

Domain System Software Release Notes

Software Release 10.0
BETA2 VERSION: 4/19/88
Order No. 005809-A01

Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824

THE TERMS AND CONDITIONS GOVERNING THE SALE OF APOLLO COMPUTER INC. HARDWARE PRODUCTS AND THE LICENSING OF APOLLO COMPUTER INC. SOFTWARE PRODUCTS CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN APOLLO COMPUTER INC. AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY APOLLO COMPUTER INC. FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY BY APOLLO COMPUTER INC. WHATSOEVER.

IN NO EVENT SHALL APOLLO COMPUTER INC. BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATING TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF APOLLO COMPUTER INC. HAS BEEN ADVISED, KNEW, OR SHOULD HAVE KNOWN OF THE POSSIBILITY OF SUCH DAMAGES.

THE SOFTWARE PROGRAMS DESCRIBED IN THIS DOCUMENT ARE CONFIDENTIAL INFORMATION AND PROPRIETARY PRODUCTS OF APOLLO COMPUTER INC. OR ITS LICENSORS.

This document was formatted on a Domain node using the troff text formatter distributed with Domain[®]/OS software.

© 1988 Apollo Computer Inc. All rights reserved.

Printed in U.S.A.

First Printing: February, 1988

**This manual is dedicated to the memory of Kriss Kellerman,
a valued friend and colleague.**

Preface

This document describes standard Domain System software for the initial Beta version of Software Release (SR) 10.0. It includes an overview of new and changed functionality, a summary of bugs fixed since the last software release, a list of known errors and limitations, and a summary of changes to documentation.

The normal software installation process places a version of these release notes in each node's `/install/doc/apllo` directory. Release notes for optional Domain software are installed in a similar manner. If you are upgrading from a previous software release to SR10.0, be sure to read *Making the Transition to SR10-Based Operating Systems* (Order No. 011435) before installing SR10.

Apollo Computer Inc. reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Apollo Computer Inc. to determine whether any such changes have been made.

How to Print the Release Notes

You may print the online copy of this document.

If your installation uses the System V `lp` print daemon, use an `lp` command similar to the following:

```
lp -dprinter_name pathname
```

where *pathname* is the pathname of the release notes, usually

Apollo and Domain are registered trademarks of Apollo Computer Inc.

3DGMR, Aegis, D3M, DGR, Domain/Access, Domain/Ada, Domain/Bridge, Domain/C, Domain/ComController, Domain/CommonLISP, Domain/CORE, Domain/Debug, Domain/DFL, Domain/Dialogue, Domain/DQC, Domain/IX, Domain/Laser-26, Domain/LISP, Domain/PAK, Domain/PCC, Domain/PCI, Domain/SNA, Domain/X.25, DPSS, DPSS/MAIL, DSEE, FPX, GMR, GPR, GSR, NCK, NLS, Network Computing Kernel, Network Computing System, Network License Server, Open Dialogue, Open System Toolkit, Personal Super Workstation, Personal Workstation, Series 3000, and Series 4000 are trademarks of Apollo Computer Inc.

UNIX is a registered trademark of AT&T.

ETHERNET is a registered trademark of Xerox Corp.

IBM is a registered trademark of International Business Machines.

PostScript is a registered trademark of Adobe Systems.

Ada is a registered trademark of the U.S. Government (Ada Joint Program Office).

MULTIBUS is a trademark of Intel Corp.

NFS is a registered trademark of Sun Microsystems Inc.

Interleaf is a trademark of Interleaf, Inc.

IMAGEN is a registered trademark of IMAGEN Corp.

/install/doc/apollo/os.v10.0__notes.

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

prf pathname -pr printer_name -npag -point 10

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

lpr -P printer_name pathname

CONTENTS

1.	Changes to System Software	1-1
1.1	New Software Installation Tools	1-2
1.2	New Problem Reporting Procedures	1-2
1.3	The Node and User Environment	1-3
1.3.1	Display Manager	1-3
1.3.2	The /etc Directory	1-3
1.3.3	The /usr/apollo Directory	1-3
1.3.4	Key Definitions	1-4
1.3.4.1	Key Definition Files and Initialization	1-4
1.3.4.2	Additional Key Definition Changes	1-4
1.3.4.3	Keyboard Support of 8-Bit Fonts	1-5
1.3.5	Font Support	1-5
1.3.6	8-bit Support	1-5
1.3.7	Font Height	1-6
1.3.8	Initialization, Login, and Shell Startup	1-6
1.3.9	Aegis Shell	1-6
1.3.10	The Print Environment	1-7
1.3.11	SIO Lines	1-7
1.3.12	Handling Devices and Mounting	1-7
1.3.13	Project Lists	1-8
1.3.14	Process Accounting	1-8
1.3.15	Log-In Accounting	1-8
1.3.16	Backup and rwmt	1-8
1.3.17	Korn Shell	1-8
1.3.18	DPSS Mail to UNIX mail/UUCP UNIX mail	1-8
1.3.19	Documenters' Workbench	1-9
1.3.20	SPM	1-9
1.3.21	Integrated TCP/IP	1-9
1.3.22	HoneyDanBer UUCP	1-9
1.3.23	netman and Diskless Nodes	1-9
1.3.24	Changes in Domain Internet Support	1-10
1.3.25	OS Name and the MD	1-10
1.3.26	Libraries in /lib	1-10
1.3.27	SysV terminfo Files	1-11
1.3.28	ns_helper	1-11
1.3.29	/usr/new Directory	1-11
1.3.30	'node_data	1-11
1.4	The File System and Naming	1-11
1.5	The Protection Model and ACLs	1-12
1.5.1	Other Protection Mechanisms	1-12
1.6	The Registry	1-12
1.7	The Programming Environment	1-13
1.7.1	Case Sensitivity	1-14
1.7.2	Process Invocation and Absolute Code	1-14
1.7.3	Object File Format	1-14

1.7.4	Long Names	1-14
1.7.5	Data Alignment	1-14
1.7.6	Changes in Standard Streams	1-15
1.7.7	Directories	1-15
1.7.8	System V STREAMS and TLI	1-15
1.7.9	Std_\$call Convention	1-15
1.7.10	Opening and Accessing Files that Lack Type Managers	1-15
1.7.11	malloc() and rws_\$alloc	1-16
1.7.12	New Compiler Features	1-16
1.7.13	A New Debugger	1-16
1.7.14	The Bind Utility	1-17
1.7.15	The lbr Utility	1-17
1.7.16	The apollo_\$std.h File	1-17
1.8	Printer Support	1-17
1.8.1	The Print Manager	1-18
1.8.2	The Print Server	1-18
1.8.3	The prf Command	1-18
1.8.4	The pre10q Daemon	1-19
1.9	Changes to TCP/IP Support	1-19
1.10	GPIO Support	1-20
1.11	Network Computing Support	1-21
1.12	GSR	1-21
2.	Software and Hardware Compatibility	2-1
2.1	Hardware Configuration Considerations	2-1
2.2	Object Module Compatibility	2-1
2.3	Network Compatibility	2-2
2.4	Media Compatibility	2-2
2.5	Layered Product Versions Compatible with SR10	2-3
2.6	SR9.7 Compatibility Items	2-6
2.7	Optional Products Released in Multiple Versions	2-7
2.8	Installation of Pre-SR10 Optional Products	2-7
2.9	2DGMR Unbundled at SR10	2-8
3.	Fixed Bugs, Limitations, and Known Bugs	3-1
3.1	Bugs Fixed Since SR9.7	3-1
3.1.1	Fork	3-1
3.1.2	The Bind Utility	3-1
3.1.3	Domain/OS Calls and Insert Files	3-1
3.1.4	COFF UID	3-2
3.2	Berkeley Distribution Bug Fixed	3-2
3.3	Known Restrictions at SR10	3-2
3.3.1	Booting Diskless from DN5x0 Nodes	3-2
3.3.2	EX DOMAIN_OS on DN5x0 Turbo Systems	3-2
3.3.3	Installation from Media	3-2
3.3.4	Aegis Shell	3-2
3.3.5	C Shell	3-4

3.3.6	BSD Bourne Shell	3-4
3.3.7	Aegis Commands in UNIX pipes	3-4
3.3.8	Aegis Commands	3-5
3.3.9	Common commands	3-5
3.3.10	Commands and Files in /etc	3-5
3.3.11	BSD Commands	3-6
3.3.12	SysV Commands	3-6
3.3.13	Access to New Object Types	3-6
3.3.14	'node_data	3-7
3.3.15	inprocess	3-7
3.3.16	SR9.x File Ownership and Setuid	3-7
3.3.17	8-Bit Font Support	3-7
3.3.18	8-Bit Character Support	3-8
3.3.19	Blasting Processes	3-9
3.3.20	getty On an Aegis-Only System	3-9
3.3.21	getty and uucp dialout	3-10
3.3.22	Cartridge Tape Drive	3-10
3.3.23	Printing	3-10
3.3.24	Links	3-10
3.3.25	COFF Objects and Type Managers	3-10
3.3.26	COFF objects and the DM	3-11
3.3.27	Creating .bak Files from SR9.7 Nodes	3-11
3.3.28	rbak of SR10 nodes by SR9.7	3-11
3.3.29	The stream_\$redefine Call	3-11
3.3.30	pfm_\$cleanup Call	3-12
3.3.31	UNIX nice Call	3-12
3.3.32	SysV open Call	3-12
3.3.33	BSD ttyslot Call	3-12
3.3.34	Registry Creation	3-12
3.3.35	TCP/IP	3-12
3.3.36	UUCP	3-13
3.3.37	SAX on a 30MB disk	3-13
3.3.38	GPR	3-13
3.3.39	SysV open call	3-13
3.3.40	Layered Products that Do Not Work On SR10	3-13
3.3.41	C Compilers	3-14
3.3.42	GPIO	3-14
3.3.43	Domain/Delphi	3-14
3.3.44	Open Systems Toolkit	3-14
3.3.45	DSEE Tutorial	3-14
3.4	SR10-Related SR9.7 Restrictions	3-14
3.5	Known Bugs at SR10.0	3-14
3.5.1	Base Software Installation	3-14
3.5.2	NFS Installation	3-15
3.5.3	C Compiler Installation	3-15
3.5.4	No Shell After Invol	3-15
3.5.5	DM Display of Tabs	3-15

3.5.6	Init	3-15
3.5.7	DM Boot over SIO line	3-16
3.5.8	calendar and invol	3-16
3.5.9	UNIX Default quit Character	3-16
3.5.10	Aegis Shell	3-16
3.5.11	Aegis Commands	3-16
3.5.12	Common Commands	3-17
3.5.13	/etc Commands and Files	3-17
3.5.14	SysV Commands and Utilities	3-18
3.5.15	BSD Commands and Utilities	3-18
3.5.16	telnet	3-18
3.5.17	/usr/adm/lastlog	3-18
3.5.18	Keys Set by kbm	3-18
3.5.19	Lost Keyboard Events	3-18
3.5.20	Printing	3-18
3.5.21	gm_\$init, Domain/Dialogue	3-19
3.5.22	vfmt Calls	3-19
3.5.23	smdu Insert Files	3-19
3.5.24	getpass and /bin/login	3-19
3.5.25	closedir	3-19
3.5.26	UUCP	3-19
3.5.27	GPIO	3-20
3.5.28	DDE	3-20
3.5.29	Pad Manager	3-21
3.5.30	Tools Not Available at Beta	3-22
3.5.31	DN5x0-T Cartridge Tape Access	3-22
3.5.32	DN5x0 Initialization Messages	3-22
3.5.33	Documentation Bugs	3-22
3.5.33.1	GPR	3-22
4.	Installing SR10	4-1
4.1	The /install Directory	4-1
4.2	/etc/invoke_script	4-2
4.3	Beta Installation Considerations	4-2
4.4	Media Types	4-2
4.4.1	Magnetic Tapes	4-3
4.4.2	Streaming Cartridge Tapes	4-3
4.4.3	Floppy Disks	4-3
4.5	Release Configurations	4-3
5.	Documentation	5-1
5.1	Introduction to SR10 Documentation	5-1
5.1.1	Organization of the Documentation Set	5-1
5.1.2	Locating Information in the Documentation Set	5-2
5.1.3	Manuals at SR10	5-3
5.1.3.1	Key SR10 Manuals	5-4
5.1.3.2	SR10 and Pre-SR10 Manuals	5-4
5.2	Changes to Documentation	5-8

5.3	UUCP Configuration Instructions	5-8
5.4	Undocumented GPR Features	5-9
5.4.1	New GPR Subroutine Calls	5-9
5.4.2	Support for the Domain Dial Box	5-11
A.	Changes to Commands and Calls	A-1
A.1	Changes to Commands	A-1
A.1.1	DM Commands	A-1
A.1.2	Commands in /com	A-1
A.1.2.1	Obsolete Commands	A-1
A.1.2.2	Changed Commands	A-2
A.1.2.3	New Commands	A-3
A.1.3	Commands in /usr/apollo/bin	A-3
A.1.4	Domain Extensions to /etc	A-6
A.1.5	TCP/IP Support	A-9
A.1.5.1	Administrative Facilities Files in /etc	A-9
A.1.5.2	SysV TCP/IP Commands and Files	A-10
A.1.5.3	Aegis TCP/IP Commands and Utilities	A-11
A.1.6	SysV Commands	A-11
A.1.6.1	Obsolete or Unsupported Commands	A-11
A.1.6.2	Changed Commands	A-12
A.1.6.3	New Commands	A-12
A.1.7	Changes to BSD Commands and Files	A-14
A.1.7.1	Obsolete or Unsupported Commands	A-14
A.1.7.2	Changed Commands and Files	A-14
A.1.7.3	New Commands	A-15
A.1.7.4	UUCP Support	A-17
A.2	Changes to Calls	A-17
A.2.1	SysV Calls, Subroutines, and File Formats	A-18
A.2.1.1	Obsolete or Unsupported Calls	A-18
A.2.1.2	New and Changed Features	A-18
A.2.2	BSD System Calls, Subroutines, and File Formats	A-23
A.2.2.1	Obsolete or Unsupported Calls	A-23
A.2.2.2	New and Changed Features	A-23
A.2.3	Domain/OS System Calls	A-28
A.2.3.1	The name_\$ and ios_\$ Interfaces	A-28
A.2.3.2	New prf_\$ Calls	A-29
A.2.3.3	New loader_\$ Calls	A-29
A.2.3.4	Changes to gpr_\$ Calls	A-30
A.2.3.5	changes to pad_\$ Calls	A-30
A.2.3.6	Miscellaneous New and Changed Calls and Insert Files	A-30
A.2.3.7	Superseded Interfaces	A-31
A.3	Changes to /systest and /sys	A-31

Chapter 1: Changes to System Software

Domain/OS Software Release 10 (SR10) is a general release of Domain[®] system software. It includes significant new functionality, performance enhancements, and bug fixes for previous releases. Additionally, at SR10 the standard system software contains UNIX functionality that was previously supported by the separate Domain/IX[™] product. SR10 provides additional functionality, performance enhancements and bug fixes to the software that was supplied in previous releases of Domain/IX.

The SR10 release of the Domain operating system provides major benefits to Apollo's customers in the form of an extended Aegis[™] environment and enhanced support in the Domain kernel for UNIX functionality. It contains three separate operating environments that can run independently of each other and concurrently. The Aegis environment provides all the functionality of the Aegis operating system, including the SR10 extensions. The BSD and SysV environments provide users with enhanced 4.3 Berkeley Software Distribution and AT&T System V Release 3 UNIX environments, respectively. The SysV environment is compatible with the System V, Release 3 SVID (Base OS, Base Libraries, and Library Extensions) and includes support for System V STREAMS.

It is important to note that each of these environments runs without relying on the presence of any other. Neither of the UNIX environments, for example, requires a /com directory to run properly.

Major changes and enhancements made to standard Domain system software since SR9.7, include the following:

- Changes in node initialization and log-in procedures
- Enhancements to system administration
- Improvements in the file system
- Changes in the process model
- A new executable binary format
- Enhancements to UNIX support
- New and changed commands
- TCP/IP is now standard in all environments

This document describes changes made between SR9.7 and SR10 and does *not* describe changes made in previous software releases.

If you are upgrading existing SR9.7 or earlier-based systems to SR10, be sure to read *Making the Transition to SR10 Operating System Releases* (Order No. 011435) before installing new software.

This chapter briefly describes changes made to Domain/OS system software. This section is *not* intended to be a complete and detailed description of all changes at SR10. Instead, it covers all areas of change and often refers you to *Making the Transition to SR10 Operating System Releases* or other manuals for detailed information. Additionally, we have included a listings of all new and changed commands and calls in Appendix A of this document.

1.1 New Software Installation Tools

At SR10 we provide a completely new set of software installation procedures that are both easier to use and more powerful than the previous installation scripts. For complete information on the new procedures, refer to *Installing Software with Apollo's Release and Installation Tools*. The major changes from a user perspective are:

- An increase in speed of installation. The new installation tools are based on compiled code rather than shell scripts. Thus, they are significantly faster than the old procedures.
- A more flexible user interface. For a user wanting a standard system or product configuration, the new tools provide a complete system of defaults to eliminate time-consuming product or option selection. For a system administrator or user wishing to tailor a network or node environment to special site or personal requirements, the tools allow for a high degree of customization.
- Integration of system software and layered software product installation. Formerly, installation of more than one product, or of system software and layered software products, had to be done on a serial basis; that is, all the objects in one software product had to be loaded before another product could be selected or configured. The new tools allow users to make all product selections, including selection of system software options and the choice of a default network or node environment, before any software is actually loaded to the target node. A user or system administrator can configure the node or nodes to be installed during the day and run the installation itself during the night.
- Simultaneous installation to multiple nodes. A system administrator can install SR10 software on multiple nodes simultaneously, if the nodes to be installed are running the Server Process Manager (SPM) program.

In addition to providing the new Release and Installation functionality, Apollo provides an updated set of pre-SR10 installation scripts in the `/install/sr9.7_compatibility` directory of the Authorized Area. Use the scripts in this directory to install SR9.7 based optional software products that are compatible with the SR10 version of the Domain/OS operating system and are not being re-released with SR10. The procedures are described in Chapter 5 of *Installing Software with Apollo's Release and Installation Tools*.

1.2 New Problem Reporting Procedures

As of SR10, we have implemented a new format for reporting software problems, called the Apollo Product Report. At the same time, we have improved our ability to

track and respond to product reports. A new command, `/usr/apollo/bin/mkapr`, that generates an Apollo Product Report replaces the `crucr` command, which is now obsolete. The `mkapr` command is available in all Domain/OS environments.

1.3 The Node and User Environment

We've made significant Enhancements to the workstation environment that affect system administration and use. The following sections briefly describe these modifications.

1.3.1 Display Manager

We have made the following changes to the Display Manager:

- The DM now handles text lines of up to 1024 characters, including pathnames up to 1023 characters.
- The DM can now handle 8-bit fonts; see "Font Support," below.
- The DM now creates files of type `unstruct`, unstructured files without a Domain streams header, instead of type `uasc` unstructured ASCII with a streams header.
- The DM no longer sets the execute bit on files that it creates.
- The DM login prompt is now "login:".

1.3.2 The /etc Directory

All systems, even if they have only the Aegis environment, have an `/etc` directory. This directory contains many of the administrative commands and facilities that were located in the `/com` or `/sys` directories. The exact contents of the `/etc` directory depends on the installed environments. Any commands in `/etc` that were in the `/com` directory in previous releases are still available in that directory, usually through links.

Because the `/etc` directory includes node-specific files such as `/etc/ttys`, each node, disked or diskless, has a `'node_data/etc` directory that contains files that are node-specific and the `/etc` directory entries are links to these files.

1.3.3 The /usr/apollo Directory

We now install a `/usr` directory in all environments and all systems, even if they have only the Aegis environment, now have a `/usr/apollo` directory. It contains certain common environment commands and Domain extensions to the UNIX environment. It also includes C include files for Domain system calls, as well as other added-value files.

The `/usr/apollo` directory contains the following subdirectories:

<code>/usr/apollo/bin</code>	This directory contains Domain commands that apply to all three environments or extend the UNIX environment. It includes commands that were previously in the <code>/com</code>
------------------------------	---

directory but are also applicable to the UNIX environments, new commands that apply to all three environments, and commands that are specific to UNIX under Domain/OS. While commands that were in the /com directory in previous releases are still available in that directory, usually through links. /usr/apollo/bin is now on the default search paths for all shells.

/usr/apollo/lib

This directory contains binaries and related files that UNIX users do not run directly. Examples include the cc compiler that the /bin/cc program calls. The ftn compiler is also in this category.

/usr/apollo/include

This directory contains Domain/OS system call C include files with function prototypes. Both /sys5.3/usr/include and /bsd4.3/usr/include contain directories, called apollo, that are links to this directory. Versions of the Aegis system call C insert files that do not have function prototypes are located in the /sys/ins directory for compatibility purposes.

See *Making the Transition to SR10 Operating System Releases* for more information on the /usr/apollo directory and its effects on all three user environments.

1.3.4 Key Definitions

We have made several enhancements to the key definitions at SR10; these are described in the following subsections.

1.3.4.1 Key Definition Files and Initialization

At SR10, we have changed the files in the /sys/dm directory from which the Display Manager (DM) automatically loads the key definitions at login. The exact files that are used depend on the specific keyboard and environment. For detailed information on key definition initialization, see the *Using Your Environment* and *Managing System Software* manuals for your environment(s).

AT SR10 key definitions support 8-bit characters. To accommodate this change, new files are used to store the binary versions of the most recent key definitions. For more information on these changes and their effects, see *Making the Transition to SR10 Operating System Releases*.

1.3.4.2 Additional Key Definition Changes

You can now define the SHIFT/key and CTRL/key combinations for the following keys for Model II (including international) keyboards: BACK SPACE (bs), ESC (esc), RETURN (cr), and DELETE (del). You can also define CTRL/key combinations for all left keypad, (l1-lf), right keypad (r1-r6) and numeric keypad keys. You cannot define any of these key combinations for Model I (old-style low-profile) or earlier keyboards.

Several standard key definitions have been changed. These changes are described in detail in *Making the Transition to SR10 Operating System Releases*. (Key definitions are still user-modifiable.)

1.3.4.3 Keyboard Support of 8-Bit Fonts

At SR10 we provide support for the composition of 8-bit characters, that is, characters in an 8-bit font where the the topmost bit is set. You can compose 8-bit characters on all keyboards through user-settable compose and alt key functions. We also supply a new command, `/usr/apollo/bin/kbm`, that displays the current keyboard characteristics and sets the compose key and the multi-national long and short shift keys.

1.3.5 Font Support

The following changes have been made in font support at SR10.

1.3.6 8-bit Support

SR10 includes new support for 8-bit fonts, that is, fonts that provide up to 256 characters. The following font sets are now supplied in 8-bit format and conform to the ISO 8-bit character set commonly known as Latin-1, which includes all the characters used in Western European languages.

- Courier family
- `din_f7x11`
- `f5x9`
- `f7x13`
- `f7x13.b`
- Helvetica family
- Times family
- Legend family
- Std family

The remaining standard fonts do not include 8-bit characters and are still supplied in 7-bit font format.

Because users may wish to continue using the keyboard character encoding of the pre-SR10 national fonts, we continue to provide the 7-bit versions of these fonts. However, we also provide a set of tools in `/usr/apollo/bin`, such as `french_to_iso`, that convert files written with the 7-bit format national fonts into 8-bit ISO Latin-1 format. (Note that these tools are SysV Bourne Shell scripts and therefore require SysV software to run.)

The SysV Bourne and Korn shells and the Aegis shell provide 8-bit support, as does DPSS/Mail. The keyboard supports 8-bit character input, and we provide tools for

creating 8-bit fonts and for converting between 7-bit and 8-bit fonts.

Other features of the new font support include:

- Mono- and proportionally-spaced fonts
- Font file field to specify bytes per character
- Minimum and maximum character code field
- Bounding box coordinates around character
- Underlining specification

Programs that "knew" about the old format of fonts, i.e., included the `smdu.ins` file in their program, will NOT automatically work at SR10.

For detailed information on using 7-bit and 8-bit fonts, see *Making the Transition to SR10 Operating System Releases*. See Chapter 3 of these notes for a detailed list of restrictions on 8-bit fonts.

1.3.7 Font Height

The fonts in the Courier, Times, and Helvetica families are slightly higher than they were at SR9.7.

1.3.8 Initialization, Login, and Shell Startup

The node initialization, login, and shell startup sequences and procedures have been revised. The initialization sequence is designed to be independent of the windowing system and conforms more closely to the UNIX startup mechanisms. Login and startup procedures have added flexibility.

At SR10, the `init` process runs as Process 1, instead of the DM. It executes the file `'node_data/etc/rc` and invokes `getty` to initialize the serial lines and start the DM or SPM. Also, each node's default environment and `systype` are set by a `/etc/envron` file, while a `user_data/envron` file specifies each user's default environment and `systype`. See the User's guide for your environment, *Making the Transition to SR10 Operating System Releases*, and the *Managing System Software* manuals for detailed information on the initialization, login, and shell startup procedures.

1.3.9 Aegis Shell

The following changes have been made to the Aegis shell:

- The shell now runs `user_data/sh/startup` in the home directory for every interactive version of the shell that is run.
- The shell now runs `user_data/sh/login` any time it is invoked from a login program such as `/com/login`, `siologin`, `crp` (without `-me`). More specifically (for programmers) the shell can be made to run its login program by invoking it with `argv[0]` set to `'-sh'`.
- The shell now checks that any program or shell script it invokes has execute permission for the current user.

- Environment variables can be defined at shell startup time by using `-Dvariable[=value]`. This feature can be used to make startup scripts branch depending on how they were invoked.
- The shell now supports 8-bit character input and output.
- The shell now supports the use of a CSR (command search rules) environment variable. If this variable exists it is expected to contain a list of blank-separated pathnames that are used to determine where to look for executable files. Changes to the search path via the `csr` command will change this variable and changes to the variable will be reflected in the command search rules reported by the `csr` command. However, unlike the `csr` command, changes to the CSR variable will *not* be inherited by the parent of a shell script.
- All internal shell commands now output their errors to `errout`. As a result, error messages from internal commands can be redirected.
- `Stdout` and `errout` can now be merged into a single file via `(> foo >? foo)`.
- The shell will continue searching through its search rules until it finds an executable file that matches the specified name. Thus if you have a shell script called `ld` earlier in your search path than `/com/ld`, and that shell script is not executable, the shell will execute `/com/ld` and will not give you an error. However if it cannot find a copy of `ld` anywhere else in your search path, then you will get an error about the lack of execute permissions.
- The shell no longer prints a one line status after a fault or undefined global error. This information can be obtained by using the `tb` command.

1.3.10 The Print Environment

Changes to the print environment at SR10 include enhancements to the `prf` command to provide queue and job control, a new interface for adding drivers to `prsvr`, and a print manager that allocates and coordinates print resources in a network. See the "Changes to Printer Support" section of this chapter for more information.

1.3.11 SIO Lines

We now provide full UNIX tty support (`getty` and `init`) for remote login over SIO (Serial I/O) lines, as well as `siomonit` and `siologin`.

UNIX SIO line software now fully supports all UNIX line protocol semantics. The Aegis SIO software continues to support the Aegis semantics. As a result, the `/dev/siox` files and the corresponding `/dev/ttyxx` files refer to the same devices, except that the state of the DCD line is always ignored on open by `/dev/siox` devices. For a detailed description of this change and its effects, see *Making the Transition to SR10 Operating System Releases*.

1.3.12 Handling Devices and Mounting

via `/dev`" Both SR10 UNIX environments support handling devices in the `/dev`

directory and building special files with the **mknod** command.

1.3.13 Project Lists

Project lists (group sets) are fully supported at SR10 and may be used in all environments. At SR9.7, project list entries matched only ACL entries of the exact form `%project_name.%.%`. At SR10, the project list check substitutes each project in the user's project list into the Subject Identifier (SID) and compares each resulting SID in doing an ACL check. See *Making the Transition to SR10 Operating System Releases* for more details.

1.3.14 Process Accounting

We provide local process accounting with `/etc/sa` and `/etc/accton` in all environments. The on-line manual pages provide details.

1.3.15 Log-In Accounting

We provide facilities to log login events. By editing the `'node_data/etc/login_log.conf` file you can select to log any combination of the following types of logins: DM, window, SPM, and sio line. You can also specify the file that will contain the logged information.

1.3.16 Backup and `rwmt`

We've made changes to the `rbak` and `wbak` commands to support a mixed pre-SR10 and SR10 network. We've also added support to `tar` and `cpio` for archiving typed files.

The `wbak` and `rbak` utilities have been enhanced to allow backup output to be written to a streams object and restore input to be read from a streams object. Streams objects include files, standard input, and standard output. An option has also been added to the `rwmt` utility to disable byte swapping when data is written to magnetic tape. These features provide additional flexibility in the `wbak` process, including the ability to use a disk file as an intermediate staging area for backup output. (The 'disable byte swapping' feature is needed when using `rwmt` to write `wbak` output from a disk file to magnetic tape.)

`Rbak` will no longer restore protected subsystems or `setuid` objects unless the user has permissions to do so.

1.3.17 Korn Shell

We've added support for the Korn shell in both the BSD and SysV environments.

1.3.18 DPSS Mail to UNIX mail/UUCP UNIX mail

SR10 includes the ability to gateway from DPSS mail to UNIX mail and to UUCP. A

field in the person entries in the registry allow users to specify a mail address. The DPSS mail product provides a command, `/com/edsd`, that allows you to edit the entries in a directory of subscribers.

The SR10 BSD and SysV environments support 4.3BSD mail and AT&T System V, Release 3 mail, respectively. Both forms of mail use the `sendmail` utility as a base.

1.3.19 Documenters' Workbench

Both UNIX environments of SR10 Domain/OS support Documenters' Workbench (DWB), including the `ditroff` utility. The Documenter's Workbench is now an optional product. UNIX text formatting tools such as `troff` are no longer part of base SR10 System V software; the BSD environment still includes all tools that are part of the Berkeley distribution.

1.3.20 SPM

At SR10, you can configure a system so that the Server Processor Manager (SPM) prevents unauthorized users from creating processes on or logging into a node. To do so, create a `'node_data/spm_control` file that lists the authorized users. See *Making the Transition to SR10 Operating System Releases* or the *Managing System Software* manuals for more information on this subject.

1.3.21 Integrated TCP/IP

At SR10, TCP/IP software is a part of the base operating system. One version operates in all environments, and the previous layered product is now obsolete. See *Making the Transition to SR10 TCP/IP* for details.

1.3.22 HoneyDanBer UUCP

A single version of UUCP, known as HoneyDanBer, operates in both UNIX environments. It is a more robust implementation providing enhancements such as spool directory trees and better security and error handling. SR10 UUCP communicates over communications lines with most versions of UUCP, including Apollo pre-SR10 versions and other vendors' HoneyDanBer and other UUCP implementations. We also supply a version of HoneyDanBer UUCP that runs on SR9.7 systems and will work correctly on a network with SR10 UUCP. See *Making the Transition to SR10 Operating System Releases* for more information on running UUCP on mixed SR10/SR9.7 networks.

1.3.23 netman and Diskless Nodes

At SR10 the `/sys/net/netman` program automatically executes the commands in the shell script `/sys/net/netman.rc` when a remote node attempts to boot diskless. This script creates and initializes the resources that are required to support the remote node. `/sys/net/netman.rc` is actually a link to either `/sys/net/netman.com_sh` or `/sys/net/netman.bin_sh`, depending upon the default node environment specified during

installation.

We no longer supply the `/install/create_partner` script that was included in previous releases. However, the diskless list mechanism is still in effect.

1.3.24 Changes in Domain Internet Support

The maximum number of hops allowed in a Domain internet has been increased to sixteen provided that every network in the internet meets minimum network throughput performance guidelines. The minimum transmission rate over any single link must be at least one megabit per second with a maximum end-to-end delay of 100 ms. Refer to *Planning Domain Networks and Internets* for descriptions of the internet configurations and equipment that you can use to ensure adequate internet performance.

When you install SR10 on Domain internet routers, you must reassign network numbers to the router's principal and alternate networks. Refer to *Managing Domain/OS and Domain Routing in an Internet* for procedures that describe how to assign network numbers.

1.3.25 OS Name and the MD

The Operating System is now called Domain/OS, not Aegis. As a result, the OS file in the `/saun` directories has been changed to `domain_os` (with an underscore replacing the slash in the name). You can now start the operating system from the mnemonic debugger (MD) by using the following command:

```
EX DOMAIN_OS
```

1.3.26 Libraries in /lib

We have changed the way in which libraries in the `/lib` directory are loaded. The `/etc/sys.conf` file now allows you to specify whether libraries must be loaded into global space or can be loaded into user space, whether the global libraries should be loaded on system initialization or when first needed at runtime, and whether the library is optional (that is, whether the system should initialize successfully if the library is not found).

The following libraries in the `/lib` directory are obsolete at SR10: `/lib/tbllib`, `/lib/unixlib`.

The following libraries are new at SR10:

- libc** The `libc` and `clib` libraries now contain the data and entry points that were previously in the `clib` and `unixlib` libraries.
- rgylib** The registry library; it is required on each system by the new distributed registry.
- cpp** The C preprocessor library; this is not a global library, but a link to `/usr/lib/cpp`.

dblib The debugging library; this is used by the traceback command and for cross-process debugging.

We also supply four libraries in the `/sr9.7_compatibility/sr9.7_executables/lib` directory that should be installed on SR9.7 systems in a network with SR10 systems. The `kg.lib`, `loader.lib`, `prlib`, and `name.lib` libraries enable programs that reference SR10 entry points to run on the SR9.7 nodes. They contain interludes that convert SR10 entry points into SR9.7 entry points. The `prlib` library should replace the standard SR9.7 `prlib`.

1.3.27 SysV terminfo Files

We have made a change to save disk space at SR10. We have created the directory `/usr/lib/terminfo.local` that contains a set of terminfo files that are most commonly used. When `sys5.3 curses` looks for a terminfo file, it first checks to see if the file is under the new directory. If so, it uses that file. If not, it goes on to use the original `/usr/lib/terminfo` directory. Warnings or errors are never generated about missing files in `terminfo.local` (`terminfo.local` is about 78 blocks). You can install `terminfo.local`, `terminfo` or both locally or as links to a centralized directory.

1.3.28 ns_helper

At SR10, `ns_helper` cannot run on diskless nodes. Additionally, the `ns_helper`'s database files have been moved from their SR9.7 location in the directory tree. *Making the Transition to SR10 Operating System Releases* describes procedures for converting an SR9.7 `ns_helper` to SR10.

1.3.29 /usr/new Directory

The `/usr/new` directory contains unsupported miscellaneous tools and commands from the Berkeley 4.3BSD distribution. You can optionally install `/usr/new` when you install either the BSD or SysV environment. It is then put in a single directory under `/usr` (so that its pathname does not include a systype).

1.3.30 'node_data

The `'node_data` directory has been rearranged to put more files into subdirectories. New subdirectories include `system_logs` for logging and dump files, `sysmp` for system temporary files, etc for node-specific `\tc` objects.

1.4 The File System and Naming

The operating system contains both a new file system and a newer, more robust directory structure. The kernel and libraries are completely case sensitive at SR10. Component names in a pathname can be up to 255 characters long, and pathnames can be up to 1023 characters long.

SR10 software can interpret, but no longer create, UNIX symbolic links. All UNIX commands and system calls that manipulate symbolic links now perform equivalent operations on Aegis links.

All directories at SR10 contain the following entries:

- . (dot, the current directory)
- .. (dotdot, parent of the current directory)

The usage of the characters ~ (tilde) and (backslash) at the start of pathnames has been changed. The ~ character (representing the HOME or naming directory) must now be separated from the next component in the pathname by a / character. You can no longer use \ for the parent directory; you must use

1.5 The Protection Model and ACLs

Protection of file system objects is now based on both UNIX protections and the Access Control List (ACL). SR10 ACLs have a simplified set of access rights (pwr_xk), including a new right (k - keep) that allows you to specify an object as not deletable or renamable, regardless of the permissions associated with the containing directory. Each object has a set of required permissions that are stored in such a way as to increase performance. You can add finer degrees of protection to an object with extended ACLs, which are essentially like pre-SR10 ACLs.

The SR10 protection model integrates ACLs and the UNIX protections (modes) more tightly than at previous releases. If you do not use any of the ACL extension features, protection in the two Domain/OS UNIX environments works exactly as you would expect it to work in a UNIX environment, down to the differences in protection inheritance between 4.3BSD and SysV.

Pre-encoded or "canned" ACLs, including the `%.backup.% acl`, are no longer required. Also, the ACL inheritance mechanism has been expanded to allow you to specify UNIX permission inheritance via the initial file and directory ACLs.

See *Making the Transition to SR10 Operating System Releases* for more information on the changes to the protection model and its implications, including its implications for backup and for access on a mixed SR9.7 and SR10 network.

1.5.1 Other Protection Mechanisms

We've added a command, `/etc/lprotect` that allows a node user to protect a node from access by a remote process running as root. We have added a file `'node_data/node_owners` that enables you to specify which users can run `lprotect` on the node and `sigp` any process on the node. Also, a `'node_data/spm_control` file can control the users gaining access to the node via the SPM. See *Making the Transition to SR10 Operating System Releases* for more information on these facilities.

1.6 The Registry

The SR10 registry is based on Apollo's Network Computing SystemTM (NCS); it

comprises a server and a database of naming and account information. Since the registry can be replicated, it scales well to all sizes of networks. In addition, the registry data is highly available, and easily propagated, which is not always the case in an environment that uses a single user account information file.

You can decentralize administration of the registry, if you wish, by changing the ownership of organizations in the registry database. Registry information is manipulated by the `edrgy` command. Access to most of the functionality of the `edrgy` tool is limited to the owner of the entries, typically system administrators, but users can change certain fields of their own entries.

Because the registry uses NCS, each registry site must also run the local location broker daemon, `llbd`. The standard startup script templates include lines that will automatically start this process. Also, at least one node on each network must run the global location broker, `glbd`.

At SR10, we use the UNIX password encryption algorithm to encrypt passwords stored in the registry. For compatibility, passwords are also maintained in SR9.7 encryption.

Apollo's implementation of the two UNIX environments includes the `/etc/passwd` and `/etc/group` files, which contain user account and group information, respectively. These are typed read-only files and are generated by the registry daemon automatically.

1.7 The Programming Environment

Several changes have been made in the programming environment at SR10. These include the following:

- Support for case-sensitive naming
- Single program-per-process invocation model
- COFF object module format
- Support for long names
- New ACL interface semantics
- Changes in standard IOS stream numbering
- Data alignment support in compilers
- Changes in symbolic link support
- Changes in the `malloc` and `rws_malloc` algorithm
- A new debugger

The following subsections summarize these changes. See Chapter 4 of *Making the Transition to SR10 Operating System Releases* for detailed information on changes to the programming environment.

1.7.1 Case Sensitivity

At SR10, the kernel and libraries are case sensitive with respect to pathnames and symbol names. Programs that depend on case-insensitive behavior do not work at SR10. However, we have provided a downcase mode (controlled by the `DOWNCASE` environment variable) as a temporary compatibility measure. Support for `DOWNCASE` will be removed at the next major release.

1.7.2 Process Invocation and Absolute Code

At SR10, the default process invocation model has changed to one in which a new program is always invoked in a new process. This change permits more general support for non-relocatable (absolute) code. While Apollo will continue to support relocatable code and in-process program invocation, we have converted all Apollo software to work when invoked in a separate process.

As part of this change, Domain C, Pascal, and FORTRAN compilers now produce absolute code by default, which will make most programs run faster. You can still produce position-independent (relocatable) code by specifying the `-pic` option on the command line.

Also, at SR10 UNIX shells do not support in-process invocation of commands or programs; that is, they do not support the `inprocess` variable. The Aegis shell still supports in-process invocation, however, when the `INPROCESS` variable is defined.

1.7.3 Object File Format

SR10 compilers generate a new object format which is an extended version of the AT&T Common Object File Format (COFF) standard. COFF object modules are not compatible with the "obj" format object modules produced by previous releases. To make full use of SR10 program development tools, you should recompile all source files. Programs that have knowledge of pre-SR10 object module format must be changed. It is not possible to bind old (obj) and new (COFF) modules into a single object. However, we provide an obj-to-COFF conversion tool which is documented in *Making the Transition to SR10 Operating System Releases*.

1.7.4 Long Names

At SR10, component names can be up to 255 characters long, and pathnames can be up to 1023 characters long. Previous limits were 32 characters for a leaf (component) name and 256 characters for a pathname. New `name_$` system calls have been added to support long names.

1.7.5 Data Alignment

Programs that make use of data structures in which data is naturally aligned, (that is, where the address boundary of each object is a multiple of its length) run faster than programs with data that is not naturally aligned. We, therefore, urge programmers to use naturally aligned data structures in their programs. We have added new facilities

to the Pascal and C compilers to support natural alignment.

1.7.6 Changes in Standard Streams

At SR10, both Aegis and UNIX COFF programs are invoked with three file descriptors, or streams, open, and the Aegis errin stream is no longer supported. Obj objects (pre-SR10 format) are still invoked with four streams. The rules for various situations at SR10 are in *Making the Transition to SR10 Operating System Releases*.

1.7.7 Directories

The Domain/OS naming calls that are new at SR10 and the UNIX directory calls return the . (dot, this directory) and .. (dotdot, parent of this directory) entries.

1.7.8 System V STREAMS and TLI

The SysV programming environment supports System V streams, part of the Network Extensions to System V Release 3. It provides a framework for developing communication services. SysV also supports the Transport Layer Interface (TLI), which provides a protocol-independent interface to the transport layer of communication services developed with STREAMS. See *Getting Started with SysV STREAMS*, *Programming with SysV STREAMS*, and *Programming with the SysV Transport Layer Interface* for detailed information on System V STREAMS and TLI.

1.7.9 Std_\$(call Convention

The std_\$(call calling convention is considered obsolete at SR10. We retain the Aegis system call insert files with std_\$(call in /sys/ins/?.ins.c, but we do not commit to maintaining these files in future releases. We now use the function prototyping feature of ANSI C to provide the same capability with cleaner syntax. Released Aegis system calls have new insert files written in C which contain ANSI function prototypes. The new C insert files reside in /usr/include/apollo/?.h for better integration with the rest of the C library.

1.7.10 Opening and Accessing Files that Lack Type Managers

The Domain IOS streams facility has been modified so that it is possible to open files that do not have type managers. In the past, attempts to open such files through streams (ios_\$(open, Pascal/FTN open, UNIX open(2), etc.) always failed. Since UNIX programs do I/O to files *only* via streams, they could not access such files.

At SR10, the IOS streams facility will open a file of a type that has no manager if the type is listable at the node that is opening the file, that is, if the lty command will display that type. (You can make a type listable by using the crty command.) The streams facility will then treat the file exactly as if it were of type "unstruct", an unstructured file without a streams header.

1.7.11 malloc() and rws_\$alloc

The calls `malloc()` and `rws_$alloc` now attempt to reserve the specified amount of backing store on disk. If the requested amount of disk space isn't available, then the calls fail. This behavior ensures that a program will not get a "disk full" fault when it touches data in the allocated area; disk storage is always reserved for allocated memory. As a result of these changes, increased disk space must be available to run programs that do large mallocs or space allocation. This space is made available for reallocation when the program makes the corresponding `free` or `rws_$release` call; it is released when the program terminates.

We have also provided a module to help in debugging malloc calls. `/usr/apollo/lib/libmalloc.dbg` is a library containing versions of `malloc`, `free`, `realloc`, and `calloc` that you can `inlib` if there is an apparent problem with any of these calls. This module is essentially the same version of `malloc` that's supplied in the `libc` global library, but compiled with the full optional checking and statistics gathering turned on. It does fairly stringent checking on each call to `malloc` or `free`; if it discovers storage corruption, it prints a message about what it discovered on `stderr`, and calls `abort()` to produce a `proc_dump`.

1.7.12 New Compiler Features

In addition to generating COFF files and absolute code, the language compilers at SR10 also provide many new and modified options for improved efficiency and UNIX compatibility.

The FORTRAN, C, and Pascal compilers support 8-bit characters in comments and strings. However, they do not support characters with the uppermost bit set in identifier names, and flag attempts to use such characters as errors; therefore, identifiers must be in 7-bit ASCII.

Domain C supports function prototypes as defined in the proposed ANSI standard and in the C++ language. Domain C also allows you to replace certain UNIX mathlib calls with inline code by including the `<builtins.h>` header file. These built in routines provide faster execution times than the corresponding library calls.

Compiler changes for SR10 are fully detailed in the SR10 Domain Language Reference manuals. An overview of the changes is also available in the release notes for these products.

1.7.13 A New Debugger

SR10 replaces `DEBUG` with a new debugger, `Domain/DDE`, which can debug both `obj` and `COFF` format objects. `Domain/DDE` provides many new features such as a menu-driven interface, language-sensitive expression evaluation, multiprocess debugging, and cross-node debugging.

The new debugger also has macros that simulate the old `DEBUG` and `dbx`. The macros to simulate `Domain/Debug` syntax syntax; these macros are located in `/sys/debug/old_debug_macros` and `/sys/debugdbx_macros`, respectively.

DDE cannot debug programs built before SR9.5. For full details see the *Domain Distributed Debugging Environment (Domain/DDE) Reference*.

1.7.14 The Bind Utility

We've made the following changes to the bind utility for SR10:

- The binder now handles only those objects produced by SR10 compilers, SR10 linkers (/com/bind or /bin/ld), or SR10 archivers (com/Blbr or /bin/ar). The binder will not handle any object files compiled prior to SR10.
- The bind utility supports both a -runtime option to control the UNIX run-time system call semantics and a -systype option to control pathname semantics. See the *Domain Binder and Librarian Reference* for details on the options.
- At SR10, you must use the -mergebss bind option whenever you have compiled a C program with /bin/cc or by using /com/cc with the -bss option. When you bind with the -mergebss option, you cannot run the output file through the binder again.
- The binder can now mark object modules to be run with a non-default stack size. (The default is 256k).

1.7.15 The lbr Utility

We've made the following changes to the librarian utility, lbr, for SR10:

- The lbr utility handles only those objects generated by SR10 compilers, SR10 linkers (/com/bindd or /bin/d) or SR10 archivers (/com/lbr or /bin/ar). The lbr utility now generates library files in UNIX archive (ar) file format.
- The format of -list output is now the same as ar -tv format.

1.7.16 The apollo_\$std.h File

The UNIX include file /usr/include/apollo_\$std.h is currently supported for use by the /com/cc compiler. In the UNIX environments, the C preprocessor automatically defines the constants specified in this file. The /com/cc compiler automatically includes the file. However the /com/cc compiler does not issue any warning if the file is missing. Apollo does not commit to support this file in the next major release. Future releases of /com/cc will not automatically include this file, and we will not ship the file as part of the software distribution.

1.8 Printer Support

With the release of SR10, the print architecture has been updated. Two new components, prmgr (print manager) and pre10q (pre-SR10 daemon), have been added to prf and prsvr. The print manager coordinates user print requests generated via prf with printer resources provided by prsvr. The pre-SR10 daemon, pre10q, permits printing pre-SR10 jobs in an SR10 environment. This can occur when a network is operating in a mixed environment (some nodes at SR10, some at pre-SR10 levels). The

following sections describe some of the changes in printer support. For detailed information on configuring and running printer facilities, see the *Managing System Software* manual for your environment.

1.8.1 The Print Manager

The print manager controls one or more print servers. With SR10, it is no longer possible to queue jobs to a non-existent printer. The print manager checks each job to ensure it has a printer with the specified name. If none is found, it rejects the job.

Print servers are "bound" to a print manager by a new option in the print server configuration file called "prmgr_site", which identifies the print manager by the manager's logical name. The print manager logical name can be the same as the spooling node name.

1.8.2 The Print Server

Several new directives have been added to prsvr, and several old ones obsoleted. The following options are now obsolete:

cpi	Replaced by the point option
pageno_column	Use the page_headers, header_string, and footer_string options
page_length	Use the paper_length option
page_width	Use the paper_width option
plot_mode	Ignored by prsvr
print_length	Use the paper_length option
print_width	Use the paper_width option
sio_line	Use the io_device_name option

When the print server starts, it prints a startup page indicating the default settings for a print job. If the print server configuration file contains obsolete directives, they are listed on the startup page. Modify the configuration file to eliminate obsolete components and restart the print server. See the *Managing System Software* manual for your environment for detailed information on the print server and its configuration file options. The help file `/sys/help/prsvr/config.hlp` also documents all configuration options.

1.8.3 The prf Command

The prf command includes the following new options: `-pre10`, `-dialog`, `-check`, `-list_printers`, `-list_sites`, and `-sig_printer`. The `-pre10` and `-check` options together determine both whether the job is queued to an SR9.7 or SR10 print server and the errors that are reported. If you specify `-pre10`, the print request will only be queued to an SR9.X server and you will not get an error message if the requested printer does not exist. If you specify `-check`, the print request will only be queued to an SR10

server and you will get an error message if the requested printer does not exist. If you use neither switch, the job is queued to any available server, and you will not get an error message if the requested printer does not exist.

1.8.4 The pre10q Daemon

The `pre10q` daemon handles queueing of requests from SR9.X nodes to an SR10 print server. The system administrator starts this process to monitor the SR9.X print queue. See chapter 3 of *Printing in the Aegis Environment* or the anaging System Software manual for your environment for details.

1.9 Changes to TCP/IP Support

At SR10, our TCP/IP product is completely restructured. Previously, we offered two TCP/IP products, Domain TCP/IP for the Aegis environment and Domain/IX 4.2BSD TCP/IP for the Domain/IX environment. While these products provided the same basic capabilities, they differed in their configuration processes and application programs. The products also differed because Domain TCP/IP was an optional software package for Aegis users, and Domain/IX 4.2BSD TCP/IP was bundled with Domain/IX operating system software.

At SR10, TCP/IP has a single implementation that follows the 4.3BSD UNIX model and works in all three environments: BSD, Aegis, and SysV. All TCP/IP software, including the 4.3BSD network extension, is included in the standard software distribution, and the TCP/IP layered products are no longer supported under SR10. (However, you can operate a mixed network of SR10 and pre-SR10 TCP/IP hosts.) Pre-SR10 TCP/IP enhancements have been carried forward to preserve functionality. Other upward-compatible enhancements have been introduced to improve TCP/IP performance and administration. In addition, SR10 TCP/IP incorporates numerous bug fixes.

At SR10, programs and utilities associated with the Domain TCP/IP product have been replaced with their 4.3BSD counterparts. For Domain TCP/IP users, this means following new TCP/IP configuration procedures and learning a slightly different user interface for TCP/IP application programs. For Domain/IX users, the transition is to the 4.3BSD model.

TCP/IP internet administration at SR10 also follows the 4.3BSD UNIX model. All administrative files reside in the `/etc` directory. The new 4.3BSD name daemon, `named`, greatly simplifies host-name/address resolution in large TCP/IP internets. A new Apollo option to the `rwho` daemon allows a master node to track all users in the local network or in an entire Domain internet.

Host administration is based on the `/etc/rc` startup file. See *Making the Transition to SR10 TCP/IP* for details about the Domain/OS TCP/IP implementation. See *Configuring and Managing TCP/IP* for administrative information; in particular, see Procedure 3-2, "Configuring a TCP/IP Host or Gateway" for the specific configuration steps that are required before you can use TCP/IP.

For information about using TCP/IP application programs, see *Using TCP/IP Network Applications*. Further documentation for TCP/IP programs and utilities can be found

in the *BSD Programmer's Reference and Managing System Software for BSD*.

All Domain network controllers support TCP/IP host and gateway operations. The device drivers for these controllers are resident in standard software. The EtherController-MB, a special-purpose controller that operates through a GPIO interface, also supports TCP/IP host and gateway operations. The driver for the EtherController-MB is available in the COM_ECMB optional software package.

At SR10, there is one programming interface to TCP/IP, the 4.3BSD socket interface, which now supports raw sockets. Although the socket interface is BSD functionality, it can be used in all three environments. A new manual which describes using the socket interface, *Programming with 4.3BSD Socket Calls*, will be available after the SR10 release. For the present, consult the *Domain Programming Tools Reference* (11010-A00) and the *BSD Programmer's Reference* for information about socket calls.

Berkeley has changed the BSD TCP/IP socket interface between 4.2BSD and 4.3BSD. (The data structure returned by `gethost` was changed.) SR10 TCP/IP conforms to the 4.3BSD standard. Therefore, although an application that was built using 4.2BSD should work under 4.3BSD, it cannot be changed and recompiled using SR10 software without first modifying it to conform to the new interface.

Making the transition to SR10 TCP/IP is a simple exercise for Domain/IX 4.2BSD TCP/IP users, but it is more complex for Domain TCP/IP users. *Making the Transition to SR10 TCP/IP* provides information to aid users of Domain TCP/IP in converting to the BSD model. These tools include program- and file-mapping tables that compare the SR10 and pre-SR10 TCP/IP products. *Configuring and Managing TCP/IP* contains detailed information about setting up TCP/IP hosts and internets.

1.10 GPIO Support

The following general changes occur in Domain GPIO with SR10:

- All `ddfs` (device descriptor files) must be rebuilt with the SR10 version of the `crddf` command.
- All `ddfs` are now character-special devices.
- Global entry points are now case sensitive for case-sensitive languages such as C.
- A new GPIO procedure has been added to support the acquisition of devices in a streams type manager. The procedure is `pbu_$acquire_stream`.

The following GPIO commands and procedures have been changed or added with SR10:

- `aqdev` (changed)
- `crddf` (changed)
- `pbu_$acquire` (changed)
- `pbu_$acquire_stream` (added)

Making the Transition to SR10 Operating System Releases describes in detail the effects of these changes on GPIO components that are provided as part of standard system software, and the GPIO release notes detail all changes in the GPIO product.

1.11 Network Computing Support

While most Network Computing functionality supported by SR10 is not new, the Network Computing Kernel (NCK) software is now required for the operation of both the registry and the new hardcopy system; therefore, we are repeating the information that was included in the SR9.7 Release Notes.

The Network Computing Kernel is a component of the Network Computing System (NCS), a set of tools for heterogeneous distributed computing. Applications can use NCS to distribute both data and program execution among different types of processors and across one or more heterogeneous networks. NCS tools run on Domain workstations and are designed to be portable to other processors.

NCK provides the run-time support for distributed computing, including the run-time support for the registry and hardcopy servers. In addition to NCK, the Network Computing System includes the Network Interface Definition Language (NIDL) and compiler, used to develop distributed applications. Unlike NCK, NIDL is an optional product and is therefore not included with SR10. Concurrent Programming Support (CPS) is support for multitasking in a single process.

For more information regarding these products, refer to the *Network Computing System Reference* and the *Concurrent Programming Support Reference*.

1.12 GSR

GSR is being rereleased at SR10 with a new stroke font in `/sys/gsr/gsr.stroke_font`. The SR9.7 font will not work on SR10. Likewise, the SR10 font will not work on SR9.7.

Chapter 2: Software and Hardware Compatibility

This chapter describes the compatibility between SR10 software and previously released software. It lists the layered software product releases that can successfully run on SR10 and SR9.7 nodes in a network that includes workstations running both base releases. It also lists certain hardware limitations. *Making the Transition to SR10 Operating System Releases* provides more detailed compatibility information.

2.1 Hardware Configuration Considerations

The following Hardware Configuration restrictions and recommendations apply to Domain/OS SR10.

- SR10 Does not support DN100, DN400, DN420, or DN600 systems. These workstations will work under SR9.X releases, including SR9.7. Apollo will continue to support these machines under SR9.X releases at least until the next major release of the operating system (some time in 1989). SR10 will work on all other node types, including the DN460 and DN660 systems.
- SR10 requires a minimum memory size of two megabytes. We recommend that systems have a memory size of three or four megabytes for increased performance.
- Because of a change in the storage allocation algorithm described in Chapter 2, many SR10 applications will not run correctly on a system with a nearly full disk. Attempts to run with too little free disk space will result in disk full errors. As a general rule the operating system requires at least 500K bytes per user process on all systems except the Series 10000.
- Because of the requirement for free space and an increase in system software size, stand alone workstations with 50MB or less of disk space may require additional storage, depending on your computing needs.

Apollo offers many upgrades and add-on offerings to provide additional memory and storage space. Additionally, we offer "020" cpu and memory upgrades for 3XX and 5XX systems. We have significantly reduced the price of many of these upgrades. See your sales representative for information on these upgrade programs.

2.2 Object Module Compatibility

The object module format has been standardized on Common Object File Format (COFF) at SR10. SR9.2 object modules that execute successfully on systems running SR9.5 and that were not compiled for SR8 (i.e., not compiled for System III and 4.1BSD, or AUX) will run on SR10, subject to the same compatibility restrictions that affect SR9.5 object modules.

We no longer support AT&T System III or Berkeley 4.1BSD call semantics. Object modules that were stamped with the sys3 or BSD4.1 systype cannot run on SR10. This

restriction includes C programs which were compiled and bound by pre-SR10 tools under Aegis with no systype specification (which also received a systype stamp of sys3.) They must be ported to the new call semantics and rebuilt with a currently supported systype.

Object modules compiled under SR9.5 through SR9.7 will execute on systems that run SR10, subject to the compatibility restrictions discussed in detail in *Making the Transition to SR10 Operating System Releases*.

SR10, does not support compiling a module targeted to run on pre-SR9.5 systems (i.e., you can't produce or debug SR9.2 object modules on SR10). However, the SR10 FORTRAN, Pascal, and C language releases do include versions of the SR9.7 compilers that contain bug fixes; you can use these compilers on SR10 (or SR9.7) systems to generate code that will run on SR9.7 systems. Also, you can copy existing SR9.7 development tools onto an SR10 system and use them to develop applications targeted for systems with software releases from SR9.5 through SR9.7. The SR10 debugger can work with SR9.5 and later object modules.

2.3 Network Compatibility

Nodes running SR10 can communicate with and access files on nodes running all previous releases. Nodes running SR9.7 can communicate with and access files on nodes running all previous releases *and* nodes running SR10. There can be some loss of information in the transformation of Access Control Lists (ACLs) between SR10 and pre-SR10 nodes, but all changes that occur result in more restrictive access rights rather than less restrictive.

Nodes running pre-SR9.7 software releases cannot access files on SR10 nodes but they can communicate with SR10 nodes for informational purposes. Thus, sitting at an SR9.5 node, you cannot access files on an SR10 node; you can, however, run some network commands and such non-file operations as bldt and pst.

We recommend that if you must run pre-SR10 software on a network with SR10 systems, you should update all pre-SR10 systems to SR9.7.

2.4 Media Compatibility

Changes to on-disk structures at SR10 require that you reformat a node's disk with the invol utility before you install SR10. (We've made changes to invol, so you must use the SR10 version of this command.)

All disk volumes (floppy and removable hard disks) generated on systems running SR5 or later releases can be mounted on an SR10 system. No volume that has been initialized as an SR10 file system can be mounted on any pre-SR10 system. However, the SR10 invol program has an option that will generate pre-SR10 formatted volumes.

You cannot use a pre-SR10 node as a source node for SR10 installations, and you cannot run SR10 installation software on a pre-SR10 node.

2.5 Layered Product Versions Compatible with SR10

The following tables list layered product versions that can be used on SR9.7 and SR10 nodes in a network that includes both releases. The layered product version numbers reflect the latest versions that work with the corresponding operating system, *not* the version that was shipped with that release. Note that products that will be available after SR10 are shown in parentheses. Later versions of optional products are also compatible, unless otherwise stated in those products' release notes.

The table is *not* intended to help you determine which layered products are available for purchase. For specific information about purchasing layered products, see your local Sales Representative.

<i>Comparison of Layered Product Versions</i>			
Product	SR9.7 Version	SR10 Version	Comment
A-ADD-ETH	?	Obsolete	In base OS
ALIS	1.10L	Not available	Not available at SR10 FCS
Domain/Access	2.0	3.0	Older versions do not work on SR10
Domain/Ada	1.0/1.1	1.5	Post SR10 FCS release; an interim version is available
Domain Assembler	Not available	1.0	Newly supported at SR10
Domain/C (CC)	4.89	(SR10)	SR10 release includes a bug fix version of the 9.7 compiler that runs on SR10 nodes and generates code for pre-SR10 systems
Domain C++	1.2	1.2	Post-SR10 FCS
Domain/CORE	9.5	9.5	
Domain/CommonLISP	2.10	2.2	Post-SR10 release; an interim version is available
D3M	5.5	6.0	Version 5.5 must run on 9.7 nodes in a mixed network of SR9.7 and SR10 systems
Domain/Dialogue	2.0	2.01	Older versions do not work on SR10

<i>Comparison of Layered Product Versions</i>			
Product	SR9.7 Version	SR10 Version	Comment
Domain/Delphi	1.0	1.0	SR10 books are unavailable at Beta; see "Restrictions" section of Chapter 3 for loader restriction
Domain FORTRAN	9.66	10.00	SR10 release includes a bug fix version of the 9.7 compiler that runs on SR10 nodes and generates code for pre-SR10 systems
Domain/IX	9.5	Obsolete	In base OS
Domain/Laser 26	2.2	Obsolete	In base OS
Domain/LU6.2	1.0	1.1	Older versions do not work on SR10
Domain/PAK	2.1	3.0	Older versions do not work on SR10
Domian Pascal	7.38	(SR10)	SR10 release includes a bug fix version of the 9.7 compiler that runs on SR10 nodes and generates code for pre-SR10 systems
Domain/PCC	2.1	3.0	Version 2.10 runs at SR10, except for diagnostic; Version 3.0 is post-SR10 FCS
Domain/PCC_Remote	1.1	Obsolete	Included in Domain/PCC
Domain/PCE	1.0	3.0	Version 1.0 runs at SR10, except for diagnostic; Version 3.0 is post-SR10 FCS
Domain/PCI-1	3.0	4.0	Post-SR10 FCS
Domain/PCI-ETHERNET	?	4.0	Post-SR10 FCS
Domain/PCI-RING	1.0	4.0	Post-SR10 FCS
Domain/SNA 3270	2.0	2.1	Older versions do not work on SR10
Domain/SNA 3770	1.1	1.2	Older versions do not work on SR10
Domain X.25	2.3	2.3	

<i>Comparison of Layered Product Versions</i>			
Product	SR9.7 Version	SR10 Version	Comment
Domain 4014	2.0	2.1	Older versions do not work on SR10; At beta, the version 2.1 does not have release notes
Domain 5080	2.0/3.0	3.0	DOWNCASE must be set to true at SR10
DPSS/Mail	1.1	2.0	Older versions do not work on SR10
DSEE	3.2	3.3	Version 3.2 must run on 9.7 nodes in a mixed network of SR9.7 and SR10 systems
Documenter's Workbench	Not Available	2.0	Newly supported at SR10
ECMB	7.2	7.2	
EtherBridge	1.4	Obsolete	In base OS
EtherController-AT	2.0	tbd	
2DGMR	2.0	2.1	Version 2.0 requires 7-bit fonts; Version 2.0 supports 8-bit fonts; an interim version of 2.1 is available
3DGMR	2.5 or 2.6	2.6	
GKS	1.0	1.0	
GPIO	9.6.1	10.0	Older versions do not work on SR10
GSR	2.0	2.1	Only change between 2.0 and 2.1 is in stroke font structure; Version 2.1 is required for using stroke fonts on SR10
Interleaf	2.5.13	unavailable at beta	An interim version is available
NFS	1.0	2.0	Older versions do not work on SR10
Network Interface Definition Language	1.0 (plus patch)	1.0 (plus patch)	See "Installation of Pre-SR10 Optional Products", below, for SR10 installation requirements

<i>Comparison of Layered Product Versions</i>			
Product	SR9.7 Version	SR10 Version	Comment
Network License Server	1.0	1.0	
Open Dialogue	1.0	1.1	Post-SR10 FCS Release
PHIGS	1.0	1.0	
PostScript	1.0	unavailable at Beta	
PS_PRINT	?	Obsolete	In base OS
RJE	2.1	2.2	Unavailable at Beta
SPE Board Driver	1.2	2.0	Older versions do not work on SR10
TCP/BSD4.2	3.0	Obsolete	In base OS
TCP/IP	3.0	Obsolete	In base OS
TEK 4692	1.01	Obsolete	In base OS
TML++	1.0	1.0	
TranScript	Not available	2.0	Newly supported at SR10
Versatec	2.1	2.1	See "Installation of Pre-SR10 Optional Products", below, for SR10 installation requirements

2.6 SR9.7 Compatibility Items

The SR10 release includes a directory, `/sr9.7_compatibility`, that contains a variety of files and tools required for compatibility between SR10 and SR9.7 systems. This directory is divided into two subdirectories, as follows:

sr9.7_executables This directory contains programs which you should run on SR9.7 systems that are connected to networks that have SR10 systems. In most cases, they should replace software that was shipped at SR9.7. These facilities include a version of Honey-DanBer UUCP that will work with SR10 HoneyDanBer UUCP, software for printing in a mixed SR10/SR9.7 environment, NCS support, and a tool for converting between SR9.7 and SR10 registries. It also provides libraries that allow you to run SR10 compilers on SR9.7 nodes.

compat_with_sr9.7 This directory contains sr9.7 software that enables an SR10 system to generate "obj" format object modules that will run on pre-SR10 systems. These include versions of the librarian and various UNIX program development tools. 9.7-compatible compilers can also be put in this directory when you install SR10 C, Pascal, and FORTRAN.

This directory also contains tools that run on SR10 nodes to provide compatibility with objects generated on SR9.7 systems. These tools include the `cvtname` command that converts between SR9.7 and SR10 case representation. (At beta, `cvtname` is in the `/com` directory).

Note that the SR10 release *does not* include SR9.7 insert files or fonts. If you wish to use these on an SR10 system, you must obtain the required tools from an existing SR9.7 system.

Making the Transition to SR10 Operating System Releases describes all SR9.7 compatibility items in detail and discusses the considerations for generating pre-SR10 executable software on SR10 systems.

2.7 Optional Products Released in Multiple Versions

The Domain Software Engineering Environment (DSEE) and D3M optional products are being released at the same time as SR10 in two versions, one that runs on SR10 systems and one that runs on SR9.7 systems. In these cases, the versions of the optional product that were available before the release of SR10 will not work properly in an environment that also includes SR10 nodes, and you must now install the new SR9.7-compatible version on all SR9.7 nodes.

The installation software will only install the correct version of these products. That is, it will produce an error message if you attempt to install the SR10-compatible version on an SR9.7 system or if you attempt to install the SR9.7-compatible version on an SR10 system. For more details, see *Installing Software with Apollo's Release and Installation Tools*.

2.8 Installation of Pre-SR10 Optional Products

The `/install/sr9.7_compat` directory contains tools for installing SR9.7-based optional software products that are compatible with SR10 and are not being re-released with SR10. If the optional product uses the standard installation script described by Chapter 5 of *Installing Domain Software*, you should use the `sr9.7_compat` installation tools in place of the tools that were supplied with the optional product software when you install the software on an SR10 node.

If you are installing any pre-SR10 layered products that are compatible with SR10 and are not being rereleased at SR10 onto an SR10 node, and you are *not* using the tools from `/install/sr9.7_compat`, you must set the `INPROCESS` and `DOWNCASE` environment variables before running the installation script.

Some layered products that were shipped before SR10, including the Network Interface Definition Language (NIDL), require the existence of a `/com` directory before they can be installed. At SR10, the `/com` directory is no longer automatically created when you install only UNIX environments. Therefore, a system that does not include the Aegis environment might not have a `/com` directory. In this case, you must create the `/com` directory before you install the optional product.

For the Versatec product, SFW-VERS, version 2.1, you must modify the build_ddf.sh script if you install the software in an SR10 system. Change the procedure names VERS_\$INT, VERS_\$INIT and VERS_\$CLEANUP to vers_\$int, vers_\$init, and vers_\$cleanup.

2.9 2DGMR Unbundled at SR10

2DGMR was made a layered product when we released Version 2.0, prior to SR10. A revised version of this product (Version 2.1) runs on SR10. 2DGMR Version 1.0 was included in previous releases of software, but is not provided or supported on SR10. For more information, contact your Customer Services representative.

Chapter 3: Fixed Bugs, Limitations, and Known Bugs

This chapter documents bugs that we have fixed since SR9.7 as well as User Change Requests (UCRs) and Apollo Problem Reports (APRs) we have addressed. The chapter also documents all known limitations and bugs in SR10 standard software.

We also ship, in hard copy only, a final bugs list. This list includes any bugs discovered, or fixed, after this document was completed. Consult that document, also shipped with the media kit, to ensure that you know of the final disposition of all bugs that are known to exist or that are fixed in this release.

3.1 Bugs Fixed Since SR9.7

The following subsections document bugs fixed since SR9.7 in the Domain/OS operating system (kernel and libraries), the Display Manager, and various commands and utilities.

NOTE: *At beta we have documented only a few of the many fixed bugs. this list will be complete by FCS.*

3.1.1 Fork

The UNIX fork () call now works after a GPIO device has been acquired; however, the GPIO device is not accessible to the the child process. In order for a device to be accessible to child processes after forking, the device must be a GPIO global device.

3.1.2 The Bind Utility

We have fixed the following bind bugs since SR9.7:

- Prior to SR10, the bind utility could not resolve UNIX symbolic links containing uppercase letters. We've fixed this for SR10. The bind utility now resolves UNIX symbolic links in a case sensitive manner.
- Prior to SR10, if a C program defined a malloc function (but no calloc function), then calloc would call the system malloc (stored in clib) instead of the user-defined malloc. We've fixed this problem at SR10. Now, calloc will call the user-defined malloc instead of the system-defined malloc.

3.1.3 Domain/OS Calls and Insert Files

The typedef of pgm_\$argv in /sys/ins/pgm.ins.c was corrected from an array of pointers-to-int to an array of pgm_\$argv_ptr, which are pointers to pgm_\$arg. As a result, all C sources that using pgm_\$argv will need to be fixed to not coerce args to (int *).

3.1.4 COFF UID

A SR9.7 install bug caused the type uid for COFF to be wrong. All compilers that generate COFF output use the correct UID, 322.0, and all SR10 software uses the correct UID. As a result, if you do a `/com/ld -type` command on a directory containing COFF objects from an SR9.7 system, the command may report a type of `type?`, instead of `coff`. This is the only side effect of the SR9.7 bug, and you can correct this problem by using the following command to install the correct type UID on the SR9.7 node: `crty coff -u 322.0 -r`.

3.2 Berkeley Distribution Bug Fixed

We have fixed the following bug in the Berkeley Software Distribution:

4.2BSD and 4.3BSD `/usr/include/malloc.h` declare the `utmp` struct `ut_time` field as a long. There is no guarantee on any UNIX system that a time can fit into a long. We therefore declare the field as a `time_t`.

3.3 Known Restrictions at SR10

The following subsections describe restrictions and limitations at SR10.

3.3.1 Booting Diskless from DN5x0 Nodes

You cannot use a DN550, DN560, DN 570, or DN580 (non-turbo) node with a Revision 16 ring board as the disked partner to a diskless SR10 node. You must upgrade your ring board to Revision 17. Earlier revision ring boards do not have this problem.

3.3.2 EX DOMAIN_OS on DN5x0 Turbo Systems

You cannot use the `MD EX DOMAIN_OS` command on DN5x0 Turbo systems with the FPX floating point accelerator; instead you must use the `EX AEGISFR` command. If you use `EX DOMAIN_OS`, the node proms will not load the FPX microcode.

3.3.3 Installation from Media

The procedures for installing from distribution media always leave a `/com` directory on the node. If you are only installing UNIX environments, you can delete the `/com` directory tree after you have installed all software, including all layered products, from the distribution media.

3.3.4 Aegis Shell

The following restrictions apply to the Aegis shell.

- Any program or shell script that the Aegis shell invokes must give execute permission to the current user. This should not effect most SR9.7 scripts that were created

by the DM because the DM set the execute bit prior to SR10. However at SR10 the DM does not set the execute bit. Therefore, you must use the `edacl`, `chacl`, or `chmod` command to set the bit on any shell script created by the DM at SR10.

- Error input redirection from shell scripts and obj format files will continue to work. However COFF files do not have an error input (`errin`) stream and an attempt to directly redirect their error input (for example, if run by a shell script whose error is redirected) will result in an error message from the shell.
- Stdout and `errout` may be merged into a single file via (`> foo >? foo`). However since the buffering from a particular command may be different in one stream than in the other there is no guarantee that the messages will be placed in the file in the correct order. (Traditionally `errout` is not buffered and `stdout` often is when directed to a file.)
- Commands immediately after an arithmetic expression on the same line, such as

```
a := 1; args foo
```

will not work if the command has anything in it that could be interpreted as an arithmetic operator.
- The shell does not fail gracefully if the number of legal pipes in a single command line (approximately 120) is exceeded.
- In some instances the 'select' builtin will treat arguments as numbers even though they have been quoted.
- There is currently no clean way to loop through an arbitrary number of arguments on the command line. However the following script will work for everything except null arguments:

```
#!/com/sh
eon
#
# Note that this will exit the loop early if there are any 0 length arguments.
#
for i := 1 to 99999
  arg := ^"set -c args ^i"
  if eqs ^arg "" then exit endif

  # Do your stuff here.
  args ^arg
endfor

#Here's the explanation, from the inside out.
#
#   args ^i      -> ^1
#   set -c args ^1 -> <argument#1>
#
# set -c is like saying 'sh -c' except that it's faster and takes place
# at the current level. This would have worked with "sh -c" as well. We
# are taking advantage of the fact that command line arguments after -c are
# reparsed, so we get one extra level of parsing.
#
# Note: If you KNOW that none of your arguments have any spaces in them,
# you can make this process much simpler by just saying:
#
#   for arg in ^*
#     ...
#   endfor
```

3.3.5 C Shell

You cannot set use the C shell limit command to set the stack size.

The maximum token size in the C shell is 1020 characters; therefore you cannot use a file or directory name that exceeds this limit.

3.3.6 BSD Bourne Shell

The BSD Bourne shell does not give an error if you try to do multiple while redirections.

3.3.7 Aegis Commands in UNIX pipes

An Aegis command may fail if it is used in a pipeline in a UNIX shell. For example, if you use pipe the output of /bin/cat into /com/fpatb while in a UNIX shell, fpatb will not search the entire contents of the file.

3.3.8 Aegis Commands

The following restrictions exist in Aegis Commands. Also check the "Common Commands" restriction list for commands that are also available through the /usr/apollo/bin directory.

- | | |
|-----------------|---|
| ld of SR10 /dev | If you issue a /com/ld command at an SR9.7 node to list one or more entries in an SR10 node's /dev directory and specify an option that requests system type information (the -a or -st options), the output may include unexpected characters, including non-printing and control characters. |
| netstat | The netstat -config command does not report the presence of floppy disk drives on DN3000 and DN4000 systems because the floppy controller is part of the winchester controller. |
| xdmc | If you crp on a remote node and try to execute a DM command on the remote node by using xdmc in the crp window, the DM command executes on the local node. To execute a DM command on the remote node, enter:

<code>xdmc command > /dev/display</code>

in the crp window. It will have the side effect of popping a window on the remote node, but it will work. |

3.3.9 Common commands

The following restrictions apply to commands that are in the /usr/apollo/bin directory; any restriction also applies to the version of the command that is in the /com directory, if there is one.

- | | |
|-----|---|
| crp | Crp from pre-SR9.7 nodes to SR10 nodes does not work because pre-SR9.7 software does not understand the SR10 directory format; crp <i>does</i> work from SR9.7 nodes to SR10 nodes. You cannot use crp -me from an SR10 node to a pre-SR9.7 node if your working directory or naming directory is an SR10 object. You can use crp -me to a pre-SR9.7 node if you first change your naming and working directories to pre-SR10 directories.

If an SR10 node's 'node_data has tight protections, users from SR9.7 nodes may not be able to crp onto the node because crp creates a mailbox directly in the 'node_data directory. |
| tb | The tb -args command is no longer supported due to limitations of the COFF format. Use the debugger to get the values of passed arguments. |

3.3.10 Commands and Files in /etc

The following restrictions apply to common commands and files in the /etc directory. See the "BSD Commands" and "SysV Commands" sections for any restrictions to /etc commands that are environment-specific.

- show_lc** The **show_lc** command does not work on obj format files.
- nodestat** The **nodestat -c** command does not report the presence of floppy disk drives on DN3000 and DN4000 systems because the floppy controller is part of the winchester controller.

3.3.11 BSD Commands

The following restrictions exist in BSD commands. Note that some restrictions that are listed for BSD commands may also apply to the corresponding SysV command.

- finger** The **finger** command on a large network with many users can take a long time because it reads the entire registry.
- hostid** The **hostid** command does not work.
- ls** Because of the changes in ACLS and protections at SR10, the **ls** command always reports that files on SR9.7 systems have extended ACLs.
- makewhatis** The **makewhatis** command does not work because we do not supply unformatted man pages.

3.3.12 SysV Commands

The following restrictions exist in SysV commands. Note that some restrictions that are listed for SysV commands may also apply to the corresponding BSD command.

- hostid** The **hostid** command does not work.
- ls** Because of the changes in ACLS and protections at SR10, the **ls** command always reports that files on SR9.7 systems have extended ACLs.
- nice** The mapping of nice values into the underlying Domain/OS priority mechanisms is not one-to-one. While this doesn't affect processes' relative priorities, it may result in variations between the values as set by **nice** and those reported by **ps**.
- rm** **rm -r** does not always remove a directory in a foreign (non-Apollo) file system; it fails if the current working directory and the directory to be removed have the same Date-time modified stamp.

3.3.13 Access to New Object Types

Pre-SR10 nodes cannot recognize object types that are new at SR10. As a result, it is impossible for SR9.7 nodes to access or manipulate SR10 character or block special

devices or to execute COFF object modules. The new types include:

- cmpexe
- coff (uid differs from SR9.7)
- console
- debug_* (eight types)
- dev_tty
- display
- group
- log
- nuls
- org
- osio
- passwd
- spe
- spmio
- spmio_9x
- uds

3.3.14 'node_data

At beta, the 'node_data directory, and all its subdirectories *except* tmp and usrtmp must have their protections set so that all protections are inherited in an Aegis manner, that is, from the directories.

3.3.15 inprocess

The INPROCESS environment variable is no longer supported in UNIX shells.

3.3.16 SR9.x File Ownership and Setuid

SR10 does not recognize (and cannot change) the UNIX owner and group on pre-SR10 files. Therefore you cannot run setuid programs on an SR9.x volume from an SR10 system. Also, if you use `cpf -sacl` or `cpt -sacl` to copy any files from a pre-SR10 node to an SR10 node, the owner of the copy will be set according to the SR10 inheritance rules, and will not be the owner perceived on the SR9.7 node.

3.3.17 8-Bit Font Support

The following restrictions exist at beta in support for 8-bit fonts. See the next section for additional restrictions on 8-bit characters.

- The fonts listed in the "Font Support" section of Chapter 1 are supplied only in 8-bit format, and cannot be used on SR9.7 systems.
- Domain/Dialogue supports the new 8-bit font format. However, it does not support 8-bit characters. Therefore, it is currently still limited to regular ASCII character set.

You can use the `/sys/dm/cvt_font` utility to convert 7-bit fonts to 8-bit format that will work with the DM and Domain/Dialogue. Therefore, you can save any existing 7-bit fonts, or create new fonts using the SR9.7 version of `edfont`. You can then convert these fonts to 8-bit format and have fonts with identical appearances in both formats. In this case you may have to resolve conflicting font names.

3.3.18 8-Bit Character Support

The following restrictions currently exist in support for 8-bit characters. In the following descriptions, the term "8-bit character" refers to a character where the eighth bit is set.

- If you use an SR9.7 node to edit a file that contains 8-bit characters, the top bit of the characters in the lines that get edited are lost, and therefore the 8-bit characters are garbled. Similarly, if you copy a file whose name contains 8-bit characters to a 9.7 node, you must change the name to one that only contains ASCII characters.
- Pre-SR10 nodes cannot correctly read 8-bit characters.
- User login names, node names, group names, and organization names can only include lowercase ASCII letters (a-z), numbers (0-9), and underscores (_).
- Some SysV C Shell utilities will output 8-bit characters, while others do not. The shell terminates completely if you use a European character as input.
- The BSD shells do not accept 8-bit characters on input or output.
- Although most SysV shells and the Aegis shell support 8-bit characters on output, sorted output may not always appear in the expected order. Both SysV and Aegis sort all characters in numeric code order, but the two environments interpret the 8-bit characters differently. SysV interprets any 8-bit character as being negative, so Latin-1 characters sort *before* regular ASCII. Aegis considers the 8-bit characters to be positive; and, since their code values are higher than those for ASCII, Aegis orders Latin-1 characters *after* regular ASCII.
- DSEE does not support filenames that include 8-bit characters.
- 8-bit support is included in the SR10 print server. However, only Postscript printers support the full 8-bit character set. Some printers, such as Imagen printers, support only a subset, and some printers may require modifications.

For example, if you have a printer that uses a daisy wheel or other mechanical impact device, you might have to replace the current font with one that includes the Latin-1 characters. Similarly, you might have to load a language-specific PROM for a dot-matrix printer in order to generate the same characters on paper that you see on screen. If your printer font does not include a Latin-1 character that is in your file, the system prints a blank.

- At Beta, the following Latin-1 characters do not print (you get a blank) when using a PostScript printer.

copyright sign
soft hyphen
registered trademark
macron
ring above, degree sign
plus/minus
superscript two
superscript three
superscript one
fraction one-quarter
fraction one-half
fraction three-quarters
capital icelandic ETH
capital Y with acute accent
capital icelandic THORN
lowercase icelandic ETH
lowercase Y with acute accent

- At Beta, several Latin-1 characters do not print correctly when using a PostScript printer. The following table lists the Latin-1 character and the character that actually is printed

Latin-1 Character	Printed Character
—	—
not sign	~
micro sign	u
multiplication sign	*
division sign	/
lowercase icelandic THORN	

- Because the Domain C, Domain Pascal, and Domain FORTRAN compilers adhere to existing standards, none allow 8-bit characters in identifier names. All flag as errors any attempts to use European characters in identifier names. However, you can include 8-bit characters in strings and comments.

3.3.19 Blasting Processes

Blasting a process (by using the Aegis `sigp -b` or `DM dq -bFR` command, or by responding yes in response to a `logout "do you want to blast" query`) leaves many system resources in an indeterminate and unreliable state. If you must blast a process, you should then shut down to the mnemonic debugger level and reboot, or at least exit to the boot shell, as soon as possible after blasting. This will restore the system resource state.

3.3.20 getty On an Aegis-Only System

Although the `getty` utility is in the common `/etc`, you cannot use it to control SIO lines

on a node where only Aegis is installed; you must use the Aegis siologin utility.

3.3.21 getty and uucp dialout

SR10 does not support uucp dialout on a loine owned by getty.

3.3.22 Cartridge Tape Drive

You must issue a command to rewind a cartridge tape, such as the `rbak -rewind command`, before you can access the cartridge tape.

3.3.23 Printing

Any applications which have bound a pre-SR10 `/lib/prflib` into the module will not run properly on an SR10 node. These applications should be rebound with the `-inlib /lib/prflib` switch and should not have the `prflib` bound in.

Ordinarily, in a mixed environment of SR10 and pre-SR10 systems the `/sys/print` directory must be located on an SR9.7 node; otherwise, users on SR9.7 nodes cannot queue jobs. However, the `/sr9.7_compatibility/sr9.7_executables` directory contains versions of `/com/prf` and `/lib/prflib` that overcome this restriction. If you install these items on the SR9.7 nodes in place of the original `prf` and `prflib`, you can then put the `/sys/print` directory on an SR10 node and queue successfully from the SR9.7 nodes. In all cases, the print manager and print server can run on SR10 or SR9.7 nodes.

If you find it necessary or desirable to put the spool directory on an SR10 volume, then you must copy `/lib/prflib` and `/com/prf` from the `/sr9.7_compatibility/sr9.7_executables` directory to the SR9.7 nodes on your network.

At Beta, Headers and footers are printed only on PostScript printers.

3.3.24 Links

Domain/OS soft links are not objects. As a result, the links themselves do not have protections (although the objects to which the links point, of course, do). Similarly, they do not have date/time stamps. The UNIX `stat` call and `ls/lR` command report the date-time modified of the directory that contains the link as the links modification time.

3.3.25 COFF Objects and Type Managers

Objects compiled by using the standard SR10 tools are in COFF format and will not execute under previous releases. If you write a type manager and want it to be usable from SR9.x systems, do not use the COFF compiler and tools. The managers must be in "obj" object format to interoperate with SR9.x nodes and allow you to share files of that type between SR10 and SR9.x nodes. However, if you do not want the type manager to be usable with SR9.7 nodes, you should use the COFF tools. Note that you should specify position independent code for COFF managers.

Normally, if you create a new type and manager on any node and create files with that type, other nodes can see that object even if you have not installed the type manager on them. If the other node lacks the new type manager, it tries to use the type manager from your node (where the file was found). However, if you use COFF for that new type manager and the other node is running a pre-SR10 release, that node cannot load your COFF type manager and cannot read your new files.

3.3.26 COFF objects and the DM

If you use an SR10 compiler under any of the following conditions, the resulting COFF file will appear to be an ascii-readable file and the DM will attempt to open a pad on it if you try to read the file.

- Build on a SR9.7 node and send the output to an SR10 node
- Build on a SR9.7 node and send the output to an SR9.7 node
- Build on a SR10 node and send the output to a SR9.7 node

3.3.27 Creating .bak Files from SR9.7 Nodes

It is possible for a program on an SR9.7 node to use `ms_$mk_permanent` (or any other call that generates .bak files) and cause an SR10 node to create a backup file. If the original file name exceeds 28 characters, the call would normally create a .bak name that exceeds 32 characters. However, to ensure that the SR9.7 node that is creating the file can also access it, the software truncates the original file name to ensure that the backup file name, including .bak, does not exceed 32 characters. The filename is *not* truncated if a program on an SR10 node makes the call.

3.3.28 rbak of SR10 nodes by SR9.7

The SR9.7 version of `rbak` cannot restore SR10 nodes. The `/sr9.7_compatibility/sr9.7_executables` directory contains a version of `rbak` that runs on SR9.7 systems and can restore SR10 nodes.

3.3.29 The `stream_$redefine` Call

You can execute a `stream_$redefine` call to convert an unstructured format file (such as COFF or unstruct type files) to a file with a streams header (such as `uasc` or `obj`) before writing to the file. However, you cannot redefine an unstructured file to a streams file after writing to it. This limitation is the result of the differences in streams files and unstructured files. Streams-format files begin with a header, while unstructured files do not. Once the unstructured file is written to, the `redefine` operation cannot create the required streams header.

Note that the `stream_$` interface, including the `stream_$redefine` call is superseded by the `ios_$` interface.

3.3.30 pfm_\$cleanup Call

At SR10 `pfm_$cleanup` does not automatically release the cleanup handler at end of normal execution. If you issue a `pfm_$cleanup` call you must now always explicitly issue a `pfm_$rls_cleanup` call at the end of normal execution.

3.3.31 UNIX nice Call

The mapping of nice values into the underlying Domain/OS priority mechanisms is not one-to-one. While this doesn't affect processes' relative priorities, it may result in variations between the values as set by nice and those reported by ps.

3.3.32 SysV open Call

The sysV open call `O_SYNC` option has no effect.

3.3.33 BSD ttyslot Call

The BSD `ttyslot` call returns the slot number of a tty in the `/etc/utmpfile`, not the `/etc/ttys` file.

3.3.34 Registry Creation

You can now create a registry for the first time by using the new `create_rgy` command. This command is shipped only in the `/install` directory on the boot media, and is *not* copied across the network by the normal installation procedure. If you use this command, you will automatically get passwords on all the canned accounts. The password is `-apollo-`. The account for user has the password `-apollo-` also, but if you login using the full text `user.none.none`, you do not need a password.

If you want to use your existing 9.7 registry, you must use `/sr9.7_compatibility/sr9.7_executables/com/cvtrg` to create an SR10 registry from your SR9.7 registry. This program runs on SR9.7 nodes only. At beta you may want to make a copy of your SR9.7 registry first and then run `cvtrgy` against that to create your SR10 registry.

3.3.35 TCP/IP

SR10 TCP/IP locates its administrative files in the `/etc` directory, while the layered SFW-TCP product that ran on Aegis-only systems in earlier releases located the administrative files in the `/sys/tcp` directory. Also, we recommend that each network have only *one* administrative node. As a result, there can be administrative problems if the TCP/IP administrative node on a network with both SR10 and pre-SR10 TCP/IP users is an Aegis-only SR9.7 node, or an SR10 node.

We recommend that, if you have both SR10 TCP/IP users and pre-SR10 Aegis-only TCP/IP users, your TCP/IP administrative node should run SR9.7 Domain/IX. (You should install Domain/IX if it currently runs only Aegis.) This way, the node can then serve both SR9.7 Aegis TCP/IP users and SR10 TCP/IP users since it will have both

/etc and /sys/tcp directories.

3.3.36 UUCP

Domain/IX (pre-SR10) UUCP can not share spool queues with the SR10 UUCP. The /sr9.7_compatibility/sr9.7_executables directory contains versions of the HoneyDan-Ber UUCP that you should install and run on SR9.7 systems in a mixed network of SR10 and pre-SR10 systems. These versions can share pool queues with SR10 UUCP.

3.3.37 SAX on a 30MB disk

/systest/sax requires approximately 11 megabytes of free disk space. As a result, it will run out of disk space on a 30MB disk or on a disk that is nearly full.

3.3.38 GPR

On DN570 and DN570T systems the maximum size of a hidden bitmap is restricted to 224 X 214 and not the documented 224 X 224.

3.3.39 SysV open call

SR10 does not support the System V Release 3 open(2) call's O_SYNC mode.

3.3.40 Layered Products that Do Not Work On SR10

The currently released versions of layered products are known not to work on SR10 systems at Beta2.

- ADA (An interim version is available on special request)
- Common Lisp (An interim version is available on special request)
- Modula 2
- RJE
- PostScript
- DPCI-1 3.0
- DPCI-Ring 1.0
- DPCC 2.1
- DPCC-Remote 1.1
- DPCE 1.0
- 2DGMR 2.0 (An interim version is available on special request)
- Interleaf (An interim version is available on special request)
- Alis

3.3.41 C Compilers

You cannot use the SR10 version of `/bin/cc` as a front end to the SR9.7 version of `/fb/com/cc`; you must use the version of `/bin/cc` that is in the `/sr9.7_compatibility` directory.

3.3.42 GPIO

If a GPIO library is in SR9.2 object module format, it cannot be loaded on an SR10 system.

3.3.43 Domain/Delphi

The Domain/Delphi loader will not work unless you first install the SR10 D3M runtime unit, not the one shipped with Domain/Delphi. You only need the Domain/Delphi loader to load new books into a Domain/Delphi database. SR10 documentation is not available through Domain/Delphi at beta.

3.3.44 Open Systems Toolkit

Managers must be compiled or bound as position independent code.

At SR10 there are additional trait constant files in the `/sys/traits` directory. These files have a suffix of `.2`, for example `ios_dir_trait.2`, and include currently-unreleased additional operations that are used by certain software products at SR10. You should continue to bind type managers to the trait constant files without the suffixes, for example `ios_dir_trait`.

3.3.45 DSEE Tutorial

The DSEE Tutorial's Libraries Discussion Topic will omit figures or redraw the figures at the wrong time if you replay a segment.

3.4 SR10-Related SR9.7 Restrictions

SR9.7 systems cannot execute shell scripts that are in unstruct format files (as opposed to usac files).

3.5 Known Bugs at SR10.0

The following subsections describe known bugs at SR10.

3.5.1 Base Software Installation

The following installation bugs are known at beta:

Closed ACLs and `/etc` If you install with closed ACLs, several files in `/etc` currently get protection mode `777`.

<code>/etc/syslog.conf</code>	The installation procedure does not create a default <code>/etc/syslog.conf</code> file. The man page provides help on creating it.
<code>/etc/inetd.conf</code>	The template <code>/etc/inetd.conf/fR</code> file does not include an entry for the BSD <code>ntalkd</code> daemon.
Media Installation	You cannot install from the distribution media onto a DSP server node; the node must be a workstation with a full display.
DTEK Release Notes	Release notes are not available for the layered DTEK 4014 product at beta.
<code>install++</code>	The <code>install++</code> tool runs the installation procedure even if you abort from the configuration procedure.

You should also check Chapter 4 of these release notes for any additional installation information procedures that are required as a result of beta bugs.

3.5.2 NFS Installation

The NFS installation procedure prints the following message:

"check the file `/etc/rc.tcp` for instructions on how to run TCP/IP"

The file name is incorrect; it should be `/etc/rc.local`.

3.5.3 C Compiler Installation

The C compiler always creates a `/sr9.7_compatibility` directory, even if your configuration that does not include this tree.

3.5.4 No Shell After Invol

If you install on an involed node, boot it up, and then log in you may get no shell the first time if your shell is a link to another node. This is because your node hasn't found the naming server yet and thus doesn't know any node names.

3.5.5 DM Display of Tabs

The DM currently converts a tab character into 8 times the width of a space character; as a result the spacing of text with tabs is incorrect. The DM will use a spacing of 8 times the width of the character "n" by FCS.

3.5.6 Init

If the DM fails to load, `init` continually attempts to load it.

On a node where the SPM runs in place of the DM, i.e., a server node without a graphic display, the SPM quit command should shut down the SPM, leaving the system

in the boot shell. Instead, the init process automatically restarts the SPM.

3.5.7 DM Boot over SIO line

You cannot boot the DM over an SIO line.

3.5.8 calendar and invol

The `calendar` and `invol` standalone utilities (`saus`) truncate messages that are longer than 80 characters.

3.5.9 UNIX Default quit Character

When the standard keyboard or user-defined key definitions are not in effect, for example in the vt100 emulator or over an SIO line, the default quit character is CTRL-? instead of CTRL.

3.5.10 Aegis Shell

The following bugs exist in the Aegis shell.

- Aegis wildcards do not support `./<wildcard>` at Beta2.
- A leading period (.) in a pathname that includes wildcards is interpreted as meaning the current working directory, even if it is not followed by a slash separator. Thus, `.csh?*` is interpreted as meaning any object in the working directory with a name that starts with `csh`.

3.5.11 Aegis Commands

The following commands in the `/com` directory currently have bugs. You should also check in the "Common Commands" and "/etc Commands and Files" bug listings below for bugs additional commands.

<code>chuvol</code>	The <code>chuvol</code> command is not installed at beta. (This command is a link to <code>/etc/chuvol</code> , which is not installed.)
<code>mtvol</code>	The <code>mtvol</code> command may report the following error even though it mounts the volume correctly: "unable to add entry to mount table"
<code>netstat</code>	The <code>netstat</code> command returns without an error message if it does not find the node for which information was requested.
<code>tctl</code>	The <code>tctl</code> command does not parse the command line arguments correctly.
<code>vsize</code>	The <code>vsize</code> command is not installed at beta. (This command is a link to <code>/usr/apllo/bin/vsize</code> , which is not installed.)

3.5.12 Common Commands

The following bugs exist in commands that are in the `/usr/apollo/bin` directory; any bug also applies to the version of the command that is in the `/com` directory, if there is one.

- chacl** You cannot use the `chacl` command to `setuid`.
The `chacl` functionality for setting the owner, group and organization does not work correctly.
- crp** You cannot login when using `crp` from the single user shell, that is when in a shell that is invoked from the bootshell. This is true both for the `crp` command with no options or `crp -login user_name`. You can use `crp -me`.
If you use the UNIX `man` command when in a `crp` window, the working is set to the `man` directory.
- dbacl** The `dbacl` command allows you to add users but does not update its display to include the new information. It also has problems setting the local access only bit, with initial file acls, and with directory acls.
- emt** You cannot use `emt` if you do not have a `/com` directory.
- systype** `systype` currently returns a "not an object file" error message if you do not have read permission on the file.
- vsize** The `vsize` command is not installed at beta.
- wbak** You can get the following error when running `wbak`:
 ?(vfmt) Error writing to stream 3, status 1080001
This error is caused by `wbak` attempting to write to both standard stream 3 and the error stream.
An attempt to back up an object where the full path exceeds 1024 characters results in a guard fault. Note that this bug applies to the resolved path, not the absolute pathname; absolute pathnames cannot exceed 1023 characters.

3.5.13 /etc Commands and Files

The following bugs exist in commands and files that are located in `/etc`:

- chuvol** The `/etc/chuvol` command is not installed at beta.
- getty** `/etc/getty` displays a banner saying `bsd` under all environments, including SysV. The banner may be garbled when you use a dialup line. `Getty` is still working in both these cases.
- nodestat** The `/etc/nodestat` command returns without an error message if it does not find the node for which information was requested.

writed The /etc/writed daemon is not installed at beta.

3.5.14 SysV Commands and Utilities

The following bugs exist in SysV commands. Note that some bugs that are listed for BSD commands may also apply to the corresponding SysV command.

find find -local does not currently work
vi Vi does not work over dialup lines.

3.5.15 BSD Commands and Utilities

The following bugs exist in BSD commands. Note that some bugs that have been reported for SysV commands may also apply to the corresponding BSD command.

df The df command cannot see devices that are mounted on SR9.7 systems because df now looks for devices in /etc/mnttab; at SR9.7, only NFS supported this file.
nm The nm command does not work with obj format files. The BSD nm -og command lists the archive name, instead of the names of the objects that make up the archive, with the symbols and values.
man -f and apropos The man -f and apropos commands do not work. To make them work, move /usr/lib/whatis to /usr/man/whatis.
vi The BSD implementation of vi has the known Berkeley distribution bug that causes macros to destroy file text.

3.5.16 telnet

The telnet delete key does not work.

3.5.17 /usr/adm/lastlog

Although the beta software installs a /usr/adm/lastlog file, we do not support it.

3.5.18 Keys Set by kbm

If you use the kbm command to set a key specification such as the compose key, third party software cannot reset the meaning of that key.

3.5.19 Lost Keyboard Events

D3000, DN4000 and DN5xx-T can lose typed characters, for example if you type "abcdefg" you may only get "abc".

3.5.20 Printing

the following bugs are known in the printing software:

- The following lowercase characters do not currently print correctly on a PostScript printer:

- i with grave accent
 - i with acute accent
 - i with circumflex accent
 - i with diaeresis

- A storage allocation bug exists in the filters in `prsvr` that use `tmllib`. This problem occurs primarily in the non-PostScript printers that process text jobs. Jobs that are sent to printers in bitmap or transparent mode do not experience this problem.

When you use a filter in the print server that uses `tmllib`, storage is not released. The lost storage for each text job is approximately the size of the text file. Eventually the disc will become full. To correct this condition restart `prsvr`. This will free the allocated storage.

- The example programs for writing drivers and filters are not included in this release of the software.

3.5.21 `gm_$init`, Domain/Dialogue

The `gm_$init` call fails the first time it is called, but works the second time. Domain/Dialogue uses this call to initialize graphics areas; therefore some Domain/Dialog applications may fail.

3.5.22 `vfmt` Calls

The `vfmt` calls do not properly handle a decimal value of `-32768` when using the `%wpd%$` format string. The resulting output is `-4294934528`.

3.5.23 `smdu` Insert Files

The `smd_$` interface is superseded. Although the beta software includes `/sys/ins/smdu.ins.xxx` insert files, they will not be shipped at the FCS.

3.5.24 `getpass` and `/bin/login`

The UNIX `getpass()` call fails from an SIO line when there is no controlling tty. As a result `/bin/login` fails to get the password in such cases.

3.5.25 `closedir`

The UNIX `closedir` call memory faults on a NULL pointer.

3.5.26 UUCP

Because the UUCP lock directory is local per node, if two users linked to the same

uucp site node attempt to uucp or uux to the same remote system at the same time, there may be a sequence number conflict.

UUUCP does not currently support communication over TCP, TLI, or X.25.

3.5.27 GPIO

If you have global GPIO devices loaded, you must exit to the boot shell prior to shutting the machine down.

3.5.28 DDE

The following bugs are known in the Domain/DDE debugger:

- The debugger interprets `-type` and `-on` arguments appearing in the program invocation as its own arguments. To work around, invoke the debugger with no arguments and use the Domain/Dialogue interface to the `DEBUG` command.
- These commands do not work well at Beta2:
 - `TRACE`
 - `WATCHPOINT`
- The following command/option combinations do not work at Beta:
 - `GO [-signal expression | -ignore]`
 - `STEP [-signal expression | -ignore]`
- The following additional problems are known:
 - All references to names in the program are case sensitive. C is case correct, but in other languages, use lower case.
 - `SEARCH` with no arguments (repeat search) does not work.
 - No source display if the dde window is totally obscured when dde is invoked.
 - The `AGAIN` key does not work from the debugger input pad until after the cursor has entered the debugger display area once.
 - Some keys cannot be defined using the `KEY` command. This is so they can retain the user's DM key bindings.
 - You cannot use the debugger to assign conformant arrays in Pascal.
 - Argument information is sometimes unavailable in FORTRAN.
 - If the debugger's request to load the program fails, the debugger says the program exited, without giving an appropriate error message.
 - If watching or tracing is on, `STEP -OVER` steps into routine.
 - Variables declared in unnamed C inner scopes cannot be referenced. (This is caused by a compiler bug).

- If the debugger is tracing or watching at statement granularity, STEP -INSTRUCTION does not produce the right results. If it is tracing or watching at statement granularity in a particular routine, STEP does not produce the right results.
- Displays may be an annoying color if you don't use the default color map.
- The debugger cannot find the correct address for variables in registers in code ranges that have been optimized away. (This is caused by a compiler bug).
- If you are debugging a obj format file and set your environment to a module, then no source will be displayed and line number references will not be recognized. Use env \proc instead.
- If you scroll the transcript up to look at something, then move the cursor into the display area, the transcript jumps back to the bottom.
- The debugger cannot reference variables that are not in the current environment, e.g., local variables in a different stack frame.
- The DM tn (to_next_window) command will not go to the display area.
- If the following warning appears occasionally, it can be ignored:
(Warning) Found more than one symbol called "-----"
- Named string constants have type char (obj only).
- Syntax error reporting in Pascal is poor.

Use the following procedure to report any additional bugs that you may discover in the debugger:

1. Run the debugger with **prop echo -pretty** or **prop echo -graphic**, if possible, to transcript the commands given via menus, keybindings, and macros.
2. Include in your bug report:
 - The version of the debugger; use the version command
 - A description of the problem
 - A transcript of the whole debug session (if possible)
 - All the applicable error log entries (from 'node_data/system_logs/debug_error_log')
 - A small test program, if possible

3.5.29 Pad Manager

The pad manager will ignore a use_font in a frame; it can't put back original font after the frame is closed and deleted.

If you do an conditional pad get (ios_\$get with ios_\$cond_opt/) and something is obtained (that is, something was stacked in the input pad waiting to be read), then henceforth input to the pad is read immediately and more stacked input moves to the

transcript before it should.

3.5.30 Tools Not Available at Beta

dbx, prof, and gprof are not in beta but will be available by FCS.

3.5.31 DN5x0-T Cartridge Tape Access

An attempt to access a cartridge tape drive on a DN5x0-T (Turbo) system that does not have a cartridge tape drive and does have a disk drive crashes the node.

3.5.32 DN5x0 Initialization Messages

DN5x0 Systems may not display the Apollo logo and may clip the first character of the startup banner at boot time.

3.5.33 Documentation Bugs

The following documentation errors exist at beta2.

- | | |
|----------------|--|
| chacl | You must use I and not i to specify the ignore bit when you use /usr/apollo/bin/chacl. The man page incorrectly indicates that you should use the lowercase character. |
| General Errors | The UNIX man pages currently have several errors where they refer to unsupported facilities and hardware such as /dev/kmem, 3b2 hardware, etc. |
| Underscores | In some cases, underscores may appear to be "strange". |

3.5.33.1 GPR

When using gpr_\$text on a DN3000 high resolution monochrome workstation with a large number of font files, (more than approximately 40) there is a small probability that a spurious "bad font file" (symbol gpr_\$bad_font_file) status will be returned, and the text will not be drawn.

This problem is most likely to occur when GPR is being used either in borrow mode, or in direct mode with little or no other activity on the screen.

Chapter 4: Installing SR10

SR10 provides a completely new set of tools for installing Domain software. *Installing Software with Apollo's Release and Installation Tools* provides detailed instructions for using these tools to install standard system software on Domain workstations and servers. However, see "Beta Installation Considerations" below for installation information that applies at Beta. This chapter contains information on the /install/IR directory used by the new installation tool and about other installation-related tools. It describes any special installation considerations that apply at Beta. An it provides a list of media on which this release is distributed.

4.1 The /install Directory

As part of the change to the install procedures, we have changed the use and contents of the node /install directory. The top-level /install directory on the average user's node no longer contains installation scripts. Instead, it contains the following:

- a baseline directory. This directory holds baseline files for the node on which it is located. A baseline file is a record of the state of the node, in terms of products installed and options selected, as of its most recent installation. The new install tools create a new baseline file, and optionally purge the old files, every time they are run.

Because the existence of a baseline file on a node speeds up subsequent installations of software to that node, we recommend that users not delete this file or the /install directory in which it is stored.

- a configuration directory (optional - created by the node owner). This directory contains configuration files. Configuration files are formatted lists of products and product options that the install tool uses as input to determine what objects to load during an installation. They are created by the config or the install++ tool, or you can use default configuration files provided by Apollo with the software.
- a doc directory, which contains the apollo directory. The /install/doc/apollo directory contains the SR10 software release notes, release notes for optional products, and *Making the Transition to SR10 Operating System Releases*. This directory also contains an informal file of answers to commonly asked questions about SR10.
- a preserve.list file that contains a list of files on this node that are not to be changed by the installation.
- a not_installed file that contains a list of files that could not be installed during the last installation session.
- a not_installed.temp file that contains a list of files that could not be installed during the last installation session. The install tool goes back and forth between this file and the previous one. At the end of an installation, this file is deleted.

If a node or file server is used as an Authorized Area - that is, a source area for installations over the network - its /install directory contains all the files and directories listed above, as well as the following objects:

- Release Index subdirectories. These directories' names all begin with "ri.". Each release index directory is associated with a single released product. The product may be the base OS software with the three operating environments, or it may be a layered product like Domain FORTRAN or Domain/Dialogue. Its release index directory contains all the objects belonging to the product, including the binary release index that is the blueprint for that product's installation.
- an /install/tools directory containing the complete set of SR10 installation tools.
- an /install/help directory containing online help files for the tools in the /install/tools directory.
- an /install/overrides directory for files to control the configuration choices available to ordinary users. We ship some files in this directory.
- an /install/templates directory containing 'canned' configuration and Authorized Area selection files provided by Apollo.
- an /install/sr9.7_compat directory containing SR9.7 style install scripts. These scripts are for use with optional products that can run on SR10 nodes but are not being re-released with SR10.
- an /install/<site> directory created by the site system administrator and containing site-specific configuration, override and Authorized Area selection files.

4.2 /etc/invoke_script

A new script /etc/invoke_script invokes either an Aegis shell script or a Bourne shell script, depending upon the environment or environments installed on the system. This tool is useful if you wish to use shell scripts to install non-Apollo software and cannot be sure which environments will be available on the node where the software is being installed.

4.3 Beta Installation Considerations

Appendix A of *Installing Software with Apollo's Release and Installation Tools* describes how to install the Domain/OS operating system on the first SR10 node in your network. If the disk on which you create the first Authorized Area is big enough to hold the optional software products shipped with the Beta release, the Appendix A procedure also allows you to install optional software on the first node. If you do not install optional software as part of the procedure in Appendix A, follow the directions for creating an authorized area in Chapter 3 of *Installing Software with Apollo's Release and Installation Tools*.

4.4 Media Types

We distribute SR10 on magnetic tapes, 1/4-inch streaming cartridge tapes, 5-1/4-inch

floppy disks, or 8-inch floppy disks as indicated in the following subsections.

4.4.1 Magnetic Tapes

Magnetic tape distributions of SR10 are not available at the Beta release.

4.4.2 Streaming Cartridge Tapes

Streaming cartridge tape distributions of SR10 consist of the following tapes:

CRTG_STD_SFW_VOL1 SR10.0
CRTG_STD_SFW_VOL2 SR10.0
CRTG_STD_SFW_VOL3 SR10.0
CRTG_STD_SFW_VOL4 SR10.0

4.4.3 Floppy Disks

Floppy disk distributions of SR10 are not available at the Beta release.

4.5 Release Configurations

SR10 is shipped with several prespecified configurations for installation, as described in detail in *Installing Software with Apollo's Release and Installation Tools*. In particular, there are configurations that install all Domain software or subsets that are likely to be of interest to users with local disks of various sizes.

Chapter 5: Documentation

This chapter describes the new documentation at SR10. We start with an introduction that gives you an overall view of the SR10 documentation set and describes changes we've made in the set at SR0. The rest of the chapter describes the changes in individual documents.

5.1 Introduction to SR10 Documentation

This introduction contains the following sections:

- Organization of the set at SR10
- Changes in the documentation to help you find information more easily
- SR10 books and their relation to earlier versions

5.1.1 Organization of the Documentation Set

We've organized the SR10 documentation set to fit the single operating system/multiple environment strategy of Domain/OS. Our goal was to create environment-specific manuals for the Aegis, BSD, and SysV environments whenever that made the information easier to access and use, but to provide one manual for the three environments in all other cases. We provide Aegis, BSD, and SysV versions of the:

- Task-oriented user's guides (*Using Your Environment*)
- Command references (*.... Command Reference*)
- Programmer's references (*.... Calls or Programmer's Reference*)
- System administration guides (*ManagingSystem Software*)

For basic shell command, call, or system administration information, refer to the manual for the environment(s) you will use. For basic tutorial information (*Getting Started with Domain/OS*), software installation information, SR10 transition information, Display Manager information, and any application information such as debugging or networking tools such as TCP/IP, refer to the application-specific manuals. These are appropriate for any environment. Documentation for hardware products and optional products is also appropriate for any environment.

For example, if your site plans to install and run BSD in your network, you will be using environment-specific manuals such as *Using your BSD Environment*, the *BSD Command Reference*, and the *BSD Programmer's Reference*. You'll also use application-specific manuals such as the *Domain Distributed Debugging Environment (DDE) Reference* and the *Domain Display Manager Reference*. (Note that Display Manager (DM) commands now appear in this separate DM manual, which is relevant for whatever operating environment you are using.) Beginning users can use *Getting*

Started with Domain/OS.

At SR10 we sell documentation in packages customized for any one environment, or for all three environments. These custom or general packages are available for general users, programmers, or system administrators.

5.1.2 Locating Information in the Documentation Set

SR10 provides a number of features to help you more easily locate the information you need. They are:

- Identifying information on manual spines
- *Domain Documentation Master Index/Domain Documentation Quick Reference*
- On-line manuals list available through man or help
- Domain/Delphi
- System call information available through man or help
- Man pages available through the HELP key
- Improved indexes in individual manuals

As you will note when you unpack your SR10 manuals, we now provide a built-in binding for our user manuals. Besides holding the book together and eliminating the need to order separate binders and labels, the binding mechanism provides a spine for each manual. The spine identifies the manual and allows you to easily locate books on a shelf.

The spine of each manual contains the manual's title and an icon that shows the manual's type. You can use the icons to help you organize your manuals on a shelf. For example, you might want to put all the software tools manuals together. You can do this easily by grouping the books that contain a tool icon. (The *Domain Documentation Quick Reference*, described next, shows all the icons and the books in each icon category.)

The *Domain Documentation Master Index* and *Domain Documentation Quick Reference* are two new tools for locating the proper manual in your hardcopy documentation set. The *Master Index* is an index of topics related to our software and hardware, and a reference to the manuals that discuss these topics. The *Master Index* also contains a complete list of manuals with their current part numbers and a description of each manual. This manual and the *Domain Documentation Quick Reference* replace the *Technical Publications Overview*.

The *Domain Documentation Quick Reference* is a folding card that also lists all the manuals in the documentation set, by category and icon. Use this card to help organize your bookshelf and get the right information when you need to order new manuals. For several releases, a similar list of manuals and their current revision numbers has been available online through the help manuals command. Now you can now also access this file by using either the UNIX man manuals or the Aegis help manuals command. (man manuals does not work at beta.)

The *Domain Documentation Master Index* is a hard-copy method of locating information in the documentation set.

At SR10, a number of manuals are available through the optional Domain/Delphi product, our online documentation retrieval system. Domain/Delphi provides sophisticated keyword search mechanisms, allowing you to browse online to retrieve online information about a particular subject, or access information sequentially as you do in a hard-copy book. You do not even need to have on hand hard-copy versions of manuals available through Domain/Delphi. At SR10, the following manuals are available through Domain/Delphi.

SR10 documentation is not available through Domain/Delphi at Beta.

- (to be added)

At SR10, you can access online documentation about programming calls for any environment via the help or man commands. Prior to SR10, only UNIX programming call man pages were available on line, via the man command, and summaries of the OS call interfaces were available through the Help facility. Now, you can also use the helpFR or man command to access detailed manual pages for Domain/OS calls. For example, to get information on the ios_\$open call from an Aegis shell, you would enter

```
help call ios_$open
```

Note that you must specify the word call before the call name. To get a man page for the same call from a UNIX shell, you would type

```
man ios_open
or
man a ios_open
```

where a is the section specifier for Domain/OS calls. Note that you should not include the \$ (dollar sign) character in the call name; This eliminates the need to escape the \$, which has special meaning in most UNIX shells.

We also provide "intro" section pages for each set of calls that can be accessed by just specifying the prefix, for example:

```
man a ios
or
help calls ios
```

We have changed the default definition of the HELP key in UNIX environments so that the key will read man pages rather than help files.

At SR10 we also paid particular attention to the individual indexes in each manual. To help you locate release-related information, the SR10 Standard Release Document now contains an index. (Not at Beta, however)

5.1.3 Manuals at SR10

This section highlights some of the most important SR10 manuals. We also list the

Domain/OS manuals that are new or revised at SR10, and indicate the equivalent versions of earlier manuals.

5.1.3.1 Key SR10 Manuals

The SR10 documentation set contains a number of new and revised manuals. This section details some of the manuals that are critical for upgrading to the new release, and using the new software effectively.

Making the Transition to SR10 Operating System Releases is the first book you should read about SR10. Plan to read this book before you plan your upgrade to SR10. An important source of information for this book was the SR10 transition experiences of beta test users and internal Apollo users. So we think it should be particularly useful in guiding you through your transition. Install SR10 only after reading this manual thoroughly!

SR10 provides a powerful and flexible new set of installation tools. *Installing Software with Apollo's Release and Installation Tools* is required reading before using these tools to install standard or layered software.

The *Domain Documentation Master Index*, mentioned previously, is a critical manual because it guides you to the manuals that are most appropriate for getting information about each component of Domain/OS and its three operating environments.

5.1.3.2 SR10 and Pre-SR10 Manuals

This section lists the Domain/OS manuals that are new or revised at SR10, and compares them with the pre-SR10 document set. We include this information because, in addition to revising many manuals, we have changed the organization of the document set to reflect the new structure of Domain/OS. In some cases the manual has changed name and/or part number between SR9.7 and SR10, but still provides the same type of information. In other cases, we have reorganized information into different manuals. For example, the *Domain System Command Reference* no longer exists; the *Domain Display Manager Command Reference* and *Aegis Command Reference* replaced this manual. Other manuals are completely new at SR10. Also, our part number scheme has changed somewhat, as described next.

Prior to SR10, manuals typically had a six-digit part number that stayed the same from release to release. Manual part numbers at SR10 have a six-digit base number and a three-digit suffix. For hardcopy documentation, the three-digit suffix begins with A. Typically, we increment the suffix every time we revise a manual. So, for example, the SR10 version might be 011022-A00 and the next revision might be 011022-A01. This new numbering scheme will enable us to maintain two versions of one manual at the same time.

When we mention manual part numbers in our documentation, we usually only mention the base number, and not the suffix. When you see the base number, we are referring to the most recent version of the manual. However, when we show the entire part number, we're referring to the specific version that had that part number. In the manual correspondence table that follows, we show complete part number information.

The following table lists the Domain/OS software manuals that are new at SR10. It lists the manual name and part number at SR10, and the equivalent name and part number at SR9.7. It does not list manuals for layered products that have been revised at SR10. You can see a full list of manuals available at SR10 FCS date by entering help manuals in an Aegis shell or man manuals in a UNIX shell. (man manuals does not work at beta.)

<i>Domain/OS Documentation Changes at SR10</i>			
SR10 Manual	SR10 Part Number	Corresponding Pre-SR10 Manual	Pre-SR10 Part Number
Making the Transition to SR10 Operating System Releases	011435-A00	Making the Transition to SR9.5	009492
Installing Software with Apollo's Release and Installation Tools	008860-A00	Installing Domain Software	008860, Rev 01
Getting Started with Domain/OS	002348-A00	Getting Started with your Domain System Getting Started with your Domain/LX System	002348, Rev 04 008017, Rev 01
Domain Display Manager Command Reference	011418-A00	Domain System Command Reference	002547, Rev 01
Domain Documentation Master Index	011242-A00	Technical Publications Overview	002685, Rev 07
Domain Documentation Quick Reference	002685-A00		
Using TCP/IP Network Applications	011021-A00	Using ftp and telnet	008867, Rev 00
Using Your Aegis Environment	011021-A00	Domain System User's Guide	005488, Rev 01
Aegis Command Reference	002547-A00	Domain System Command Reference	002547, Rev 04
Aegis Utilities Reference	009414-A00	Domain System Utilities Reference	009414, Rev 04
Using Your BSD Environment	011020-A00	Domain/LX System User's Guide	005803, Rev 00
BSD Command Reference	005800-A00	Domain/LX Command Reference for BSD4.2	005800, Rev 04
Using Your SysV Environment	011022-A00	Domain/LX System User's Guide	005803, Rev 00

<i>Domain/OS Documentation Changes at SR10</i>			
SR10 Manual	SR10 Part Number	Corresponding Pre-SR10 Manual	Pre-SR10 Part Number
SysV Command Reference	005798-A00	Domain/IX Command Reference for BSD4.2	005798, Rev 01
UNIX Text Processing	011018-A00	Domain/IX Text Processing Guide Domain/IX Text Editors Quick Reference	005802, Rev 01 005804, Rev 01
Managing Aegis System Software	010852-A00	Administering Your Domain System	001746, Rev 06
Managing BSD System Software	010853-A00	System Administration for Domain/IX BSD4.2	009355, Rev 00
Managing SysV System Software	010851-A00	System Administration for Domain/IX SysV	009356, Rev 00
Managing Domain Routing and Domain/OS in an Internet	005694-A00	Managing the Domain Environment in an Internet	005694, Rev 01
Planning Domain Networks and Internets	009916-A00	Planning Domain Networks and Internets	009916, Rev 00
Managing the NCS Location Broker	011895-A00	New at SR10	N/A
Configuring and Managing TCP/IP	008543-A00	Configuring and Managing TCP/IP	008543, Rev 01
Making the Transition to SR10 TCP/IP	011717-A00	New at SR10	N/A
Domain/OS Call Reference	007196-A00	Domain System Call Reference	007196, Rev 01
Programming with Domain/OS Calls	005506-A00	Programming with General System Calls	005506, Rev 01
BSD Programmer's Reference	005801-A00	Domain/IX Programmer's Reference for BSD4.2	005801, Rev 01
SysV Programmer's Reference	005799-A00	Domain/IX Programmer's Reference for SysV	005799, Rev 01
Getting Started with SysV STREAMS	012276-A00	New at SR10	N/A
Programming with SysV STREAMS	012275-A00	New at SR10	N/A

<i>Domain/OS Documentation Changes at SR10</i>			
SR10 Manual	SR10 Part Number	Corresponding Pre-SR10 Manual	Pre-SR10 Part Number
Programming with the SysV Transport	012205-A00	New at SR10	N/A
Domain/OS Programming Environment Reference	011010-A00	Domain/IX Support Tools Guide	009413, Rev 00
		Domain Binder and Librarian Reference	004977, Rev 02
Domain Distributed Debugging Environment (Domain/DDE) Reference	011024-A00	Domain Language Level Debugger Reference	001525, Rev 04
Printing in the Aegis Environment	011774-A00	New at SR10	N/A

5.2 Changes to Documentation

The rest of this chapter contains information that is not currently supplied in published documentation and will be included in the appropriate manuals at a later date.

5.3 UUCP Configuration Instructions

You should do the following when configuring UUCP. These instructions are not inclusive, but contain Apollo-specific configuration information.

- Tty/sio ports used by uucp must be configured to be compatible with the attached modem. e.g., baud rate, bits per character, parity.
- Although /dev/siox and /dev/ttyx (where x is the port number) refer to the same physical port, they are handled differently. Operations on /dev/siox ignore DCD status while default operations on /dev/ttyx adhere to the UNIX conventions (such as requiring DCD to be high before completing an open of the device). The uucp programs open the device with the O_NDELAY flag, which causes the open to return without waiting for DCD. This enables either /dev/siox or /dev/ttyx devices to be used for both dialing out or receiving calls. However, it is recommended that /dev/ttyx devices be used exclusively for uucp (including cu and tip). This enables device locking to function properly.
- To monitor the tty/sio port for incoming calls, add an entry to the /etc/ttys file to start /etc/getty. It is recommended for the reasons stated above, that /dev/ttyx be used for getty. A port monitored by /etc/getty cannot be used to dialout on unless getty is suspended first. This can be done using a script.
- The uucp programs look for the file, /usr/lib/uucp/sitename, to determine the uucp sitename. This is an ascii file containing the uucp sitename and must be created by the system administrator. If this file does not exist, the sitename is set to the nodename returned by the gethostname call.
- The login account for sites calling into the uucp site node must have its shell defined to be /usr/lib/uucp/uucico.real.
- The lock directory used by the uucp programs (including cu and tip) is /usr/spool/locks. It is installed as a link to 'node_data/locks on both uucp user nodes and uucp site nodes. Old programs that use /usr/spool/uucp for locking should be modified to use the new directory.
- Some versions of uucp, including the sr9.7 Domain/IX version, send the login sequence with even parity by default. The sr10 login program expects no parity. To change the parity of the login sequence sent, add the expect/send sequence of "" P_ZERO before the login expect/send sequence in the L.sys file. For example, the file could look like:

```
ogin:--ogin: uucp ssword: uucp  
hammer Any ACU 9600 T5551212 "" P_ZERO ""
```
- Uuencode does not preserve aegis file type information. Uudecode always produces files of type unstruct. Therefore, to transfer an obj type file, use the obty command to change the type to unstruct. Uuencode the file, transfer it to the destination

system, and uudecode it. Use the obty command to modify the type to obj.

You should refer to *Managing SysV System Software* or *Managing BSD System Software* for information on UUCP. The following documents also contain helpful information about HoneyDanber UUCP:

- *AT&T UNIX System V User's Guide*FR, Chapter 8, pp. 51-77
- *AT&T UNIX System V System Administrator's Guide*, Chapter 9, pp. 1-44
- *UNIX System Security*, Patrick H. Wood and Stephen G. Kochan, Hayden Books, Chapter 6, pp. 168-186
- "HoneyDanBer UUCP - Bringing UNIX Systems into the Information Age", Bill Rieken and Jim Webb, *;login:*, May/June 1986, pp. 27-36 and *;login:*, July/August 1986, pp. 10-35

5.4 Undocumented GPR Features

This section describes GPR features that were added at SR9.7 but have not yet been incorporated into the two GPR manuals, *Domain Graphics Primitives Resource Call Reference* (Order No. 007194) and *Programming with Domain Graphics Primitives* (Order No. 005808). These features include eight new subroutine calls and support for the Domain Dial Box.

5.4.1 New GPR Subroutine Calls

The following new calls were added to GPR at SR9.7. They will be added to the next release of the GPR manuals:

- `gpr_$inq_background`
- `gpr_$inq_blank_timeout`
- `gpr_$inq_event_data`
- `gpr_$inq_foreground`
- `gpr_$inq_plane_mask_32`
- `gpr_$set_blank_timeout`
- `gpr_$set_raster_op_mask`
- `gpr_$make_bitmap_from_array`

The data types used with these calls are existing types described in *Domain Graphics Primitives Resource Call Reference* or *Domain/OS Call Reference*. Any new data types are simple extensions of existing types. For example, the `gpr_$mask_32_t` type is a set with 32 elements. C and FORTRAN programmers should use 4-byte integers rather than 2-byte integers to store the elements of this set:

- `Gpr_$inq_background` is a function that returns the background color of the window open to the current process. The background color is returned in `gpr_$pixel_value_t` format. This is a 4-byte integer. This function can be used in

only direct and frame modes. It returns the following error if it is used in borrow mode: no such rectangle property key (graphics/rectangle manager).

- **Gpr_\$inq_blank_timeout** returns the time period that the display manager waits before it shuts off the screen following the last input event. This routine is similar to the shell command `scrto`. The time period is returned in the timeout parameter. The data type for this parameter is listed in `/sys/ins/base.ins.pas`. This data type is described in detail in the *Domain/OS Call Reference*.
- **Gpr_\$inq_foreground** is a function that returns the foreground color of the window open to the current process. The foreground color is returned in `gpr_$pixel_value_t` format. This is a 4-byte integer. This function can be used in only direct and frame modes. It returns the following error if it is used borrow mode: no such rectangle property key (graphics/rectangle manager).
- **Gpr_\$inq_plane_mask_32** returns the plane mask for the current bitmap. Mask, in `gpr_$mask_32_t` format, is the current mask. This is a set with up to 32 elements. Use `gsr_$set_plane_mask_32` to set a plane mask.
- **Gpr_\$set_blank_timeout** establishes the time period that the system waits before it shuts off the screen. The time period is measured from the last input event. This routine is similar to the shell command `scrto [n]`. Set the time period with the timeout parameter. The data type for this parameter is `time_$clock_t`, which is listed in `/sys/ins/base.ins.pas`. This data type is described in detail in the *Domain/OS Call Reference*.
- **Gpr_\$set_raster_op_mask** provides a way to set raster operations on several planes with just one call. `Pl_mask`, in `gpr_$mask_32_t` format, establishes the plane mask. This is a set with up to 32 elements. `Op`, in `gpr_$raster_op_t` format, establishes the raster operation. This is a 2-byte integer. The raster operations established with this call affect only the primitives currently specified with `gpr_$raster_op_prim_set`.
- **Gpr_\$inq_event_data** returns the time that an event occurred, and, if the event is a dial event, the dial number and the dial value. The dial number and dial value are returned in `gpr_$event_data_t` format. The two input parameters are `event_type` and `length`. `Event_type` is the type of event in `gpr_$event_t` format.

`Length` is the size in bytes of `event_data` parameter. The output parameters are `event_data`, `length_ret`, `time_stamp` and `sts`. `Event_data` is the data structure (variant record) used to return the event data. FORTRAN programmers may implement this data type by declaring a two element array of 4-byte integers. The first element is the dial number, and the second element is the dial value.

`Length_ret` is the number of bytes returned by the call. `Time_stamp` is the time that the event occurred in `time_$clock_t` format. This data type is described in detail in *Programming with Domain/OS Calls*. `Sts` is the status parameter.

- **Gpr_\$make_bitmap_from_array** provides a way to generate a bitmap from an array. The bitmap is stored in main memory in external bitmap file format. This feature enables BLT operations to be performed with the new bitmap, which is advantageous if read/write pixel operations done with arrays is limiting.

To deallocate this type of bitmap, use `gpr_$deallocate_bitmap`. Since this type of bitmap resides in main memory, it would also be deallocated upon calling `gpr_$terminate`.

{ `Bgpr_$make_bitmap_from_array` is used to create a bitmap descriptor pointing to a given memory address (containing the image data). }

```
PROCEDURE gpr_$make_bitmap_from_array (  
  IN   size      : gpr_$offset_t;      { X,Y size of bitmap }  
  IN   attribs   : gpr_$attribute_desc_t; { Attributes bitmap is to have }  
  IN   groups    : integer;            { Number of groups }  
  IN OUT g_headers : gpr_$bmf_group_header_array_t;  
                                     { Group headers }  
  OUT  bitmap    : gpr_$bitmap_desc_t;  { Returned bitmap descriptor }  
  OUT  status    : status_$t           { Returned status }  
); EXTERN;
```

5.4.2 Support for the Domain Dial Box

The dial box is an interactive graphics input device that allows the user to dynamically manipulate various display parameters with GPR and 2D GMR applications. GTCO Nomad and Apollo dial boxes are both supported. The dial box enhances the capabilities provided by a mouse or digitizing tablet. When used with customer applications and Apollo dial device utilities, the dial box provides precise zoom, pan, 3D-rotation, and other user-selectable graphics capabilities.

Dial Server

The dial server allows for dial device data to be integrated into the GPR input environment. This data is then available for GPR, 2DGMR, and Domain/Dialogue applications. For Domain/Dialogue dial box information see the online release notes `/install/doc/apollo/dialog.v2.01` that are available with this release. For 2DGMR information see the *Programming with Domain 2S Graphics Metafile Resources* (Order number 5097-A00) manual.

The dial box server is a process that waits for data from dial boxes connected to SIO ports. It also waits for command requests from a mailbox.

The server passes dial box data to processes that have enabled input for dial boxes via GPR functions.

Dial requests (commands) are used for initializing a dial box on an SIO port and closing/releasing an SIO port. These requests are passed to the server through the mailbox.

You may connect up to three dial boxes to a node. When multiple boxes are connected, they may be distinguished by the dial numbers that are assigned when the box is initialized.

Starting the Dial Server and Sending Requests

To start the dial server, execute the following Display Manager command:

```
cps /sys/dial_server -n dial_server
```

After the dial box process is started, you must initialize each dial box. Use the following shell command to initialize a dial box:

```
/usr/apollo/bin/dial_server_request [-Init port<p>] [-First_dial <n> (optional)]
```

Possible port numbers are 1, 2, and 3 depending on the number of SIO ports on the node. First_dial is the number assigned to the first dial. The default first dial number is 1.

The dials on the GTCO box are numbered sequentially as follows.

```
first_dial      first_dial + 1    first_dial + 2
first_dial + 3  first_dial + 4    first_dial + 5
first_dial + 6  first_dial + 7    first_dial + 8
```

The dials on the Apollo Dial Box are numbered sequentially as follows:

```
first_dial      first_dial + 4
first_dial + 1  first_dial + 5
first_dial + 2  first_dial + 6
first_dial + 3  first_dial + 7
```

Dial numbers may range from 1 to 99, and you can assign the first dial of any dial box to any number in that range. When you initialize a dial box, include the first dial argument, or the first dial will be dial 1. This occurs with every box that you initialize. If you initialize three boxes without explicitly setting the first dial number, all the boxes will have the first dial as dial 1. In this case, you will not be able to distinguish input among the boxes.

If you use two dial boxes, the following shell commands initialize the first box on Port 1, with the knobs numbered from 1 to 9, and the second box initialized on Port 3 with the knobs numbered from 20 to 29. In this case, you can distinguish what box produced the input by the dial numbers:

```
/usr/apollo/bin/dial_server_request -init port1 -first_dial 1
/usr/apollo/bin/dial_server_request -init port3 -first_dial 20
```

Provided that your request is accepted, the following messages are displayed:

```
>>Server Response: ....Request Accepted: Initialize Port 1, First dial = 1..
>>Server Response: ....Request Accepted: Initialize Port 3, First dial = 20..
```

If the server is not running, the following error message is displayed:

*****Could not open up mailbox channel*****

Other errors, such as port contention/inaccessibility problems with the dial box, the cables, or the request itself, are indicated by one of the following error messages:

>>Server Response:Request Failed: Box not connected or dead box....

>>Server Response:Request Error: Port not available....

>>Server Response:Request Error: Could not open port....

Closing an SIO Port

To close an SIO port, use the following request:

```
/usr/apollo/bin/dial_server_request [-close port <p>]
```

For example, to close SIO Port 1, enter the following command:

```
/usr/apollo/bin/dial_server_request -close port 1
```

Provided that your request is accepted, the following message is displayed:

>>Server Response:Request Accepted: Close Port 1....

Using Dial Boxes with GPR

To enable GPR to recognize dial events, use `gpr_$enable_input`. A new event type, `gpr_$dial`, has been added to the `gpr_$event_t` data type for this purpose.

Use `gpr_$event_wait` or `gpr_$cond_event_wait` to wait for dial events. If the event was a dial event, `gpr_$inq_event_data`, a new GPR call, returns the dial number and the dial value. The dial number identifies the dial, and the dial value indicates the amount that the knob was turned.

Testing the Dial Box

The dial box verification procedures are included in the `/systest` directory. The tests include `dial_driver_test` and `dial_server_test`.

`dial_driver_test` allows you to check dial box functionality without initializing the dial server. `dial_server_test` allows you to test the box with the server initialized (use the initialization command described above). The tests return positive integers when a dial is turned to the right and negative integers when a dial is turned to the left. The tests also specify the dial being used. To run the tests, enter the `/systest` directory and type:

```
dial_server_test <RETURN>  
dial_driver_test <RETURN>
```

Dial Box Startup Scripts

All the SR10 startup scripts include a commented command to start up the dial server. Instructions in the file direct you to uncomment this command if you want the server to start automatically.

Dial Box Example

The following Pascal fragment shows how to enable and accept input from a dial box:

```
gpr_$enable_input(gpr_$dial, keys, st);
IF(st.all <> 0) THEN writeln(' gpr_$dial could not be enabled!');

REPEAT
  discard(gpr_$event_wait(ev_type, ev_char, ev_pos, st));
  IF(st.all <> 0) THEN writeln('GPR_$inq_event_data failed - retlen = ',
    retlen, 'status = ', st.all);
  IF(ev_type = gpr_$dial) THEN
    BEGIN
      gpr_$inq_event_data(ev_type, sizeof(ev_data), ev_data, retlen, time, st);
      IF(st.all <> 0) THEN writeln('GPR_$inq_event_data failed - retlen = ', retlen,
        'status = ', st.all)
      ELSE writeln('GPR Event: dial#:', ev_data.dial_number, ' dial value = ',
        ev_data.dial_value);
    END;
UNTIL (count > 1);
```

Appendix 1: Changes to Commands and Calls

This appendix lists the commands and calls that have been added, changed, and deleted since SR9.7.

A.1 Changes to Commands

The following subsections list new, changed, and obsoleted commands since SR9.7.

A.1.1 DM Commands

The DM log-in command no longer accepts the **-h** or **-p** options.

A.1.2 Commands in /com

The following subsections describe changes to the Aegis commands that are located in the **/com** directory. As of SR10, several commands in the **/com** directory are simply links to files in either the **/usr/apollo/bin** or **etc** directory. See the sections on these directories for more details.

A.1.2.1 Obsolete Commands

The following commands and utilities are no longer supported as of SR10, or will not be supported in future releases:

aqdev	The aqdev command is supplied in the /com directory at SR10 for changeover compatibility reasons only. See the GPIO release notes for more details.
bind_sr9.2	Development of software for pre-SR9.5 systems is not supported at SR10
cmacct	Replaced by the edrgy utility
cmppo	Replaced by the edrgy utility
crrgy	Replaced by the rgy_create installation utility
crucr	Replaced by the mkapr command
debug	Replaced by the dde utility
edacct	Replaced by the edrgy utility
edppo	Replaced by the edrgy utility
fppmask	Obsoleted due to single-program-per-process support; replace with the fpp_\$control Domain/OS call.

lbr_sr9.2	Development of software for pre-SR9.5 systems is not supported at SR10
login	If you do not specify the -c option, the /com/login command now invokes the default login shell, and not necessarily the Aegis shell.
lrgy	Replaced by the rgy_admin utility's lrep command
mrgrgy	Replaced by the rgy_merge utility
old_edfont	Replaced by the new edfont utility
prfd	Use prf -dialog
rldev	The rldev command is supplied in the /com directory at SR10 for changeover compatibility reasons only. See the GPIO release notes for more details.
salrgy	Replaced by the rgy_admin utility
scrch	Replaced by the /usr/apollo/scrattr command
sh_8	The standard Aegis shell, sh, now provides 8-bit support
vctl	This command is replaced by stty

A.1.2.2 Changed Commands

The following commands have been changed between SR9.7 and SR10. Refer to the corresponding reference document or Help file for details of the new version of the call.

acl	The acl command has been changed to incorporate the SR10 protection mechanisms.
crtobj	Several options have been added and deleted. Several options have been added to provide support for European languages and time/date formats
edacl	The edacl command has been changed to incorporate the SR10 protection mechanisms.
lusr	The -me option is now the default and a new -allnodes switch has been added.
nd	The nd command is now built in to the Aegis shell; /com/nd is a shell script. As a result, programs that pgm_\$invoke /com/wd or /com/nd will no longer work - they will have to use the appropriate system calls instead.
sigp	Any process with an id that has p rights on the 'node_data/node_owners file of the node where the process executes can signal any process on the node. Otherwise, the process can only signal processes with the same SID.

- tb** The **tb** command has been extended. It now reads the process dump file for tracebacks of dead processes, and includes all functionality that was previously provided by the **/systest/ssr_util/fmpd** program. The available options have been changed and expanded. the **-args** option is no longer supported. (**/com/tb** is a link to **/usr/apollo/tb**.)
- wbak** The **/wbak** command takes a new argument, **-presr10** allows you to make a tape on sr10 that is readable on an SR9 system by the standard pre-SR10 rbak. There are separate **/com/wbak** and **/usr/apollo/bin/wbak** commands with different interfaces.

A.1.2.3 New Commands

The following commands are new as of SR10:

- cvtname** Tool for conversion between SR9.7 and SR10 case representation. This command is in **/com** only at Beta. It will be installed in the **/sr9.7_compatibility/compat_with_sr9.7** directory at FCS.
- dde** A new language level debugger which replaces the debug tool (**/com/dde** is a link to **/usr/apollo/dde**.)
- lst** Lists information about a storage tree

A.1.3 Commands in **/usr/apollo/bin**

The **/usr/apollo** directory contains Domain/OS commands that are useful in all environments. The following commands, servers, files, and utilities are also available through the **/com** directory or are new as of SR10 and are specific to Domain/OS.

The commands in the following list that were in the **/com** directory in previous releases are still available in that directory; however, many are now links to these files. In cases where the **/com** command expands derived wild cards there are different versions of the command in the **com** and **usr/apollo** directory, and the version in **/usr/apollo/bin** directory conforms to UNIX syntax conventions and does not expand wildcards. Commands in **/usr/apollo/bin** that are new at SR10 are *not* available through the **com** directory

- bltd** Displays the time at which the operating system was built (**bltd(1)**)
- chacl** Changes the access control list, intended for UNIX users; new at SR10 (**chacl(1)**)
- cpacl** Copys the access control list, intended for UNIX users; new at SR10 (**cpacl(1)**)
- cpscr** Copys the current display to a file (**cpscr(1)**)
- crddf** Creates, displays or modifies a Device Descriptor File (**crddf(1)**)
- crp** Create a remote process; Note: you If you use **crp** in a UNIX shell and include a command line in the **crp** invocation, you must quote any wild cards that you include in the command line. (**crp(1)**)

crpad	Creates a transcript pad and window (crpad(1))
crtv	Creates a new type (crtv(1))
dbacl	Domain/Dialogue based tool for changing protections
dde	High-level debugger; new at SR10 (dde(1))
dial_server_request	Initializes a dial box (ial_server_request(1))
dlty	Deletes a type (dlty(1))
edfont	Edits a character font; revised at SR10 to edit 8-bit fonts; (edfont(1))
edmtdesc	Edits a magnetic tape descriptor file (edmtdesc))
emt	Emulates a dumb terminal (emt))
esa	Displays the address of external symbol (esa))
french_to_iso	Converts an old-format French font to an 8-bit ISO standard font format; new at SR10 (french_to_iso)
fst	Prints fault status information; at SR10 this command is only useful if the INPROCESS variable is set and all commands are run in-process. In all other cases, the tb command provides the information formerly provided by fst. (fst(1))
german_to_iso	Converts an old-format German font to an 8-bit ISO standard font format; new at SR10 (german_to_iso)
intm	Installs a type manager (intm(1))
inty	Installs a type
kbm	Displays the keyboard characteristics and sets the key assignments for the compose, long_shift, and short_shift functions; new at SR10 (german_to_iso) (inty(1))
las	Lists objects mapped into the address space (las(1))
lbr2ar	Converts an Aegis obj lbr library format file to UNIX COFF ar archive format; new at SR10
lcm	Loads a color map (lcm(1))
llib	Lists installed libraries (llib(1))
llkob	Lists locked objects (llkob(1))
lsacl	Lists the access control list, intended for UNIX users; new at SR10 (lsacl(1))
lty	Lists installed types (lty(1))
mkapr	Creates an Apollo Product Report; new at SR10

mksinit	Make a SysV STREAMS initialization file from a master.d file; new at SR10
mt	The BSD mt program
nor.dan_to_iso	Converts an old-format Norwegian/Danish font to an 8-bit ISO standard font format; new at SR10 (nor.dan_to_iso)
obj2coff	Converts obj format binaries (produced by pre-SR10 compilers) to COFF format; new at SR10
prf	Queues a file for printing; there are separate /com/prf and /usr/apollo/bin/prf commands with different interfaces. (prf(1))
rbak	Restores a backup file that was generated by wbak (rbak(1))
rwmt	Reads or writes magnetic tapes; there are separate /com/rwmt and /usr/apollo/bin/rwmt commands with different interfaces. (rwmt(1))
scrattr	Shows screen attributes, provides the same information provided in previous releases by /com/scrch ; new at SR10 (scrattr(1))
scrt0	Sets the screen time-out for display blanking (scrt0(1))
stcode	Translates the status code value to a text message (stcode(1))
strinfo	Print information about currently running (SysV) Streams; new at SR10 (strinfo(1))
swedish_to_iso	Converts an old-format Swedish font to an 8-bit ISO standard font format; new at SR10 (Swedish_to_iso)
swiss_to_iso	Converts an old-format Swiss font to an 8-bit ISO standard font format; new at SR10 (swiss_to_iso)
systype	Displays a file's UNIX version stamp; was in /etc in previous releases (systype)
tb	Traces back a program after a fault; this command has been extended. It now reads the process dump file for tracebacks of dead processes, and includes all functionality that was previously provided by the /systest/ssr_util/fmpd program. It also provides information previously provided by the fst command. The available options have been changed and expanded.
tpm	Sets the mouse or touchpad mode (tpm(1))
trconf	Configure a (SysV) Stream with trace modules; new at SR10
trmon	Monitor a Stream with trace modules after configuration; new at SR10
ts	Displays the module name and time stamp stored in an Apollo object module; previously in /systest/ssr_util

tz	Sets or displays the system time zone (tz(1))
uk_to_iso	Converts an old-format UK font to an 8-bit ISO standard font format; new at SR10
wbak	Writes a backup tape; there are separate /com/wbak and /usr/apollo/bin/wbak commands with different interfaces. The /wbak command takes a new argument, -presr10 allows you to make a tape on sr10 that is readable on an SR9 system by the standard pre-SR10 rbak. (wbak(1))
vsize	Sets or displays VT100 window settings (vsize(1))
vt100	VT100 terminal emulator (vt100(1))
xdmc	Executes a display manager command (xdmc(1))

A.1.4 Domain Extensions to /etc

The etc directory now contains Domain/OS administrative commands and files and is common to all three environments. The following commands, servers, files, and utilities were previously located in either the /com directory or the /sys directory on both Aegis and Domain/IX systems, or are new as of SR10 and are specific to Domain/OS.

The commands in the following list that were in the /com directory in previous releases are still available in that directory; however, many are now links to these files. In cases where the /com command expands derived wild cards there may be different versions of the command in the com and etc directories with different semantics. Some items in the following list are links to entries in the /node_data/etc directory.

chuvol	Changes the UID of a volume
copyright	A file containing copyright statements that are displayed by /etc/rc
cpboot	Copies the system boot file
crpty	Create a pseudo-tty
ctnode	Catalogs a node; there are separate /com/ctnode and /usr/apollo/bin/ctnode commands with different interfaces
daemons	A directory of files that control the daemons started by etc/rc; a link to 'node_data/etc/daemons; new at SR10
dm_or_spm	Starts the DM or SPM process, depending upon whether the system has a display; new at sr10
dmmsg	A file containing the message displayed in the DM output window on login; new at SR10
drm_admin	Data replication manager administration tool; formerly in /sys/ncs /etc/ncs/drm_admin

edns	Edits the ns_helper database
edrgy	Edits the registry database; new at SR10
environ	A file that specifies the default values of the SYSTYPE and ENVIRONMENT variables for the node.
environment	Displays the current environment and installed environments (This command is not available at Beta)
find_orphans	Finds uncataloged objects; The /etc/find_orphans command has been changed so that it can now take a list of orphans that is generated by salvol and catalogs the objects. It should be run each time you boot a node; the /etc/rc template now includes this procedure.
glbd	Global location broker daemon (/etc/ncs/glbd); formerly in /sys/ncs
group	Group file information; this is now a read-only file provided through the extensible streams mechanism
import_passwd	Imports the contents of standard UNIX /etc/passwd and /etcgroup files into the registry
invoke_script	Invokes one of two scripts, an Aegis shell script or a Bourne shell script, that are specified as input arguments, depending upon the environment(s) installed on the system.
invol	Initializes a disk volume; the invol utility has been rewritten at SR10
lb_admin	Location broker administration tool; formerly in /sys/ncs; (/etc/ncs/lb_admin)
lcnet	Lists connected networks
lcnode	Lists connected nodes
llbd	Local location broker daemon; formerly in /sys/ncs (/etc/ncs/llbd)
login_log.conf	This file (a link to 'node_data/etc/login_log.conf) is used to specify which login-related events should be logged and, optionally, the name of the log file.
lprotect	Protects the local node from remote root access; new at SR10
magic	New at SR10
mkdevno	Creates a device number; new at SR10
netmain	Network statistics tool
netmain_chklog	Cleans up bad netmain log files
netmain_note	Places a message in the network error log
netmain_srvr	Network statistics server; formerly in /sys/net
netsvc	Sets or displays network services

nodestat	Displays node statistics; /com/netstat is now a link to this file
obty	Sets or displays the type of an object; there are separate /com/obty and /usr/apollo/bin/obty commands with different interfaces
org	Provides information about organization members, similar to the group file; it is provided through the extensible streams mechanism. The entry for a single organization can span multiple lines; in this case each line but the last is terminated by an escaped newline. New at SR10
passwd	Passwd file information; this is now a read-only file provided through the extensible streams mechanism
probenet	Probes the network and displays error statistics
rgy_admin	Registry administration tool; new at SR10
rgy_merge	Tool for merging registries; new at SR10
rgyd	Registry daemon; new at SR10
rtchk	Tests traffic between adjacent routers
rtstat	Displays internet router statistics
rtsvc	Sets or displays internet routing service
salacl	Salvages an Access Control List; this utility takes a new -l option that lists the UIDs of the ACL files that are merged.
salvol	Salvages a disk volume; this command has been revised at SR10, and now takes a set of command line arguments when run from a shell
server	Runs the program or command specified by its arguments, by default with process id user.server.none; new at SR10.
show_lc	A compatibility tool which shows which obsolete system calls are being made by an object module; new at SR10
shutspm	Shuts down SPM on a node
syncids	Synchronizes UNIX user IDs on disk to the definitions in the registry to ensure consistency; this command should only be necessary in extraordinary circumstances; new at SR10
sys.conf	Global library configuration file, a link to 'node_data/etc/sys.conf'; new
sys_sh	The shell executed by init to interpret /etc/rc; new at SR10
templates	A directory containing template configuration files, such as a template /etc/rc
uctnode	Uncatalogs a node; there are separate versions of the uctnode command in the com and /usr/apollo/bin directories with different

interfaces.

uctob	Uncatalogs an object; there are separate versions of the uctob command in the com and /usr/apollo/bin directories with different interfaces.
ulkob	Unlocks an object; there are separate versions of the ulkob command in the com and /usr/apollo/bin directories with different interfaces.
uuid_gen	Generates an ASCII UUID (Universal Unique Identifier) (/etc/ncs/uuid_gen)

A.1.5 TCP/IP Support

At SR10, all three environments include the BSD version of TCP/IP communications protocol support. Each environment also provides a set of tools and utilities that use TCP/IP as an underlying communications protocol. The following sections list the commands and files provided for TCP/IP support. For more information on these facilities, see the TCP/IP documentation.

A.1.5.1 Administrative Facilities Files in /etc

The following facilities and files are common to all three environments and have been put in the **/etc** directory. Note that the section numbers follow the SysV section numbering scheme. See *Making the Transition to SR10 TCP/IP* for information on changes between the previous release of Domain/IX BSD TCP/IP and SR10 TCP/IP.

arp	(arp(1m))
dtcb	Dump TCP control block program; a debugging tool (dtcb(1))
ftpd	(ftpd(1m))
gateways	This file contains information about routing gateways (gateways(4))
gettable	(gettable(1m))
hostns	Maps /etc/hosts to /etc/named (hostns(1m))
hosts	(hosts(4))
hosts.equiv	Specifies equivalent hosts for login purposes (hosts.equiv(4))
hostname	(hostname(1))
htable	(htable(1m))
ifconfig	(ifconfig(1m))
inetd	(inetd(1m))
inetd.conf	(inetd.conf(4))
mbd	Memory buffer dump; a troubleshooting utility (mbd(1))

mkhosts	(mkhosts(1m))
named	(named(1m))
named.boot	(named.boot(4))
named.ca	(named.ca(4))
named.hosts	(named.hosts(4))
named.local	(named.local(4))
named.rev	(named.rev(4))
networks	(networks(4))
nshost	Maps <i>/etc/named</i> to <i>/etc/hosts</i> (nshost(1m))
ping	(ping(1m))
protocols	(protocols(4))
resolv.conf	Contains information used by named (resolv.conf(4))
rexecd	(rexecd(1m))
rlogind	(rlogind(1m))
route	(route(1m))
routed	(routed(1m))
rshd	(rshd(1m))
rwhod	(rwhod(1m))
services	(services(4))
syslogd	(syslogd(1m))
tcpd	Replaces <i>tcp_server</i> used by pre-sr10 Aegis TCP/IP (tcpd(1m))
telnetd	(telnetd(1m))
tftpd	An adaptation of the MIT Project Athena implementation of the tftp protocol (tftpd(1m))
trpt	TCP remote packet trace utility; a debugging tool (trpt(1m))

A.1.5.2 SysV TCP/IP Commands and Files

The following TCP/IP commands are supported in the SysV environment in the */usr/bin* directory

ftp	(ftp(1c))
logger	(logger(1))
netstat	(netstat(1))

rlogin	(rlogin(1c))
remsh	The BSD rsh utility (remsh(1c))
ruptime	(ruptime(1c))
rwho	(rwho(1c))
telnet	(telnet(1c))
tftp	An adaptation of the MIT Project Athena implementation of the tftp protocol (tftp(1c))
whois	(whois(1))

A.1.5.3 Aegis TCP/IP Commands and Utilities

The following TCP/IP commands are supported in the Aegis environment in the /com directory

ftp	
tcpstat	The BSD netstat command
telnet	

A.1.6 SysV Commands

The following sections list commands that run in the SysV environment that are obsolete, changed, or added as of SR10. It includes commands added to support AT&T System V Release 3 UNIX.

A.1.6.1 Obsolete or Unsupported Commands

As of SR10 all SysV text formatting facilities are part of The Documentors Workbench optional layered product. Therefore the following commands are no longer provided as part of the base software: **nroff**, **troff**, **eqn**, **neqn**, **tbl**, **mm**, **mmt**, **checkmm**, **mvt**, **cm**, **ptx**, **diffmk**, **hyphen**.

Due to export restrictions we do not support the following commands: **xsend**, **xget**, **enroll**, **crypt**, **makekey**.

The following commands are no longer supported in SR10:

addroot	Obsoleted by changes to the protection mechanism (addroot(1m))
crpasswd	Obsoleted by changes to the registry (crpasswd(1m))
cvtumap	Conversion from SR8 naming is not supported by SR10 (cvtumap(1m))
fix_cache	Obsoleted by changes to the protection mechanism (fix_cache(1m))
flush_cache	Obsoleted by changes to the protection mechanism (flush_cache(1m))

mkptnr	netman now performs all functions that were done by this command (mkptnr(1m))
sup	Obsoleted by changes to the protection mechanism (sup(1m))
update_slave	Obsoleted by changes to the registry system (update_slave(1m))

A.1.6.2 Changed Commands

The following commands were significantly changed between SR9.7 and SR10:

cp	This command has several added options and extensions. (cp(1))
cpio	This command can now handle typed files. (cpio(1))
date	This command can now set the date. The software system date and time is changed immediately, but the hardware clock (used to generate Domain UIDS) is actually updated when the node reboots. (date(1))
ld	The ld loader no longer invokes /com/bind, but is a separate utility. Ld now follows full UNIX semantics; as a result, archives must now be the final argument to the command. Ld can now mark object modules to be run with a non-default stack size. (ld(1))
ls	The ls command takes a new option, -T that causes it to display the types of all objects. (ls(1)) (mkdir(1))
mkdir	The SysV mkdir command also now accepts the System V Release 3 -m and -p options. (mkdir(1))
ps	The ps command can now display process status information for remote nodes. (ps(1))
tar	This command can now handle typed files. (tar(1))
time	The time command now splits cpu time up evenly into system time and user time. Previously it reported all cpu time as user time. (time(1))

A.1.6.3 New Commands

The following commands are supported in SR10 SysV that were not previously supported or are new with System V Release 3. This list includes BSD commands supported by SysV, but does not include the Apollo extensions in /etc and /usr/apollo/bin

300	(300(1))
450	(450(1))
captainfo	(captainfo(1m))
clone	Open any minor device on a SysV STREAMS driver (clone(7))

dump	(dump(1))
f77	(f77(1))
genc	(genc(1m))
getopt	(getopt(1))
getty	The SysV environment uses the 4.3BSD version of this command (/etc/getty(1m))
glossary	(glossary(1))
greek	(greek(1))
infocmp	(infocmp(1m))
init	The SysV environment uses the 4.3BSD version of this command (/etc/init(1m))
ksh	(ksh(1))
list	(list(1))
locate	(locate(1))
log	(log(7))
mcs	(mcs(1))
mkdev	The SysV mkdev command takes options to create Apollo-specific devices. (/etc/mkdev(1m))
mkinit	(mkinit(1m))
mknod	(/etc/mknod(1m))
mt	The SysV environment now supports the BSD version of the mt utility. (mt(7))
ports	(ports(7))
prof	(prof(1))
prtconf	(prtconf(1m))
sa	(sa(7))
sendmail	The SysV environment now supports the BSD version of sendmail . (sa(1m))
starter	(starter(1))
strace	(strace(1m))
strclean	(strclean(1m))
streamd	(streamd(1m))
strerr	(strerr(1m))

suid_exec	(suid_exec(1m))
sxt	(/dev/sxt(7))
syncids	(syncids(1m))
timex	(timex(1))
usage	(usage(1))
uucheck	(uucheck(1m))
uucico	(uucico(1m))
uucleanup	(uucleanup(1m))
uusched	(uusched(1m))
Uutry	(Uutry(1m))
uuxqt	(uuxqt(1m))

A.1.7 Changes to BSD Commands and Files

The following subsections list commands that run in the BSD environment that are obsolete, changed, or added as of SR10.

A.1.7.1 Obsolete or Unsupported Commands

Due to export restrictions we do not support the **crypt** or **makekey** commands.

The following commands are no longer supported in SR10:

addroot	Obsolete by changes to the protection mechanism (addroot(1m))
crpasswd	Obsolete by changes to the registry (crpasswd(1m))
cvtumap	Conversion from SR8 naming is not supported by SR10 (cvtumap(1m))
eyacc	Eyacc is not supported by 4.3BSD (eyacc(1))
fix_cache	Obsolete by changes to the protection mechanism (fix_cache(1m))
flush_cache	Obsolete by changes to the protection mechanism (flush_cache(1m))
mkptnr	netman now performs all functions that were done by this command (mkptnr(1m))
ratfor	(ratfor(1))
sup	Obsolete by changes to the protection mechanism (sup(1m))
update_slave	Obsolete by changes to the registry system (update_slave(1m))

A.1.7.2 Changed Commands and Files

The following commands were significantly changed between SR9.7 and SR10:

cp	This command has several added options and extensions. (cp(1))
date	This command can now set the date. The software system date and time is changed immediately, but the hardware clock (used to generate Domain UIDS) is actually updated when the node reboots. (date(1))
ld	The ld loader no longer invokes /com/bind, but is a separate utility. Ld now follows full UNIX semantics; as a result, archives must now be the final argument to the command.
printcap	The installation tools now put a /etc/printcap file on each node, and do not create printcap as a link to an administrative node file. Therefore, if you use BSD printing facilities you must now recreate the appropriate links or administer each node's printcap file. (The file format is not changed.) (/etc/printcap(8))
ps	The ps command can now display process status information for remote nodes. (ps(1))
tar	This command can now handle typed files. (tar(1))
time	The time command now splits cpu time up evenly into system time and user time. Previously it reported all cpu time as user time. (time(1))

A.1.7.3 New Commands

The following commands programs and files are supported in the SR10 BSD environment and were not supported in SR9.7 and earlier Domain/IX 4.2BSD. Note that this list does not include SysV administrative facilities that are available in the BSD environment due to the common /etc directory

ac	(ac(8))
atq	(atq(1))
atrm	(atrm(1))
biff	(biff(1))
coffdump	An extended version of the SysV dump command; SR10 does not support the 4.3BSD dump command. (coffdump(1))
compress	(compress(1))
comsat	(comsat(8c))
error	(error(1))
f77	(f77(1))
file	The BSD environment uses the System V version of the file command. (file(1))

finger	(finger(1))
getty	(tctty(8))
gprof	(gprof(1))
init	(init(8))
ksh	(ksh(1))
logger	(logger(1))
makedev	(/dev/makedev(8))
mkdev	The SysV mkdev command takes options to create Apollo-specific devices. (/etc/mkdev(8))
ntalkd	The new talk daemon supplied as part of 4.3BSD (ntalkd(8))
pdx	(pdx(1))
ping	(ping(8))
pix	(pix(1))
pmerge	Pascal file merger (pmerge(1))
prof	(prof(1))
rdist	(rdist(1))
reboot	The /etc/reboot command works for all nodes except for sau2 (DN3xx) and sau5 (DN5xx non-turbo) systems. On these systems, it acts as a halt command. (/etc/remote(8))
remote	(/etc/remote(8))
sa	(sa(8))
sccs	(sccs(1))
shutdown	(/etc/shutdown(8))
struct	(struct(1))
syslogd	(syslogd(8))
tcopy	(tcopy(1))
tftp	An adaptation of the MIT Project Athena implementation of the tftp protocol (tftp(1c))
timed	(timed(8))
trpt	(trpt(8c))
vacation	(vacation(1))
vgrind	(vgrind(1))

vwidth	(vwidth(1))
w	(w(1))
whois	(whois(1))
xsend	(xsend(1))

A.1.7.4 UUCP Support

At SR10, we support only the System V HoneyDanBer version of UUCP in both the BSD and SysV environments. The following BSD commands have been replaced by their System V HoneyDanBer equivalents:

- uucico.8c
- uuclean.8c
- uucp.1c
- uucpd.8c
- uulog.1c
- uuname.1c
- uupoll.8c
- uuq.1c
- uusend.1c
- uusnap.8c
- uux.1c
- uuxqt.8c

The following HoneyDanBer UUCP commands have been added to the BSD environment:

- uucheck.8c
- uucleanup.8c
- uusched.8c
- uustat.1c
- uuto.1c
- uux.1c

A.2 Changes to Calls

Domain/OS now supports the AT&T System V Release 3 and the 4.3 Berkeley Software Distribution environments, and the supported UNIX system calls conform to those releases. In addition, SR10 supports many UNIX calls that were unavailable in

Domain/IX, and several Aegis calls have been added or changed.

A.2.1 SysV Calls, Subroutines, and File Formats

The following subsections list the SysV calls that are obsolete at SR10, as well as the features that have been added or changed since SR9.7.

A.2.1.1 Obsolete or Unsupported Calls

We do not support the `crypt(3x)` call due to export restrictions.

The following system calls are obsolete. Apollo does not commit to support these calls in future software releases.

<code>default_acl</code>	This call is no longer required. At SR10 it is a noop. (<code>default_acl(2)</code>)
<code>set_sbrk_size</code>	This call is no longer required. At SR10 it is a noop. (<code>set_sbrk_size(2)</code>)
<code>soft_link</code>	This call is now identical to <code>symlink(2)</code> . Both now support soft links by using textual links. (<code>soft_link(2)</code>)

A.2.1.2 New and Changed Features

The following system calls, subroutines, file formats, and miscellaneous features are supported in the SR10 SysV environment that were not in SR9.7 and earlier Domain/IX SysV, or are significantly changed since SR9.7.

<code>abort</code>	Terminates a FORTRAN program (<code>abort(3f)</code>)
<code>abs, iabs, dabs, cabs, zabs</code>	A FORTRAN absolute value (<code>abs(3f)</code>)
<code>acct</code>	Per-process accounting file format (<code>acct(4)</code>)
<code>acos, dacos</code>	FORTRAN arccosine intrinsic function (<code>acos(3f)</code>) (<code>aimag(3f)</code>)
<code>aint, dint</code>	FORTRAN integer part intrinsic function (<code>aint(3f)</code>)
<code>asin, dasin</code>	FORTRAN arcsine intrinsic function (<code>asin(3f)</code>)
<code>atan, datan</code>	FORTRAN arctangent intrinsic function (<code>atan(3f)</code>)
<code>atan2, datan2</code>	FORTRAN arctangent intrinsic function (<code>atan2(3f)</code>)
<code>bool: and, or, xor, not, lshift, rshift</code>	FORTRAN bitwise Boolean functions (<code>bool(3f)</code>)
<code>checklist</code>	List of file systems processed by <code>fsck</code> and <code>ncheck</code> (<code>checklist(4)</code>)
<code>conjg, dconjg</code>	FORTRAN complex conjugate intrinsic function (<code>conjg(3f)</code>)
<code>cos, dcos, ccos</code>	FORTRAN cosine intrinsic function (<code>cos(3f)</code>)
<code>cosh, dcosh</code>	FORTRAN hyperbolic cosine intrinsic function (<code>cosh(3f)</code>)

ctio	Cartridge tape I/O (ctio(4))
dim, ddim, idim	Positive difference intrinsic functions (dim(3f))
dirent	File system independent directory entry (dirent(4))
dprod	Double-precision product intrinsic function (dprod(3f))
dup2	Duplicates an open file descriptor (dup2(3c))
exp, dexp, cexp	FORTRAN exponential intrinsic function (exp(3f))
filehdr	File header for common object files (filehdr(4))
fstab	File system table (fstab(4))
ftype: int, ifix, idint, real, float, sngl, dble, cmplx, dcmplx, ichar, char	Explicit FORTRAN type conversion (ftype(3f))
getarg	Returns a FORTRAN command-line argument (getarg(3f))
getdents	Reads directory entries and puts them in a file (getdents(2))
getenv	Returns a FORTRAN environment variable (getenv(3f))
getmsg	Gets the next message off a stream (getmsg(2))
getoid, geteoid	Gets the organization identity (getoid(2))
getorgent, getorgid, getorgnam, setorgent, endorgent	Gets the organization file entry (getorgent(3))
gettydefs	The speed and terminal settings used by getty (gettydefs(4))
getut: getutent, getutid, getutline, pututline, setutent, endutent, utmpname	Accesses a utmp file entry (getut(3c))
iargc	Returns the number of command-line arguments (iargc(3f))
index	Returns the location of a FORTRAN substring (index(3f))
inittab	Script for the init process (inittab(4))
issue	Issues an identification file (issue(4))
jagent	Host control of windowing terminal (jagent(5))
layers	Protocol used between host and windowing terminal under layers(layers(5))
ldahread	Reads the archive header of a member of an archive file (ldahread(3x))
ldclose, ldaclose	Closes a common object file (ldclose(3x))
ldfcn	Common object file access routines (ldfcn(4))
ldfhread	Reads the file header of a common object file (ldfhread(3x))
ldlread, ldlininit, ldlitem	Manipulates line number entries of a common object file function (ldlread(3x))

ldlseek, ldnlseek	Seeks to line number entries of a section of a common object file (ldlseek(3x))
ldohseek	Seeks to the optional file header of a common object file (ldohseek(3x))
ldopen, ldaopen	Opens a common object file for reading (ldopen(3x))
ldrseek, ldnrseek	Seeks to relocation entries of a section of a common object file (ldrseek(3x))
ldsgetaname	Retrieves the file name of a member of an archive file (ldsgetaname(3x))
ldsgetname	Retrieves the section name for a common object file entry entry (ldsgetname(3x))
ldsgetstring	Retrieves a string from the common object file string table (ldsgetstring(3x))
ldshread, ldnsread	Reads an indexed/named section header of a common object file (ldshread(3x))
ldsseek, ldnsseek	Seeks to an indexed/named section of a common object file (ldsseek(3x))
ldtbindx	Computes the index of a symbol table entry of a common object file (ldtbindx(3x))
ldtbread	Reads an indexed symbol table entry of a common object file (ldtbread(3x))
ldtbseek	Seeks to the symbol table of a common object file (ldtbseek(3x))
len	Returns the length of a FORTRAN string (len(3f))
limits	File header for implementation-specific constants (limits(4))
linenum	Line number entries in a common object file (linenum(4))
lockf	Record locking on files (lockf(3c))
log, alog, dlog, clog	FORTRAN natural logarithm intrinsic function (log(3f))
log10, alog10, dlog10	FORTRAN common logarithm intrinsic function (log10(3f))
malloc, free, realloc, calloc, mallopt, mallinfo	Fast main memory allocator (malloc(3x))
master	Master configuration database (master(4))
max, max0, amax0, max1, amax1, dmax1	FORTRAN maximum-value functions (max(3f))
mclock	Returns FORTRAN time accounting (mclock(3f))
mil: ior, iand, not, ieor, ishft, ishftc, ibits, btest, ibset, ibclr, mvbits	FORTRAN Military Standard functions (mil(3f))

min, min0, amin0, min1, amin1, dmin1 FORTRAN minimum-value functions (**min(3f)**)

mmap, msync, munmap, mprotect, madvise, mset, mclear Berkeley's proposed memory management system calls. The implementation is based on proposals published by Berkely. These calls are provided in Domain/OS as extensions to the SysV environment. (**mmap(3)**)

mod, amod, dmodFORTRAN remaindering intrinsic functions (**mod(3f)**)

monitor Prepares execution profile (**monitor(3c)**)

nlist Gets entries from the name list (**nlist(3c)**)

pnch File format for card images (**pnch(4)**)

poll SysV STREAMS facility input/output multiplexing (**poll(2)**)

prof Profile within a function (**prof(5)**)

profil Execution time profile (**profil(2)**)

putmsg Sends a message on a stream (**putmsg(2)**)

rand, irand, srand Random number generator (**rand(3f)**)

reloc Relocation information for a common object file (**reloc(4)**)

rename Renames a file; this call now will replace the target if it is a file; at SR9.7, rename would not replace an existing target file. (**rename(3f)**)

rfmaster Remote File Sharing name server master file (**rfmaster(4)**)

round: anint, dnint, nint, idnint FORTRAN nearest integer functions (**round(3f)**)

scnhdr Section header for a common object file (**scnhdr(4)**)

scr_dump Format of curses screen image file (**scr_dump(4)**)

setoid, seteoid, setroid Organization ID (**setoid(3)**)

setreoid Sets the real and effective organization ID (**setreoid(2)**)

sign, isign, dsign FORTRAN transfer-of-sign intrinsic function (**sign(3f)**)

signal Specifies FORTRAN action on receipt of a system signal (**signal(3f)**)

sigset, sighold, sigrelse, sigignore, sigpause Signal management (**sigset(2)**)

sin, dsin, csin FORTRAN sine intrinsic function (**sin(3f)**)

sinh, dsinh FORTRAN hyperbolic sine intrinsic function (**sinh(3f)**)

sio Serial I/O (**sio(4)**)

sqrt, dsqrt, csqrt FORTRAN square root intrinsic function (**sqrt(3f)**)

statfs, fstatfs	Gets file system information (statfs(2))
stdio	Standard buffered input/output package (stdio(3s))
stime	Sets the time (stime(2))
strcmp: lge, lgt, lle, llt	String comparison intrinsic functions (strcmp(3f))
syms	Common object file symbol table format (syms(4))
system	Issues a shell command from FORTRAN (system(3f))
system	System configuration information table (system(4))
t_accept	Accepts a connect request (t_accept(3n))
t_alloc	Allocates a library structure (t_alloc(3n))
t_bind	Binds an address to a transport endpoint (t_bind(3n))
t_close	Closes a transport endpoint (t_close(3n))
t_connect	Establishes a connection with another transport user (t_connect(3n))
t_error	Produces an error message (t_error(3n))
t_free	Frees a library structure (t_free(3n))
t_getinfo	Gets protocol-specific service information (t_getinfo(3n))
t_getstate	Gets the current state (t_getstate(3n))
t_listen	Listens for a connect request (t_listen(3n))
t_look	Looks at the current event on a transport endpoint (t_look(3n))
t_open	Establishes a transport endpoint (t_open(3n))
t_optmgmt	Manages options for a transport endpoint (t_optmgmt(3n))
t_rcv	Receives data or expedited data sent over a connection (t_rcv(3n))
t_rcvconnect	Receives the confirmation from a connect request (t_rcvconnect(3n))
t_rcvdis	Retrieves information from a disconnect request (t_rcvdis(3n))
t_rcvrel	Acknowledges receipt of an orderly release indication (t_rcvrel(3n))
t_rcvudata	Receives a data unit (t_rcvudata(3n))
t_rcvuderr	Receives a unit data error indication (t_rcvuderr(3n))
t_snd	Sends data or expedited data over a connection (t_snd(3n))
t_snddis	Sends a user-initiated disconnect request (t_snddis(3n))
t_sndrel	Initiates an orderly release (t_sndrel(3n))
t_sndudata	Sends a data unit (t_sndudata(3n))
t_sync	Synchronizes the transport library (t_sync(3n))

t_unbind	Disables a transport endpoint (t_unbind(3n))
tan, dtan	FORTTRAN tangent intrinsic function (tan(3f))
tanh, dtanh	FORTTRAN hyperbolic tangent intrinsic function (tanh(3f))
times	The times call now reports 50% of cpu time as user time and 50% of cpu time as system time. It previously reported all cpu time as user time. (times(2))
timezone	Sets the default system time zone (timezone(4))
ttyslot	Finds the slot in the utmp file of the current user (ttyslot(3c))
ulimit	Gets and sets user limits (ulimit(2))
umount	Unmounts a file system (umount(2))
unistd	File header for symbolic constants (unistd(4))
ustat	Gets file system statistics (ustat(2))
xtproto	Multiplexed channels protocol used by xt(7) driver (xtproto(5))
putpwent	Writes password file entry (putpwent(3c))

A.2.2 BSD System Calls, Subroutines, and File Formats

The following subsections list the BSD calls that are obsolete at SR10, as well as the features that have been added or changed since SR9.7.

A.2.2.1 Obsolete or Unsupported Calls

We do not support the **crypt(3)** call due to export restrictions.

The following system calls are no longer required and are obsolete. At SR10 they are supplied as no-ops. Apollo does not commit to support these calls in future software releases.

default_acl	This call is no longer required. At SR10 it is a noop. (default_acl(2))
set_sbrk_size	This call is no longer required. At SR10 it is a noop. (set_sbrk_size(2))
soft_link	This call identical to symlink fR(2) . both now support soft links by using textual links. (soft_link(2))

A.2.2.2 New and Changed Features

The following system calls, subroutines, file formats, and miscellaneous features are supported in SR10 BSD that were not in SR9.7 and earlier Domain/IX 4.2BSD, or are significantly changed since SR9.7.

abort	Abnormal termination (abort(3f))
access	Determines the accessibility of a file (access(3f))
acct	Execution accounting file (acct(5))
alarm	Executes a subroutine after a specified time (alarm(3f))
asinh, acosh, atanh	Inverse hyperbolic functions (asinh(3m))
bessel functions: j0, j1, jn, y0, y1, yn	These are of two kinds for integer orders (bessel(3f))
bit	The and, or, xor, not, rshift, lshift bitwise functions (bit(3f))
chdir	Changes the default directory (chdir(3f))
chmod	Changes the mode of a file (chmod(3f))
cons	Console interface (cons(4))
ctio	Cartridge tape I/O (ctio(4))
ctime, localtime, gmtime, asctime, timezone	Convert date and time to ASCII (ctime(3))
dbx	The dbx symbol table information (dbx(5))
disktab	Disk description file (disktab(5))
dump, dumpdates	Incremental dump format (dump(5))
erf, erfc	Error functions (erf(3m))
etime, dtime	Returns elapsed execution time (etime(3f))
exit	Terminates process with status (exit(3f))
fdate	Returns the date and time in an ASCII string (fdate(3f))
fl	Console floppy interface (fl(4))
fmin, fmax, ffrac, dfmin, dfmax, dfrac, inmax	Returns extreme values (fmin(3f))
flush	Flushes output to a logical unit (flush(3f))
fork	Creates a copy of this process (fork(3f))
fseek, ftell	Repositions a file on a logical unit (fseek(3f))
getarg, iargc	Returns command line arguments (getarg(3f))
getc, fgetc	Gets a character from a logical unit (getc(3f))
getcwd	Gets the pathname of the current working directory (getcwd(3f))
getdents	Reads directory entries and puts them in a file (getdents(2))
getenv	Gets the value of environment variables (getenv(3f))
getfsent, getfsspec, getfsfile, getfstype, setfsent, endfsent	Gets a file system descriptor file entry (getfsent(3))

getlog Gets a user's log-in name (**getlog(3f)**)

getoid, geteoid Gets the organization identity (**getoid(2)**)

getopt Gets the option letter from argv (**getopt(3)**)

getorgent, getorgid, getorgnam, setorgent, endorgent Gets the organization file entry (**getorgent(3)**)

getpid Gets the process ID (**getpid(3f)**)

getttyent, gettyenam, setttyent, endttyent Gets the tty's file entry (**getttyent(3)**)

gettytab Terminal configuration data base (**gettytab(5)**)

getuid, getgid Gets the user or group ID of the caller (**getuid(3f)**)

getusershell, setusershell, endusershell Gets legal user shells (**getusershell(3)**)

hostnm Gets the name of the current host (**hostnm(3f)**)

icmp Internet Control Message Protocol (**icmp(4p)**)

idate, itime Returns the date or time in numerical form (**idate(3f)**)

copysign, drem, finite, logb, scalb copysign, remainder, (ieee(3m))

index, rindex, lnlblk, len Tells about character objects (**index(3f)**)

ioinit Changes f77 I/O initialization (**ioinit(3f)**)

ip Internet Protocol (**ip(4p)**)

kill Sends a signal to a process (**kill(3f)**)

ldahread Reads the archive header of a member of an archive file (**ldahread(3x)**)

ldclose, ldaclose Closes a common object file (**ldclose(3x)**)

ldfhread Reads the file header of a common object file (**ldfhread(3x)**)

ldlread, ldlinit, ldlitem Manipulates line number entries of a common object file function (**ldlread(3x)**)

ldlseek, ldlseek Seeks to line number entries of a section of a common object file (**ldlseek(3x)**)

ldohseek Seeks to the optional file header of a common object file (**ldohseek(3x)**)

ldopen, ldaopen Opens a common object file for reading (**ldopen(3x)**)

ldrseek, ldrseek Seeks to relocation entries of a section of a common object file (**ldrseek(3x)**)

ldsgetaname Retrieves the file name of a member of an archive file (**ldsgetaname(3x)**)

ldsgetname	Retrieves the section name for a common object file entry entry (ldsgetname(3x))
ldsgetstring	Retrieves a string from the common object file string table (ldsgetstring(3x))
ldshread, ldnshread	Reads an indexed/named section header of a common object file (ldshread(3x))
ldsseek, ldnsseek	Seeks to an indexed/named section of a common object file (ldsseek(3x))
ldtbindex	Computes the index of a symbol table entry of a common object file (ldtbindex(3x))
ldtbread	Reads an indexed symbol table entry of a common object file (ldtbread(3x))
ldtbseek	Seeks to the symbol table of a common object file (ldtbseek(3x))
link	Makes a link to an existing file (link(3f))
loc	Returns the address of an object (loc(3f))
long, short	Integer object conversion (long(3f))
lp	Line printer (lp(4))
madvise	One of Berkeley's proposed memory management system calls. The implementation is based on proposals published by Berkely. (madvise(3))
malloc, free, falloc	Memory allocator (malloc(3f))
map3270	Database for mapping ASCII keystrokes into IBM 3270 keys (map3270(5))
mkdir	The mechanism for assigning the initial file ACL and initial directory ACL to the directory created by the mkdir command has been changed. The mkdir now works identically to the /com/crd command; that is, the initial file ACL and initial directory ACL are now inherited from the parent directory. (mkdir(2))
mmap, msync, munmap, mremap	Berkeley's proposed memory management system calls. The implementation is based on proposals published by Berkely. (mmap(3))
monitor, monstartup, moncontrol	Prepares execution profile (monitor(3)) Multiple precision integer arithmetic (mp(3x))
mprotect	One of Berkeley's proposed memory management system calls. The implementation is based on proposals published by Berkely. (mprotect(3))
mset, mclear	Two of Berkeley's proposed memory management system calls. The implementation is based on proposals published by Berkely.

(mset(3))

dbm_open, dbm_close, dbm_fetch, dbm_store, dbm_delete, dbm_firstkey,
dbm_nextkey, dbm_error, dbm_clearerr Database subroutines
(ndbm(3))

nlist Gets entries from name list (nlist(3))

perror, gerror, ierrno Gets system error messages (perror(3f))

plot: openpl et al. The f77 library interface to plot (3X) (plot(3f))

profil Execution time profile (profil(2))

putc, fputc Writes a character to a FORTRAN logical unit (putc(3f))

qsort Quick sort (qsort(3f))

rand, drand, irand Returns random values (rand(3f))

random, drandm, irandm Better random number generator (random(3f))

rename Renames a file (rename(3f))

res_mkquery, res_send, res_init, dn_comp, dn_expand Resolver routines (resolver(3))
(resolver(5))

setoid, seteoid, setroid Organization ID (setoid(3))

setreoid Sets real and effective organization ID (setreoid(2))

siginterrupt Allows signals to interrupt system calls (siginterrupt(3))

signal Changes the action for a signal (signal(3f))

sigreturn Returns from signal (sigreturn(2))

sio Serial I/O (sio(4))

sleep Suspends execution for an interval (sleep(3f))

cbirt, sqrt Cube root, square root (sqrt(3m))

stat, lstat, fstat Gets file status (stat(3f))

statfs, fstatfs Gets file system information (statfs(2))

stdio Standard buffered input/output package (stdio(3s))

symlink This call now supports soft links by using textual links. (symlink(2))

syslog, openlog, closelog, setlogmask Controls system log (syslog(3))

system Executes a UNIX command (system(3f))

time, ftime Returns system time (time(3c))

times The times call now reports 50% of cpu time as user time and 50%
of cpu time as system time. It previously reported all cpu time as
user time. (times(3c))

<code>topen, tclose, tread, twrite, trewin, tskipf, tstate</code>	f77 tape I/O (<code>topen(3f)</code>)
<code>tp</code>	Mag tape formats (<code>tp(5)</code>)
<code>traper</code>	Traps arithmetic errors (<code>traper(3f)</code>)
<code>trapov</code>	Traps and repairs floating-point overflow (<code>trapov(3f)</code>)
<code>trpfpe, fpecnt</code>	Traps and repairs floating-point faults (<code>trpfpe(3f)</code>)
<code>ttynam, isatty</code>	Finds the name of a terminal port (<code>ttynam(3f)</code>)
<code>ttys</code>	Terminal initialization data (<code>ttys(5)</code>)
<code>ualarm</code>	Schedules a signal after a specified time (<code>ualarm(3)</code>)
<code>unlink</code>	Removes a directory entry (<code>unlink(3f)</code>)
<code>usleep</code>	Suspends execution for a time interval (<code>usleep(3)</code>)
<code>va</code>	Benson-Varian interface (<code>va(4)</code>)
<code>vgrindefs</code>	vgrind's language definition database (<code>vgrindefs(5)</code>)
<code>vhangup</code>	Virtually "hangs up" the current control terminal (<code>vhangup(2)</code>)
<code>wait</code>	Waits for a process to terminate (<code>wait(3f)</code>)

A.2.3 Domain/OS System Calls

At SR10, several calls and interfaces have been superseded.

A.2.3.1 The `name_$` and `ios_$` Interfaces

The `name_$name_t` and `name_$pname_t` data types have been superseded to support long names. The `name_$long_name_t` supports leaf names up to 255 characters long (not including a null terminator), and `name_$long_pname_t` supports pathnames up to 1023 characters long (not including a null terminator). In order to use the new longer names without breaking existing code, calls that supplied the old short names and did not accept buffer length arguments have been superseded by new calls with "_lc" suffixes added to their names. The superseded calls are still supported at SR10, but may not be available in future releases. The following table summarizes the changes.

Interface Changes to Support Long Pathnames	
Superseded Calls	Replacement Calls
ios \$get_dir	ios \$get_dir_lc
ios \$inq_path name	ios \$inq_path name_lc
name \$extract_data	name \$extract_data_lc
name \$get_ndir name \$get_ndir cc	name \$get_ndir_lc
name \$get_path name \$get_path cc	name \$get_path_lc
name \$get_wdir name \$get_wdir cc	name \$get_wdir_lc
name \$read_dir name \$read_dir cc	name \$read_dir_lc
name \$read_link	name \$read_link_lc

The new default file system object type is **unstruct**, an unstructured file without a streams header, rather than **uasc**, an unstructured ASCII file with a streams header. That means, if **ios_\$create** is called to create an object of type **uid_\$nil**, it will result in creating an object of type **unstruct**.

The **ios** facilities can open and access files that do not have type managers, as long as the node that is opening the file has the type listed. The **ios** facilities will then treat the file as an unstructured type.

The **ios_\$errin** and **stream_\$errin** constants are not supported in COFF programs.

The interface, including the **ios_\$read** and **ios_\$write** calls, is now interruptable.

A.2.3.2 New prf_\$ Calls

The following new **prf** calls allow you to program with the new Domain print architecture. For more information on the SR10 Domain print architecture, see *Printing in the Domain Environment*. *Printing in the Domain Environment* is not available at Beta1.

- prf_\$edit_job** Edits a print job
- prf_\$get_printers** Returns a list of printers on the network or by specified print manager
- prf_\$get_sites** Returns a list of print managers
- prf_\$read_queue** Gets a list of print jobs in the queue
- prf_\$signal_printer** Signals a printer to stop, suspend, or continue a job

A.2.3.3 New loader_\$ Calls

The following **loader_\$** calls are new at SR10.

loader_\$load
loader_\$cond_load
loader_\$inlib
loader_\$clear_inlib
loader_\$kg_lookup
loader_\$kg_define
loader_\$lookup_start_addr
loader_\$lookup_stack_size
loader_\$inquire_image
loader_\$inquire_section
loader_\$inquire_stat
loader_\$inquire_known_lib

A.2.3.4 Changes to gpr_\$ Calls

The gpr_\$ calls now support both 7-bit and 8-bit fonts. This support does not change the call interfaces and does not require any change in existing programs.

A.2.3.5 changes to pad_\$ Calls

Two new pad_\$ calls are being released at SR10, pad_\$isa, which determines whether a stream supports pad operation, and pad_\$isa_dm_pad which determines whether the stream is open on a local DM pad. The pad_\$isa_dm_pad call is particularly useful in determining whether it is appropriate to pop up a pad, in cases where a program may be run from a process that is crped onto a remote node and might pop a pad on the remote node.

A new pad_\$bw5_1024x800 display type has been added to the pad_\$inq_disp_type call to support the small monochrome monitor on DN3000 and DN4000 workstations.

A.2.3.6 Miscellaneous New and Changed Calls and Insert Files

A new ec2_\$wait_slow_io call has been added that behaves like ec_\$wait_svc except that it recognizes when a signal has been designated to terminate active I/O via the BSD sigvec call. This call is provided for compatibility with 4.3BSD. Use it for a serial I/O device, for example in a manager, or wherever an indefinite wait can occur that should be interruptible by a signal.

A new osinfo_\$get_rev call returns the information contained in the Domain/OS build string. This information identifies the date, revision and version of the OS

The mbx_\$start_helper call is released at SR10.

The old 7-bit font format has been removed from the insert files for the `smdu` interface. There is a new set of insert and header files for the 8-bit format fonts; these files are `/sys/ins/fontn.ins.pas` `/sys/ins/fontn.ins.ftn` `/usr/include/apollo/fontn.h`.

A.2.3.7 Superseded Interfaces

As of SR10, the `smd_$` and `streams_$` interfaces have been superseded. The `ios_$` and `ios_dir_$` interfaces supersede the `streams_$` calls, and the functions provided by the `smd_$` calls are now better implemented by using `pad_$` and `graphics` calls.

A.3 Changes to `/systest` and `/sys`

The `ts` command has been moved from the `/systest/ssr_util/` directory to the `/usr/apollo/bin` directory.

`/systest/ssr_util/fmpd` is now obsolete. All of its functionality is now reproduced by `/usr/apollo/tb`.

`ctboot` has been moved from the `/sys` directory to the `/systest` directory.

