools

There are two basic approaches for installing SR10.4 for the first time in an existing SR10.x-based network.

The first approach is to load and install SR10.4 from the distribution media on an **involed** (initialized) node. Use the procedure in Chapter 1 of the 008860-A03 version of the installation manual. (**Involing** an existing node prior to installing SR10.4 is optional. See Section 2.4). In addition to installing an operational configuration of SR10.4 on the node, this procedure creates an Authorized Area on the node. The Authorized Area contains the SR10.4 installation tools and the SR10.4 product. You can then use this Authorized Area to install SR10.4 across the network on other **involed** nodes (Chapter 2 of the installation manual) or on non-**involed**, SR10.x nodes (Chapter 5).

The other approach is to load SR10.4 and the SR10.4 installation tools from distribution media into an existing Authorized Area, without **involing** the node. To do this, use the procedure "Loading Products from Media into an Authorized Area" in Chapter 5 of the installation manual. This procedure overwrites the current installation tools in the Authorized Area with the SR10.4 tools. You can then install SR10.4 from this Authorized Area across the network on other **involed** nodes (Chapter 2) or on non-**involed** nodes ("Configuring Products in an Authorized Area" and "Installing Products from an Authorized Area," Chapter 5). At any point you can also install SR10.4 on the Authorized Area node itself, but this is not required before performing network installs of SR10.4 from this Authorized Area.

2.3 Disk invol Not Required

Installation of SR10.4 on an existing node does not require disk initialization (**involing**) but you may wish to **invol** your node to take advantage of new functionality such as disk quotas.

2.3.1 If You Don't invol

If you do not invol the disk, be sure to use the **config** utility to reconfigure before installing SR10.4, since the configuration questions for SR10.4 are substantially different from those for earlier SR10 releases. *Do not* install SR10.4 using a configuration file created for a pre-SR10.4 version of Domain/OS.

Use the SR10.4 config and install tools to configure and install SR10.4. Do not use the SR10.3 versions.

2.4 Appropriate SAUs for SR10.4 Installation

At SR10.4, support is discontinued for SAUs 2, 3, 4, 5, and 6. See Section 5.2 of this document for lists of supported and unsupported SAUs.

2.5 SR10.4 Installed with Closed ACLs

Unlike earlier versions of Domain/OS, which gave you the configuration option of an open or closed protection scheme, SR10.4 is unconditionally installed with closed Access Control Lists (ACLs). This means that system file ACLs limit write access to the root user only.

Templates are provided in:

install/templates/apollo/os.v.10.4/setprot_open_aegis
install/templates/apollo/os.v.10.4/setprot_open_bsd4.3
install/templates/apollo/os.v.10.4/setprot_open_sys5.3

install/templates/apollo/os.v.10.4.p/setprot_open_aegis
install/templates/apollo/os.v.10.4.p/setprot_open_bsd4.3
install/templates/apollo/os.v.10.4.p/setprot_open_sys5.3

to enable you to open the system file ACLs. These templates are used in conjunction with the **/etc/setprot** command. You will need to be logged in as **root** for this to work.

2.6 Changes to the SR10.4 Release Index

We have made changes to the SR10.4 Release Index to be more consistent in the way that we preserve files in DOMAIN. The following files are now installed and the files which they would replace are renamed by appending the current date to the filename:

/etc/printcap
/etc/snmpd.conf

/sys/dm/startup_login
/sys/dm/startup_login.1280bw
/sys/dm/startup_login.1280color
/sys/dm/startup_login.191
/sys/dm/startup_login.768
/sys/dm/startup_login.color

/sys/node_data/startup
/sys/node_data/startup.1280bw
/sys/node_data/startup.1280color
/sys/node_data/startup.191
/sys/node_data/startup.768
/sys/node_data/startup.color
/sys/node_data/startup.spm

/sys/node_data/etc/login_log.conf
/sys/node_data/etc/syslog.conf

/sys/node_data/etc/dm_display/color_map
/sys/node_data/etc/dm_display/color_map.rgb
/sys/node_data/etc/dm_display/color_map.rgb332
/sys/node_data/etc/dm_display/color_map.rgb444 (a88k only)
/sys/node_data/etc/dm_display/color_map.rgb666

/usr/lib.uucp/Devconfig
/usr/lib.uucp/Devices
/usr/lib.uucp/Dialcodes
/usr/lib.uucp/Dialers
/usr/lib.uucp/Maxuuscheds
/usr/lib.uucp/Maxuuxqts
/usr/lib.uucp/Permissions
/usr/lib.uucp/Poll
/usr/lib.uucp/Sysfiles
/usr/lib.uucp/Systems

/bsd4.3/usr/lib/Mail.rc

If you do not want this to happen, you should add the filenames of those files you want to preserve to the file **/install/preserve.list**. This change applies to both m68k and a88k machines.

The following files are installed only if they do not exist on the target node. (for example, in the case of a freshly initialized disk):

/etc/hosts
/etc/hosts.equiv
/etc/hosts.equiv_link
/etc/hosts_link
/etc/networks
/etc/networks_link
/etc/protocols

/etc/rgy/passwd_override

/sau7/cmd (m68k only)

/sau8/cmd (m68k only)

/sau9/cmd (m68k only)

/sys/dm/fonts/std
/sys/dm/fonts/std.1280bw
/sys/dm/fonts/std.1280color
/sys/dm/fonts/std.191
/sys/dm/fonts/std.color

/sys/net/diskless_list

/sys/node_data/device_numbers
/sys/node_data/spm_control

/sys/node_data/cron/at.deny
/sys/node_data/cron/cron.deny
/sys/node_data/cron/crontab
/sys/node_data/cron/queuedefs
/sys/node_data/cron/at/lasttimedone
/sys/node_data/cron/crontabs/root

/sys/node_data/etc/fstab
/sys/node_data/etc/inetd.conf
/sys/node_data/etc/mnttab
/sys/node_data/etc/node_owners
/sys/node_data/etc/phones
/sys/node_data/etc/remote
/sys/node_data/etc/stackd.conf
/sys/node_data/etc/ttys
/sys/node_data/etc/utmp

/sys/siologin/siomonit_file

/bsd4.3/usr/lib/aliases.dir
/bsd4.3/usr/lib/aliases.pag

/bsd4.3_usr/spool/news

/sys5.3/usr/lib/aliases.dir
/sys5.3/usr/lib/aliases.pag

/sys5.3_usr/spool/mqueue

2.7 Canned Selection Files and Configurations

SR10.4 is shipped with 13 pairs of selection and override files, plus one configuration file that works with any pair of selection/override files. Use these files to install your software. *Installing Domain Software* (008860-A03) describes selection, override, and configuration files and their purposes in detail. The following subsection describes the components that are specified by the selection files. These descriptions are followed by tables listing the product components and their sizes. The last subsection describes what product components are loaded into the AA and can consequently be installed with the configuration file **/install/templates/apollo/os.v.10.4/cf.os.**

Selection files for the various Domain/OS sizes reside in //<*authorized_area*>**/install/templates/apollo/os.v.10.4**. The names of the selection files are as follows:

aa.aegis_bsd4.3_large
aa.aegis_bsd4.3_medium
aa.aegis_large
aa.aegis_medium
aa.aegis_small
aa.aegis_small_prog
aa.aegis_sys5.3_large
aa.aegis_sys5.3_medium
aa.bsd4.3_large
aa.bsd4.3_medium
aa.large
aa.sys5.3_large
aa.sys5.3_medium

2.7.1 Selection Component Descriptions

Following are brief descriptions of the components that make up the Domain/OS product.

sysboot	The program that loads Domain/OS into memory.
SAU/N base	A directory containing the required stand-alone utilities.
SAU/N diagnostics	Offline hardware diagnostics used for troubleshooting hardware problems.
sau_sys help	Help files for offline hardware diagnostics.
systest	A directory containing various online system tests and exercisers.
systest/ssr_util	A directory containing unsupported field service utilities.
install utilities	Domain/OS commands used by one or more of the installation tools.
domain_examples	A directory containing online programming examples.
standard fonts	The standard system fonts (installed on every node) are in the directory **/sys/dm/fonts.**
optional fonts	All 8-bit fonts not considered standard. Nonstandard fonts include the families 'charter' and 'new century schoolbook' in all sizes (8, 10, 12, 14, 18, 24) and styles (roman, bold, italic and bolditalic), oblique and boldoblique styles of 'courier' and 'times' in all sizes, 'courier' in all styles and all but standard sizes (8, 14, 18, 24), all standard font families in size 8, an Old-English font, some graphics fonts (chess, symbol), and unusual point sizes (7, 9) for some of the standard fonts. These are in

/sys/dm/fonts.

16-bit fonts

Kanji and Hangul fonts, containing JIS codes and 7-bit ASCII codes below 127. These are also in /sys/**dm/fonts.**

lib

The Domain/OS libraries.

etc base

This component contains all the /**etc** commands that are common to all three environments or are identical in both the BSD and SysV environments. However, it does not contain the /**etc** commands required to run TCP/IP.

etc/tcp

This component contains the /**etc** commands that are required to run TCP/IP.

guaranteed commands

A common set of commands present on every node, regardless of environments installed. These are needed to run installation scripts for third-party software. This command set is a subset of **/sys5.3/bin** and is installed there for all environments.

com

A directory containing a large set of Aegis environment commands.

sys base

The top-level Domain/OS system directory.

sys/help

A directory containing help files for the Aegis environment.

sys/ins

Apollo-specific include files for Aegis software development.

sys/source

A directory containing sources for bit-pad support, the **emt** command, and models for implementing **siorf/siotf** on a non-Apollo system.

usr base

Base software utilities for use in all three environments. They should be present on all nodes but do not have to be local to every node.

usr/new

A directory containing a set of user contributed commands from the BSD distribution of the UNIX operating system.

usr/games

A directory containing a collection of games, including games from the SysV and BSD distributions and games developed by Apollo.

usr/apollo/include

A directory containing C include files for Domain/OS calls with function prototypes.

usr/apollo/man

A directory containing manual pages with detailed descriptions of the Domain/OS calls.

bsd4.3 base

All commands and files that are specific to the BSD environment except those included in the bsd4.3 **etc** or bsd4.3 **usr** components.

bsd4.3 etc — BSD environment commands that reside in **/etc** and either have nonidentical counterparts or no counterparts in a SysV environment.

bsd4.3 usr — BSD environment commands that reside in **/usr** and either have nonidentical counterparts or no counterparts in a SysV environment.

sys5.3 base — All commands and files that are specific to the SysV environment except those included in the sys5.3 **etc** or sys5.3 **usr** components.

sys5.3 etc — SysV environment commands that reside in **/etc** and either have nonidentical counterparts or no counterparts in a BSD environment.

sys5.3 usr — SysV environment commands that reside in **/usr** and either have nonidentical counterparts or no counterparts in a BSD environment.

2.7.2 Selection Component Tables

The following tables list the software components that are loaded into your Authorized Area if you use the predefined selection files. They also specify the sizes of each component that is installed. They should help you determine the particular selection file that is most appropriate for your use and disk sizes. Table 1 covers the small and medium sized selections for SAUs 7, 8, 9, 11, 12, and 14. Table 2 covers the large selections for SAUs 7, 8, 9, 11, 12, and 14. Table 3 covers the small and medium selections for SAU10. Table 4 covers the large selections for SAU10.

Note that these tables give the total size of the Authorized Area and the size of the software that will be installed on your node if you choose one of the standard templates. You can reduce the size of the software that is installed on the node by using a customized configuration file instead of the one supplied in **install/templates/apollo/os.v.10.4/cf.os**. If you use a customized configuration, the messages displayed during the **config** operation indicate the amount of disk space used by your selections.

Each row in the tables corresponds to a selection component determined by the release index. The row identifies the directory that contains the software to be installed. However, some directories, such as **/etc**, are split among several selections, and some selections determine the software that is installed in several directories.

Each column corresponds to a particular predefined selection file. For example, the AVM column defines the contents of the **aa.aegis_sys5.3_medium** selection file. The following table key lists the meanings of the one-character selection-file identifiers.

NOTE: The AA sizes given in Tables 2-2 and 2-3 are for SAU7, SAU8, and SAU9 machines (dn2500, dn3000, dn 3010, dn3500, dn4000, and dn4500). Actual installed sizes for SAU11, SAU12, and SAU14 machines (425s, 425t, 425e, 433s, 400s, 400t, 400dl, and dn5500) may be up to 10% larger than specified in these tables. This is due to the fact that the SAU11, SAU12, and SAU14 machines use a 4KB page size for disk I/O (as opposed to the 1KB page size used by the SAU7, SAU8, and SAU9 machines).

NOTE: You'll need approximately 12MB of free disk space on the node on which you are running the install program. This allows for space required by the installation processes during runtime. If you are running the install program on the same node on which you are installing Domain/OS, you'll need about 12MB of free disk space *in addition to* the actual size of the configuration as shown in the following tables.

Table Key:

A = aegis
B = bsd4.3
V = sys5.3
S = small
M = medium
L = large
P = prog (for programmers)

TABLE 2-1. AA Size for Small and Medium Selections (SAUs 7,8,9,11,12,14)

Component	**Size (MB)**	**Selection File Code**						
		AS	ASP	AM	BM	VM	ABM	AVM
sysboot	0.01	X	X	X	X	X	X	X
SAU/N base	5.7	X	X	X	X	X	X	X
SAU/N diagnostics	2.5							
systest	9.0							
systest/ssr_util	2.8							
install utilities	1.0	X	X	X	X	X	X	X
domain_examples	3.5							
optional fonts	1.4			X	X	X	X	X
lib	11.2	X	X	X	X	X	X	X
etc base	10.0	X	X	X	X	X	X	X
guaranteed commands	.3	X	X	X	X	X	X	X
com	3.3	X	X	X			X	X
sys base	28.0	X	X	X	X	X	X	X
sys/help	3.0							
sys/ins	1.0		X	X			X	X
sys/source	0.3							
usr base	8.9	X	X	X	X	X	X	X
usr/apollo/include	0.9		X	X	X	X	X	X
usr/apollo/man	3.4							
usr/X11	18.9							
usr/games	3.1							
usr/new	4.4							
bsd4.3 base	1.5				X		X	
bsd4.3 usr	15.7				X		X	
sys5.3 base	2.0					X		X
sys5.3 usr	16.7					X		X
Total (Approximate)	158.5	68.5	70.4	71.8	84.7	86.2	89.0	91.0

NOTE: An additional minimum of 12 MB of free space must be available during the installation from media.

Sizes for SAU11, SAU12, and SAU14 machines may be up to 10% larger. (See note on page 2-9)

TABLE 2-2. AA Size for Large Selections (SAUs 7,8,9,11,12,14)

Component	**Size (MB)**	**Selection File Code**					
		AL	BL	VL	ABL	AVL	ABVL
sysboot	0.01	X	X	X	X	X	X
SAU/N base	5.7	X	X	X	X	X	X
SAU/N diagnostics	2.5	X	X	X	X	X	X
systest	9.0	X	X	X	X	X	X
systest/ssr_util	2.8	X	X	X	X	X	X
install utilities	1.0	X	X	X	X	X	X
domain_examples	3.5	X	X	X	X	X	X
optional fonts	1.4	X	X	X	X	X	X
lib	11.2	X	X	X	X	X	X
etc base	10.0	X	X	X	X	X	X
guaranteed commands	.3	X	X	X	X	X	X
com	3.3	X			X	X	X
sys base	28.0	X	X	X	X	X	X
sys/help	3.0	X			X	X	X
sys/ins	1.0	X			X	X	X
sys/source	0.3	X			X	X	X
usr base	8.9	X	X	X	X	X	X
usr/X11	0.9	X	X	X	X	X	X
usr/apollo/include	3.4	X	X	X	X	X	X
usr/apollo/man	18.9		X	X	X	X	X
usr/games	3.1		X	X	X	X	X
usr/new	4.4		X	X	X	X	X
bsd4.3 base	1.5		X		X		X
bsd4.3 usr	15.7		X		X		X
sys5.3 base	2.0			X		X	X
sys5.3 usr	16.7			X		X	X
Total (Approximate)	158.5	95.4	132.3	133.8	139.9	141.4	158.5

NOTE: An additional minimum of 12 MB of free space must be available during the installation from media.
Sizes for SAU11, SAU12, and SAU14 machines may be up to 10% larger. (See note on page 2-9)

Table Key:

A	=	aegis
B	=	bsd4.3
V	=	sys5.3
S	=	small
M	=	medium
L	=	large
P	=	prog (for programmers)

TABLE 2-3. AA Size for Small and Medium Selections (SAU10)

Component	Size (MB)	Selection File Code						
		AS	ASP	AM	BM	VM	ABM	AVM
sysboot	0.01	X	X	X	X	X	X	X
SAU10 base	6.7	X	X	X	X	X	X	X
SAU10 diagnostics	24.6							
systest	8.3							
systest/ssr_util	2.8							
install utilities	1.0	X	X	X	X	X	X	X
domain_examples	3.5							
optional fonts	1.4			X	X	X	X	X
lib	8.4	X	X	X	X	X	X	X
etc base	11.3	X	X	X	X	X	X	X
guaranteed commands	.3	X	X	X	X	X	X	X
com	3.3	X	X	X			X	X
sys base	17.6	X	X	X	X	X	X	X
sys/help	1.9							
sys/ins	1.1		X	X			X	X
sys/source	0.3							
usr base	9.4	X	X	X	X	X	X	X
usr/apollo/include	1.2		X	X	X	X	X	X
usr/apollo/man	3.4							
usr/X11	8.4							
usr/games	3.1							
usr/new	5.8							
bsd4.3 base	1.9				X		X	
bsd4.3 usr	15.7				X		X	
sys5.3 base	2.3					X		X
sys5.3 usr	16.7					X		X
Total (Approximate)	160.6	58.0	55.7	57.1	70.3	71.9	74.7	76.3

NOTE: An additional minimum of 12 MB of free space must be available during the installation from media.
Sizes for SAU11, SAU12, and SAU14 machines may be up to 10% larger. (See note on page 2-9)

TABLE 2-4. AA Size for Large Selections (SAU10)

Component	Size (MB)	Selection File Code					
		AL	BL	VL	ABL	AVL	ABVL
sysboot	0.01	X	X	X	X	X	X
SAU10 base	6.7	X	X	X	X	X	X
SAU10 diagnostics	24.6	X	X	X	X	X	X
systest	8.3	X	X	X	X	X	X
systest/ssr_util	2.8	X	X	X	X	X	X
install utilities	1.0	X	X	X	X	X	X
domain_examples	3.5	X	X	X	X	X	X
optional fonts	1.4	X	X	X	X	X	X
lib	8.4	X	X	X	X	X	X
etc base	11.3	X	X	X	X	X	X
guaranteed commands	.3	X	X	X	X	X	X
com	3.3	X			X	X	X
sys base	17.6	X	X	X	X	X	X
sys/help	1.9	X			X	X	X
sys/ins	1.1	X			X	X	X
sys/source	0.3	X			X	X	X
usr base	9.4	X	X	X	X	X	X
usr/apollo/include	1.2	X	X	X	X	X	X
usr/apollo/man	3.4		X	X	X	X	X
usr/X11	8.4	X	X	X	X	X	X
usr/games	3.1		X	X	X	X	X
usr/new	5.8		X	X	X	X	X
bsd4.3 base	1.9		X		X		X
bsd4.3 usr	14.7		X		X		X
sys5.3 base	2.5			X		X	X
sys5.3 usr	16.7			X		X	X
Total (Approximate)	160.6	196.1	134.8	167.8	141.4	141.8	160.6

NOTE: An additional minimum of 12 MB of free space must be available during the installation from media.
Sizes for SAU11, SAU12, and SAU14 machines may be up to 10% larger. (See note on page 2-9)

2.7.3 Software Loaded into the Authorized Area

The following subsections describe what is loaded (or not loaded) into the Authorized Area for each of the canned selection files for base software that we ship, and provide information on the size of the software that is loaded.

2.7.3.1 Small Aegis (aa.aegis_small)

This is a minimum Aegis environment and does not include any tools for program development. You get the following:

- The /SAU/N directory
- The **/sys5.3/bin** guaranteed commands used to install third-party applications
- The **/usr/apollo/bin** commands
- The Apollo network administration utilities (**cpboot**, **edns**, **lcnet**, **netmain**, **probenet**), routing tools, and registry tools
- Support for printing but not in a mixed network (SR9.7 with Domain/OS)

You do not get the following:

- **/domain_examples**
- **/systest** or **/systest/ssr_util**
- **/sys/help**
- **/sys/source**
- A large set of optional fonts
- TCP/IP administration utilities
- TCP/IP user utilities (such as **ftp**, **telnet**)
- Font editing utilities
- Some of **/com**, including:
  - — Open System Toolkit utilities (**crty**, **crtyobj**)
  - — Serial line communication commands (**em3270.*xxx***, **siorf**, **siotf**)
  - — Spelling checker software (**fserr**)
- Any programming tools, including the high-level debugger (**dde**), **/com/db**, or any include files (**/sys/ins** or **/usr/include**)

The small Aegis selection (**aa.aegis_small**) requires approximately 64 MB (50 for SAU10 nodes). See column AS in Tables 2-1 and 2-3.

2.7.3.2 Small Aegis for Programmers (aa.aegis_small_prog)

This is a minimum Aegis environment with support for software development. You get everything described in Small Aegis (**aa.aegis_small**), with these additions:

- The high-level debugger (**dde**) and **/com/db**.
- All of these include files:
  - — **/sys/ins** (***.ins.*** files for Domain/OS calls)
  - — **/usr/include/apollo** (***.h** files for Domain/OS calls)

— **/usr/include** (***.h** files for BSD or SysV calls)

The small Aegis selection for programmers (**aa.aegis_small_prog**) requires approximately 66 MB (53 for SAU10 nodes). See column ASP in Tables 2-1 and 2-3.

2.7.3.3 Medium Aegis (aa.aegis_medium)

This is a more complete Aegis environment. You get everything described in Small Aegis for Programmers (**aa.aegis_small_prog**), with these additions:

- Support for printing in a mixed (SR9.7 with Domain/OS) network
- The large set of optional fonts
- TCP/IP administration utilities
- TCP/IP user utilities (**ftp, telnet**)
- The font editing utilities
- All of standard **/com**, including these:
  - — Open System Toolkit utilities (**crty, crtyobj**)
  - — Serial line communication commands (**em3270.***xxx*, **siorf, siotf**)
  - — Spelling checker software (**fserr**)

The Aegis medium selection (**aa.aegis_medium**) requires approximately 69 MB (53 for SAU10 nodes). See column AM in Tables 2-1 and 2-3.

2.7.3.4 Large Aegis (aa.aegis_large)

This selection includes everything available in an Aegis environment. In addition to the things in Medium Aegis (**aa.aegis_medium**), it picks up the following:

- Hardware diagnostics
- **/systest,** including **/systest/ssr_util**
- **/domain_examples**
- **/sys/help**
- **/sys/source**

The Aegis large selection (**aa.aegis_large**) requires approximately 100 MB (109 for SAU10 nodes). See column AL in Tables 2-2 and 2-4.

2.7.3.5 Medium BSD (aa.bsd4.3_medium) and Medium SysV (aa.sys5.3_medium)

These are fairly light BSD or SysV environments. They support program development but do not include manual pages. You get:

- The **SAU/N** directory
- The **/sys5.3/bin** guaranteed commands used to install third-party applications (These are part of standard SysV environment anyway)

- The Apollo network administration utilities (**cpboot**, **edns**, **lcnet**, **netmain**, **probenet**), routing tools, and registry tools.
- All standard **bsd4.3** or **sys5.3** trees except where noted below
- **/usr/apollo/bin** commands
- The large set of optional fonts
- The high-level debugger (**dde**)
- TCP/IP administration files and utilities
- TCP/IP utilities (such as **ftp**, **rlogin**)
- **/usr/include** (***.h** files for BSD or SysV calls)
- **/usr/include/apollo** (***.h** files for Domain/OS calls)
- Support for UNIX mail
- Support for UNIX printing
- Support for UNIX program development (**ld**, **make**, **sccs**),

You do not get the following:

- **/domain_examples**
- **/systest** or **/systest/ssr_util**
- Support for Domain hardcopy (printing)
- Font editing utilities
- **/usr/apollo/man** (manual pages for Domain/OS calls)
- **/usr/man**
- **/usr/games**
- **/usr/new**
- Support for UUCP
- Support for BSD or SysV graphics
- Support for BSD or SysV text processing (**nroff**, **troff**)

The BSD medium selection (**aa.bsd4.3_medium**) requires approximately 80 MB (68 for SAU10 nodes). See column BM in Tables 2-1 and 2-3. The SysV medium selection (**aa.sys5.3_medium**) requires approximately 84 MB (68 for SAU10 nodes). See column VM in Tables 2-1 and 2-3.

2.7.3.6 Large BSD (aa.bsd4.3_large) and Large SysV (aa.sys5.3_large)

These selections include everything available in the respective environments. They pick up all the things listed as not included for the medium UNIX environments above.

The large BSD selection (**aa.bsd4.3_large**) requires approximately 119 MB (130 for SAU10 nodes). See column BL in Tables 2-2 and 2-4. The large SysV selection (**aa.sys5.3_large**) requires approximately 122 MB (134 for SAU10 nodes). See column VL in Tables 2-2 and 2-4.

2.7.3.7 Combination Medium Selection Files

The medium combined selections, Aegis/BSD (**aa.aegis_bsd4.3_medium**) and Aegis/SysV (**aa.aegis_sys5.3_medium**), are direct concatenations of the individual ones listed above except that they do not include the font utilities that medium Aegis (**aa.aegis_medium**) includes.

The combined Aegis and BSD medium selection (**aa.aegis_bsd4.3_medium**) requires approximately 85 MB (72 for SAU10 nodes). See column ABM in Tables 2-1 and 2-3. The combined Aegis and SysV medium selection (**aa.aegis_sys5.3_medium**) requires approximately 88 MB (76 for SAU10 nodes). See column AVM in Tables 2-1 and 2-3.

2.7.3.8 Combination Large Selection Files

The large combined selections, Aegis/BSD (**aa.aegis_bsd4.3_large**), Aegis/SysV (**aa.aegis_sys5.3_large**), and Aegis/BSD/SysV (**aa.large**), include everything available in the member environments.

The combined large selection for all three environments (**aa.large**) requires approximately 144 MB (157 for SAU10 nodes). See column ABVL in Tables 2-2 and 2-4.

2.8 Media Types

We distribute SR10.4 on streaming cartridge tapes and magnetic tapes as follows:

Cartridge tape distributions:

m68k version:

CRTG_STD_SFW_1
CRTG_STD_SFW_2
CRTG_STD_SFW_3
CRTG_STD_SFW_4
CRTG_STD_SFW_BOOT_1

a88k version:

CRTG_STD_SFW_1
CRTG_STD_SFW_2
CRTG_STD_SFW_3
CRTG_STD_SFW_4
CRTG_STD_SFW_BOOT_1

Magnetic tape distributions:

m68k version:

MT_STD_SFW_1
MT_STD_SFW_2
MT_STD_SFW_3
MT_STD_SFW_4
MT_STD_SFW_5
MT_STD_SFW_6
MT_STD_SFW_7
CRTG_STD_SFW_BOOT_1

a88k version:

MT_STD_SFW_1
MT_STD_SFW_2
MT_STD_SFW_3
MT_STD_SFW_4
MT_STD_SFW_5
MT_STD_SFW_6
MT_STD_SFW_7
CRTG_STD_SFW_BOOT_1

Chapter 3: Documentation

This chapter lists the documents that are new or have been revised since SR10.3. It also includes any changes or corrections to documentation that we were not able to update for this release. For a list of manuals that were new or revised at SR10.3, see Chapter 3 of the SR10.3 Release Notes in **/install/doc/apollo/os.v.10.3__notes.**

Note that release documents for optional products can be found online in the **/install/doc/apollo** directory.

3.1 New or Revised Documents

New and revised documents that we are introducing at this release include:

Domain/OS System Administration Guide	(019001-A00)
Domain/OS System Administration Reference	(019208-A00)
Domain/OS Standards Compliance Document	(019207-A00)
Installing Domain Software [former title: Installing Software with Apollo's Release and Installation Tools]	(008860-A03)
Printing in the Domain/OS Environment: System Administrator's and Programmer's Guide	(011774-A02)
Managing NCS Software	(011895-A02)
Configuring and Managing TCP/IP	(008543-A03)
Using TCP/IP Network Applications	(008667-A01)
Domain/OS BSD Command Reference	(005800-A02)
Domain/OS SysV Command Reference	(005798-A02)
Domain/OS BSD Programmer's Reference	(005801-A01)
Domain/OS SysV Programmer's Reference	(005799-A01)
Domain/OS Call Reference Volume 1	(007196-A01)
Domain/OS Call Reference Volume 2	(012888-A01)
Domain/OS Call Reference Volume 3	(019409-A00)
Administering the Optical Disk Library System on Domain/OS	(018904-A00)
HP Series 6300 Model 20GB/A Optical Disk Library System CE Installation Guide for Apollo Sites	(018905-A00)
Domain Distributed Debugging Environment Reference	(011024-A01)
Porting your GPR Application to Xlib	(019579-A00)
Using the X Window System (For Xdomain; for Xapollo information, see	(B1171-90043)

Using the X Window System on Apollo Workstations (015213-A02))	
HP VUE System Administration Manual	(B1171-90044)
HP VUE User's Guide	(B1171-90042)
HP VUE Configuration Guide for Domain/OS Systems	(B1171-90046)
HP Series 6100 Model 700/S CD-ROM Drive User's Guide	(A1999-90002)
HP Series 6100 Model 700/S CD-ROM Drive Service Handbook	(A1999-90003)

3.2 Tech Pubs Connection Telephone Number Change

Please note that the Tech Pubs Connection telephone number given in pre-SR10.4 manuals *for calls from outside the U.S.* has been changed. The new number is:

(508) 256-6600 ext. 4965

3.3 New and Updated Manual Pages and Help Files

The following sections provide lists of manpages that are new or have been revised for this release.

3.3.1 New or Revised Domain SysCall Man Pages

The following Domain SysCall man pages are new or revised at SR10.4:

cache_intro.a
cd_cvd.a
cd_defs.a
cd_drec.a
cd_idmap.a
cd_nmconv.a
cd_ptrec.a
cd_pvd.a
cd_type.a
cd_xar.a
ctm_alloc_pv.a
ctm_find_color.a
ctm_inc_use_count.a
ctm_inq_curr_color_map.a
ctm_intro.a
ctm_mark_read_only.a
ctm_release_pv.a
ctm_set_curr_color_map.a
fault_intro.a
fpp_control.a
ios_intro.a
ios_open.a
mbx_put_rec.a
mbx_put_rec_cond.a
ms_addmap.a
ms_addmapx.a
ms_crmapl.a
ms_crtemp.a
ms_intro.a
ms_maplx.a
ms_relock.a
ms_reprotect.a
pad_intro.a
prf_edit_job.a
prf_get_printers.a
prf_get_sites.a
prf_intro.a
prf_name_print.a
prf_queue_file.a
prf_set_option.a

prf_signal_printer.a
prf_stream_print.a
sio_inquire.a
sio_intro.a
vec_dmult_sub_vector.a
vec_dmult_sub_vector_i.a
vec_imult_add_vector16_i.a
vec_imult_sub_vector.a
vec_imult_sub_vector16.a
vec_imult_sub_vector16_i.a
vec_imult_sub_vector_i.a
vec_mult_sub_vector.a
vec_mult_sub_vector_i.a

3.3.2 New or Revised Aegis Help Files

The following Aegis Help files are new or revised at SR10.4:

acadmin.hlp
arp.hlp
audit.hlp
bind.hlp
calls.hlp
cddrec.hlp
cdfsd.hlp
cdfsmount.hlp
cdmntsuppl.hlp
cdptrec.hlp
cdvd.hlp
cdxar.hlp
chuvol.hlp
cpf.hlp
cpt.hlp
crp.hlp
crpad.hlp
ctnode.hlp
dcalc.hlp
dde.hlp
dmlock.hlp
dmtvol.hlp
drm_admin.hlp
dtcb.hlp
ed.hlp
edacl.hlp
edmtdesc.hlp
edns.hlp
edquota.hlp
ex.hlp
fmt/commands.hlp
ftn.hlp
ftp.hlp
glb_obj.txt.hlp
glb_site.txt.hlp
glbd.hlp
invol.hlp
lb_admin.hlp
ld.hlp
limits.hlp
llbd.hlp
lprot.hlp
lprotect.hlp
ls_tv.hlp
lvolfs.hlp
manuals.hlp
mbd.hlp
mkdev.hlp
mtvol.hlp
named-xfer.hlp
netstat.hlp
netsvc.hlp
nrglbd.hlp
nshost.hlp

pio.hlp
prf.hlp
prsvr/config.hlp
pst.hlp
quotaon.hlp
rbak.hlp
resolver.hlp
route.hlp
rpccp.hlp
rpcd.hlp
salvol.hlp
set.hlp
setprot.hlp
sigp.hlp
spm.hlp
stcode.hlp
tb.hlp
tcp.hlp
tcpd.hlp
tcpstat.hlp
tctl.hlp
telnet.hlp
trpt.hlp
uctnode.hlp
uuid_gen.hlp
uuidgen.hlp
uuidname.txt.hlp
vt100/unix.hlp
vt100.hlp

3.3.3 New or Revised SysV Man Pages

The following SysV man pages are new or revised at SR10.4:

ar.1
at.1
bsh.1
catdump.1
cc.1
cddrec.1
cdptrec.1
cdvd.1
cdxar.1
cpp.1
crp.1
crpad.1
csh.1
cxref.1
date.1
dde.1
df.1
dmlock.1
f77.1
ftp.1c
gencat.1
iconv.1

ksh.1
ld.1
lex.1
lint.1
login.1
lp.1
mail.1
make.1
netstat.1
newaliases.1
pax.1
prf.1
ps.1
rbak.1
stcode.1
stty.1
tar.1
tb.1
telnet.1c
tftp.1c
uucp.1c
uuencode.1c
uustat.1c
uuto.1c
uux.1c
vi.1
vt100.1
who.1
yacc.1
access.2
alarm.2
chdir.2
chmod.2
chown.2
chroot.2
close.2
creat.2
domain.2
dup.2
exec.2
exit.2
fcntl.2
fork.2
fsync.2
getgroups.2
getpid.2
getuid.2
ioctl.2
kill.2
link.2
lseek.2
madvise.2
mkdir.2
mknod.2
mmap.2
mount.2
mprotect.2
nice.2
open.2
pause.2
pipe.2
poll.2
quota_read.2
read.2
readlink.2
rename.2
rmdir.2
setgroups.2
setpgid.2
setsid.2
setuid.2
sigaction.2
signal.2
sigpending.2
sigprocmask.2
sigsuspend.2
stat.2
statfs.2
swapon.2
time.2
times.2
truncate.2
ulimit.2
umask.2
uname.2
unlink.2
utime.2
wait.2
write.2
abort.3c

abs.3c
assert.3x
bessel.3m
bsearch.3c
catclose.3c
catgets.3c
catopen.3c
cfgetispeed.3
cfgetospeed.3
cfsetispeed.3
cfsetospeed.3
cftime.3x
clearenv.3
clearenv.3c
clock.3c
conv.3c
ctermid.3s
ctime.3c
ctype.3c
curses.3x
cuserid.3s
dial.3c
difftime.3c
directory.3x
div.3c
domain.3
drand48.3c
dup2.3c
erf.3m
exp.3m
fclose.3s
flockfile.3s
floor.3m
fopen.3s
fread.3s
frexp.3c
fseek.3s
ftw.3c
funlockfile.3s
gamma.3m
getc.3s
getclock.3
getcwd.3c
getenv.3c
getgrent.3c
getlogin.3c
getopt.3c
getorgent.3c
getpass.3c
getpwent.3c
gets.3s
gettimer.3
hsearch.3c
hypot.3m
intro.3
isnan.3c
localeconv.3c
lsearch.3c
malloc.3c
matherr.3m
memory.3c
mkfifo.3
mktime.3c
mktimer.3
msem_init.3
msem_lock.3
msem_remove.3
msem_unlock.3
nl_langinfo.3c
pathconf.3
perror.3c
popen.3s
printf.3s
pthread_attr_create.3p
pthread_attr_delete.3p
pthread_attr_getstacksize.3p
pthread_attr_setstacksize.3p
pthread_cancel.3p
pthread_cleanup_pop.3p
pthread_cleanup_push.3p
pthread_cond_broadcast.3p
pthread_cond_destroy.3p
pthread_cond_init.3p
pthread_cond_signal.3p
pthread_cond_timedwait.3p
pthread_cond_wait.3p
pthread_condattr_create.3p
pthread_condattr_delete.3p
pthread_create.3p
pthread_detach.3p

pthread_equal.3p
pthread_exit.3p
pthread_getspecific.3p
pthread_join.3p
pthread_keycreate.3p
pthread_mutex_destroy.3p
pthread_mutex_init.3p
pthread_mutex_lock.3p
pthread_mutex_trylock.3p
pthread_mutex_unlock.3p
pthread_mutexattr_create.3p
pthread_mutexattr_delete.3p
pthread_once.3p
pthread_self.3p
pthread_setasynccancel.3p
pthread_setcancel.3p
pthread_setspecific.3p
pthread_testcancel.3p
pthread_yield.3p
putc.3s
putenv.3c
puts.3s
qsort.3c
raise.3c
rand.3c
reltimer.3
remove.3c
rmtimer.3
scanf.3s
setbuf.3s
setchrclass.3x
setclock.3
setjmp.3c
setlocale.3c
siglongjmp.3
sigsetjmp.3
sigsetops.3
sinh.3m
sleep.3c
stdarg.3c
stdio.3s
string.3c
strtod.3c
strtol.3c
sysconf.3
system.3s
tcdrain.3
tcflow.3
tcflush.3
tcgetattr.3
tcgetpgrp.3
tcsendbreak.3
tcsetattr.3
tcsetpgrp.3
thread_abort.3t
thread_cleanup.3t
thread_create.3t
thread_handle_signals.3t
thread_info.3t
thread_inhibit.3t
thread_resume.3t
thread_self.3t
thread_set_priority.3t
thread_startup.3t
thread_state.3t
thread_suspend.3t
thread_terminate.3t
threadp_get.3t
threadp_init.3t
threadp_set.3t
tmpfile.3s
tmpnam.3s
trig.3m
tsearch.3c
ttyslot.3c
ungetc.3s
unlocked_getc.3s
unlocked_putc.3s
vprintf.3s
wctomb.3c
acct.4
aliases.4
domain.4
float.4
fstab.4
gettytab.4
glb_obj.txt.4
glb_site.txt.4
group.4
limits.4

locale.4
mnttab.4
passwd.4
profile.4
resolver.4
term.4
terminfo.4
utmp.4
uuidname.txt.4
acadmin.1m
arp.1m
buildlang.1m
cdfsd.1m
cdfsmount.1m
cdmntsuppl.1m
chuvol.1m
ctnode.1m
domain.1m
drm_admin.1m
dtcb.1m
edns.1m
edquota.1m
glbd.1m
invol.1m
lb_admin.1m
llbd.1m
lprot.1m
lprotect.1m
ls_tv.1m
mbd.1m
mkdev.1m
mount.1m
named-xfer.1m
netsvc.1m
nodestat.1m
nrglbd.1m
nshost.1m
quotaon.1m
rlogind.1m
route.1m
rpccp.1m
rpcd.1m
sendmail.1m
setprot.1m
spm.1m
stcode.1m
swap.1m
swapon.1m
syslogd.1m
tcpd.1m
tftpd.1m
timed.1m
timedc.1m
trpt.1m
uctnode.1m
uucheck.1m
uucico.1m
uuclean.1m
uuid_gen.1m
uuidgen.1m
uusched.1m
pio.7
tcp.7
termios.7

3.3.4 New or Revised BSD Man Pages

The following BSD man pages are new or revised at SR10.4:

ar.1
bsh.1
catdump.1
cc.1
cddrec.1
cdptrec.1
cdvd.1
cdxar.1
cpp.1
crp.1
crpad.1
csh.1
date.1
dde.1
df.1
dmlock.1
f77.1
ftp.1c
gencat.1
iconv.1
ksh.1
ld.1
lex.1
lint.1
ln.1
login.1
mail.1
make.1
man.1
netstat.1
pax.1
prf.1
ps.1
rbak.1
sort.1
stcode.1
tar.1
tb.1
telnet.1c
tftp.1c
uucp.1c
uuencode.1c
uustat.1c
uuto.1c
uux.1c
vi.1
vt100.1
who.1
yacc.1
accept.2
access.2
acct.2
adjtime.2
bind.2
brk.2
chdir.2
chmod.2
chown.2
connect.2
creat.2
domain.2
execve.2
getdents.2
getgroups.2
gethostname.2
getitimer.2
getpeername.2
getrlimit.2
getrusage.2
getsockname.2
getsockopt.2
gettimeofday.2
intro.2
kill.2
link.2
lseek.2
madvise.2
mkdir.2
mknod.2
mmap.2
mount.2
mprotect.2

open.2
pipe.2
ptrace.2
quota_read.2
read.2
readlink.2
recv.2
rename.2
rmdir.2
send.2
setgroups.2
setpgid.2
setsid.2
sigaction.2
sigpause.2
sigpending.2
sigprocmask.2
sigreturn.2
sigsetmask.2
sigstack.2
sigsuspend.2
sigvec.2
socketpair.2
stat.2
statfs.2
swapon.2
symlink.2
truncate.2
unlink.2
utimes.2
wait.2
write.2
abort.3
abs.3
assert.3
atof.3
bsearch.3
catclose.3
catgets.3
catopen.3
cfgetispeed.3
cfgetospeed.3
cfsetispeed.3
cfsetospeed.3
clock.3
ctime.3
ctype.3
difftime.3
div.3
domain.3
execl.3
exit.3
fclose.3s
flockfile.3s
floor.3m
fopen.3s
fread.3s
fseek.3s
funlockfile.3s
getenv.3
getgrent.3
getlogin.3
getmntent.3
getorgent.3
getpass.3c
intro.3
localeconv.3
malloc.3
matherr.3m
memory.3
mkfifo.3
mktime.3
nice.3c
nl_langinfo.3
pathconf.3
perror.3
printf.3s
pthread_attr_create.3p
pthread_attr_delete.3p
pthread_attr_getstacksize.3p
pthread_attr_setstacksize.3p
pthread_cancel.3p
pthread_cleanup_pop.3p
pthread_cleanup_push.3p
pthread_cond_broadcast.3p
pthread_cond_destroy.3p
pthread_cond_init.3p
pthread_cond_signal.3p
pthread_cond_timedwait.3p
pthread_cond_wait.3p

pthread_condattr_create.3p
pthread_condattr_delete.3p
pthread_create.3p
pthread_detach.3p
pthread_equal.3p
pthread_exit.3p
pthread_getspecific.3p
pthread_join.3p
pthread_keycreate.3p
pthread_mutex_destroy.3p
pthread_mutex_init.3p
pthread_mutex_lock.3p
pthread_mutex_trylock.3p
pthread_mutex_unlock.3p
pthread_mutexattr_create.3p
pthread_mutexattr_delete.3p
pthread_once.3p
pthread_self.3p
pthread_setasynccancel.3p
pthread_setcancel.3p
pthread_setspecific.3p
pthread_testcancel.3p
pthread_yield.3p
qsort.3
raise.3
rand.3c
remove.3
scanf.3s
setbuf.3s
setlocale.3
setuid.3
siglongjmp.3
signal.3c
sigsetjmp.3
sigsetops.3
sleep.3
stdarg.3
stdio.3s
string.3
sysconf.3
system.3
tcdrain.3
tcflow.3
tcflush.3
tcgetattr.3
tcgetpgrp.3
tcsendbreak.3
tcsetattr.3
tcsetpgrp.3
thread_abort.3t
thread_cleanup.3t
thread_create.3t
thread_handle_signals.3t
thread_info.3t
thread_inhibit.3t
thread_resume.3t
thread_self.3t
thread_set_priority.3t
thread_startup.3t
thread_state.3t
thread_suspend.3t
thread_terminate.3t
threadp_get.3t
threadp_init.3t
threadp_set.3t
tmpfile.3s
tmpnam.3s
unlocked_getc.3s
unlocked_putc.3s
varargs.3
vprintf.3s
wctomb.3
intro.4
pio.4
tcp.4p
termios.4
aliases.5
float.5
gettytab.5
glb_obj.txt.5
glb_site.txt.5
limits.5
locale.5
resolver.5
types.5
utmp.5
uuidname.txt.5
limits.7
math.7
stddef.7

acadmin.8
arp.8c
buildlang.8
cdfsd.8
cdfsmount.8
cdmntsuppl.8
chuvol.8
ctnode.8
drm_admin.8
dtcb.8
edmtdesc.8
edns.8
edquota.8
glbd.8
invol.8
lb_admin.8
llbd.8
lprot.8
lprotect.8
ls_tv.8
mbd.8
mkdev.8
mkdevno.8
mount.8
named-xfer.8c
netsvc.8
nodestat.8
nrglbd.8
nshost.8
quotaon.8
rlogind.8c
route.8c
rpccp.8
rpcd.8
sendmail.8
setprot.8
spm.8
stcode.8
swap.8
swapon.8
tcpd.8
tftpd.8c
timed.8
timedc.8
trpt.8c
uctnode.8
uucheck.8c
uucico.8c
uuclean.8c
uuid_gen.8
uuidgen.8
uusched.8c

3.3.5 kbm Manual Page

Toggle mode can be specified for the mouse buttons (m1-m16). Users of trackball devices may find this useful. Note that mouse buttons can only be specified with the **-T** option.

3.3.6 Changes to Domain/PCC Documentation

If you use the alternate address range for a DPCC board on your Series 400 system, disregard the instructions about not changing the DMA channel jumpers in the manual *Installing Domain/PCC in Your Series DN3000/DN4000.*

For Series 400 systems, select the alternate DMA channel 7 by moving the two DMA jumpers to the leftmost pins. Refer to Figure 6 in the manual for the location of the DMA jumper pins. For more information about alternate jumper configurations for Series 400 option boards, refer to the "HP Apollo 9000 Model 400s and 433s Configuration Worksheet", which ships with Series 400 systems.

3.3.7 Correction to SR10.3 Release Notes

Section 1.6.3 of the SR10.3 *Domain System Software Release Notes* (017957-A01) introduced a new Display Manager feature (**shut_lock**), but gave an incorrect pathname for the file. The correct pathname should be **/sys/node_data/etc/dm_display/shut_lock**.

3.4 New GPR-to-X Porting Manual and Examples

Customers interested in porting Domain Graphics Primitives Resource (GPR) application programs to Xlib can read the new manual *Porting Your GPR Application to Xlib.*

The manual is organized similarly to the GPR programming manual, *Programming with Domain Graphics Primitives* (005808-A00). It compares GPR routines with Xlib routines and shows which GPR routines have Xlib equivalents and which ones do not. The manual also describes how to simulate some GPR features in Xlib when a direct Xlib correspondent does not exist.

In the process of writing this manual, we have ported many of the original GPR online examples to Xlib. These programs (in C only) are online under **/domain_examples/gpr-to-x_examples**.

3.5 Changes to X Window System Documentation

The following subsections describe changes to *Using the X Window System on Apollo Workstations* (015213-A02):

Change p. 2-3 by adding this paragraph just before Section 2.2.1:

> Note that you may have problems starting X at boot time using the xdm client, which was not available when this manual was written. This manual assumes that

the DM is running.

Change p. 2-6 to add the following comment just after the sentence that begins "Your site administrator":

> Beware that the .login file is run on every login, including rlogin/telnet from a remote system, so you might not want to start X from the .login file at your site.

Change p. 2-7 to add this note at the bottom of the page:

> Be sure to check that Xapollo is not already running before you issue the "startx" command.

Change p. 2-12 to add this note after paragraph 2:

> You have to put the cursor in a DM pad to return from a "dmio -off" command. The "dmio" command can only be used with the DM. Input is sent to the DM when the cursor is in a DM pad or the cursor is in the root window and the DM owns the root. The "dmio" command won't work if X owns the root and no DM windows are visible: in this case, you will have to issue the "dmwin" command and then use "xdmc dmio -on" to restart DM window management.

Change p. 4-9 to add this after the sentence "The remove subcommand undoes the assignment on a per-key basis":

> The -e (expression) subcommand is also useful, as in the following command that adds F0 as a meta key:

```
    -xmodmap -e "add mod1 F10"
```

Change p. 4-9 to insert these lines after "clear Mod1" in the lines from the xmodmap.kb2sample files:

```
!
! "Relabel" key F0 (aka F10) as Meta_L
keycode 0x93 = Meta_L
!
! Make F0 act as the Meta  modifier
add Mod1 = Meta_L
! make Alt_L also be Meta
add Mod1 = Alt_L
```

Change p. 3-33 to include this sentence after the one beginning "You can put these escape sequences":

> **seltek** and **selvt** allow the user to switch command input/output to the desired window without using the pop-up xterm menus.

Change p. 3-33 to remove the word "line" in the third line from the bottom, which begins with "alias seltek."

3.6 Changes to HP Series 6100 Model 700/S CD-ROM User's Guide

(The changes described in this section are applicable to the SR10.3 version of the *HP Series 6100 Model 700/S CD-ROM User's Guide* (MFG No. A1999-90602). If you have the SR10.4 version of this manual (MFG No. A1999-90609), the changes listed here have been incorporated therein, and you can ignore this section.)

For SR10.4, we've made some changes to information in the HP Series 6100 Model 700/S User's Guide. These changes update the prerequisites for using the drive on your system type and explain how to mount a CD-ROM drive that's connected to an SR10.4 system. The changes are as follows:

Delete the last two sentences in the paragraph that precedes Table 1.

In Table 1-1, "Prerequisites for Reading from a CD-ROM Drive on Domain/OS Systems", make the following changes under the column "Minimum OS plus PSK" for the systems indicated in the "System" column:

- For all systems except Series 10000, change the information listed under to column "Miminum OS plus PSK" to "SR10.3 + LFC50BAD (SR10.3.5)".
- For Series 10000TX systems, change "SR10.3.p + TBD" to "SR10.3 + LFC50BBD (SR10.3.5)".
- Change "Series 10000TX", listed under the System column, to "Series 10000".
-

Delete footnotes 3 and 4 at the bottom of Table 1-1.

Insert the following text before Table 1-2:

> Table 1-2 shows the minimum Boot ROM revisions required if you want to boot software from the CD-ROM drive. Hewlett-Packard does not suport bootable systems on Domain, but certain third parties or Independant Software Vendors (ISVs) may provide bootable software on a CD-ROM disc. Use the information in Table 1-2 if you have a bootable CD-ROM disc from one of these vendors. Note that the CD-ROM drive can only boot to the systems that are listed in Table 1-2.

In Table 1-2, "Boot ROM Revisions for Booting from a CD-ROM Drive on Domain/OS

Systems", change the minimum boot ROM revision for booting from a Series 2500 from 4.0 to 4.1.

Delete the notice immediately following Table 1-2.

On page 1-14, under the subsection "Connecting the CD-ROM Drive", replace the existing notice with the following text:

> NOTICE: You cannot use a CD-ROM drive in a Series 35xx, 4000, or 4500 system that uses a non-SCSI cartridge tape drive. You must either remove the ctape controller from the system or replace your non-SCSI ctape drive with a SCSI ctape drive.

On page 1-18, insert the following text before Step 1:

> NOTICE: Systems that run the SR10.4 operating system and connect the CD-ROM drive to SCSI controller 0 are preconfigured with CD-ROM device files named "cdrom" through "cdrom_5" (as shown in Table 1-8). If your system meets these criteria, go directly to Step 3 without performing Steps 1 and 2.

In Step 2 on page 1-19, delete the last sentence on the page and replace it with the following paragraph:

> To find the minor_device_number, refer to Table 1-9 or 1-10, depending on your operating system. Use the table to cross-reference the system's SCSI controller number with the CD-ROM drive's target address. For example, for an SR10.4 system with a SCSI controller number = 0 and a CD-ROM drive SCSI ID = 3, the minor_device_number = 6144.

On page 1-20, change the title of Table 1-9 to "SR10.3 minor_device_number Matrix".

Add the following table and notice after the notice on page 1-20:

Table 1-10. SR10.4 minor_device_number Matrix

Drive Target	**SCSI Controller Number**			
Address	**0**	**1**	**2**	**3**
0	0	32768	65536	98304
1	2048	34816	67584	100352
2	4096	36864	69632	102400
3	6144	38912	71680	104448
4	8192	40960	73728	106496
5	10240	43008	75776	108544
6	12288	45056	77824	110592

NOTICE: For SR10.4 systems, the SCSI controller number is usually 0, unless you have more than one SCSI controller in your system.

In Step 3 on page 1-21, replace the phrase "**llbd** and **cdfsd** files" with "**llbd**, **cdfsd**, and **xpager** (SR10.4 systems only) files".

On page 2-10, add the following step 1 before the current step 1. Change the existing steps 1 through 3 to steps 2 through 4.

1. If your system is running SR10.3 or SR10.3.p, go to step 2 without performing this step. If your system is running SR10.4, you must create a mount directory (for example, /cd_mountpoint) to define where to access the CD-ROM file system. To create a mount directory, type one of the following commands:

```
SysV:   $ mkdir mount_directory_name
BSD:    % mkdir mount_directory_name
Aegis:  $ crd mount_directory_name
```

Chapter 4: Bugs, Limitations, and SRs

This chapter describes software problems and limitations known to exist in SR10.4. See the SR10.3 release notes for information about software problems and limitations at that release.

4.1 Domain/OS Problems Fixed

New with SR10.4, we are providing a Domain Software Release Bulletin (SRB) on the SR10.4 media. The SRB documents all customer reported known problems that have been resolved with SR10.4. It also covers some associated optional products such as the compilers. A complete list of the products described in the SRB can be found in the software release contents of the SRB. The SRB is located in the following files:

/install/doc/apollo/os.v.10.4__software_release_bulletin (m68k version)
/install/doc/apollo/os.v.10.4.p__software_release_bulletin (a88k version)

4.2 System Software Notice

See the *SR10.4 System Software Notice (Addendum to the Domain/OS System Software Release Document)* for information discovered after this Release Document went to press.

4.3 Domain/OS Bugs and Limitations

This section contain information on limitations and known bugs in Domain/OS.

4.3.1 Domain/OS Bugs

This section contains a list and description of known bugs in Domain/OS.

4.3.1.1 DM Bug

Attempts to execute the Display Manager **xi -f** command on a 9000/425e workstation with either a color or greyscale monitor fail with the following error message:

```
    (XI) /tmp/screen.gmf - Wrong display hardware from (graphics / primitives)
```

This results in a corrupted locked output file.

4.3.1.2 mrgri Problems

4.3.1.2.1 Merging X Window System Environments

At SR10.4, the X Window System environments on the m68k and PRISM systems differ substantially. As a result, attempts to merge a m68k and PRISM release index with the **mrgri** tool will produce warnings of the form:

```
Copying from //mako/mrgri_disk/install/ri.apollo.os.v.10.4/usr/X11/lib/app-defaults/XClipboard.
        to   //mako/mrgri_disk/install/ri.apollo.os.v.10.4.compexe/usr/X11/lib/app-defaults/XClipboard.
INFORMATION: Text files in primary and secondary differ. Using latest one.
```

You can ignore these warnings, and the resulting merged release index will function properly on both m68k and PRISM systems.

Note, however, that the X11R4 enhancements that are part of the m68k release will only function on m68k systems. This means that when you install from the merged release onto a DN10000 and are using the DN10000 to run the install program, you will get errors similar to these:

```
Locally running:  //mako/mrgri_disk/install/ri.apollo.os.v.10.4.compexe/install_utils/bin/ksh \
  //mako/mrgri_disk/install/ri.apollo.os.v.10.4.compexe/install_utils/mk_dde_fonts.sh //mako/disk01 \
  //mako/mrgri_disk/install/ri.apollo.os.v.10.4.compexe
//mako/mrgri_disk/install/ri.apollo.os.v.10.4.compexe/install_utils/mk_dde_fonts.sh[25]: \
2451 compound executable contains no module for machine type (process manager/loader)
ERROR:Local process call failed
status F (OS)
```

You can avoid these problems by

1. Running the install program on an m68k system and target the PRISM system or
2. Running the install program on a PRISM and selecting 'none' or configuration question.

4.3.1.2.2 Must Use -t Option to mrgri

When using **mrgri** to produce a **cmpexe** release, you must supply the **-t** option. If you do not, **mrgri** will fail to produce **cmpexe** objects correctly, and will produce error messages like the following:

```
//steamers/install/ri.apollo.nfs.v.2.2.p/sys/nfs/mountDump - name not found(stream manager/IOS)
Cmpexeing //steamers/install/ri.apollo.nfs.v.2.2.p/sys/nfs/set_flag
//steamers/install/ri.apollo.nfs.v.2.2.p/sys/nfs/set_flag - name not found(stream manager/IOS)
```

Note that the use of **-t** produces a third Authorized Area, so be sure to have sufficient disk space available.

4.3.1.3 Problem with Tab Separator

The SYS5.3 **sort** command may give incorrect results if the tab character is specified as the separator using '-t' option. In particular, if a null field is encountered (i.e., two tab characters in sequence), the field is erroneously skipped. This problem does not occur if a separator character different than tab is specified.

4.3.1.4 BSD4.3 /bin/wall Fails to Write to Users crp'd Onto a Node

The BSD4.3 version of **/bin/wall** fails to write to users who have **crp**'d onto a node. An error message of the following type is returned:

```
/dev/crp00: No such file or directory
(stream manager/IOS) error: flag not supported for this object type
```

4.3.1.5 cpio Trailer Truncated

cpio archives files alphabetically. If the last entry is empty (e.g., a temporary directory "tmp" that doesn't have any test files in it) the trailer is appended to the line with the information for "tmp", instead of starting a new line. This renders the tape non-portable to other systems, e.g., HP-UX and OSF.

4.3.1.6 telnet Problems

The following sections describe bugs in the **telnet** command.

4.3.1.6.1 Using telnet During a crp Session Sometimes Mishandles Keyboard echo/noecho

Using **telnet** during a **crp** session sometimes mishandles keyboard echo/no-echo. This is particularly apparent when trying to login to the remote system, and in some cases, the password will be echoed back to the screen inappropriately. To avoid this, use **rlogin** instead, or do not use **telnet** during a **crp** session.

4.3.1.6.2 telnetd Processes Lock Up on Bad pty

/etc/telnetd' processes will lock up in "getterminaltype" on a bad pty. This sometimes happens while creating a new **telnet** session and is related to having a corrupted pty. ptys can be corrupted by the abnormal termination of processes that use them (e.g. **xterm** and **telnet**). If this should happen, you can remake your ptys with the following command, executed as root:

```
/etc/mkdev /dev pty
```

Background jobs (started with "nohup") can lock up a pty preventing further use of that pty. If a user logs in to a node via **telnet/rlogin**, starts a background job, and then logs out, then his pty (**/dev/ttyp**?) becomes locked and future attempts to **telnet/rlogin** to that pty fail. If the user happend to have the first pty (**/dev/ttyp0**) then all future attempts to **telnet/rlogin** to that machine will fail until that job terminates. To avoid this situation, ensure that jobs started in the background either close their standard input file descriptor or re-direct it to another file.

4.3.1.6.3 Cursor Positioning Problems When in telnet Line Mode

When **telnet** is in "line" mode, the prompt is incorrectly located in the output pad and the cursor at the left hand edge of the input pad. After typing a command and return, the input text is properly moved to the output line with the prompt.

4.3.2 Domain/OS Limitations

This section contains information on limitations to Domain/OS.

4.3.2.1 'Unimplemented SVC' Error using /install/tools/install

Running **/install/tools/install** fails, returning an "Unimplemented SVC" error, when both of the following conditions are met:

- You are running the install program on a node other than a Series 400 node, and
- The node has a version of **/sys/mgrs/pio** released with an SR earlier than SR10.2.4.

Workaround: Update **/sys/mgrs/pio** from a node running SR10.2.4 or later before attempting the install. Or you can simply run the install a second time.

4.3.2.2 Must Specify Interface Configuration in /etc/rc.local

The interface configuration in **/etc/rc.local** must be properly specified in order for the name server program **/etc/named** to run correctly.

All physical interfaces present on the machine (**eth0**, **dr0**, etc.) must have a corresponding uncommented **/etc/ifconfig** statement in **/etc/rc.local**. **If /etc/rc.local** is not in **/install/preserve.list**, then the default SR10.4 **/etc/ifcon fig** statement in **/etc/rc.local** configures the **net0** interface, which works for all nodes which have only one interface (ethernet or ring). But for nodes with more than one interface, the default **/etc/rc.local** must be edited to comment the **net0** line and uncomment the appropriate lines for each<HR> specific interface present.

In addition, the local loopback interface **lo0** must also have an uncommented **/etc/ifconfig** statement in **/etc/rc.local**. If **/etc/rc.local** is not in **/install/preserve.list**, then the default SR10.4 **/etc/ifconfig** state <|,6>ment in **/etc/rc.local** will configure the **lo0** interface. But for nodes which do preserve an old **/etc/rc.local**, it must be edited to uncomment the **/etc/ifconfig** statement for **lo0.**

If the interface configuration is not correctly specified in these ways, **/etc/named** silently exits, and name server functions are not available.

4.3.2.3 DM Limitation

The DM uses the Domain mailbox subsystem to communicate with other processes. If **mbx_helper** is not running, the DM cannot terminate other processes in response to the **lo** command. If for security or other reasons you cannot permit **mbx_helper** to run on your node, you should use the X11R4 runtime environment without the DM.

mbx_helper will be started automatically during the system boot from **/etc/rc.user**. After it has been started, it can be killed only by root.

4.3.2.4 ex Command Limitation

The **ex** command is provided for system debugging purposes only. This command is not guaranteed to work properly under all circumstances and should not be used in place of the *shut* command to reboot the system.

4.3.2.5 siorf/siotf Limitation

You do not need to use the **tctl** command to set the sync and insync parameters of the SIO line when receiving a non-ASCII file. **siotf** and **siorf** recognize the types of the files being transferred and set these parameters correctly.

At SR10 and above you do need to turn off the echoing of input characters over the SIO line before implementing **siorf** and **siotf.** This can be done using the **tctl -noecho** or **stty -echo** command. This is necessary to obtain successful **siorf, siotf** handshaking (Hello,Answer_hello).

4.3.2.6 plot/tplot Limitation

The SR10.4 versions of /usr/bin/plot (BSD4.3) and /usr/bin/tplot (Sys5.3) will not run on nodes running pre-SR10.4 software; likewise, pre-SR10.4 versions of these commands will not run on an SR10.4 node.

4.3.3 Install Tools and rbak/wbak Compatibility Limitation

The pre-SR10.3 versions of the **rbak** and **wbak** commands and of the install tools are incompatible with the Series 400/425 and DN5500 workstations. Be sure to use the SR10.3, or later, versions of these tools.

Specifically, the incompatible commands are

- **/usr/apollo/bin/wbak**
- **/com/wbak**
- **/usr/apollo/bin/rbak**

- **/com/rbak**
- **install/tools/rbak_sr10**

The versions of these commands that are compatible with the Series 400/425, DN5500 workstation and other machine types are included with SR10.4. The SR10.4 versions overwrite earlier versions when you install SR10.4. Thus, the incompatibility normally should not pose a problem. Just be sure not to copy or move pre-SR10.4 versions of these commands to nodes with SR10.4 installed.

To verify that you are using the correct versions of these commands you may check their time stamps with the **ts** command. Use the following format with the **ts** command:

/usr/apollo/bin/ts *command*

Where *command* is the complete pathname of the command on which you want to check the time stamp.

The dates listed for the **rbak**, **wbak**, and **rbak_sr10** commands should be the same as, or later than the following dates:

- 1990/02/15 **rbak**
- 1990/03/19 **wbak**
- 1990/08/01 **install/tools/rbak_sr10**

4.3.4 Hardcopy Bugs and Limitations

The following subsections describe hardcopy bugs and limitations.

4.3.4.1 Hardcopy Bugs

The following sections describe hardcopy bugs.

4.3.4.1.1 prf/pre10q Problem

When using **prf** to queue a job to a print manager which is also being monitored by a **pre10q** daemon, you may see this error message :

```
?(prflib) Problem with queue - print job duplicate from pre10q daemon (US/print utility)
```

If this happens, execute

prf -r *<printer_name>*

before you try and queue the job again. You should see the job in the queue. If the job is not in the queue, and has not printed, requeue the job.

This behavior is the result of a timing window between **prflib**, the print manager and the **pre10q** daemon. The **pre10q** daemon tries to requeue SR9 print jobs and "lost" SR10 print jobs. In the case of this error, **pre10q** "requeues" the job to the print manager before **prflib** finishes. So when **prflib** finally gets its turn to queue the job to the print manager, the print manager reports that it "already has that job".

4.3.4.2 Hardcopy Limitations

The following sections describe hardcopy limitations.

4.3.4.2.1 Cannot Print Multiple Plane gmf Files on Cut Sheet Printers

Multiple plane gmf files cannot be printed on cut sheet printers such as PostScript, Impress and Tektronix printers.

4.3.5 HP-VUE Bugs and Limitations

The following sections describe bugs and limitations in HP VUE 2.01.

4.3.5.1 HP VUE Limitations

The following sections describe limitations in HP VUE 2.01.

4.3.5.1.1 Hardware Limitation

Although HP VUE will run on any system that runs SR10.4, its performance on older, slower systems may be unacceptably slow. HP recommends a DN4500 with 16 MB of memory as the minimum configuration. Series 400 systems deliver good HP VUE performance.

4.3.5.1.2 HP VUE Node Environment Link Must Point to SR10.4 Node

HP VUE runs only from a UNIX environment; therefore, an HP VUE node must have either a SysV or BSD environment installed, either as a link or a local copy. If installed as a link, the link must point to an SR10.4 node in order for HP VUE to run.

4.3.5.1.3 HP VUE Depends on Reliable TCP/IP Services

Proper functioning of HP-VUE at present depends critically on the reliability of the TCP/IP services of the network. As a consequence, an interruption of TCP/IP services may make it impossible to log into an SR10.4 system that is configured to run only HP-VUE (i.e., no DM). If your TCP/IP services are unreliable, you must fix them if you want to configure your node this way.

4.3.5.1.4 HP VUE Provides no Software Reboot Mechanism

HP-VUE contains no mechanism for a user to reboot the system through software without the use of a root login account.

4.3.5.1.5 HP VUE Login Screen Input Limitation

HP VUE may lose characters that are input during redisplay of the login screen after screen blackout. In order to prevent this, allow HP VUE to repaint the screen fully before entering characters.

4.3.5.1.6 vuelogin 'grabs' the X Server

vuelogin "grabs" the X server. This means that no other X client can use the display or keyboard. This is a security feature. You can, however, configure the system to allow X clients to output to the display while vuelogin is running by setting the grabServer resource to false.

4.3.5.1.7 Extended Login Time

HP VUE maintains detailed information about your session. When you log in, HP VUE attempts to restore this information. If your session is complicated (lots of windows, lots of clients), logging in can take a considerable amount of time. To avoid this expenditure of time, use VUE's lock button instead of logging out.

4.3.5.1.8 HP VUE 2.0 Users Must Own Their Home Directory

Users must own their home directories in order to run HP VUE 2.0. If they do not, HP VUE is unable to create its error log file **$HOME/.vue/errorlog**; this can cause a number of processing errors, and impedes subsequent troubleshooting.

4.3.5.1.9 HP VUE Style Manager Cannot Change Keyboard Tone

Domain-style keyboards do not implement tone control. Attempts,therefore, to change the keyboard tone via HP-VUE's Style Manager have no effect.

4.3.5.2 HP VUE Bugs

The following sections describe known bugs in HP VUE 2.01.

4.3.5.2.1 HP VUE Help Facility Always Displays Man Pages for Current SYSTYPE

Although the menu displayed by the HP VUE Help Facility (the "?" button on the front panel) allows you to select BSD, SysV or Aegis man pages for display, HP VUE always displays the man page for your current SYSTYPE.

4.3.5.2.2 HP VUE Occasionally Miscomputes Pathnames

Occasionally, HP-VUE mis-computes the full pathname of a file. Instead of the correct dds-style pathname of **//hostname/dir/file**, VUE computes an ip-style pathname of the form: **hostname:/dir/file.**

4.3.5.2.3 Vuefile 'Change To' builds remote pathname wrong

In HP-UX VUE, remote file systems are mounted in **/nfs/hostname**. The file manager "Change To" dialog accepts the syntax **hostname:/path** and converts this to **/nfs/hostname/path**. When an HP-UX file system is mounted to a Domain box via this mechanism, the path name constructed by the file manager is **/nfs/hostname//nfs/hostname/path**. The path label in the file manager looks correct, **//hostname/path**, but file manager operations fail due to the path expansion. If the syntax **/nfs/hostname/path** is entered at the "Change To" prompt, all works correctly. Use // to navigate, not **hostname:**.

4.3.6 xdm Problem

If the **/usr/lib/X11/xdm/Xservers** file looks like the following, per instructions for running **Xdomain** with **xdm**, the system hangs and you have to reboot:

```
#:0 secure /bin/nice --10 /etc/Xapollo -K /usr/X11/lib/keyboard/keyboard_quit.config -D1 S+R+
:0 local /etc/Xdomain bc
```

Depending on how many switches you use on **Xapollo**, things happen a little differently, but basically **Xapollo** and **Xdomain** both run, the cursor behaves strangely, and **Xdomain** hangs. A workaround is to leave a space between the # and the :0 and the line will be "officially" commented out. Or, you can just remove the line.

4.4 NCS 2.0 Limitations and Bugs

NCS 2.0 has the following known bugs and limitations:

- Applications written with the NCS 2.0 API cannot make use of the Global Location Broker (GLB); therefore, NCS 2.0 API applications must not call any **lb_$** subroutines.
- Although applications written with the NCS 2.0 API can comprehend UUIDs in the 1.5.1 format, an application written with the NCS 1.5.1 API that converts a UUID in the NCS 2.0 format into string representation will drop significant information from the UUID, because NCS 1.5.1 UUIDs contain fewer significant bits than NCS 2.0 UUIDs.

  A consequence of this problem is that NCS 1.5.1 administrative tools cannot manipulate NCS 2.0 UUID entries in the Location Broker databases.
- Nodes running versions of NCS prior to 1.5.1 should not share a Location Broker cell with nodes running NCS 2.0. In particular, pre-NCS-1.5.1 versions (for example, those shipped at or before Domain/OS SR10.1) of the **glbd** daemon and the **drm_admin** command cannot interoperate with their NCS 2.0 counterparts.

- The **rpccp** command accepts commands which depend on the OSF DCE naming and authentication services, but does not execute these commands. Only the **rpccp** commands **exit, help, quit, show mapping,** and **remove mapping** have any effect at NCS 2.0.

- The **clean** command of the **lb_admin** utility, if run on a host that does not support every protocol family, will not recognize some valid addresses.

  For instance, if **lb_admin** is run on an IP-only host, it will request permission to remove all DDS addresses from the database, giving a message like the following:

  ```
      object = 3805055be000.0d.00.00.91.5b.00.00.00
        type = 3805055cf000.0d.00.00.91.5b.00.00.00
   interface = mandelbrot/block "mandelbrot block server" @ <invalid address> global

        Invalid Address Family.  Delete? n
  ```

  You should not delete the entries for the unsupported protocol; answer "n", as shown.

  Workaround (and best general practice): If possible, run **lb_admin** on a host that supports every protocol family.

4.4.1 Fixed Bugs in NCS 2.0

The following bugs have been fixed since the last release of NCS:

- NCS daemons now fail gracefully if the **tcpd** daemon dies. If a socket fails repeatedly, the runtime will close it and discontinue listening on it. If the failing socket was the only socket on which the server was listening, a communications failure exception is raised.

- IP addresses specified in the **glb_site.txt** file are now interpreted correctly.

4.4.2 NFS 2.3 Limitations

The following sections describe limitations in NFS 2.3.

4.4.3 NFS 2.3/Threads Limitation

Domain NFS 2.3 is not guaranteed to be thread safe: accessing remote (NFS) files from a SR10.4 pthread may not always work entirely correctly. A patch to correct the problem will be available shortly after the release of SR10.4; contact your customer representative for details on ordering this patch.

This patch consists of two files:

/sys/mgrs/nfs_gate
/lib/rpclib

The patch corrects these problems as follows:

1. It ensures that a thread will always access the correct per-thread copy of errno.
2. It limits entry into the NFS type manager code to one thread at a time.

Note that while the second item guarantees thread safety it does not allow simultaneous remote i/o by multiple threads within a process.

4.5 DSEE Version 4.0 Bug

DSEE V4.0, when running with the X/motif interface (the default interface at V4.0), mishandles CTRL-C and CTRL-Q characters entered in the transcript pad area, causing DSEE to fault. When this happens, the DSEE process continues to run, but the transcript pad area becomes blank. Prior to the transcript area becoming blank, the following message is displayed:

```
X toolkit error:  Select failed
```

To fix this problem, please install patch number pd92_m0394 (m68k) or patch number pd92_p0323 (a88k).

4.6 CD-ROM Limitations

CD-ROM bootability is not supported on DN4000 or DN10000 nodes.

Programs that have an obj format may not be directly executed from the CD-ROM.

4.7 X11 R4 Limitations

- **uwm** is not supported with X11R4. Use **mwm** instead. In general, users are encouraged to move to **mwm**, since **uwm** is no longer shipped by MIT, and is no longer supported by many vendors. SR10.3 and SR10.4 startup scripts that reference **uwm** should be modified to use **mwm**.
- Both the new and the old **xdm** Xsession files should be modified to remove the statement **/usr/apollo/bin/kbm** if they are to be used with the borrow mode server (Xdomain). The executable **/usr/apollo/bin/kbm** has been known to cause problems with the Xdomain server.
- X11R4 runs only on SAU types 7, 8, 9, 11, 12, and 14. Use the **bldt** command to determine your SAU type.
- The X11R4 shipped with SR10.4 is a RUNTIME only environment consisting of X11R4/Motif1.1 shared libraries, X11R4/Motif1.1 clients, an X11R4 borrow mode server (Xdomain), and HP-VUE 2.0 It does not include an X11R4 or Motif1.1 BUILD environment needed to develop new X11R4/Motif1.1 applications from source. Instead, the X11R4/Motif1.1 header files and archived libraries have been broken out of the base software releases (SR10.4) and into a layered product known

as the "User Environment Developer's Kit (UEDK) v.1.1 (LAC4BBAD)".

For those still wishing to develop on X11R3, the X11R3 build environment shipped with SR10.3 is still available as an install option in SR10.4

4.8 HP/DDE Bugs and Limitations

The following subsections describe the status of bugs and limitations in HP/DDE.

4.8.1 HP/DDE Limitations

The following sections describe limitations in HP/DDE.

4.8.1.1 SR9.7 Compatibility Code Removed in SR10.4

SR10.4 removes some SR9.7 compatibility code:

- Support in HP/DDE for **/com/debug** compatibility commands is removed.
- HP/DDE and HPC no longer support debugging OBJ files. Under SR10.4, you cannot debug an OBJ program.

4.8.1.2 Restriction on HP/DDE and FORTRAN I/O Statements

If you attempt to step through a FORTRAN READ, WRITE, or INQUIRE statement with an ERR, END, or IOSTAT specifier, HP/DDE may lose control of the target program. With an IOSTAT specifier, this problem can occur even though no apparent transfer of control results from the statement's execution.

To avoid loss of control, set breakpoints at the statements specified by ERR and/or END, and, if there is an IOSTAT parameter, at the next sequential statement.

Suppose, for example, that your program contains the following statement:

```
READ (5, 100, END = 900, IOSTAT = RSTAT) X, Y, Z
```

Before stepping to the READ statement, set breakpoints at statement 900 and at the statement following the READ statement.

4.8.2 Bugs in HP/DDE

HP/DDE contains the following known bugs:

- If a procedure has no statements in it, HP/DDE can not set a breakpoint on the procedure name.
- Argument information is sometimes unavailable in FORTRAN.
- HP/DDE cannot find correct addresses for variables in registers in code ranges that have been removed in optimization.
- HP/DDE cannot print the contents of virtual addresses from F8000000 to FFFFFFFF. A request such as:

```
dde> print ˆinteger32(16#fc070000)ˆ
```

results in the following error message:

```
ˆ: ?(dde) No read access to virtual address fc070000
```

4.9 SoftBench Version 1.01 Limitations

SoftBench 1.01 includes limited support for Domain SR 10.4. The limitations are as follows:

- SoftBench Version 1.01 does not work correctly with the X11R4 borrow-mode server (Xdomain), and is not supported running in that configuration. We recommend you use the X11R2 share-mode server (Xapollo) only.
- *SoftBench is not supported onSR10.4 when HP VUE 2.01 is installed on the same machine.* SoftBench Version 1.01 and HP VUE 2.01 collide with each other at install time due to their having some common files. To avoid such a collision when installing SR10.4 (using install++ or config), do not select the x11_runtime component when asked which X11 components to install. This will ensure that VUE 2.01 does not get installed.
- SoftBench Version 1.01 on Domain is not completely compatible with the A.02.01 release on HP-UX and SunOS. The online file **/usr/softbench/README.A.01.A.02**, supplied with the A.02 product, provides detailed information on the interoperability problems you may encounter when using SoftBench Version 1.01 on Domain/OS with A.02 SoftBench on HP-UX and SunOS.

4.10 Hardware Bugs and Limitations

4.10.1 Problems Common to all HP Apollo 9000 Series 400/425 systems

The first command you issue to the system after booting, either a command to the Phase II shell (the boot shell) or a log-in attempt, is sometimes not interpreted correctly. If the first command is a Phase II shell command, the error message "Unknown command" is displayed. If the first command is a log in attempt, you receive a "login incorrect" message. If this happens, simply reissue the command to the Phase II shell or attempt to log in again; subsequent command executions work correctly.

4.10.2 Floppy Disk Limitations

On Model 425e workstations, you must format a 3.5-inch floppy disk with the **invol** program before using it on the resident floppy drive.

Note also that the 425e floppy drive supports high density floppy disks (not low density).

4.10.3 Logical Volume Limitation

The maximum logical volume size for striped disks on the DN10000 is 16-GB. For other nodes with 4-KB disk block size (SAU11, 12, and 14) it is 4-GB, and it is 2-GB on SAU7, 8, and 9 nodes.

4.10.4 General External Device Limitation

Always turn on any external devices that are connected to your system unit before you turn on the system unit. This practice allows the system diagnostics to test these devices on power up.

Conversely, always turn off your system unit before you turn off any external devices that are connected to your system unit. This practice prevents possible system hangs.

4.10.5 Problems with Series E and F (Meerkat Displays)

On 4-plane and 8-plane MK1 and MK3 systems under X, the hardware look up table (LUT) is changed one color at a time at the rate of 60 colors per second. This may result in slow performance when activating clients (for example via **mwm**), which use large private colormaps, or when using clients which change colormaps frequently.

On 4 and 8-plane color MK1 systems, an **XCopyArea**() request of a **pixmap** to the screen may flash with extraneous colors, even when the color map is not changed. Unlike other display devices supported by the R4 server, the MK1 is an **XYPixmap** device. **XCopyArea**() from a **pixmap** to the screen is done a plane at a time, resulting in momentary color artifacts. Other approaches would severly degrade performance. GPR exhibits the same behavior when displaying color bitmaps.

4.10.6 Model 425e Graphics Limitation

The 425e does not support planar access to the frame buffer. If **gpr_$remap_color_memory** or **gpr_$remap_pixels** are used to set plane mode access, the error "gpr_$wrong_display_hardware" is returned. The 425e does support pixel oriented access to the frame buffer. As a result, the optional product DTEK 4014 does not run on a Model 425e workstation; also, the DM command **xi** and the OS command **cpscr** do not function properly.

4.10.7 Network Limitation

On Series 400t and DN5500 systems, the 802.5 Token Ring controller cannot be set to unit 1. You must use unit 0 setting.

4.10.8 Model K1388 Dial Set Notes (SR 5003014050)

The following sections provide information on using the Model K1388 Dial Set, and on starting up **/sys/dial_server.**

4.10.8.1 Using The Model K1388 Dial Set Connected to the 3-Port SIO Connector

If you want to use a Dial Set connected to a 3 port sio connector, you must connect the Dial Set to the connector before booting the node. (Dial Sets can, however, be connected to the P2 system serial port while the system is running.)

4.10.8.2 Starting /sys/dial_server from /etc/rc.user

You can start /sys/dial_server at system startup using the following procedure:

1. Uncomment the following lines in **/etc.rc.user** as shown:

```
#  dial server:
#  Used in conjunction with the dial_server_request command.
#
  if [ -f /sys/dial_server ]; then
       (echo " dial_server       /sys/dial_server &
  fi
```

2. Enter this command in ~/user_data/startup*.*:

```
cps /com/dial_server_request -init port1
```

4.11 PSK Q3-91 SR10.3 Product Support Kit Release Document Bug

Section 1.9.2.4 of the PSK Q3-91 SR10.3 Product Support Kit Release Document gives, as an example of the use of input serial devices with the Xdomain server, a sample **X0devices** file entry specifying a Summagraphics II 1812 Graphics Tablet as an input device. This example is in error; the Summagraphics Graphics Tablet was not supported in the Q3PSK. (It is, however, supported in SR10.4.)

4.12 Programming Limitations

- The new 68040 MOVE16 instruction must be preceded by a NOP in order to work correctly. (Domain compilers never generate a MOVE16 instruction.)
- Indirect accesses, where the intermediate access (to fetch the address) is to a serialized page causes the CPU to lock up. Avoid this construct. (Domain compilers do not generate this construct.)
- Certain instructions (**FMOVE, FMOVEM, MOVEM**) can cause double writes to occur, even to serialized locations, unless they are preceded and followed by a NOP. Beware of this when writing GPIO drivers.
- The **FScc -(Ay)** instruction does not work. Avoid this construct. (Domain compilers do not generate this construct.)

4.13 HP OmniBack Limitation

Versions 1.2 and 2.0 of HP OmniBack are not supported in this release. To run HP OmniBack under Domain/OS SR10.4, you must upgrade to HP OmniBack Version 2.01.

4.14 SAX Limitations

The following limitations should be noted when running the SAX utility.

- If the ID of a SCSI cartridge tape is not set to 0, SAX fails.
- If you run the SPE test using SPE version 2.1 or earlier, you must create a link from **/dev/pio** to **/dev/spe_pio_ddf**.

4.15 Interleaf Limitation

Interleaf TPS4 on Domain/OS 68040 workstations requires a software patch from Interleaf. Contact Customer Support at Interleaf for the patch.

Interleaf TPS4 does not display on the Model 425e. The 425e does not support planar access to the frame buffer, and TPS4 requires planar access for display.

4.16 Patches

Patches are classified as follows:

PRESERVED

Deliverables of the patch are not overwritten by the install. If you want the functionality of the patch and the current release, apply the patch, then install this release.

There are NO patches to SR10.3 in this category at SR10.4.

NOT PRESERVED

Deliverables of the patch are overwritten by the install and this release does not include the source changes of the patch. You cannot have both the patch and the functionality of this release on the same system.

There are NO patches to SR10.3 in this category at SR10.4.

INCLUDED

The patch source changes were included in the code used to build this release. The functionality of the patch is delivered in this release. Do not apply the patch; the desired functionality is built into this release.

4.16.1 Patches Included in Domain/OS SR10.4

ALL patches to SR10.3 up to and including the following patch numbers are INCLUDED in SR10.4.

m68k	pd92_m0392
a88k	pd92_p0322

For information on the classification of later patch numbers, see the Release Notes for the patch in question.

4.17 PSKs Included in Domain/OS SR10.4

All functionality added in PSKQ3_91 (SR10.3.5), and all bug fixes from its patch releases, have been incorporated in SR10.4.

Chapter 5: SR10.4 Compatibility

5.1 Overview

This chapter documents incompatibilities between SR10.4 and previous SR10 releases.

5.2 Hardware Platforms

- SR10.4 does not run on SAU 2 -- SAU 6.

Unsupported Machine Types		
SAU	**Node Name**	**CPU Type**
sau2	**dn300**	**68010**
	dn320	**68010**
	dn330	**68020**
sau3	**dsp80**	**68010**
	dsp90	**68020**
sau4	**dn460**	**68010**
	dn660	**68010**
sau5	**dn550**	**68010**
	dn560	**68020**
	dn570	**68020**
	dn580	**68020**
sau6	**dn560T**	**68020**
	dn570T	**68020**
	dn580T	**68020**
	dn590T	**68020**

- SR10.4 runs on SAU 7 -- SAU 14.

Supported Machine Types		
SAU	**Node Name**	**CPU Type**
sau7	**dn3500**	**68030**
	dn4000	**68020**
	dn4500	**68030**
sau8	**dn3000**	**68020**
	dn3010	**68020**
sau9	**dn2500**	**68030**
sau10	**dn10000**	**Prism**
sau11	**9000/425s**	**68040**
	9000/425t	**68040**
	9000/425e	**68040**
	9000/433s	**68040**
sau12	**9000/400s**	**68030**
	9000/400t	**68030**
	9000/400dl	**68030**
sau14	**dn5500**	**68040**

5.3 SR9.7 Compatibility

- SR10.4 does not provide support for moves from SR9.7 to SR10.4; if you want to move from SR9.7 to SR10.4, you should move to a previous version of SR10 first.

  Of course, you can choose to install SR10.4 from scratch; but SR10.4 does not include any information on making a transition from SR9.7 to SR10.4. Also, SR10.4 does not provide conversion tools, like the registry conversion tools, nor does it contain the SR9.7 compatibility directory. And much of the SR9.7 compatibility code has been removed, as is described under the next bullet.

- SR10.4 removes some SR9.7 compatibility code:

  - The INPROCESS environment variable has been removed, and mark/release support, designed for executing programs inprocess, has also been removed. Since inprocess has been removed the **aqdev**, **rldev** and **pbu_$acquire()**, **pgm_$invoke()**, **pbu_$release()** method of using GPIO drivers has also been removed. Note, however, that **pbu_$acquire()**, **appl...**, **pbu_$release()**, continue to work as before.

  - The ability to mount and read SR9.7 volumes (disks and floppies) has been removed. You can still access SR9.7 volumes across the network as long as they're mounted on a pre-SR10.4 node. But you can't mount them on an SR10.4 node.

  - Support in DDE for **/com/debug** compatibility commands has been removed.
  - Execution of SR9.7 object files continues to be supported, as long as they do not reference features that have been removed.
  - Execution of object files that are stamped sys5 or bsd4.2 continue to be supported, and the top level sys5 and bsd4.2 links are created at install time.
  - DDE, HP/PAT, and HPC no longer support OBJ files. Under SR10.4, you cannot debug an OBJ program.
  - Support for DOWNCASE has *not* been removed.

- SR9.7 and SR10.4 can coexist in a network with both file sharing and **crp** ability. Network access is comparable to that between SR9.7 and previous versions of SR10. There are, however, some SR10 capabilities not available from an SR9.7 node. Some examples:
  - SR9.7 can't see "long" SR10 file/path names;
  - Some SETUID functionality doesn't work between SR9.7 and SR10 nodes;
  - Copying an object (or an acl) from an SR10 node to an SR9.7 node and then back produces something different than what you started with (that is, SR9.7 can't represent some state).

5.4 SR10 Compatibility

- SR10.4 is fully binary compatible with SR10 based binaries that do not use discontinued SR9.7 compatibility code.
- SR10.4 is fully source compatible for all PASCAL and FORTRAN programs.
- SR10.4 may require minor source or makefile changes when recompiling in order to move C programs to ANSI-C.

  The default development environment for compiling C programs under UNIX is ANSI-C: the ANSI C include files, the ANSI C preprocessor, and ANSI C semantic behavior in the Unix libraries. SR10.4 does not provide the **/sys/ins/*.c** include files. You must use the **/usr/apollo/include/*.h** files.

  The 89.1 compilers continue to run on SR10.4, but you are encouraged to use Domain C Version 6.8 or later, Domain FORTRAN Version 10.8 or later, or Domain Pascal Version 8.8 or later.

  Programs that have never been modified to take advantage of the ANSI C features can modify their makefiles to use the **-A nansi** switch that provides straight K&R C. Programs that take advantage of ANSI-C features such as function prototypes may require further source modification, since SR10.4 provides full ANSI C compliance and the Domain C Version 6.8 or later compilers do not support the partial ANSI C that has been supported in previous compilers.

For more information on ANSI C, see the SR10.3 Release Notes, and the ANSI C section in Chapter 1 of these release notes.

- SR10.4 does not require that you **invol** your node, but you may wish to do so to take advantage of new functionality, such as disk quotas.
- The Model 20GB/A Autochanger is not directly accessible from nodes running a pre-SR10.4 release.
- Code that is compiled with a standards compliance compiler symbol explicitly defined (**_POSIX_SOURCE**, **_XOPEN_SOURCE**, or **_AES_SOURCE**), will not execute on an SR10.3 node. See the section "Generating Standards Compliant Applications" in Chapter 1.

5.5 Kernel Changes Affecting Compatibility

- Unix mapping calls:

  NOTE: The Domain/OS unit of mapping and protection is the segment (32K, 128K, 512K) and not the page.

  mmap() and **mremap()** now handle a protection of PROT_NONE (reference to the segment(s) in question will result in an access violation). Unless PROT_NONE is specified to these calls, read access is automatically granted to the region in question.

  - **mmap()**
    MAP_ANON: an anonymous VM area is allocated instead of a temporary file; the MAP_SHARED flag controls whether this area is copied or shared following a fork. (MAP_SHARED==true implies that parent/child share the mapping and see each others changes; MAP_SHARED==false implies that the child gets a copy of the area).

    You can map something with PROT_NONE rights.
  - **munmap()**
    You can now unmap a sub-region (in whole segments) of a region previously mmap()'d. (Prior to SR10.4 you could only unmap() an entire region: the munmap() "addr" argument was required to point to the start of a mapping (as returned by mmap()) and the "len" argument was ignored.)
  - **mprotect()**
    You can now set the protection on a region to PROT_NONE. You can change the protection on a sub-region (in whole segments) of an mmap()'d region. (Prior to SR10.4 you could only mprotect() an entire region that was mmap()'d: the

mprotect() "addr" argument was required to point to the start of a mapping (as returned by mmap()) and the "len" argument was ignored.)

- **mremap()**
  (This call is an Apollo extension.)

  As in pre-SR10.4 releases, you must pass to mremap() a VA returned by a prior call to mmap(). The entire region is unmapped and a new mapping is established. (If you've previously munmap()'d *sub-regions* from the original mapping, the behavior of the system is uncertain --- it attempts to unmap the "contiguous" area remaining at the specified VA.)

  An advantage of this over pre-SR10.4 behavior is that mremap() now attempts to establish the new mapping at the VA specified to it -- that is, the VA of the current mapping (the one being released).

- **madvise()**
  You can now madvise() a sub-region of a region mmap()'d earlier. (Prior to SR10.4, the "addr" argument was required to point to the start of a mapping (as returned by mmap()).)

- **ms_$** mapping calls (Note: this bullet lists only changes to the *supported* calls.)

- **ms_$addmap**
  The **ms_$addmap** call is used to obtain additional mappings to an object already mapped. Prior to SR10.4, there was a restriction that mappings obtained via this call had to be released prior to the release of the original mapping on which they were based.

  At SR10.4, this restriction has been removed; the mappings can be released in any order.

- **ms_$addmapx**
  This call is new at SR10.4. The only difference between it and ms_$addmap is that it accepts a "newva" argument that specifies where the additional mapping is to be established (whereas ms_$addmap picks a location on its own).

  Look in **/usr/include/apollo/ms.h**, and **/us/sys/ins/ms.ins.pas** for this call's signature.

- **getrusage**
  The BSD getrusage() system call now returns correct values for the following fields:
  .ru_majflt
  .ru_minflt
  .ru_inblock
  .ru_outblock
  (For the meanings of these fields, see the BSD Programmer's Manual).

- **setrlimit**
  The BSD setrlimit() call now supports RLIMIT_RSS. This option limits the amount of memory that a process is permitted to use (i.e., its working set).
- The "ulimit -m" command to the Korn shell is functional at SR10.4.
- **-sparse_vm** switch
  At SR10, Domain/OS started backing anonymous virtual memory (for example, that returned by C's malloc() or Pascal's new()) by disk blocks.

  Shortly thereafter, a "sparse_vm" binder switch was added that allowed one to indicate that a program's static variables should instead be treated the pre-SR10 way. That is, disk blocks would only be allocated as the variables were accessed/used. This was primarily provided for the use of Fortran programs with large, static arrays. (Due to an unintentional side effect of the underlying implementation, this flag also caused malloc()'d space to be allocated in a sparse manner.)

  At SR10.4, the meaning of this flag was enhanced to cover space allocated by C's malloc() as well as Pascal's new(). Specifically, a program can be bound as follows:

  ```
  /bin/ld -A sparse_vm -o foo.bin foo.o
  ```

  to cause disk blocks for its malloc()/new() space to be allocated only as that space is accessed/touched.

- Various limits
  - A node can now support as many as 20 diskless children.
  - The number of locks that a node can support is now between 2048 and 8192. The precise number is a function of the machine type and amount of main memory installed. (More memory implies more locks can exist.)
    NOTE: this change was actually made at SR10.3.
  - An individual process can now hold as many as 500 locks. Only a limited number of processes (currently, about 1/8th the maximum number supported on a machine) can have this many; the remainder are still limited to 70.
    NOTE: this change was actually made at SR10.3.
  - The maximum file size on SAUs 10, 11, 12, 14 (4K-page machines) has been increased from 2 to 4 gigabytes. On other machines, the limit remains 2 gigabytes.
- **OS Paging File**
  During boot, the OS now attempts to automatically allocate paging file disk blocks as needed, instead of complaining that the paging file is too small.

  The invol program still requires you to set the size of the OS paging file. This setting determines the *initial* size of the paging files, which then grows or shrinks as required.

- **stat**() system call
  For files accessed across NFS, "correct" values have been returned for the inode and device fields of the stat structure.

  Additionally, a new field has been added to the stat structure that, if non-zero, has been set to a value that identifies the host/machine on which the file in question resides. (Currently, this field is set for files residing on native Domain/OS nodes or accessed across NFS.) This field is 2 longwords in size and can be accessed as **.st_hostid[0]** and **.st_hostid[1]}.**

- **statfs**() System Call
  The statfs() call now works correctly for file systems accessed across NFS.

- File Locking Changes
  There are two "modes" of file locking provided by Domain/OS. In simple terms, the difference between them is the following:

  - NR-XOR-1W locks allow *either* multiple readers (possibly from different nodes) *or* a single writer (and no readers) to have access to the file at a given time. This type of access (i.e., these kinds of locks) can only be obtained by using Domain/OS (i.e., non-Unix) calls.

  - COWRITERS locks allow readers *and* writers to access a file at the same time. Previous to SR10.4 the ONLY restriction was that if there is a write lock on the file, then ALL locks must originate from the same node. For example: if a process on node A is writing a file (not necessarily on node A), only processes on node A can access that file -- for either read or write. This kind of access (i.e., these kinds of locks) are automatically utilized by the various Unix calls (e.g., open, creat), and can also be obtained optionally via Domain/OS calls.

    At SR10.4, enhancements have been made to the COWRITERS locking mechanism (i.e., that used by Unix system calls).

    - At SR10.4, the only restriction on a file open for both reading and writing is that all processes **writing** to the file must reside at the same node.
    - In such a scenario (readers and writers from different nodes), the amount of time before reading processes will see changes made by a writing process is undetermined. The OS does, however, attempt to propagate the changes within a small number of minutes.
    - This new behavior is provided *only* when any involved nodes (the file's home node, any reading nodes, any writing nodes) are running SR10.4.

    With this enhancement it becomes possible, for example, to sit at one node and examine a log file being produced by a program executing at a different node.

- **Other File System Related Changes** At SR10.4, the "modified" and "accessed" times of files being accessed by Unix programs changes with *each* write() and read() system call. Likewise, the length of a file being written (which used to be maintained in user-space libraries until the file was closed) is now correctly maintained in the kernel, where it will be correctly reported by stat() calls performed by other processes (e.g., **/bin/ls -l**);

  This behavior change only takes affect when all involved nodes (that is, any node accessing the file as well as the node at which it resides) are running SR10.4.

  For remotely accessed files, there can be some delay in "observing" these attribute changes. For example, consider a file residing at node A being written from node B and being **stat**()'d from node C. Changes made by node B are held there and are only "pushed" through to node A (the home node) every few minutes. During this interval, accesses from node A, C, or other nodes will continue to see the old attributes.

- **Swapping, Quotas, /etc/fstab, /etc/mnttab**, or **/etc/mtab** NOTE: In the following discussion, whenever /etc/mtab is mentioned, both /etc/mtab and /etc/mnttab are meant (they're really the same file).

  - **/etc/swap, /etc/swapon**
    At SR10.4, the appropriate /etc/mtab entry is updated when swapping is enabled on a volume. (Prior to SR10.4, the /etc/swap and /etc/swapon commands didn't update /etc/mtab at all.)

    When swapping is enabled, the options field (mnt_opts, the 4th field) is changed (probably from "rw") to "sw". When swapping is disabled, the options field is changed back to either "rw" or "rq", depending on whether disk quota checking is also enabled on the volume.

    An exception to this behavior is the boot volume (used initially for swapping), which, after Domain/OS boots, has an "rw" in its /etc/mtab entry. This was done in order to prevent user confusion at seeing an "sw" on their boot file system. (See the next point; under older Unixes, mountable file systems and swapping volumes were two completely different things.)

  - Recall that under Domain/OS, swapping space is dynamically allocated from ordinary, mounted file systems. Therefore, a given volume can have both swapping and disk quota checking enabled on it. Whichever of these operations (/etc/swapon or /etc/quotaon) is executed last determines the encoding left in the /etc/mtab (/etc/mnttab) file: "sw" or "rq". (That is to say, each of these operations overwrites the mnt_opts field).

  - The bsd4.3 "mount -a" command mounts all disks specified with a type of "rw" or "ro" in /etc/fstab.

In prior releases, this command failed if it encountered an /etc/fstab entry of "rq" (e.g., "rw,rq"). (It didn't recognize "rq" ... although it did understand "quota"). At sr10.4, the mount command understands "rq" as well.

Therefore, you can now create an /etc/fstab entry of the form:

```
/dev/dsk/W0d0s4 /tmp/ww4 4.3 rq,rw,sw 0 0 0
```

and have

mount -a	Mount that disk (BSD version of mount command only)
swapon -a	Enable that disk for swapping
quotaon -a	Enable quota checking on that device

- Prior to SR10.3, only a single space was allowed between fields in /etc/fstab entries. At SR10.4, multiple spaces or tabs are permitted.

- **Device Numbers** At SR10.4, the device number encodings have been changed. This was done primarily to accommodate the optical disk jukebox product, which supports more disks than could be accomodated in the old scheme.

  Domain/OS uses two types of device numbers: an on-disk representation and a representation utilized by various Unix calls (stat, mknod, etc.). For historical reasons, these have been different, although at SR10.4 they have the same bit encodings.

  For on-disk device numbers:

  - Prior to SR10.4, device numbers were encoded on disk as a 16-bit **dev_number_$t**:

5 bits	major device number
11 bits	minor device number

For block devices (e.g., /dev/dsk/...), device numbers were broken down further as follows:

5 bits	controller type
3 bits	controller number
4 bits	drive number
4 bits	logical volume number

This device number was returned via the unsupported **file_$get_attr_info** and **file_$get_attributes** calls (in the **.r_dev_num** and **.dev_num** fields).

- At SR10.4, a new 32-bit low-level device number exists, **dev_number32_$t**:

13 bits	major device number
19 bits	minor device number

For block devices (e.g., /dev/dsk/...), device numbers were broken down further as follows:

10 bits,	
3 bits	controller type
4 bits	bus number (currently unused)
4 bits	controller number
7 bits	drive number
4 bits	logical volume number

This new device number is returned via the above-mentioned **file_$** calls (in **.r_dev_num32** and **.ndev_num** fields).

NOTE: Be aware that the old device number still exists, both on disk and in the structure returned by the above file_$ calls. With one exception, both device numbers encode the same information, and applications can continue to use the "old one". The exception occurs in the case of those devices that cannot be represented in the old encoding (e.g., too large a controller or drive number).

Whenever the SR10.4 OS reads a device file containing the old, 16-bit device number (e.g., from a remote pre-SR10.4 node), it creates a corresponding 32-bit number to return in all relevant (file_$) calls.

For the "Unix" device numbers returned by stat(), accepted by mknod(), dissected by the major/minor macros and assembled by the makdev macro:

- Prior to SR10.4, a 32-bit "Unix" device number had the following representation:

23 bits	major device number
9 bits	minor device number

  For block devices (e.g., /dev/dsk/...), they are broken down further as follows:

20 bits	node number at which device resided
3 bits	controller type
2 bits	controller number
3 bits	drive number
4 bits	logical volume number

- At SR10.4, we switched to a "Unix" device number that had the same encoding as the internal, on-disk one discussed above (excpt for the high bit set to 1; see below):

1 bit	(set to 1),
13 bits	major device number
19 bits	minor device number

  For block devices (e.g., /dev/dsk/...), they are broken down further as follows:

1 bit	(set to 1)
9 bits	---stat() hashes a node-ID into field (see below)
3 bits	controller type
4 bits	bus number (currently unused)
4 bits	controller number
7 bits	drive number
4 bits	logical volume number

One (supposed) virtue of the pre-SR10 encoding was that, for block devices, it contained a full node-ID, making the device number unique across the network. Two objects were known to reside on the same volume IFF their device numbers were the same. (This property was unique to Domain/OS.) Although a separate host-ID field has been added to the stat structure, SR10.4 stat() tries to preserve this behavior by hashing a node-ID into as many free bits as remain (only 9). Therefore, two files with the same device number only *probably* reside on the same volume.

Note that both stat(), and the SR10.4 macros that construct device numbers (makdev), set the msb of their result. This assures that the macros separate device numbers into their major and minor number constituents according to either the pre-SR10.4 scheme (msb == 0) or the SR10.4 scheme (msb == 1).

Because of this, pre-SR10.4 versions of some device number processing commands (such as /etc/mknod) do run correctly under SR10.4. Other commands, however, may have problems. One example is a pre-SR10.4 version of **/bin/ls -l /dev/dsk** executed on a SR10.4 node. It receives new-format device numbers from its stat() call and proceeds to parse them into major/minor constituents according to the old format, and ends up displaying garbage.

The SR10.4 version of **/bin/ls** pointed at a pre-SR10.4 device operates correctly.

- **Sync** Prior to SR10.4, the **/bin/sync** command (and the sync() system call) only force wrote pages for files that were still open for write.

  At SR10.4, sync() has been changed to correctly write all modified pages resident in memory. For remote objects, however, the pages are just written across the network to their home node; they are *not* forced to disk there.

- **Reference Counts on Directories** At SR10.4, reference counts on directories are recorded and reported (e.g., **/bin/ls -l**) the "Unix" way. That is, a directory's reference/link count includes:

  1 for each time it is cataloged in a parent directory (normally once);
  1 for its "." self reference ;
  1 for each sub-directory's ".." reference back to it.

  Therefore, an "empty" directory created by **/bin/mkdir** will produce a reference count of 2. Creating a subdirectory underneath it will change its reference count to 3, and so forth.

The SR10.4 salvol corrects the reference counts on all directories as they are processed (thus converting a pre-SR10.4 volume to a SR10.4 one). Similarily, a SR10.3 salvol will switch them back to a pre-SR10.4 count.

Although it might appear confusing if SR10.4 is run on a node with pre-SR10.4 style reference counts (e.g., as left by a SR10.3 salvol), there is no danger of losing directories. The object deletion algorithms, normally based on a reference count going to 0, were "tweaked" to recognize this case and to refrain from deleting a directory prematurely.

Specifically, a directory's reference count is normally decremented when a subdirectory is deleted (its ".." reference). If that parent's directory == 1, however, (as it would on a SR10.3 disk), the decrement is not performed.

Running SR10.3 on a disk that had earlier been run under SR10.4 or salvol'd by the SR10.4 salvol produces a different problem. Because that OS is unprepared for directory reference counts > 1, the directory may not actually be deleted when the user requests it. (Although the name is removed, the directory object lives on since its reference count is still > 0).

For these reasons, we strongly recommend that you be careful about switching between SR10.3 and SR10.4, and that you give some thought to which version of **salvol** you use,as follows:

- If a disk is to be mounted under SR10.4 and it was LAST mounted under SR10.3 or **salvoled** by a SR10.3 **salvol**, it should first be **salvoled** by a SR10.4 **salvol.**
- If a disk is to be mounted under SR10.3 and it was LAST mounted under SR10.4 or **salvoled** by a SR10.4 **salvol**, it should first be **salvoled** by a SR10.3 **salvol.**

Miscellaneous Changes made for POSIX Compliance

- **rmdir()** on open directories
  POSIX requires (and tests for) the following property:

  If a directory opened with **opendir()** is deleted with **rmdir()**, it is not actually deleted until it is closed via **closedir()**. Although the directory can be read in this state (via **readdir()**), the "." and ".." entries should not be returned. This differs from pre-SR10.4 behavior.

- The following calls now fail with ENOENT if a pathname of length 0 is passed to them.

  access
  chdir
  unlink

chmod
chown
mknod
link
rename
mkdir

- Prior to SR10.4, the system rejected certain types of malformed pathnames with an EINVAL error. (Examples are: pathnames containing illegal characters and incorrect 'node_data syntax of malformed EVs embedded within pathnames.) At SR10.4, these are reported in several cases as ENAMETOOLONG.

- The following calls in the SYS5 environment will fail with ENOENT if a pathname of length 0 is passed to them:

  open
  creat
  truncate

- The write call now correctly returns ENOSPC on disk-full conditions and EDQUOT on disk-quota-exceeded errors.
- The access system call now requires a valid **amode** argument. If an illegal value is passed, the call fails with an EINVAL error.
- The link system call now translates **name_$file_not_directory** (not operating on a directory) Domain/OS status codes into ENOTDIR instead of into EXDEV.
- The **rename** system call now returns EINVAL if the the new pathname contains the old one as a prefix (prior to SR10.4 it used to return EISDIR in the SYS5 environment).

  In general, a better job is being done of detecting illegal cases (src contained in tgt, tgt contained in src).

  The rename system call no longer allows cross-volume/device operations to be performed; it returns an EXDEV error in such cases.

  The rename system call does a "better" job when pointed at symbolic links and directories.

Miscellaneous Behavior Changes

- Under SR10.4, the **/com/sald** command is able to "repair" many types of directory damage/inconsistencies that the SR10.3 version was not able to. (The improvements actually reside within the OS kernel.)

- Prior to SR10.4, only one process at a time could access a raw disk (accessed, for instance, via **open O_RDWR /dev/dsk/W0d0s4**).

  At SR10.4, the same rules that apply to ordinary files also apply to raw disks.

  Specifically, multiple programs can "open" the same raw disk; or a program can "open" the raw disk, fork and have both it and its child access the disk.
- Opendir() does not perform an **acl** check when it is executed upon an SR10.3 directory. In the case of SR10.3 directories, the **acl** check is performed when a subsequent readdir() call is executed.
- A new file created by creat() will have its DTM, DTU, and DTA (Unix date and time modified, date and time accessed, date and time attributes changed) all set to identical values.
- Prior to SR10.4, force-write operations on directories (via, say, files opened in O_SYNC mode) didn't always work correctly (although the probability of failure was very low). This has been corrected at SR10.4.

Changes Made for Standards Compliance The following changes were made for standards compliance. For these changes to be observed, the application must be compiled with the appropriate standards compliance flags. Unless otherwise noted, changes made for POSIX also apply to XPG and AES. Changes made for XPG also apply to AES.

- XPG/3 Changes

  cuserid The function generates a character representation of the name associated with the effective user ID of the process.

  kill If pid is -1, the sending process is sent the signal along with all other processes for which the sending process has permission to send the signal. Prior to SR10.4, the sending process was not sent the signal.

- POSIX Changes

  fcntl After a successful F_GETLK request, the value of l_whence will be SEEK_SET.

  fcntl.h If the application defines the compiler symbol _POSIX_SOURCE, struct flock is declared as follows:

```
struct flock {
   short    l_type;       /* Type of lock. */
   short    l_whence;     /* Flag for starting offset. */
   off_t    l_start;      /* Relative offset in bytes. */
   off_t    l_len;        /* Size; if 0 then until EOF. */
   long     l_sysid;
   pid_t l_pid;       /* Process ID of the lock owner. */
}
```

5.6 Shell Changes

In SR10.4 the Bourne shell (**/bin/sh**) has been replaced by the Kornshell (**ksh**). There are minor incompatibilities between the Bourne shell and the Kornshell. This may cause existing shell scripts to behave differently, or to misbehave. Chapter 21 of *The KornShell Command and Programming Language* by Morris I. Bolsky and David G. Korn discusses portability and compatibility issues between various versions of the Bourne shell and **ksh**.

SysV and BSD versions of the Bourne shell are provided with SR10.4 as **/sys5.3/bin/bsh** and **/bsd4.3/bin/bsh**. You can use one of these shells to execute scripts that are incompatibile with **ksh**.

The Domain/OS Bourne shell uses the **SHENV** environment variable to indicate the shell script to be executed when invoked. The Kornshell uses the **ENV** environment variable for this purpose.

In SR10.4, the shell library (**/lib/shlib**) is bound into **/com/sh**. The previous version of **/lib/shlib** remains so that a SR10.4 node can correctly execute **/com/sh** that resides on a pre-SR10.4 node. Also, **/com/sh** has been recompiled in COFF. This implies that only three streams are available instead of four.

Chapter 6: Eight-bit Native Language Support (NLS) in SR10.4

6.1 Overview of Internationalization on Domain/OS

At SR10.4 Domain/OS provides eight-bit Native Language Support (NLS), which includes standard ASCII characters and Western European character sets. This support enables programmers to write applications that operate in multiple languages, any of which can be be specified at runtime. The Domain/OS NLS implementation is based on international functions defined in **ANSI x3.159 Programming Language C, IEEE Std 1003.1-1989**, and **X/Open Portability Guide (December, 1988: XPG3)**.

Domain/OS NLS consists of a set of new and updated C library functions and related commands. Within the framework of these NLS system interfaces, Domain/OS provides mechanisms by which:

- Programmers can develop international applications to work in many different languages and conform to different cultural conventions.
- The runtime environment of a program can be set up to provide the correct processing of native language text and cultural data.
- Eight-bit coded character sets are supported, which meet the requirements of major Western European languages.

This chapter provides an introduction to internationalization concepts and terminology, an overview of developing international software, a discussion on the process of localizing software, and a description of creating and maintaining message catalogs.

6.2 Internationalization Concepts

For programmers who are unfamiliar with internationalization concepts, the following sections describe

- Internationalization Terminology
- Data Transparency
- Collation
- Character Classification
- Date and Time Conversions
- Numeric and Monetary Formatting
- Program Messages

- Answering Yes and No

6.2.1 Internationalization Terminology

- **Character Set/Codeset:** A collection of characters with assigned code values. For example, ASCII contains a specified group of characters, each of which has an assigned value in the set.
- **Code Position/Code Point:** The numeric value a character is assigned within a codeset. For example, A is at code position 0x41 in ASCII.
- **Internationalization:** The process of generalizing programs or systems so that they can handle a variety of languages, character sets, and national customs.
- **I18N:** An abbreviation for the word "Internationalization" (it begins with I, followed by 18 letters, and ends with N).
- **Localization:** The process of providing language-specific or country-specific information and support for programs. Localization can be done in two ways. In the first, programs are altered to add in the specific information that a given country or language needs. In the second, programs are internationalized so that they can use tables of localized data that are bound in at runtime; the programs are *not* tailored to specific locales. The second approach speeds and simplifies the development of systems that meet local user needs, and reduces the maintenance of those systems.

6.2.2 Data Transparency

ASCII supports only a limited number of languages. New codesets have been created that include a much larger variety of characters. One feature is common to all the new sets: they need all eight bits of a byte to encode the characters. All programs must treat characters as basic data units, and therefore they cannot alter any bits of the byte(s).

6.2.3 Collation

English sorting rules are among the simplest of any language: each letter sorts to one, and only one, place. ASCII makes things even simpler by encoding the characters in order. Other languages include a variety of collation methods. Here are a few examples:

- **Primary/Secondary.** In this system, a group of characters all sort to the same primary location. If there is a tie, a secondary sort is applied.
- **One-to-Two Character Mappings.** This system requires that certain single characters be treated as if they were two. For example, in German, ß (scharfes-S) is collated as if it were "ss."
- **N-to-One Character Mappings.** Some languages treat a string of characters as if it were one single collating element. For example, in Spanish, the "ch" and "ll" sequences are treated as their own elements within the alphabet. Dictionaries have separate sections for them (that is, there are entries for *a, b, c, ch, d,* and so on). The following words are in correct Spanish order:

canto
construir
curioso
chapa
chocolate
dama

- **Don't-Care Character Mappings.** In some cases, certain characters may be ignored in collation. For example, if a hyphen (-) were defined as a don't-care character, the strings *re-locate* and *relocate* would sort to the same place.

In addition to these collation rules, some languages use basically the same rules as English, but still need more than a plain ASCII sort. For example, in Danish, there are three characters that appear *after* z in the alphabet. This means that you cannot assume that the range [A-Z,a-z] includes every letter.

6.2.4 Character Classification

The new characters that are necessary to support languages besides English need classification. For European languages you just need to expand the set of "alpha" characters to include the additional letters. There are a few exceptions, however. For example, the German alphabet has a lowercase letter that has no uppercase equivalent. Therefore, **islower** would return TRUE on this letter, while **toupper** would return the original character.

6.2.5 Date and Time Conventions

Users around the world express dates and times with different formatting conventions. When specifying day and month names, Americans generally express using this format:

Tue, May 22, 1990

while the French would use this format:

Mardi, 22 mai 1990

An internationalized system gives users access to *their* language's names. People also express numeric dates in different ways, even within a single country. These examples show common methods for formatting dates. These formats, however, are not the only way to write the date in the listed country:

3/20/90	American: month/day/year order
20/3/90	British: day/month/year order
20.3.90	French: day.month.year order

As with dates, there are many conventions for expressing the time of day. Americans use the 12-hour clock with its a.m. and p.m. designations, while most other people in Europe and Asia use the 24-hour clock for written times.

In addition to the 12-hour/24-hour clock differences, punctuation for written times can vary. For example:

3:20 p.m.	American
15.20	German

With different date and time formats come different time zones, which can vary in 1-hour, 30-minute, or even 15-minute increments.

6.2.6 Numeric and Monetary Formatting

The characters used to format numeric and monetary values vary from place to place. For example, Americans use a period (.) as the radix character (that is, the character that separates whole and fractional quantities), and a comma (,) as a thousands separator. In many European countries, these definitions are reversed.

For example:

	Numeric Formats
1,234.56	American; comma as thousands separator; period as radix character
1.234,56	French: period as thousands separator; comma as radix character

	Monetary Formats
$1,234.56	American; dollars
kr1.234,56	Norwegian; krona
DM 1.234,56	German; marks

Note that users may need more than two places for fractional digits with monetary amounts.

6.2.7 Program Messages

One of the most basic user needs is the ability to interact with the system in the local language. This means that it must be possible to see all program messages in the local language and for the program to accept input in that language. Often, programs are written with the English messages hard-coded into the program. In an internationalized system, the messages are put in a separate module and replaced with calls to a messaging system.

6.2.8 Answering Yes or No

Many programs ask questions that need a positive or negative response. Those programs typically look for the English string literals "yes" or "no." An internationalized program lets users enter the words that are appropriate to their language.

6.2.9 Codeset Changes

One of the most sweeping changes for I18N support is the addition of new codesets, or the expansion old ones, to include non-English characters. Because so many programs rely on ASCII in one way or another, all commonly used sets begin with ASCII and then build from there.

6.2.10 Eight-Bit Sets

Codesets that use all eight bits of a byte can support European, Middle Eastern, and other alphabetic languages. Domain/OS supports the most popular standard series called ISO 8859/1.

ISO 8859/1 is often called **Latin-1**. It includes the characters necessary for Western European languages, such as French, German, Italian, and Spanish. Latin-1 is arranged so that it includes ASCII characters at their traditional 0x0-0x7f code positions, and then puts the additional characters the Western European languages need in positions 0xa0-0xff.

Some assumptions that are valid for plain ASCII are not valid for Latin-1 and other new sets. For example, while ASCII letters are arranged in English alphabetical order, the additional letters in Latin-1 are not in any language's order. This means you can no longer compare the numerical value of two characters to decide which collates first. Instead, you have to build tables that describe a character's collation position independently of its encoded value.

Another now-invalid assumption is that the character set can be sorted one, and only one, way. You can sort Latin-1 using French collation rules, or Danish, or other Western European language, and your results will depend on the rules in force when you run your sort. For example, both German and Swedish include the a-umlaut character (code position 0xe4). In German, a-umlaut sorts with a's, while in Swedish it is considered a separate letter and appears after *z* in the alphabet. If you sort Latin-1 text using German rules, a-umlaut sorts one place, and if you sort that same text using Swedish rules, it sorts to a different place.

6.3 Overview of Developing Internationalized Applications

The development of international programs involves placing appropriate NLS-based library calls in programs and using NLS-based utilities to perform other language-dependent tasks.

6.3.1 Processing Language-Dependent Information

Programmers should use appropriate calls to process language- and locale-specific data. This data is kept in separate system language tables, which define various language-dependent entities such as collating sequences, character classification, and date/time formats. The NLS library calls enable the programmer to make this information available to the programs at runtime. Programming tasks include:

1. Calling the **setlocale** function to establish a program's runtime behavior for a specific language, territory, and codeset.
2. Specifying library calls to retrieve appropriate language-specific information. For example, one could use the **toupper** and **tolower** routines for upshifting and downshifting characters according to the rules of the language specified in LANG, (or one could the **setlocale(3c)** call to select the appropriate piece of the program's locale).
3. Specifying library calls to access locale-specific information. Locale-specific data define conventions that differ among locales where a single language is spoken.

   For example, to access the correct radix character, one could use **sscanf**, which reads string values that represent floating point and decimal numbers from a character string and interprets the correct radix character for the language and locale specified in LANG.
4. Specifying the **nl_langinfo** call to retrieve additional locale-specific data, such as the name and abbreviation for each day of the week and month of the year.

Domain/OS provides the following NLS-based library calls:

TABLE 6-1. NLS-Based Library Calls

Functions		
catclose	catgets	catopen
fprint	fscanf	isalnum
isalpha	iscntrl	isgraph
islower	isprint	ispunct
isspace	isupper	nl_langinfo
printf	atof	scanf
setlocale	sprintf	sscanf
strcoll	strftime	strtod
strxfrm	tolower	toupper
vfprintf	vprintf	vsprintf

Included in these calls are language-dependent and locale-specific routines for: upshifting and downshifting characters, identifying character traits, comparing strings, and locale-specific formatting. Also included are calls that enable programmers to use message catalogs.

Reference documentation for these interfaces is available online and in the appropriate programmer's reference: *SysV Programmer's Reference* (order number 005799-A01) and *BSD Programmer's Reference* (order number 005801-A01).

6.3.2 Providing Language-Specific Environments

In addition to the library calls, Domain/OS provides mechanisms to:

- Set up a language-specific environment that specifies the desired native language. LANG can be set by the user or system administrator.
- Build locale-specific databases with the **buildlang** utility.
- Create message catalogs to store user messages apart from the program logic. These messages can be translated into different languages and retrieved by the program at runtime. The message catalog tools include:
  - — The **gencat** utility

    The **gencat** utility merges the message text source file(s) *msgfile* into a formatted message catalog *catfile*. The file *catfile* will be created if it does not already exist.
  - — The **catdump** utility

    The **catdump** utility reverses the effect of **gencat**; takes a formatted message catalog and makes a modifiable message source file. Note that **catdump** is a non-standard command.
  - — The **catopen**, **catgets**, and **catclose** library calls

    These library calls enable programmers to open, close, and retrieve messages from message catalogs.
- Convert coded characters. The **iconv** utility converts the encoding of characters in *file* from one coded character set to another and writes the results to standard output.

Reference documentation for these commands is available online and in the appropriate command reference: *SysV Command Reference* (005798-A02) and *BSD Command Reference* (005800-A02). Reference for the buildlang command is available in the Domain/OS System Administrator's Reference (019207-A00).

The following sections provide more detailed descriptions of the localization process, the message catalog system, building locale-specific databases, and an overview of I18N programming modifications.

6.4 Overview of Localization

Because an internationalized system is capable of presenting information in a variety of ways, there has to be a mechanism for individual sites or users to declare which variety they want to see. This mechanism is called a **locale.**

A locale consists of three parts: language, territory, and codeset. Some standards require users to specify the language segment only, and some standards make no requirements at all about what to specify. Most implementations encourage users to include all three parts. For example, English is spoken in the U.S. and Great Britain, but the two countries use different date, time, and monetary formats. To specify the locale you want, you must include language and territory.

It is not sufficient to give just the territory because in the previous example, the language is implied by the territory. But there are numerous countries in which multiple languages are spoken. For example, Switzerland has four official languages: French, German, Italian, and Romansh. Designating a locale such as Switzerland would not give the system enough information about how the system should interact with the user.

The codeset segment of the locale specifies the name of the code which assigns values to the characters contained in the set. (For example the Latin-1 codeset is expressed as iso88591.)

6.4.1 Locale Naming Conventions

A locale name specifies language, territory, and codeset. ANSI C has specified the C locale, which defines the behavior an American-based system produces without internationalization support.

Domain/OS uses the following outline for its locale names:

```
ll_TT.codeset
```

where:

- ll is the language name. Examples:

en	English
fr	French
de	German (from Deutsch)

- TT is the territory name. Examples:

US	United States
NL	The Netherlands
ES	Spain

- codeset is the name of the codeset. On Domain/OS, this segment should not exceed **_POSIX_NAME_MAX**, as defined in **limits.h**. Examples:

  ASCII
  iso88591

These locale names are based on the following standards:

- Language names follow: ISO 639 *Code for the Representation of Names of Languages*
- Territory names follow: ISO 3166 *Code for the Representation of Names of Countries*

At SR10.4, Domain/OS provides the following Latin-1, iso88591-based locales:

TABLE 6-2. Latin-1 iso88591-based locales

Locale Names	Country	Language
da_DK.iso88591	Denmark	Danish
de_DE.iso88591	Germany	German
en_GB.iso88591	Britain	British English
en_US.iso88591	US	American English
es_ES.iso88591	Spain	Spanish
fi_FI.iso88591	Finland	Finnish
fr_FR.iso88591	France	French
fr_CA.iso88591	Canada	Canadian French
is_IS.iso88591	Iceland	Icelandic
it_IT.iso88591	Italy	Italian
nl_NL.iso88591	Netherlands	Dutch
no_NO.iso88591	Norway	Norwegian
pt_PT.iso88591	Portugal	Portuguese
sv_SE.iso88591	Sweden	Swedish

6.4.2 Locale Categories

Although assigning a value to LANG is the most common way to set a locale, there may be times when you want to assign a particular value to a smaller piece of it. The standards define these smaller categories:

- LC_COLLATE -- Controls collation
- LC_CTYPE -- Controls character classification (ctype functions)
- LC_NUMERIC -- Controls numeric formatting
- LC_MONETARY -- Controls monetary formatting
- LC_TIME -- Controls date and time

As with LANG, all of these categories are environment variables to which you can assign locale names. However, you can add an additional field (*@modifier*) to names for these categories. This allows you to select a specific version of locale-specific data.

For example, a locale might sort data two ways: in dictionary order and in telephone-book order. Suppose the standard setup for this locale uses dictionary order, but you need to use telephone-book order. You might set your environment variables this way:

```
% setenv LANG fr_FR.iso88591
% setenv LC_COLLATE fr_FR.iso88591@phone
```

The explicit setting of LC_COLLATE overrides LANG's implicit setting of that portion of the locale.

There are no standards for the contents of the *modifier* field. Examples of conventions that may be used are "fold" and "nofold" for collation.

6.4.3 Location of Locale-Specific Data

Domain/OS breaks locale-specific data into two groups:

- **Environment tables.** This includes information like month and day names, date formats, limited currency and numeric formatting info, and the character classification information for the **ctype** functions and the collation order for the named locale. These tables are located in:

  /usr/nlslib/<locale>/locale.def

- **Message catalogs.** These are the message strings that programs use. These catalogs are in:

  /usr/nlslib/<locale>/name.cat

There is a separate mechanism for finding message catalogs. The standards define an environment variable called NLSPATH for this job. Domain/OS sets its default value to:

NLSPATH=/usr/nlslib/%L/%N.cat

where %L gets filled in with the current locale name, and %N gets filled in with the value of the <name> argument to **catopen()**.

6.4.4 Limitations of Locale Variables

LANG and the LC_* categories allow you the freedom to set the locale the way *you* want it. But they don't protect you from mistakes. There's nothing to protect you from setting LANG to, say, a Swedish locale, and LC_CTYPE to a French locale. It's likely, though, that the results would not be what you intend.

There is no way to tie locale information to data. This means that the system has no way of knowing what locale you had set when you created a file, and so won't prevent you from processing that data in inappropriate ways later. For example, suppose LANG was set to a German locale when you created file **foo**. Suppose you then reset LANG to a Swedish locale. You now find that German data in **foo** will be sorted according to Swedish rules; so, for example, the a-umlaut character will be considered a separate letter and appears after *z*. (It will not sort with the a's, as you would expect in German). There are no mechanisms to prevent you from resetting LANG or warnings that doing so may cause confusing results.

6.5 Changing the User's Locale

The **setlocale** function is used to determine/set/query a program's national language locale. Together with the environment variables LANG and NLSPATH, **setlocale** can be used to change the user's locale from one national language to another.

6.5.1 Setting Environment Variables

An environment variable *LANG* is a mechanism by which users can specify requirements for program localization. It defines language, territory, and codeset. A unique value of *LANG* is defined for each supported language/territory/codeset combination. Each *LANG* setting includes instances of collating sequence, character conversion, character classification and langinfo tables and message catalogs.

LANG uses the Domain/OS locale naming convention and specifies the language, territory and codeset as follows:

```
language[_territory[.codeset]]
```

where the length of the entire string should not exceed [NL_LANGMAX] characters. (NL_LANGMAX is defined in **limits.h**.)

Either an individual user or a system administrator can set *LANG*. If a system administrator does the work, it's likely that he/she will be setting up the default locale for an entire site. Users still have the freedom to override the default. Following is an example of setting LANG in a C shell:

```
% setenv LANG fr_FR.iso88591
```

This example sets the locale to French for the shell in which it is invoked and all child processes of that shell. If you want another shell to have a different locale, you can reset LANG in that particular shell.

6.5.2 Defining LANG and NLSPATH in Startup Scripts

The NLSPATH environment variable provides the locations of message catalogs, in the form of a search path, and the naming conventions associated with message catalog files.

It is recommended that both LANG and NLSPATH be defined in a startup script along with the other user specified environment variables. (For example, the user might put them in */etc/profile* and export the environment variables after specifying them.)

To set the locale for English/ISO 88591 - Latin 1 code set (default), depending on the environment, the user would add the following lines:

- In BSD csh:(also can be used in /etc/profile)
  ```
  setenv LANG en_US.iso88591.
  setenv NLSPATH /usr/nlslib/%L/%N.cat:/usr/nlslib/%N/%L
  ```
- In SYS5 ksh: (also can be used in /etc/profile)
  ```
  LANG=en_US.iso88591
  NLSPATH=/usr/nlslib/%L/%N.cat:/usr/nlslib/%N/%L
  ```

NLSPATH, in the case above, is set for **catopen(3c)** calls to look for the named catalog and referenced by %N, in:

/usr/nlslib/<*LANG env var*>/<*named catalog*>.cat

and if not found, then in: **/usr/nlslib/**<*named catalog*>/<*LANG env var*>

The internal default search NLSPATH of the message catalog system is:
/usr/nlslib/%L/%N.cat

6.5.3 Example

For example, assume a user application specifies:

- **catd = catopen("product", 0);**
- NLSPATH=/usr/nlslib/%L/%N.cat:/usr/nlslib/%N/%L
- LANG is set to **en_US.iso88591**

  The file *product.cat* would be searched for as follows:

  — Using NLSPATH, the file product.cat searched for as:
  /usr/nlslib/en_US.iso88591/product.cat

  — If not found, the file searched for as:
  /usr/nlslib/product/en_US.iso88591.

  The latter case may be used by applications that prefer to have their various message catalogs grouped under a single program directory, with one message catalog per supported language.

6.5.4 Using the setlocale Function

An internationalized program localizes its runtime behavior for a specific language, territory, and codeset by calling the **setlocale** function to set the program's locale. The **setlocale** function provides the needed information by:

- Explicitly setting the locale for the program, or
- Returning the current value of a named locale category.

The **setlocale** function has this format:

setlocale*(category, locale)*;

where:

— *category* is a constant defined in *<locale.h>* (LC_COLLATE, LC_TYPE, LC_NUMERIC, LC_MONETARY, LC_TIME, LC_ALL).

— *locale* is a pointer to a string containing a hard-coded locale.

There are three ways to set the program locale using the *setlocale* function:

1. **setlocale***(category, string)*
   Sets a specific *category* in the program locale to a specific value of *string*. For example,

   setlocale(LC_ALL, “fr_FR.iso88591”)

In this example, all categories of the program locale are set to the locale corresponding to the string *“fr_FR.iso88591.”* This is defined as the French language spoken in France using the **iso88591** coded character set.

If *string* does not correspond to a valid setting of *locale*, *setlocale* returns a null pointer and the program locale is not changed. Otherwise, *setlocale* returns the name of the locale.

2. **setlocale** (*category*, “*C*”)
   Sets the minimal environment for C translation; is the minimal, uninternationalized environment. This is the default locale.
3. **setlocale** (*category*, "") Sets the *category* to the implementation-defined default, and corresponds to setting the associated environment variables.

6.5.5 Hints for Using setlocale(3c)

Because culture-specific data appears in unexpected places, you may want to include **setlocale** as the first statement in all programs. In addition, some other hints for using **setlocale** are listed below.

- **Selecting a Category**: In general, call **setlocale** with the category LC_ALL rather than one of the specific categories, because determining what kinds of locale-specific information a program (and the library functions it accesses) will use can be difficult. While it is obvious that a utility like **sort** uses collation info, and so would be affected by LC_COLLATE, it might not be so obvious that it also needs the information associated with LC_CTYPE. (The LC_CTYPE setting affects **sort**’s -d and -i options.)
- **Selecting a Locale String**: Assuming you do not want to set the locale to C explicitly, it is best that the *locale* parameter be the empty string rather than a pointer to a hard-coded locale. If you use a hard-coded locale, you are in effect limiting your program’s runtime behavior to what is defined in that one locale. If you use the empty string, however, the behavior can change depending on the value of the environment variables.

Here’s a typical call to **setlocale**:

```
#include <locale.h>
main()
{
 setlocale(LC_ALL, "");
 /* program processing */
}
```

6.5.6 Selecting and Building Locale-Specific Data

This section describes how to have your program retrieve certain kinds of locale-specific information, using the **nl_langinfo** routine. The routine returns a pointer to a string containing information relevant to the particular language or cultural area defined in the program's locale. The header file <langinfo.h>, in turn, defines constants that hold date, time, monetary, numeric, and messaging information. For example,

```
nl_langinfo(ABDAY_1)
```

returns a pointer to the string "Dom" if the language identified by the current locale is Portuguese, and "Sun" if the identified language is English.

6.6 Building Locale Databases

Domain/OS provides a utility called **buildlang**, which enables programmers to build their own locale-specific databases. The **buildlang** utility takes source files containing collation and character classification information, and compiles them into binary objects, called **locale.def** files.

The **buildlang()** command has the following format:

buildlang [-n] *input_file*

buildlang -d [fform] *locale_name*

Without the **d** option, **buildlang** automatically sets up the language environment as specified by *input_file*. The **buildlang** utility reads a *buildlang* script specified in the input file, creates a file called **locale.def**, and installs the file in the appropriate directory.

There are six categories of data in the **locale.def** file, recognized by *setlocale*, which make up a language definition. They are: LC_COLLATE, LC_TYPE, LC_MONETARY, LC_NUMERIC, LC_TIME, and LC_ALL (See Locale Categories section.)

The source file which *buildlang* processes, called a **buildlang script**, consists of the same six locale categories. These files consist of a series of statements and have a specific format. Each category is composed of one or more statements. Each statement begins with a keyword followed by one or more expressions. An expression is a set of well-formed metacharacters, strings, and constants. **buildlang** also recognizes comments and separators. For a complete description of the **buildlang** utility, see the **buildlang** man page.

The following example is a section of a **buildlang script** that specifies the LC_MONETARY category for the **american** locale.

```
# language american
# code set: iso88591

langname "american"
revision "1.1"

####################################################
# Set up the LC_MONETARY  category of the table

LC_MONETARY
int_curr_symbol     "USD "
currency_symbol     "$"
mon_decimal_point   "."
      .
      .
      .

END_LC
```

6.7 The Message Catalog System

The message catalog system allows the programmer to store program messages separate from the logic of a program, to be translated into several languages, and to be retrieved at run-time, according to the language needs of each user.

To facilitate this process, Domain/OS provides:

- The **gencat** utility, which produces a message catalog from one or more message source files
- The **catopen**, **catgets**, and **catclose** functions, which enable programmers to access and retrieve messages from the message catalog.
- The **catdump** utility, which creates a message source file from an existing message catalog.

6.7.1 Message Catalog Calls

Domain/OS uses the collection of interfaces for message catalogs defined in XPG3. The system consists of three basic calls for use in application programs:

- **catopen** for opening a version of a named message catalog as determined by the current locale
- **catgets** for retrieving a specific message string from that catalog
- **catclose** for closing the named catalog

To use the message catalog system, revise traditional **printf** statements to include calls to **catopen**, **catgets**, and **catclose** to access the Message Catalog System and retrieve the program's message strings.

catopen and **catclose** are similar to the standard **open** and **close** calls, so they are not described in detail here. **catgets** has the following syntax:

```
#include <nl_types.h>
char *catgets(catd, set_id, msg_id, s)
```

where

- *catd* is a catalog descriptor that **catopen** returns.
- *set_id* defines the set within the catalog from which the string should be retrieved. Few catalogs are divided into sets, so this parameter often is set to 0 (zero).
- *msg_id* defines which message within the specified set should be retrieved. *msg_id* is an integer, but you can use **#defines** to associate a mnemonic label with an integer.
- *s* is the default string (or a pointer to that string) that should be used if *msg_id* cannot be retrieved from the catalog.

The following simple example shows how to use the messaging calls. The message to be retrieved is message 1 in set 2.

```
#include <stdio.h>
#include <nl_types.h>
main()
{
 nl_catd catd;
 /* Establish the current locale, so that the appropriate */
 /* version of "hello.cat" will be opened.                */
 setlocale(LC_ALL, "");
 catd = catopen("hello.cat", 0);
 /* Retrieve and print the message. The default text is */
 /* included in case the catalog is unavailable.        */
 printf("%s\n", catgets(catd, 2, 1, "hello, world"));
 catclose(catd);
}
```

6.7.2 Mnemonic Labels for Message Identifiers

Many developers prefer to use a mnemonic label for *msg_id* rather than an integer. You can do that by using **#defines** to associate a label with the integer *msg_id.* There are different ways to associate a label with its integer *msg_id*: for example, suppose you created a file called **prog.h** that included the following:

```
#define SET_1               1
#define FILE_NOT_FOUND ((catgets(catd, SET_1, 1, "File not found\n"))
#define CANT_OPEN           2
#define NETWORK_PROB        3
#define PERMISSION_PROB     4
```

Here's how you might use these labels:

```
#include <stdio.h>
#include <nl_types.h>
#include "prog.h"           /* include the labels file */
main()
{
 /* processing . . . */
 printf(FILE_NOT_FOUND);
 /* more processing. . . */
 printf(catgets(catd, SET_1, NETWORK_PROB, "Network failure\n"));
}
```

6.7.3 Variable Ordering of Message Parameters

The examples so far have shown how to handle the simplest case: a message string without parameters to be filled in. But many messages do have parameters, and those parameters require additional programming changes.

When text is translated, the words in the translated version often are in a different order than they were in the original. For example, in English, adjectives generally precede nouns (for example, *the white house*), while in French, they usually follow nouns (for example, *la maison blanche*). When program messages are translated, their parameters may need to be reordered to accommodate the target language's word structure, or other local conventions.

Domain/OS supports XPG3 extensions to the **printf** and **scanf** families of functions to handle the need for variable ordering of parameters. Instead of simply specifying %s or %d, expand the % indicator as follows:

  %n$

where *n* is an integer that gives the position of the argument in the argument list. For example, suppose your program includes this **catgets** call:

```
printf(catgets(catd, SET_1, VAR_NOT_FOUND,
 "Variable %1$s not found in routine %2$s\n"),
 var_name,routine_name);
```

This syntax says to fill in the first format descriptor (%1$s) with the value of the first variable listed (*var_name*), and the second descriptor (%2$s) with the second variable (*routine_name*). Now suppose that when the English message is translated, the parameters get transposed. The string that VAR_NOT_FOUND would return might look like this (in a different language):

```
    "In routine %2$s, variable %1$s not found\n"
```

This syntax would instruct **printf** to fill in the first format descriptor listed (%2$s) with the value of the second variable listed (*routine_name*), and the second descriptor (%1$s) with the first variable (*var_name*).

The advantage to adding the placement indicators is that it allows a translator to change the strings and the ordering of parameters without having to change the source code. If you have a program message with more than one parameter, you should use the *%n$* syntax.

6.7.4 Building a Message Catalog with the Gencat Utility

The **gencat** utility takes one or more source message files and produces either a new message catalog, or merges new message text into an existing catalog. **gencat** has the following syntax:

gencat *catfile msgfile [msgfile...]*

where *catfile* is the target message catalog and *msgfile* is the message source file. If *catfile* exists, the messages and sets defined in *msgfile* are added to *catfile*. If set and message numbers collide, the text in *msgfile* replaces the existing text in *catfile*. If the *catfile* does not exist, it is created.

A source message file has the following format: fields in a message line are separated by single ASCII space or tab; any additional ASCII spaces or tabs are considered as part of the subsequent field. The following example shows typical lines from a source message file:

```
$quote "
$set 1
1 "[Fatal Error:]  "
2 "[Error: %s line %d]  "
3 "[Warning: %s line %d]  "
4 "[Info: %s line %d]  "
10 "%d Errors, "
11 "%d Error, "
12 "%d Warnings, "
13 "%d Warning, "
14 "%d Info messages\n"
15 "%d Info message\n"
19 "usage: gencat catfile msgfile ...\n"
21 "$delset: set number missing.\n"
22 "$delset: can't delete undefined set %d.\n"
25 "$set: set number missing.\n"
26 "No $set directive specified; NL_SETD is the assumed set.\n"
30 "(%s): Space/tab separator required between source fields.\n"
31 "Space/tab separator required between source fields.\n"
32 "Space/tab separator expected but was not found.\n"
35 "$quote: More than a single quote character specified.\n"
36 "$quote: Space/tab separator expected; empty $quote directive assumed.\n"
40 "Unrecognized identifier - line ignored: %s\n"
41 "Unknown keyword identifier '%s'.\n"
```

The Domain/OS convention for accessing a created message catalog is to search in the **/usr/nlslib** subdirectory. You may of course also use a full pathname when specifying the catalog in the catopen call.

6.8 I18N Programming Considerations

The following sections provide additional background information and suggestions for developing I18N programs on Domain/OS

6.8.1 Data Transparency

With all the new codesets, it is no longer appropriate for programs to use the high bit in a byte as a flag, or to manipulate the high bit in any other way. Programs that do use the high bit must be modified.

6.8.2 Explicit Manipulation of the Eighth Bit

Some older programs use the high bit to save information about a character. For example, an editor might set the high bit to mark a text area which is going to be processed, or a driver might set the bit to mark characters where a delayed write can be done. Most of the programs that exhibit this behavior already have been cleaned up, but you can check for either a bitwise & (and) or bitwise ! (or) with the bit masks 0x7f or 0x80. Also look in **#include** files to see whether there are any **#defines** set to these bit masks.

6.8.3 Collation

Collation routines have been modified so that they can sort a character set in a variety of ways. There are two new functions for handling international sorting: **strcoll** and **strxfrm**. They differ from the traditional **strcmp** in that they use the sorting rules defined in a given locale rather than using the ascending machine collation order. The value of LC_COLLATE when your program runs determines the order used. **strcoll** is very similar to **strcmp**, with the same number, type, and order of parameters. But since it is table-driven, it is slower than the older function.

strxfrm is a different type of function. This function transforms the data it gets and returns a string of characters that can be given to **strcmp** to be sorted. It is useful for quick sorting when you have to compare the same set of data several times. If you have to sort a large amount of data, you might choose to run **strxfrm** on each string, and then use **strcmp** to do the actual comparison. The disadvantage of **strxfrm** is that there is no way to do the inverse operation (that is, taking the transformed string and converting it back to the original string), so you have to keep the original string around as well.

6.8.4 Character Classification

The traditional **ctype** macros have been rewritten to be functions that can handle the needs of a variety of 8-bit, alphabetic languages. These changes are not visible, so you need to make only one change to your program to get the new internationalized behavior: you must call **setlocale**. The value of LC_CTYPE when your program runs determines what character classification rules are used.

If you are not using the **ctype** functions, you must remove any sections of code that do their own case conversion or that decide what a character is based on its code value. Replace these nonstandard routines with calls to the **ctype** functions.

6.8.5 Date and Time Formatting

ANSI C defines the new function **strftime** to handle language-independent date and time strings. Use this in place of the **ctime** function to print a user-requested date. Remember that **strftime** takes different parameters than does the older routine. **strftime** includes a format string that allows you to specify the order of the data you want to print.

6.8.6 Numeric/Monetary Formatting

Many of the I/O functions have been extended to accommodate the fact that the locale controls the way numeric and monetary quantities are formatted. For example, the radix character can be the decimal point (for example, 12.34), or the comma (12,34), or something else. The **printf, fprintf, sprintf, vprintf, vfprintf, vsprintf, scanf, fscanf, sscanf, strtod**, and **atof** functions all use the formatting determined by the current setting of LC_NUMERIC and LC_MONETARY.

The only change you must make in your current programs to get this functionality is to add a **setlocale** call at the top of your programs.

6.8.7 Setting Yes/No Responses

Two strings,YESSTR and NOSTR, are defined to hold the string(s) that are appropriate for giving an affirmative or negative answer for the current locale. For example, for Spanish, they might be defined this way:

```
YESSTR=si
NOSTR=no
```

When the program is run, LANG (es_ES.iso88591) determines the values YESSTR and NOSTR return.

You use the XPG3 function **nl_langinfo** and *langinfo.h* include file to retrieve the values of these strings, as shown in the following example:

```
#include <langinfo.h>
setlocale(LC_ALL, "");
do {
 char  ans[10];
 printf(catgets(catd, SETN, YORN, "Answer y or n: "));
 gets(ans);
   }   while (strcmp(ans, nl_langinfo(YESSTR)) != 0) &&
                            (strcmp(ans, nl_langinfo(NOSTR)) != 0);
```

Chapter 7: Basic Linear Algebra Subroutines (BLAS)

BLAS stands for Basic Linear Algebra Subroutines. The BLAS set comprises the low-level vector and matrix routines upon which Linpack (the linear algebra package), eispak (the eigenvector package), etc. are based. These routines are intended to be optimized for each computer system. On Domain/OS, most of these routines are hand-coded or have hand-coded inner loops. Many are wrappers for veclib calls, which are themselves hand-coded.

The BLAS library contains basic vector routines for use in numerical software. There is some overlap in functionality between these routines and the vector library ("vec_$") calls. The principal philosophical differences are:

- The BLAS routines are more-or-less industry standard, coming from such sources as Argonne National Laboratories and the Numerical Algorithms Group in England. They are in wide use on many computer systems. Their names and interface formats are well known, and any implementation must use those names and interfaces. The vec_$ routines, on the other hand, are proprietary, and should be thought of simply as part of the definition of the "virtual machine" under Domain/OS.
- The BLAS routines support single and double precision real, and single and double precision complex data types. They do not support integer data types. The "vec_$" routines support integers but not complex data.

In addition to the generally recognized BLAS routines, blaslib provides some other commonly used numerical procedures, such as Fourier transforms.

7.1 General Description

A few preliminary comments about the BLAS culture:

Most routines come in four types, distinguished by the first letter of the name, as follows:

S - single precision (32 bits)
D - double precision (64 bits)
C - single precision complex (32 bits for each part, 64 bits total)
Z - double precision complex (64 bits for each part, 128 bits total)

(There are a few exceptions to this rule for some very old routines written before this convention was consistently established).

7.2 Organization of BLAS Routines

BLAS routines are divided into three "levels", called BLAS1, BLAS2, and BLAS3:

- BLAS1 routines perform operations such as addition, scaling, dot product, and so forth, on one-dimensional vectors only.
- BLAS2 routines perform operations involving a vector and a matrix, such as application of a matrix to a vector.
- BLAS3 routines perform operations involving two matrices, such as matrix multiplication.

There are a number of other unofficial "de-facto standard" routines that do not fit into the official BLAS design but are in common use by numerical programmers and are provided by various proprietary operating systems. Many of the common unofficial routines are included in this library.

7.3 General Usage Information for BLAS Routines

The following sections contain information pertaining to the usage of BLAS routines.

7.3.1 Result Array Should Not Overlap Any Input Array

For all vector routines, it is forbidden to have the result array overlap any input array. (This is, in fact, a general requirement of the Fortran language). For example, do not attempt to set all elements of an array to 1 by doing

```
A(1) = 1.0
CALL SCOPY(99, A(1), 1, A(2), 1)
```

expecting it to copy the first element into the second, then the second into the third, and so on. Because of pipelining that may differ among system types and software releases, this coding will behave incorrectly. Use SFILL instead.

7.3.2 Use of the 'Stride' Argument

In all BLAS routines, each vector argument is immediately followed by an integer argument known as the "stride", which tells how far to advance through the array to get from one element of the array to the next. That is, the BLAS routines can skip through an array, dealing only with every Nth element, but logically treating those elements as a vector. Normally, the stride will be given as 1. When the stride is positive, the first element that is processed is the element that is passed as the argument (or, if the entire array is passed, element number 1). When the stride is negative, the scan of the vector will be backwards, skipping elements as appropriate, and ENDING on the element that is passed as the argument (or element 1).

For example, if we say

X = SDOT(5, A, 2, B, -3)

indicating that 5 elements are to be processed from each vector, and that the arrays A and B have strides 2 and -3 respectively, the elements of A that are processed are A(1), A(3), A(5), A(7), and A(9). The elements of B that are processed are B(13), B(10), B(7), B(4), and B(1). The result is

A(1)*B(13) + A(3)*B(10) + A(5)*B(7) + A(7)*B(4) + A(9)*B(1)

This rule for treating negative strides is DIFFERENT from the rule observed in veclib. Note also that the stride is ALWAYS present. Unlike veclib, there are no implicit unit stride versions.

7.3.3 Use of the 'Leading Dimension' Argument

Matrix arguments are always immediately followed in the argument list by an integer argument giving the "leading dimension". This must be the first of the two numbers appearing in the dimension statement that declared the array. It tells the subroutine what the interval in memory is between the start of one matrix column and the start of the next. That number may be larger than the size of the "virtual matrix" that the subroutine deals with. For example, if array X is dimensioned with the statement

```
DIMENSION X(5, 5)
```

and we wish to use the upper-left 3x3 corner of it in a matrix-by-vector multiplication, we could say

```
CALL SGEMV('N', 3, 3,  1.0,  X,      5,       V1,    1,  0.0,  V2,      1)
               [M  N    A  matrix leading   vector        B  vector
                                  dimension   in   stride     out   stride]
```

The numbers M and N tell how big a matrix operation to perform, and the leading dimension tells how the columns are glued together.

NOTE: The foregoing assumes you are coding in Fortran. If not, you must lay out matrices in the equivalent way, which requires great care. Remember that Fortran uses column-order storage, also known as "first subscript varying most rapidly". In the above array X, X(2, 1) is followed in memory by X(3, 1), and X(5, 1) is followed by X(1, 2).

7.4 The BLAS Routines

The following sections list the BLAS procedures and provide synopses of the behavior of the single real version of each. In these synopses, Arguments X and Y are vectors (always followed by a stride). Arguments AA, BB and CC are matrices (always followed by a leading dimension). Arguments M, N, and K are integer sizes. Other arguments are scalars of the same type as the vectors and matrices.

NOTE:
The descriptions here are very incomplete. For BLAS2 and BLAS3 routines, especially, you should read further documentation. We particularly recommend the *Linpack User's Guide* by J.J. Dongarra, Moler, Bunch and Stewart (SIAM Press, 1979) and *A set of Level 3 Basic Linear Algebra Subprograms* (Dongarra J. J., Du Croz J.J, Duff, I. and Hammarling, S). Technical Memorandum No.88 (Revision 1), Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.

7.4.1 Standard 'BLAS1' Procedures

TABLE 7-1. Standard 'BLAS1' Procedures

STANDARD "BLAS1" PROCEDURES			
Single Real	**Double Real**	**Single Complex**	**Double Complex**
SDOT	DDOT	CDOTU	ZDOTU
		CDOTC	ZDOTC
SAXPY	DAXPY	CAXPY	ZAXPY
		CAXPYC	ZAXPYC
SCOPY	DCOPY	CCOPY	ZCOPY
SSWAP	DSWAP	CSWAP	ZSWAP
SSCAL	DSCAL	CSCAL	ZSCAL
		CSSCAL	ZDSCAL
SNRM2	DNRM2	SCNRM2	DZNRM2
SASUM	DASUM	SCASUM	DZASUM
ISAMAX	IDAMAX	ICAMAX	IZAMAX
SROTG	DROTG	CROTG	ZROTG
SROT	DROT	CSROT	ZDROT

A = SDOT(N, X, INCX, Y, INCY)

returns the sum from I = 1 to N of X(I)*Y(I). These are functions returning single real, double real, single complex, or double complex type. CDOTU and ZDOTU perform normal (unconjugated) arithmetic. CDOTC and ZDOTC take the complex conjugate of argument X.

SAXPY(N, A, X, INCX, Y, INCY)

for each I in [1..N] sets Y(I) = Y(I) + A*X(I). Procedures CAXPYC and

ZAXPYC take the complex conjugate of argument X before multiplying by A.

SCOPY(N, X, INCX, Y, INCY)
copies vector X into vector Y.

SSWAP(N, X, INCX, Y, INCY)
exchanges vectors X and Y.

SSCAL(N, A, X, INCX)
for each I in [1..N] sets X(I) = A*X(I). For precedures CSSCAL and ZDSCAL the argument A is a REAL (single or double precision) number.

A = SNRM2(N, X, INCX)
returns the square root of the sum from I = 1 to N of the squares of X(I), that is, the Euclidean norm of the vector X. For SCNRM2 and DZNRM2, the result is a REAL (single or double precision) number.

A = SASUM(N, X, INCX)
returns the sum from I = 1 to N of the absolute value of X(I). For SCASUM and DZASUM it adds the absolute values of the real and imaginary parts, and returns a REAL (single or double precision) number.

I = ISAMAX(N, X, INCX)
searches the array X and returns the index of the element with largest absolute value. For ICAMAX and IZAMAX, the sum of the absolute values of the real and imaginary parts is used. If two or more elements have the same absolute value, the first is taken. When the stride is not 1, the returned index is relative to the sequence in which the procedure looked at the vector elements, not necessarily the actual vector index. For example, if the stride is 5, and the third element that the procedure looks at turns out to be the largest, ISAMAX returns 3, even though the vector index of that element is actually 11.

SROTG(A, B, C, S)
Construct a "Givens plane rotation" - this is not a vector function. It sets C and S to the cosine and sine of the angle from the origin to the point (A, B). It overwrites A and B with possibly useful information. For complex types (CROTG/ZROTG) the behavior is more complicated; argument C is always single- or double-precision real. See documentation elsewhere.

SROT(N, X, INCX, Y, INCY, C, S)
Apply a "Givens plane rotation". X and Y are vectors of X and Y coordinates of points. This rotates those points clockwise through the angle whose cosine and sine are given by C and S. For CSROT and ZDROT it performs the equivalent complex arithmetic. See documentation elsewhere.

7.4.2 De-facto Standard Elementary Vector Procedures

These exist only in the real single and double precision forms. They can be thought of as the functions that some people believe should have been included in BLAS1.

TABLE 7-2. De-facto Standard Elementary Vector Procedures

DE-FACTO STANDARD ELEMENTARY VECTOR PROCEDURES	
Single Real	**Double Real**
SFILL	DFILL
ISAMIN	IDAMIN
ISMAX	IDMAX
ISMIN	IDMIN
SNEG	DNEG
SSUM	DSUM
VSNGL	
	VDBLE
SSADD	DSADD
SSSUB	DSSUB
SSMUL	DSMUL
SVNEG	DVNEG
SVABS	DVABS
SVADD	DVADD
SVSUB	DVSUB
SVMUL	DVMUL

SFILL(N, A, X, INCX)
sets X(I) = A

I = ISAMIN(N, X, INCX)
like ISAMAX, but looks for minimum absolute value

I = ISMAX(N, X, INCX)
like ISAMAX, but looks for maximum

I = ISMIN(N, X, INCX)
like ISAMAX, but looks for minimum

SNEG(N, X, INCX)
sets X(I) = - X(I)

A = SSUM(N, X, INCX)
returns sum of X(I)

VSNGL(N, X, INCX, Y, INCY)
converts X(I) [doubleprecision] to Y(I) [singleprecision]

VDBLE(N, X, INCX, Y, INCY)
converts X(I) [singleprecision] to Y(I) [doubleprecision]

SSADD(N, A, X, INCX, Y, INCY)
sets Y(I) = A + X(I)

SSSUB(N, A, X, INCX, Y, INCY)
sets Y(I) = A - X(I)

SSMUL(N, A, X, INCX, Y, INCY)
sets Y(I) = A * X(I)

SVNEG(N, X, INCX, Y, INCY)
sets Y(I) = - X(I)

SVABS(N, X, INCX, Y, INCY)
sets Y(I) = ABS(X(I))

SVADD(N, X, INCX, Y, INCY, Z, INCZ)
sets Z(I) = X(I) + Y(I)

SVSUB(N, X, INCX, Y, INCY, Z, INCZ)
sets Z(I) = X(I) - Y(I)

SVMUL(N, X, INCX, Y, INCY, Z, INCZ)
sets Z(I) = X(I) * Y(I)

7.4.3 De-facto Standard Double-operation Vector Procedures

These exist only in the real single and double precision forms.

TABLE 7-3. De-facto Standard Double-operation Vector Procedures

DE-FACTO STANDARD DOUBLE-OPERATION VECTOR PROCEDURES	
Single Real	**Double Real**
SSVTSP	DSVTSP
SSVMVT	DSVMVT
SSVPVT	DSVPVT
SSVTVM	DSVTVM
SSVTVP	DSVTVP
SSVVMT	DSVVMT
SSVVPT	DSVVPT
SSVVTM	DSVVTM
SSVVTP	DSVVTP
SVVMVT	DVVMVT
SVVPVT	DVVPVT
SVVTVM	DVVTVM
SVVTVP	DVVTVP
SVVVTM	DVVVTM

The names constitute a sort of Polish-notation string describing the operations. "S" means the next argument is a scalar, push it onto the stack. "V" means the next argument is a vector and stride, push it onto the stack. "P", "M", and "T" mean to compute plus, minus, or times, respectively, of the top two items on the stack, and push the result back. The final result is what is left on the stack. Of course, no stacking takes place -- this is merely a mnemonic fiction.

SSVTSP(N, A, B, X, INCX, R, INCR)
sets R(I) = A * X(I) + B

SSVMVT(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = (A - X(I)) * Y(I)

SSVPVT(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = (A + X(I)) * Y(I)

SSVTVM(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = A * X(I) - Y(I)

SSVTVP(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = A * X(I) + Y(I)

SSVVMT(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = A * (X(I) - Y(I))

SSVVPT(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = A * (X(I) + Y(I))

SSVVTM(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = A - X(I) * Y(I)

SSVVTP(N, A, X, INCX, Y, INCY, R, INCR)
sets R(I) = A + X(I) * Y(I)

SVVMVT(N, X, INCX, Y, INCY, Z, INCZ, R, INCR)
sets R(I) = (X(I) - Y(I)) * Z(I)

SVVPVT(N, X, INCX, Y, INCY, Z, INCZ, R, INCR)
sets R(I) = (X(I) + Y(I)) * Z(I)

SVVTVM(N, X, INCX, Y, INCY, Z, INCZ, R, INCR)
sets R(I) = X(I) * Y(I) - Z(I)

SVVTVP(N, X, INCX, Y, INCY, Z, INCZ, R, INCR)
sets R(I) = X(I) * Y(I) + Z(I)

SVVVTM(N, X, INCX, Y, INCY, Z, INCZ, R, INCR)
sets R(I) = X(I) - Y(I) * Z(I)

7.4.4 Standard

In the following synopses, TRANS, DIAG, and UPLO are case-insensitive letters telling how to treat the matrix AA.

For TRANS, 'N' means normal, 'T' means transpose, and 'C' means conjugate transpose (Hermitian adjoint). For real matrices, T and C are the same. The size of AA, before any transposition, is M*N.

UPLO is used for symmetric matrices. Only the upper (UPLO = 'U') or lower (UPLO = 'L') triangular part is actually given. AA is square and of size N*N.

DIAG is used for triangular matrices. 'U' means it is a "unit triangular" matrix, that is, the diagonal elements are assumed to be 1 and will not be read from the array provided. 'N' means it is not assumed to be unit triangular, and the diagonal elements will be read from the array.

TABLE 7-4. Standard 'BLAS2' Procedures

STANDARD "BLAS2" PROCEDURES			
Single Real	**Double Real**	**Single Complex**	**Double Complex**
SGEMV	DGEMV	CGEMV	ZGEMV
SGBMV	DGBMV	CGBMV	ZGBMV
		CHEMV	ZHEMV
		CHBMV	ZHBMV
		CHPMV	ZHPMV
SSYMV	DSYMV		
SSBMV	DSBMV		
SSPMV	DSPMV		
STRMV	DTRMV	CTRMV	ZTRMV
STBMV	DTBMV	CTBMV	ZTBMV
STPMV	DTPMV	CTPMV	ZTPMV
STRSV	DTRSV	CTRSV	ZTRSV
STBSV	DTBSV	CTBSV	ZTBSV
STPSV	DTPSV	CTPSV	ZTPSV
SGER	DGER	CGERU	ZGERU
		CGERC	ZGERC
		CHER	ZHER
		CHPR	ZHPR
		CHER2	ZHER2
		CHPR2	ZHPR2
SSYR	DSYR		
SSPR	DSPR		
SSYR2	DSYR2		
SSPR2	DSPR2		

SGEMV(TRANS, M, N, A, AA, LDA, X, INCX, B, Y, INCY)
sets Y to A*AA*X + B*Y, using transpose or adjoint of AA as appropriate.

SGBMV(TRANS, M, N, KL, KU, A, AA, LDA, X, INCX, B, Y, INCY)
like SGEMV, but AA is a band matrix with KL sub-diagonals and KU super-diagonals. See documentation elsewhere.

SSYMV(UPLO, N, A, AA, LDA, X, INCX, B, Y, INCY)
like SGEMV, but AA is symmetric.

SSBMV(UPLO, N, K, A, AA, LDA, X, INCX, B, Y, INCY)
like SGEMV, but AA is a symmetric band matrix.

SSPMV(UPLO, N, A, AP, X, INCX, B, Y, INCY)
like SGEMV, but AP is a "packed" symmetric matrix. See documentation elsewhere.

STRMV(UPLO, TRANS, DIAG, N, AA, LDA, X, INCX)
sets X to AA*X, using transpose or adjoint of AA as appropriate. AA is an upper or lower triangular matrix.

STBMV(UPLO, TRANS, DIAG, N, K, AA, LDA, X, INCX)
like STRMV, but AA is a band matrix with K off-diagonals

STPMV(UPLO, TRANS, DIAG, N, AP, X, INCX)
like STRMV, but AP is a "packed" triangular matrix.

STRSV(UPLO, TRANS, DIAG, N, AA, LDA, X, INCX)
sets X to inv(AA)*X, using transpose or adjoint of AA as appropriate. AA is an upper or lower triangular matrix.

STBSV(UPLO, TRANS, DIAG, N, K, AA, LDA, X, INCX)
like STRSV, but AA is a band matrix with K off-diagonals

STPSV(UPLO, TRANS, DIAG, N, AP, X, INCX)
like STRSV, but AP is a "packed" symmetric matrix.

SGER(M, N, A, X, INCX, Y, INCY, AA, LDA)
tensor dyadic product - sets AA(I,J) = AA(I,J) + A*X(I)*Y(J).

SSYR(UPLO, N, A, X, INCX, AA, LDA)
symmetric tensor dyadic product - sets AA(I,J) = AA(I,J) + A*X(I)*X(J). AA is symmetric, and only the upper or lower triangular part will be written.

SSPR(UPLO, N, A, X, INCX, AP)
like SSYR, but AP will be a "packed" symmetric matrix.

SSYR2(UPLO, N, A, X, INCX, Y, INCY, AA, LDA)
sets AA(I,J) = AA(I,J) + A*(X(I)*X(J) + X(J)*Y(I)). AA is symmetric, and only the upper or lower triangular part will be written.

SSPR2(UPLO, N, A, X, INCX, Y, INCY, AP)
like SSYR2, but AP will be a "packed" symmetric matrix.

For complex types, the procedures that deal with symmetric matrices are replaced with procedures that deal with Hermitian matrices, and the dyadic product procedure comes in two flavors:

CHEMV(UPLO, N, A, AA, LDA, X, INCX, B, Y, INCY)
like SSYMV

CHBMV(UPLO, N, K, A, AA, LDA, X, INCX, B, Y, INCY)
like SSBMV

CHPMV(UPLO, N, A, AP, X, INCX, B, Y, INCY)
like SSPMV

CGERU(M, N, A, X, INCX, Y, INCY, AA, LDA)
like SGER

CGERC(M, N, A, X, INCX, Y, INCY, AA, LDA)
like SGER, but uses complex conjugate of Y

CHER(UPLO, N, A, X, INCX, AA, LDA)
like SSYR

CHPR(UPLO, N, A, X, INCX, AP)
like SSPR

CHER2(UPLO, N, A, X, INCX, Y, INCY, AA, LDA)
like SSYR2

CHPR2(UPLO, N, A, X, INCX, Y, INCY, AP)
like SSPR2

7.4.5 Standard 'BLAS3' Procedures

In the following synopses, TRANSA and TRANSB control the transposition or Hermitian adjointing of matrices AA and BB, respectively. SIDE controls the order of the matrix multiplication of AA and BB.

TABLE 7-5. Standard 'BLAS3' Procedures

STANDARD "BLAS3" PROCEDURES			
Single Real	Double Real	Single Complex	Double Complex
SGEMM	DGEMM	CGEMM	ZGEMM
SSYMM	DSYMM	CSYMM	ZSYMM
		CHEMM	ZHEMM
SSYRK	DSYRK	CSYRK	ZSYRK
		CHERK	ZHERK
SSYR2K	DSYR2K	CSYR2K	ZSYR2K
		CHER2K	ZHER2K
STRMM	DTRMM	CTRMM	ZTRMM
STRSM	DTRSM	CTRSM	ZTRSM

SGEMM(TRANSA, TRANSB, M, N, K, A, AA, LDA, BB, LDB, B, CC, LDC)

sets CC to A*AA*BB + B*CC, using transpose or adjoint of AA and BB as appropriate.

SSYMM(SIDE, UPLO, M, N, A, AA, LDA, BB, LDB, B, CC, LDC)

sets CC to A*AA*BB + B*CC or A*BB*AA + B*CC, depending on SIDE. AA (only) is symmetric, and only its upper or lower triangle is used.

SSYRK(UPLO, TRANS, N, K, A, AA, LDA, B, CC, LDC)

sets CC to A*AA*transp(AA) + B*CC. CC is symmetric, and only its upper or lower triangle will be written. If TRANS says to transpose, it computes A*transp(AA)*AA + B*CC.

SSYR2K(UPLO, TRANS, N, K, A, AA, LDA, BB, LDB, B, CC, LDC)

sets CC to A*(AA*transp(BB) + BB*transp(AA)) + B*CC. CC is symmmetric, and only its upper or lower triangle will be written. If TRANS says to transpose, it computes A*(transp(AA)*BB + transp(BB)*AA) + B*CC.

STRMM(SIDE, UPLO, TRANSA, DIAG, M, N, A, AA, LDA, BB, LDB)

sets BB to A*AA*BB or A*BB*AA, depending on SIDE. The transpose or adjoint of AA is used if so specified by TRANSA. AA (only) is triangular.

STRSM(SIDE, UPLO, TRANSA, DIAG, M, N, A, AA, LDA, BB, LDB)

sets BB to A*inv(AA)*BB or A*BB*inv(AA), depending on SIDE. The transpose or adjoint of AA is used if so specified by TRANSA. AA (only) is triangular.

For complex types, procedures that deal with Hermitian matrices have been provided, in addition to the ones that deal with plain symmetric matrices:

CHEMM(SIDE, UPLO, M, N, A, AA, LDA, BB, LDB, B, CC, LDC)

Like CSYMM, but AA is Hermitian.

CHERK(UPLO, TRANS, N, K, A, AA, LDA, B, CC, LDC)

Like CSYRK, but the adjoint of AA is used instead of its transpose. CC is Hermitian.

CHER2K(UPLO, TRANS, N, K, A, AA, LDA, BB, LDB, B, CC, LDC)

Like CSYR2K, but the adjoint of AA is used instead of its transpose. CC is Hermitian.

7.4.6 De-facto Standard Matrix Multiply Procedures

TABLE 7-6. De-facto Standard Matrix Multiply Procedures

DE-FACTO STANDARD MATRIX MULTIPLY PROCEDURES			
Single Real	**Double Real**	**Single Complex**	**Double Complex**
SMXM	DMXM	CMXMUU	ZMXMUU
		CMXMCU	ZMXMCU
		CMXMUC	ZMXMUC
		CMXMCC	ZMXMCC
SMXMA	DMXMA	CMXMAUU	ZMXMAUU
		CMXMACU	ZMXMACU
		CMXMAUC	ZMXMAUC
		CMXMACC	ZMXMACC

SMXM(AA, NAR, BB, NAC, CC, NBC)

sets CC to AA*BB.

NAR = number of rows of matrices AA and CC.
NAC = number of columns of matrix AA, and
  the number of rows of BB.
NBC = number of columns of BB and CC.

For complex matrices, the next-to-last letter of the procedure name controls complex conjugation of matrix AA, and the last letter controls complex conjugation of matrix BB. 'C' means to use the complex conjugate of each element of the matrix. 'U' means to use the matrix elements in unconjugated form. For example, ZMXMUC sets CC to AA*CONJG(BB).

The "MXM" routines require that the arrays be tightly packed in memory -- there are no "leading" dimension" parameters. Also, there is no ability to transpose the matrices. Both of these restrictions are removed in the "MXMA" routines.

SMXMA(AA, NA, IAD, BB, NB, IBD, CC, NC, ICD, NAR, NAC, NBC)

where:

NA = stride of matrix AA down each column
IAD = stride of matrix AA across each row
NB = stride of matrix BB down each column
IBD = stride of matrix BB across each row
NC = stride of matrix CC down each column
ICD = stride of matrix CC across each row
NAR = number of rows of matrices AA and CC.
NAC = number of columns of matrix AA, and
the number of rows of BB.
NBC = number of columns of BB and CC.

The letter "A" in the name means "arbitrary spacing". These routines take extra arguments telling how the matrices are arranged in memory. The arguments NAR, NAC, and NBC tell how many elements the matrices should be considered to have. The arguments NA, NB, and NC tell how far apart in memory consecutive items down each column are located. For matrices stored in the normal manner, this will be 1. The arguments IAD, IBD, and ICD tell how far apart in memory consecutive items across each column are located. For matrices stored in the normal manner, these will be the "leading dimensions" of the matrices. If the matrices are tightly packed, these numbers will be the same as the number of rows.

Hence, SMXM(AA, NAR, BB, NAC, CC, NBC) is equivalent to SMXMA(AA, 1, NAR, BB, 1, NAC, CC, 1, NAR, NAR, NAC, NBC).

To use the transpose of an array, just exchange its "NA" and "IAD" arguments, for example.

7.4.7 Fourier Transforms

We have included some trivial complex FFT routines. They are "trivial" in the sense that they have no separate initialization operation and use no temporary workspace. The required trigonometric functions are computed "on the fly" by interpolation from internal tables. Only radix-2 arrays are permitted, that is, the array sizes must be powers of 2.

TABLE 7-7. Fourier Transforms

FOURIER TRANSFORMS	
Single Complex	**Double Complex**
TCFFTF2	TZFFTF2
TCFFTB2	TZFFTB2
TCFFTFN2	TZFFTFN2
TCFFTBN2	TZFFTBN2

The letter just before "FFT" gives the type - C for single complex and Z for double complex. The letter after "FFT" is "F" for a forward transform and "B" for a backward transform. The suffix after that is:

2 one-dimensional, array size must be a power of 2.

N2 multidimensional, array size must be a power of 2 in each dimension. The array sizes in the various dimensions are specified by means of an integer list.

The transforms are not normalized. If an array is transformed forward and back, the result will be the original array with each element multiplied by the total array size.

TCFFTF2(SIZE, A)

SIZE gives the array size. It must be a power of 2. The array A is replaced by its unnormalized complex forward discrete Fourier transform:

```
                                   SIZE
for J in [1, SIZE], new A(J) = SUM    A(K) * EXP(-2*pi*i*(J-1)*(K-1)/SIZE)
                                   K=0
```

TCFFTB2(SIZE, A)

As above, but computes the unnormalized backward transform:

```
                                   SIZE
for J in [1, SIZE], new A(J) = SUM    A(K) * EXP(2*pi*i*(J-1)*(K-1)/SIZE)
                                   K=0
```

TCFFTFN2(RANK, SIZES, A)

RANK gives the number of dimensions. SIZES is an array of length RANK listing the individual sizes in each dimension. Each entry in this list must be a power of 2. The product of these entries is the actual size of the array A. The array A is replaced by its unnormalized complex forward discrete Fourier transform:

```
 for J1 in [1, SIZE(1)],  J2 in [1, SIZE(2)], ... , Jr in [1, SIZE(r)],

                          SIZE(1)  SIZE(2)     SIZE(r)
new A(J1, J2, ... , Jr) = SUM      SUM  ...    SUM     A(K1, K2, ... , Kr) *
                           K1=0    K2=0                                  Kr=0

                 (J1-1)*(K1-1)   (J2-1)*(K2-1)        (Jr-1)*(Kr-1)
    EXP(-2*pi*i* (  ------------- + ------------- + ... + -------------  ))
                   SIZE(1)         SIZE(2)             SIZE(r)
```

The multidimensional array A is stored in Fortran subscript order, that is, the first subscript varies most rapidly.

TCFFTBN2(RANK, SIZES, A)

As above, but computes the unnormalized backward transform:

```
 for J1 in [1, SIZE(1)],  J2 in [1, SIZE(2)], ... , Jr in [1, SIZE(r)],

                          SIZE(1)  SIZE(2)     SIZE(r)
new A(J1, J2, ... , Jr) = SUM      SUM  ...    SUM     A(K1, K2, ... , Kr) *
                           K1=0    K2=0        Kr=0


                (J1-1)*(K1-1)   (J2-1)*(K2-1)        (Jr-1)*(Kr-1)
    EXP(2*pi*i* (  ------------- + ------------- + ... + -------------  ))
                   SIZE(1)         SIZE(2)             SIZE(r)
```

HEWLETT
PACKARD

Order Number
D-18901-E

**Manufacturing
Part No.
018901-A00**

018901-A00