Domain System Software Release Notes

Software Release 10.1
Order No. 005809-A03

Please read all of Chapter 7
before installing SR10.1.

Restricted Rights Notice

Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824
(508) 256-6600

First Printing: December 1988

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

Apollo Computer Inc. reserves the right to make changes in specification 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

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.

Preface

This document describes standard Domain® System software for Domain/OS Software Release 10.1 (SR10.1). For the benefit of customers who may be upgrading to SR10.1 from SR9.7, this document also discusses SR10. Customers already running SR10 may want to be selective in their reading:

Chapter 1 "An Overview of Software Release 10." Summarizes the major changes between SR10 and previous releases of Domain System software.

Chapter 2 "An Overview of Software Release 10.1." Summarizes the changes between SR10 and SR10.1.

Chapter 3 "Software and Hardware Compatibility." Discusses compatibility between SR10.1 and versions of software and hardware prior to SR10.

Chapter 4 "Bugs Fixed by SR10."

Chapter 5 "Bugs Fixed by SR10.1."

Chapter 6 "Limitations and Known Bugs." Discusses limitations and known bugs in SR10.1.

Chapter 7 "Installing SR10.1."

Chapter 8 "Documentation." Discusses changes in documentation relative to releases prior to SR10.

Appendix A "Changes to Commands and Calls." Lists the commands and system calls that have been added, changed, and deleted since SR9.7, relative to SR10.

Appendix B "Making the Transition to SR10 Operating System Releases." Contains changes, corrections and additions to *Making the Transition to SR10 Operating System Releases*.

The normal software installation process places a version of these release notes in each node's **/install/doc/apollo** directory. Release notes for optional Domain software are installed in a similar manner. If you are upgrading from a previous software release to SR10, be sure to read *Making the Transition to SR10 Operating System Releases* (Order No. 011435) before installing SR10.1.

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 SysV **lp** print daemon, use an **lp** command similar to the following:

lp –d*printer_name pathname*

where *pathname* is the pathname of the release notes, usually **/install/doc/apollo/os.v10.1__notes** (note that there are *two* underscores before **notes**).

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

prf *pathname* **–pr** *printer_name* **–npag**

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

lpr –P*printer_name pathname*

CONTENTS

List of Figures

List of Tables

Chapter 1: An Overview of Software Release 10

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 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. However, the Domain/OS kernel and system libraries are common to all environments, so you can use Domain/OS calls such as **mbx_$open** in programs that will run in a Domain/OS UNIX environment.

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

* UNIX is a Registered Trademark of AT&T in the USA and other countries.

- TCP/IP is now standard in all environments

This chapter 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 SR10.1, be sure to read *Making the Transition to SR10 Operating System Releases* before installing new software.

This chapter briefly describes changes made to Domain/OS system software. It 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 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 who wants 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 when installing across a network. 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_**compat** 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 but are not being re-released with SR10. The names of the scripts to use are given in in *Installing Software with Apollo's Release and Installation Tools*, while the procedures to be followed are described in *Installing Domain Software* (Part No. 008869 Rev. 02).

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**, generates an Apollo Product Report and 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 subsections 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."
- 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 log-in 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 depend 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.

Each node, disked or diskless, has a **`node_data/etc** directory that contains files that are node-specific. The **/etc** directory entries (such as **/etc/ttys**) are links to these files.

1.3.3 The /usr/apollo Directory

We now install a **/usr** directory in all environments; 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 C include files for Domain system calls, as well as other proprietary files.

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

/usr/apollo/bin This directory contains Domain commands that apply to all three environments or extend the UNIX environment. It includes commands that were previously in the **/com** directory but are also applicable to the UNIX environments, new commands that apply to all three environments, and commands that are specific to the UNIX environments under Domain/OS. Commands that were in the **/com** directory in previous releases are still available in that directory, usually through links. The **/usr/apollo/bin** directory 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 Guaranteed Command Set

All systems, including nodes that do not have a SysV environment installed, have a **/sys5.3/bin** directory. On systems that have only BSD or Aegis, this directory contains the following commands:

cat	**chgrp**	**chmod**	**chown**	**cmp**
cpio	**diff**	**expr**	**find**	**grep**
ln	**ls**	**mkdir**	**mv**	**rm**
sed	**sort**	**sum**	**tar**	**uniq**

In addition, we always supply a SysV Bourne shell in **/etc/sys_sh**. These files enable solution suppliers to provide a single (SysV) script that will correctly install software on all nodes, independent of the installed environments.

1.3.5 Case Sensitivity

The OS kernel and libraries are now fully case-sensitive in handling pathnames. Pathname text is now maintained case-correct, and we no longer use the colon escape character to indicate uppercase characters. This full case sensitivity eliminates the problems that users encountered when operating in both the Aegis and UNIX environments.

The OS automatically converts between colon-mapped names on pre-SR10 systems and case-correct names on SR10 nodes, and we provide additional conversion tools, which are described in *Making the Transition to SR10 Operating System Releases*.

1.3.6 Key Definitions

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

1.3.6.1 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*.

1.3.6.2 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.)

1.3.6.3 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 high 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 multinational long and short shift keys.

1.3.7 Font Support

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

1.3.7.1 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
Standard family

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

Because users may want 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.

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

- Fixed-spaced 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, (that is, 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.

1.3.7.2 Font Height

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

1.3.7.3 The edfont Utility and Icons

We have rewritten the **edfont** utility to work with *n*-bit fonts and have changed the user interface. The **edfont** utility now checks and uses all font characteristics, while the old version did not. As a result, **edfont** can now notice some errors that existed in icon files generated before SR10, and may not appear to display some old icons correctly.

1.3.8 Initialization, Login, Shell Startup

The node initialization, log-in, 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 an **/etc/environ** file, while a **.environ** file in the HOME or directory can specify 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 log-in program such as **/com/login, siologin, crp** (without the **–me** option). More specifically (for programmers) the shell can be made to run its login script 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 **-D***variable*[=*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 or program called **ld** earlier in your search path than /**com**/**ld**, and that object 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 Commands in /com Requiring –p Option

Several commands in the /**com** directory that manipulate files now require a –**p**[**airwise**] option if you specify more than two file tokens. This option was added to provide a measure of protection against inadvertent uses of these commands in UNIX shells, which expand wildcards *before* executing the command. Thus, for example, if you used **cpf** in a UNIX shell with a wildcard in the source name, and the source name expanded to three pathnames, you would copy the first file onto the second source file and lose the second file. The requirement for a –**p**[**airwise**] option makes this kind of error harder to make.

At SR10, the use of parentheses to group file names (such as **cpf** (**foo bar**) /**dir**) also requires the –**p**[**airwise**] option, just as the specification of more than one pair of tokens does. The commands themselves do the checking, rather than the command-line interpreter, since only some commands can have bad results with multiple pairs of tokens. Since a leading "(" can be part of a name, it would invalidate this protection to accept "(" as syntax without "permission" from the user in the form of -**p**[**airwise**].

1.3.11 Korn Shell

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

1.4 System Administration

The following changes primarily affect the ways in which systems and networks are administered.

1.4.1 OS Name and the MD

The operating system is now called **Domain/OS**, not Aegis. As a result, the OS file in the /sau*n* directories has been changed to **domain_os** You can now start the operating system from the Mnemonic Debugger (MD) by using the following command:

```
EX DOMAIN_OS
```

Note, however, that DN5xx-T systems with FPX boards and with MD revisions earlier than 5.7 must still use **EX AEGIS**.

1.4.2 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 run time, and whether the library is optional (that is, whether the system should initialize successfully if the library is not found).

The **/lib/unixlib** library is obsolete at SR10. In addition, the **/lib/tblib** library is provided only for Ada compatibility.

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.

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

The **/lib** directory also contains a link to the C preprocessor program.

We also supply the following 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**, 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 **prflib** library should replace the standard SR9.7 **prflib**.
- The **streams** library fixes a bug encountered when an SR9.7 system uses temporary files on an SR10 node.
- The **ddslib** library enables SR9.7 nodes to run the **rgyd**; this library should be installed on nodes that use NCS-based software to communicate with SR10 systems.

1.4.3 Paging Statistics

We have made some changes to the way we count paging activity at SR10. This information is visible through the **/com/pst** and **/bin/ps** commands.

At SR9.7, the kernel made two per-process counters available:

global faults Counted 1 for each page fault occurring in global space

private faults Counted 1 for each page fault in private address space

Neither of these counters included adjustments for "touch-ahead" pages. Also a special "sharing fault" path (page resident but not in the user's mmu tables) through the kernel did not increment these counts. As a result, the counters were not consistent, accurate, or meaningful measurements of paging activity.

At SR10, we have changed the contents and meaning of these counters. They now represent the following:

data file faults A count of the number of pages read from disk or network that are not executable. It includes "touch-ahead" pages, but does not include zero-filled (newly created) pages. This number is roughly equivalent to file I/O.

executable faults A count of the number of pages read from disk or network that are executable (have the "x" access bit on).

Sharing faults (pages that are resident, but not in the mmu tables for this process) are not counted in either case.

1.4.4 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 "Printer Support" for more information.

1.4.5 UNIX Devices

The UNIX device name for any transcript pad created by the DM is now the same as for the default pad name and the process name for the shell process. For example, if you use the default DM SHELL key to create a shell, the shell process and transcript pad name will be something like **pad06**. In this case, any output written to **/dev/pad06** will appear in the shell's transcript pad.

1.4.6 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/sio***x* files and the corresponding **/dev/tty***xx* files refer to the same devices, except that the state of the

DCD line is always ignored on open by **/dev/sio***x* devices. For a detailed description of this change and its effects, see *Making the Transition to SR10 Operating System Releases*.

The SR10 SIO software initializes the **/dev/sio***x* devices to settings different than those used in previous releases. Use the **–default** option of the **tctl** command to obtain the pre-SR10 settings. For example, to set SIO line 3 to the pre-SR10 settings, do this:

```
tctl –default –line 3
```

1.4.7 Handling Devices and Mounting

Both SR10 UNIX environments support handling devices in the **/dev** directory and building special files with the **mknod** command.

1.4.8 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 Access Control List (ACL) entries of the exact form *%.group_name.%.%*. At SR10, the project list check substitutes each group 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.

1.4.9 Process Accounting

We provide local process accounting with **/etc/sa** and **/etc/accton** in all environments. The online manual pages provide details.

1.4.10 Log-In Accounting

We provide facilities to log log-in 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.

1.4.11 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, and **tar** now supports blocking factors of 1 through 20. The performance of **tar** and **tfp** has also been improved.

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.)

The **rbak** tool will no longer restore protected subsystems or setuid objects unless the user has permissions to do so.

1.4.12 DPSS/Mail to UNIX mail and UUCP

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.

1.4.13 SPM

At SR10, you can configure a system so that the Server Process Manager (SPM) prevents unauthorized users from creating processes on or logging in to a node. To do so, edit 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. The /sys/**spm/spm** program also has two new options, **-hi[gh]** and **-lo[w]**, for specifying process priority.

1.4.14 The netman Program 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. The /sys/**net/netman.rc** directory entry 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. You will find that the first diskless boot from a given partner takes longer than previously, due to the time required to create the supporting files.

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

1.4.15 The ns_helper Server

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.

1.4.16 The `node_data Directory

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

1.4.17 Domain Internet Support

The maximum number of hops allowed in a Domain internet has been increased to 16, 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.

1.4.18 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.

1.4.19 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.

1.4.20 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 SysV **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.

1.4.21 The /usr/new and /usr/games Directories

The /usr/**new** directory contains unsupported miscellaneous tools and commands from the Berkeley 4.3BSD distribution. The /usr/**games** directory contains games from the Berkeley 4.3BSD and System V distributions, as well as games that are specific to Apollo systems. You can optionally install either or both directories when you install either the BSD or SysV environment. Each is then put in a single directory under /**usr** (so that the pathname does not include a systype).

1.4.22 DOCUMENTER'S WORKBENCH

Both UNIX environments of SR10 Domain/OS support DOCUMENTER'S WORKBENCH*, 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 SysV software; the BSD environment still includes all tools that are part of the Berkeley distribution.

1.5 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. All characters except slash (/) and nul are now allowed in component names.

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)
.. (dot-dot, 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 ".." (by itself) or "../" (as one component in a pathname) instead. However, you cannot use *filename*/.. to access the parent directory of a file (for example if a file pathname has been passed as an argument); the dot-dot convention only works relative to directories.

1.6 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 (**pwrxk**), including a new right (**k** — keep) that allows you to specify an object as not deletable or renameable, 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 ACL

* DOCUMENTER'S WORKBENCH is a trademark of AT&T.

entries, which are essentially like pre-SR10 ACL entries.

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, including down to the differences in protection inheritance between 4.3BSD and SysV.

Pre-encoded or "canned" ACL entries, including the **%.backup.%** ACL entry, 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.

1.6.1 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.

1.7 The Registry

The SR10 registry is based on Apollo Network Computing System™ (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 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.

1.8 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 standard stream numbering
- Data alignment support in compilers
- Changes in symbolic link support
- Changes in the **malloc** and **rws_$alloc** 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.

1.8.1 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.

1.8.2 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, support for in-process program invocation will be removed in the next major release. 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 and set to TRUE.

1.8.3 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*.

1.8.4 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.

1.8.5 Names with Blanks and Special Characters

Pathname components can now include all characters except slash (/) and nul; blanks are now valid name characters. This may affect programs that use the Domain/OS **vfmt_$** calls. In particular, code that uses the **vfmt_$ %a** directive without the z option will normally ignore trailing blanks.

1.8.6 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.

1.8.7 Standard Streams

At SR10, both Aegis and UNIX COFF programs are invoked with three file descriptors, or streams, open; 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*.

1.8.8 Directories

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

1.8.9 System V STREAMS and TI

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 Interface (TI), 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 Interface* for detailed information on System V STREAMS and TI.

1.8.10 UNIX Process Timing

The UNIX **times** call and commands such as **time**(1), **sa**(8), and SysV **timex** that use this call now report separately the time spent in user and supervisor mode. They previously reported all CPU time as user time. Note, however, that the division of UNIX processing between user and supervisor mode varies among manufacturers, and many of our system calls have a significant execution component in the global libraries.

1.8.11 The 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.

1.8.12 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.

1.8.13 The malloc() and rws_$alloc Calls

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

We have also provided a module to help in debugging **malloc** calls. The **/usr/apollo/lib/libmalloc.dbg** module 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 it is 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**.

1.8.14 UNIX scanf Routine

The UNIX **scanf** routine now conforms to the SVID and POSIX specifications; as a result, if a field is blank (that is, there are two sequential delimiters) the remainder of the pattern is not matched, and the remaining fields will be returned as zero or null.

1.8.15 The C Math Library

The standard C math library (**math.h**) routines that we provide in **/lib/libc** are now SVID-compliant implementations of the System V Release 3 math library routines. As a result, C math library routines may execute more slowly than they have at previous base levels. You can increase the performance of many of these routines by using the in-line versions that are provided when you specify **–D_BUILTINS** in the compiler command line, which includes the **<h.builtins>** header file in the code. However, the in-line code does not guarantee SVID compliance under all error conditions.

1.8.16 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.

1.8.17 The 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 when the routine that established the handler exits.

1.8.18 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 they 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 in-line code, as described in Section 1.8.15, "The C Math Library."

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 documents for these products.

1.8.19 A New Debugger

SR10 replaces DEBUG with a new debugger, the Domain Distributed Debugging Environment, which can debug both obj and COFF format objects. This new debugger 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 Domain/Debug™ and **dbx** syntax. These macros are located in **/sys/debug/old_debug_macros** and **/sys/debug/dbx_macros**, respectively.

The new debugger cannot debug programs built before SR9.5. For full details see the *Domain Distributed Debugging Environment Reference.*

1.8.20 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/lbr** or **/bin/ar**). The binder will not handle any object files compiled prior to SR10.
- The **bind** utility supports both a **–runtype** option to control the UNIX run-time system call semantics and a **–systype** option to control pathname semantics. See the *Domain/OS Programming Environment Reference* for details on the options.
- At SR10, you must use the **–allocbss** 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 **–allocbss** 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 256 KB.)

1.8.21 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/bind** or **/bin/ld**), 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.

1.9 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). In the Domain/OS BSD environment, communication between an SR10 client (**lp**, **lpc**, etc.) and an SR10 server (**lpd**) is accomplished by using the Network Computing System (NCS).

The following subsections 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.

1.9.1 Client/Server Communications

In the Domain/OS BSD environment, communication between an SR10 client (**lp**, **lpc**, etc.) and an SR10 server (**lpd**) is accomplished by using the Network Computing System (NCS). Therefore, in order to run **lpd**, a local location broker daemon (**/etc/ncs/llbd**) must be running on the workstation running **lpd**. To enable these daemons at boot time, create the empty files /etc/**daemons/lpd** and /etc/**daemons/llbd**. Shut down the workstation and reboot. Given that the files /**usr/lib/lpd** and /etc/ncs/**llbd** exist and are executable, they will be started as daemons by /etc/rc.

In a mixed network, you must run **lpd** on both an SR9.7 and an SR10 workstation to enable users to print from either operating system because, at SR9.7, **lpd** communicates with its clients using TCP/IP. The SR10 workstation running **lpd** must also run **tcpd** and **inetd** in order to communicate with the SR9.7 **lpd**. To enable these daemons at boot time, create the empty files /etc/**daemons/tcpd** and /etc/**daemons/inetd** and shut down and reboot.

To enable users of workstations running SR9.7 to use printers attached to SR10 workstations, and vice-versa, the /etc/**printcap** entry of the respective workstations running **lpd** must treat their counterpart as a remote machine. For example, the following **printcap** fragments would allow **lpd** on a workstation running SR9.7 to support a line printer connected to /**dev/sio1** and a laser printer connected to an SR10 workstation.

```
#/bsd4.2/etc/printcap
#this is the printcap on the 9.7 node
#the remote laser printer on SR10 node //sr10
lp26:\
      :lp=:rm=sr10:rp=lp26:sd=/usr/spool/lpd/lp26:
#the local line printer
lp:\
      :lp=/dev/sio1:sd=/usr/spool/lpd/lp:if=/usr/lib/lpf:
```

Figure 1-1. Printcap Entry on a Workstation Running SR9.7 Supporting a Printer on a Workstation Running SR10.

```
#//sr10/etc/printcap
#this is the printcap on the SR10 node
#the laser printer (prints via prf)
lp26:\
      :pc=/com/prf -pr lp26 -trans:lp=/dev/null:\
      :sd=/usr/spool/lpd/lp26:if=/usr/lib/lpf:sh:sf:pw=100:
```

Figure 1-2. Printcap Entry for Printer on a Workstation Running SR10 to be Accessed from SR9.7 Workstation in Figure 1-1.

1.9.2 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.

1.9.3 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.

1.9.4 The prf Command

The **prf** command includes the following new options:

- **–pre10**
- **–dialogue**
- **–check**
- **–list_printers**
- **–list_sites**
- **–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 option, the job is queued to any available server, and you will not get an error message if the requested printer does not exist.

1.9.5 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 *Printing in the Aegis Environment* or the *Managing System Software* manual for your environment for details.

1.10 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. For example, we have included a port of the Berkeley /etc/**named** daemon version 4.7 in this release. 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 BSD System Software*.

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 the socket interface, will be available after the SR10 release. For the present, consult the *Domain Programming Tools Reference* 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.

1.11 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.

1.12 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 *Domain/OS Call References*.

1.13 GSR

The Domain Graphics Service Routines (GSR™) product is being re-released 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: An Overview of Software Release 10.1

Domain/OS supports the Personal Workstation Series DN4500 for the first time in Software Release 10.1 (SR10.1). At SR10.1, Domain/OS also includes support for important new hardware that was added for the first time in Software Release 9.7.1.

Although SR10.1 is a "point" release, the distribution contains all of the files from SR10 and the new files from SR10.1. If you are upgrading existing SR9.7.1 or earlier-based systems to SR10.1, be sure to read *Making the Transition to SR10 Operating System Releases* before installing new software.

This Chapter briefly describes changes made to Domain/OS system software. Some of these changes are the same as those made between SR9.7 and SR9.7.1. While this Chapter will provide an overview of these common changes, it refers you to another document for details.

2.1 Hardware Support

The following subsections describe hardware supported in SR10.1 that wasn't supported by SR10.0.

2.1.1 DN4500 Personal Workstation

The Series DN4500 is a SAU7-compatible (at SR10.1) Personal Workstation containing a 33 MHz MC68030 and a 33 MHz MC68882 floating point co-processor. It has a 64 KB physical directly mapped cache and a 1 GB virtual memory space, with hardware support for 4 GB. It comes in a minimum configuration of 8 MB of memory, expandable to 32 MB.

The DN4500 is available in several graphics configurations, including 1280 × 1024 8-plane color. Mass storage options include full-height 170, 380, or 760 MB ESDI winchesters, half-height ¼ in. 60 MB SCSI cartridge tape, half-height floppy, and external ESDI disk subsystems containing either 1 or 4 drives.

2.1.2 DN3500 Personal Workstation

SAU7 support for the DN3500 and DN4000-F Personal Workstation. The Series DN3500 is a SAU7-compatible Personal Workstation featuring a 25-MHz MC68030 microprocessor equipped with separate instruction and data caches, up to 32 MB of main memory, and an IBM PC AT compatible bus, all in a desktop-sized cabinet. The display options include 15-inch and 19-inch, high-resolution monochrome monitors, and 15-inch or 19-inch color monitors in 4- or 8-plane formats.

2.1.3 DN4000-F Personal Workstation

The DN4000-F includes an 8-plane (1280 × 1024) color controller and associated 19-in. (1280 × 1024) color monitor. The PC AT®-compatible bus controller has a total display memory of 2048 × 1024 × 8 planes, with system software to support direct access to the display memory in plane mode.

The new 19-inch color monitor is a 68-Hz (1280 × 1024) display. The new monitor features higher bandwidth (170 Mhz) for increased video performance, loop-through connectors for multiple displays, and continuous input voltage. Switch selection for multiple displays and input voltage is no longer necessary.

2.1.4 Floating Point Accelerator

The FPA is based on the Weitek 3164 floating point chip. It accelerates both scalar and vector processing and is available as an option to the DN4000, DN3500, and DN4500 Personal Workstations.

Support for the DN3500, DN4000-F, and the Floating Point Accelerator is functionality that has been merged into SR10.1 from SR9.7.1, where it first appeared. This merged functionality also includes:

- Enhancements to graphics software to support new hardware. These include changes to **gprlib**, changes to the **pad** and **smd** insert files, and changes to **2DGMR**.
- New or enhanced diagnostics.
- New cartridge tape access protocols

Chapter 1 of the *Personal Workstations Hardware and Software Release Document* for Software Release 9.7.1 (Order No. 012206-A00) describes functionality that has been merged from SR9.7.1. An addendum to the SR9.7.1 release document describes support for the DN3500 Personal Workstation.

2.2 New Cartridge Tape Access Protocols

In SR10.1, **wbak** enforces the more stringent cartridge tape access protocol that was introduced in SR9.7.1. This protocol prohibits writing except at beginning of tape and end of data. Therefore, **wbak** doesn't write an explicit end of tape marker on cartridge tape or support replacement of existing tape files except at beginning of tape. The following summarizes the affects of this protocol:

- Tape files cannot be "replaced" except at the beginning of the tape.
- You can't use **wbak –f end** under SR9.7 to append to a tape created under SR10.1. However, **wbak –f** *n* will work if you know *n*, the number of the last file on the tape.
- If a tape has been written to by an SR9.7 system, then added to by an SR10.1 system, it cannot be restored under SR9.7. Do not restore tapes under SR9.7 unless you are sure they have NOT been written to under SR10.1.

2.2.1 Examples of Now-Illegal wbak Invocations

When wbak attempts to write at an illegal position on tape, SR10.1 displays the following error message:

```
?(xbak) (xxxx) xxxxxx – cannot write at this tape position (library/tfp)
```

The two examples that follow demonstrate backup scenarios that are illegal:

1. Write to tape by filenumber *n* when *n* is not 1 and *n* is not exactly 1 greater than the last tape file on tape. For example:

```
wbak.10.1 –dev ct –f 1 file1
wbak.10.1 –dev ct –f 2 file2
wbak.10.1 –dev ct –f 2 new.file2      ← illegal
```

2. Write to tape by position when position is not beginning of tape or after all known tape files. For example:

```
wbak.10.1 –dev ct –f 1 file1
wbak.10.1 –dev ct –f 2 file2
rbak.10.1 –dev ct –rewind
rbak.10.1 –dev ct –f 1
wbak.10.1 –dev ct –f cur new.file2  ← illegal
```

2.2.2 Cartridge Tape Compatibility for rbak and wbak

All SR10.1 created tapes can be restored on SR9.7 and SR9.7.1 systems. When SR9.7's **rbak** fails to find a tape file on the SR9.7.1 created cartridge tape, the message displayed will change. Typically, **rbak** fails to find a given tape file because the tape file is not on the cartridge tape. **rbak** may also fail to find a given tape file due to a variety of other conditions that prevent the data from being read correctly.

Under these conditions on an SR9.7 system with an SR9.7 tape, **rbak** will display:

```
?(rbak) File "ffff", section #n not found.
?(rbak) "Current" file does not exist. (Probably at end of volume set.)
```

Under these conditions on an SR9.7 system with an SR9.7.1 tape, **rbak** will display:

```
?(rbak) (open_input_volume) Unable to open backup file.
– read no data (OS/cartridge tape manager)
```

All SR9.7 created tapes can be restored on SR10.1 systems.

All SR10.1 created tapes can be written to on SR9.7 systems with the restriction that the writes using **wbak –f end** are not supported on SR9.7 for SR10.1 created tapes.

All SR9.7 created tapes can be written to on SR10.1 systems as long as the writes are at beginning of tape or after all existing data. This tape is considered a "hybrid" tape and is discussed in the next subsection.

2.2.3 Hybrid Tape Details for rbak and wbak

A hybrid tape is any tape that was written to under an SR9.7 system and then added to under an SR10.1 system.

Such hybrid tapes can be successfully restored under SR10.1, but cannot be restored under SR9.7. Because SR9.7 systems write an explicit end of tape marker that cannot be moved or removed when SR10.1 adds to the tape, an SR9.7 system cannot read past this end-of-tape marker to access the data added under SR10.1.

NOTE: The **rbak** utility in SR9.7 fails silently when trying to restore a hybrid tape. It is up to YOU to know that a tape has been written to by an SR10.1 system and not attempt to restore it under SR9.7.

When an SR10.1 system writes to tape sequentially after a tape file written by an SR9.7 system, that SR9.7 tape file is the last thing that SR9.7 can read. The following examples demonstrate this:

```
wbak.9.7   –dev ct –f 1    file1 (permanent end of tape under 9.7)
wbak.10.1  –dev ct –f end  file2
rbak.9.7   –dev ct –f 2    –all  ← error, cannot find it
rbak.10.1  –dev ct –f 2    –all  ← works ok

wbak.10.1  –dev ct –f 1    file1
wbak.9.7   –dev ct –f 2    file2 (permanent end of tape under 9.7)
wbak.10.1  –dev ct –f 3    file3
rbak.9.7   –dev ct –f 1    –all  ← works ok
rbak.9.7   –dev ct –f 2    –all  ← works ok
rbak.9.7   –dev ct –f 3    –all  ← error, cannot find it
```

2.2.4 Adding to the End of SR10.1-Created Tapes on SR9.7 Systems

As stated earlier, it is impossible to append to the end of an SR10.1 created tape on an SR9.7 system using **wbak –f end**.

It is possible, however, to write to the end of an SR10.1 created tape on an SR9.7 system using **wbak –f** *n*. If you know the last tape file number on the SR10.1 tape is *n*, you can issue a **wbak –f** *n+1* and successfully write to the end of the tape.

If you don't know the last tape file number on the SR10.1 tape, you can find it by rewinding the tape and using **rbak –f cur –index** to read through the tape until it fails with '–**read no data (OS/cartridge tape manager)**'.

2.2.5 Change in the SR9.7.1 wbak –no_eot Option

There is a slight semantic change to **wbak**'s **–no_eot** option. In SR9.7, specification of **–no_eot** implied "do not write an explicit end of tape marker and do not rewind." In SR9.7.1, **–no_eot** implies only "do not rewind" since it doesn't write an explicit end-of-tape marker.

2.2.6 Change in Error Message When Writing on Erased Tapes

Writing by position at the beginning of a completely erased tape and appending to completely erased tapes are not supported in either SR9.7 or SR10.1. The error message displayed changed at SR9.7.1. The SR9.7 error message in these circumstances is

```
?(wbak) (open_output_volume) Unable to open volume
– read no data (OS/ctape mgr)
```

The SR10.1 error message in these circumstances is

```
?(wbak) (open_output_volume) Unable to open volume
– end of file (library/tfp)
```

2.3 Remote Magtape Support for wbak and rbak

rbak and **wbak** now provide the option "**–rem**[*ote*] <*host*>:<*dev*>" that allows the reading and writing of **wbak**–format tapes on remote drives. In the above syntax, <*host*> must be a valid Internet host name that is recorded in /etc/**hosts** such that **getservbyname** (**getservent**(3N)) would return successfully if the host is available. The host must be running **inetd**(8), which must be able to accept an **rexec**(3) request to start /etc/**rmt**. <*dev*> is the name of the special file describing the device.

2.4 New System V, Release 3.1 Functionality

At SR10.1, Domain/OS SysV incorporates the support for eight-bit characters, international alphabets, and date formats provided by AT&T in its System V, Release 3.1 of the UNIX operating system. It also includes **nawk**, the new version of **awk**.

2.4.1 Eight-Bit Characters

At SR10.1, the following utilities permit the use of eight-bit characters:

cat
ed
egrep
expr
find
grep
ls
pg
sed
sort
vi

2.4.2 International Alphabets

This support comes in the form of

- A new utility (**chrtbl**(1M)) for creating data files of international characters,
- A new directory of files constituting a database of international character classifications (/usr/lib/**chrclass**), and
- A new function (**setchrclass**) (**ctype**(3C)) and two new environment variables (**CHRCLASS** and **LANGUAGE**) (**environ**(5)) that control the prevailing set of international characters.

The following utilities recognize the prevailing set of international characters:

cat	**egrep**	**sed**
ed	**grep**	**sort**

2.4.3 International Date Formats

This support comes in the form of

- Two new functions (**cftime** and **ascftime**) (**ctime**(3C)) and a new environment string (CFTIME) (**environ**(5)) that control the prevailing date format,
- An extended **TZ** (**timezone**(4) and **environ**(5)) environment variable that allows you to specify the days on which Daylight Savings Time start and end, and
- A new directory of files constituting a database of international date formats (**/usr/lib/cftime**).

The following utilities recognize the prevailing date format:

date	**mount**	**sort**
ls	**pr**	

2.4.4 nawk

nawk is a new version of **awk** that allows you, among other things, to pass it argument arrays on the command line. See **nawk**(1).

2.5 Domain/OS SysV STREAMS

New functionality has been added to Domain/OS SysV STREAMS at SR10.1 by way of a shared Transport Interface library, support for the socket trait, and a new utility and a new daemon. In addition, the STREAMS trace module has been modified to allow the generation of any type of message.

2.5.1 Shared TI Library

At SR10.1, we supply a shared library version of the Transport Interface (TI) library. It resides in /sys5.3/**usr/lib/libnsl**_s. The following **make**(1) rules build a TI client with the shared TI library:

```
LIBNSL_S=/sys5.3/usr/lib/libnsl_s

ti_client.o : ti_client.c
              $(CC) $(CFLAGS) –W0,–inlib,$(LIBNSL_S) –c ti_client.c

ti_client   : ti_client.o
              $(LD) $(LDFLAGS) –o $@ $@.o –A inlib,$(LIBNSL_S)
```

2.5.2 STREAMS Sockets

At SR10.1, we support the socket trait for System V STREAMS modules which implement the transport interface. This allows you to use the BSD4.3 socket interface to transport level STREAMS modules instead of calls through the TI library.

In order to use the socket interface, an address family to type mapping has to be established using **edaft**(1). Subsequently, when **socket**(2) is called, it uses this mapping to open a stream to the appropriate transport level STREAMS module. Only SOCK_DGRAM and SOCK_STREAM socket types are supported in this version.

2.5.3 New Utility and Daemon

edaft(1) allows you to edit the address-families-to-type mapfile. **stackd**(1M) is a daemon that configures STREAMS stacks.

2.5.4 Changes to Trace Module

The STREAMS trace module has been modified to allow the generation of any type of message. This functionality is accessible through a new option (**–g**) to **trconf**(1).

2.6 Other Changes in the UNIX Environments

This section describes other changes in either or both of the UNIX environments (**BSD** or SysV) at SR10.1.

2.6.1 Bourne Shell ENV Renamed to SHENV

If, either under SysV or BSD, the variable SHENV is defined in the environment, the Bourne shell performs parameter substitution on the value to generate the pathname of the startup script containing commands that the shell executes every time a new shell is invoked. No error results if the file specified by the ENV parameter doesn't exist or can't be read. SHENV replaces ENV, which had performed the same function in SR10.

2.6.2 Effect on ACLs of /bin/cp

When copying a file created under SR9.7, **/bin/cp** in both UNIX environments now constructs the ACL of the target according to the same model used when the **–conv** option is specified with **/com/cpf**, **/com/cpt**, and **/com/acl**. Please see "Conversion of SR9.7 ACLs."

2.6.3 The adjtime System Call

Domain/OS BSD now supports the 4.3BSD system call **adjtime**. **adjtime** allows you to make small adjustments to the system time, as returned by **gettimeofday**(2).

2.6.4 Changes in UUCP

SR10.1's **/usr/lib/uucp/uucico** command is equivalent to SR10.0's **/usr/lib/uucp/uucico.real**; there is no **/usr/lib/uucp/uucico.real** in SR10.1. Therefore, all **uucp** accounts should have **/usr/lib/uucp/uucico** as the log-in shell, not **/usr/lib/uucp/uucico.real**. The home directory should remain **/usr/spool/uucppublic**.

The **/usr/bin/uucp** and **/usr/bin/uux** commands now default to **–r** mode. Therefore, you must now execute **/usr/lib/uucp/uucico** manually after queuing a **uucp** or **uux** job. This is functionally equivalent to the previous **uucp/uucico** behavior but eliminates the need for two **uucico** commands.

Remote machines sending a log-in sequence to an SR10.0 or SR10.1 host running **uucp** must specify an expect/send sequence of **"" P_ZERO** (no parity) in their **L.sys** file; the default for some versions of **uucp**, including SR9.7 Domain/IX, is to send the log-in sequence with even parity. The following **L.sys** entry illustrates a typical one needed by remote machines dialing into an SR10.0 or SR10.1 host running **uucp**:

```
hammer Any ACU 9600 T5551212 "" P_ZERO  "" \r ogin:--ogin: uucp ssword: uucp
```

2.6.5 passwd, chfn, and chsh

The registry server now determines whether users seeking to change a password using **/bin/passwd** are authorized to do so. This change has implications for **passwd**, **chfn**, and **chsh**.

2.6.5.1 passwd

The **passwd** utility has been changed to prompt for "your" password, not the password of the person whose password is being changed. The revised prompt says, "Enter your password:". Users seeking to change the password of another should enter their own password at that point.

2.6.6 chfn and chsh

The registry server now decides if you have the right to change the contents of a user's GECOS information (**chfn**; same as **passwd –f**) and log-in shell (**chsh**; same as **passwd –s**). If the process from which these commands are run have no authentication information (such as **rlogin** and **crp –me**), these commands prompt for the password of the person running them, not of that to whom the information applies. They present the following prompts:

```
Authentication information needed.
Enter Your password:
```

2.6.7 SysV /bin/size

At SR10.1, the **–f** option is no longer the default. If you invoke **size** with no options, it displays the combined size of all **text** sections, the combined size of all **data** sections, the combined size of all **bss** (uninitialized data) sections, and the sum of those three sizes, all in bytes. A sample command line and its output follow:

```
$ size test

14684 + 6400 + 1032 = 30014
```

2.7 Enhancements to Floating Point Functionality

The following gives the Pascal declarations for floating point functions and procedures added at SR10.1:

- Return the positive maximum normalized single precision or double precision value:

  ```
  function fpp_$sp_max_normal : real; extern;
  function fpp_$dp_max_normal : double; extern;
  ```

- Return the positive minimum normalized single precision or double precision value:

  ```
  function fpp_$sp_min_normal : real; extern;
  function fpp_$dp_min_normal : double; extern;
  ```

- Return the positive maximum denormalized single precision or double precision value:

  ```
  function fpp_$sp_max_subnormal : real; extern;
  function fpp_$dp_max_subnormal : double; extern;
  ```

- Return the positive minimum denormalized single precision or double precision value:

  ```
  function fpp_$sp_min_subnormal : real; extern;
  function fpp_$dp_min_subnormal : double; extern;
  ```

- Return a positive single precision or double precision infinity:

  ```
  function fpp_$sp_infinity : real; extern;
  function fpp_$dp_infinity : double; extern;
  ```

- Return a single precision or double precision quiet Not-a-Number (all 1's):

  ```
  function fpp_$sp_quiet_nan : real; extern;
  function fpp_$dp_quiet_nan : double; extern;
  ```

- Return a single precision or double precision signaling Not-a-Number (all 1's except signaling bit):

```
function fpp_$sp_signaling_nan : real; extern;
function fpp_$dp_signaling_nan : double; extern;
```

- Read or set the current rounding mode of the FPU:

```
procedure fpp_$get_rounding_mode( out rm: fpp_$rounding_mode_t;
          out status : status_$t ); extern;
procedure fpp_$set_rounding_mode( in  rm: fpp_$rounding_mode_t;
          out status : status_$t ); extern;
```

- Enable or disable one or more floating point exception traps:

```
procedure fpp_$get_trap_enables( out fp_enab: fpp_$trap_enables_t;
          out status : status_$t ); extern;
procedure fpp_$set_trap_enables( in  fp_enab: fpp_$trap_enables_t;
          out status : status_$t ); extern;
```

- Cause the FPU to work in full IEEE 754 denormalized mode, or in a mode which underflows from the smallest magnitude normalized value all the way to zero:

```
procedure fpp_$get_ieee_underflow_mode( out ieee_mode : boolean;
          out status : status_$t ); extern;
procedure fpp_$set_ieee_underflow_mode( in  ieee_mode : boolean;
          out status : status_$t ); extern;
```

- Read or write the accrued exception status byte in 881/fpa/fpx/dn10000-based machines:

```
procedure fpp_$get_accrued_exceptions(
          out acc_exc : fpp_$accumulated_status_t;
          out status : status_$t ); extern;
procedure fpp_$set_accrued_exceptions(
          in  acc_exc : fpp_$accumulated_status_t;
          out status : status_$t ); extern;
```

- Set or read the internal precision mode of all mc68881 and/or dn5xx/FPX based machines (this call has no effect on any other Apollo machine):

```
procedure fpp_$get_mc68881_precision_mode(
          out pr_mode : fpp_$mc68881_precision_mode_t;
          out status : status_$t ); extern;
procedure fpp_$set_mc68881_precision_mode(
          in  pr_mode : fpp_$mc68881_precision_mode_t;
          out status : status_$t ); extern;
```

- Unconditionally set the rounding mode, ieee underflow mode, the trap enables, and, if applicable, the precision mode, to their default values. The accrued exception byte is also cleared:

```
procedure fpp_$set_default_modes( out status : status_$t ); extern;
```

The following types are new at sr10.1 to support the new fpp functions and procedures. **fpp.ins.ftn** In FORTRAN, you should declare these types as integer parameters:

```
TYPE
  fpp_$rounding_mode_t          =       (fpp_$round_to_nearest,
                                         fpp_$round_to_zero,
                                         fpp_$round_to_negative,
                                         fpp_$round_to_positive);

  fpp_$trap_enables_t           = set of (fpp_$enable_inexact,
                                         fpp_$enable_divide_by_zero,
                                         fpp_$enable_underflow,
                                         fpp_$enable_overflow,
                                         fpp_$enable_invalid_operation,
                                         fpp_$enable_signaling_nan,
                                         fpp_$enable_branch_unordered);

  fpp_$accumulated_status_t     = set of (fpp_$inexact,
                                         fpp_$divide_by_zero,
                                         fpp_$underflow,
                                         fpp_$overflow,
                                         fpp_$invalid_operation);

  fpp_$mc68881_precision_mode_t =       (fpp_$extended_precision_mode,
                                         fpp_$single_precision_mode,
                                         fpp_$double_precision_mode,
                                         fpp_$reserved_precision_mode);
```

2.8 TCP/IP Enhancements

At SR10.1, several enhancements have been made to the Apollo TCP/IP implementation. They include support for the Serial Line Internet Protocol (SLIP) and improved subnetting support. We have also improved **ftp** ASCII mode file transfer performance by a factor of two.

2.8.1 SLIP Support

The Serial Line Internet Protocol allows point-to-point connections over an RS232 serial line. To run TCP/IP over an RS232 line, define the node's RS232 interface with the ifconfig command using sl0 as the interface name, specify the network address of the local interface, then specify the address of the destination end point of the link. For example:

/etc/ifconfig sl0 *<ip-address> <dest-address>*

The Apollo TCP/IP SLIP implementation supports a single serial line, and it must use SIO 1 on nodes which have more than one serial line.

2.8.2 Subnetting Support

When a TCP/IP host starts up, it must be able to find out whether it is running in a subnetted environment and, if so, what its subnet mask is. The normal way for this information to be passed to the host is through the **ifconfig** command line(s) in the host's /etc/**rc.local** start-up file. If the host is running in a subnet, it's subnet mask is defined with the netmask parameter of the ifconfig command.

At SR10, the mask could be specified as a single hexadecimal number with a leading 0x, with a dot-notation Internet address, or as a netmask name. For example:

```
/etc/ifconfig dr0 <ip-address> netmask 0xffffff00
/etc/ifconfig dr0 <ip-address> netmask 255.255.255.0
/etc/ifconfig dr0 <ip-address> netmask defaultmask
```

If you use a netmask name, shown in the third example, the name must also be added to /etc/hosts. For example:

```
defaultmask   255.255.255.0
```

At SR10.1, subnetting has been made easier with the addition of:

- A default netmask name, defaultmask, to the standard /etc/**rc.local** start up file
- Another option, **icmp-request**, to the **ifconfig**'s netmask parameter.

With the addition of the default netmask name to /etc/**rc.local**, in most cases you do not have to edit the individual start-up files to use subnetting. You merely add the netmask name "defaultmask" and the desired mask value to the network's /etc/**hosts** file. When the hosts start up, **ifconfig** sets their subnet masks to the value found in /etc/**hosts**. If "defaultmask" is not found in /etc/hosts, **ifconfig** still succeeds but sets the mask to the default for the address class (which is the same as not having subnets). For example, the default mask for a Class C network is "255.255.255.0".

If you use a netmask name other than "defaultmask" and the name is not in /etc/**hosts** or /etc/**networks**, then **ifconfig** fails. If you use 0 as the mask, then **ifconfig** will assume the default mask for the address class.

The icmp-request option to the **ifconfig**'s netmask parameter specifies that the host should obtain its subnet mask from local gateways, using the ICMP Address Mask Request/Reply protcol. A node will respond to this request only if it is a gateway (that is, it has more than one network interface enabled) and if its own subnet mask was explicity set with the mask option to the netmask parameter (and it did not itself use the ICMP request protocol). If the requesting host receives no responses within 2 seconds, it assumes that subnets are not in use on the network.

You also can enable this ICMP Address Mask protocol by assigning the address "255.255.255.255" to the netmask name "defaultmask" in /etc/**hosts**.

The ICMP Address Mask protocol is heterogeneous. Apollo gateways that have been supplied explicit subnet masks will respond to ICMP address mask requests from other machines, and other gateways will respond to ICMP requests from Apollo hosts.

2.9 Error Logging Interface

/lib/dblib provides an interface to your applications for error logging. This interface enables the your applications to:

- Create error logging files and make entries.
- Write a stack traceback to a given stream.
- Write diagnostic information for the last fault in the current process to a given stream.

/sys/ins/**errlog.ins.pas** and /sys/**ins/errlog.ins.ftn** contain the declarations this interface requires for Pascal and FORTRAN, respectively. **/usr/include/apollo/errlog.h** contains declarations for C in function prototype notation.

2.10 Tape File Processor

The unreleased interface, **tfp**, has undergone an incompatible change and now exports a larger structure than previously. Programs that used this interface need to be recompiled at SR10.1. Please note that SR10.0 versions of **rbak**, **wbak**, and **rwmt** may not run on SR10.1, due to this incompatibility.

2.11 Network Computing Kernel

At SR10.1, all NCK daemons and utilities accept a **–version** option, which prints the version of NCK version being used. In addition, the Location Broker administrative tool and the Global Location Broker contain new features and enhancements, as described in the following subsections.

2.11.1 Changes to lb_admin

The Location Broker administrative tool **lb_admin** can now associate textual names with numerical UUIDs (Universal Unique Identifiers) and perform garbage collection of obsolete database entries. There is a new **set_timeout** command that adjusts the timeout period used by **lb_admin**, and the **use_broker** command has been slightly modified.

2.11.1.1 Associating Names with UUIDs

In addition to UUIDs, **lb_admin** can now use textual names to identify objects, types, and interfaces. In both the graphical and the command-line interfaces of **lb_admin**, these names are accepted as input and displayed as output instead of UUIDs whenever possible. If an undefined name is entered as input, **lb_admin** treats the input as a wildcard.

The associations of names with UUIDs are defined in the file /etc/ncs/**uuidname.txt** on the host where **lb_admin** is invoked and also (optionally) in the file **uuidname.txt** in the home directory of the user who invoked **lb_admin**. The **lb_admin** tool reads these files when it starts up and it uses the names defined in the files for the duration of the session.

Each UUID in a **uuidname.txt** file appears at the beginning of a line. Names associated with that UUID occupy the remainder of the line, separated by spaces or tabs. Names that contain spaces or tabs must be delimited by double quotation marks. Blank lines and lines beginning with # are ignored. A **#include** construct supports inclusion of other files in this format.

More than one name can be associated with a UUID if several names appear on one line of a **uuidname.txt** file, if a UUID appears on several lines of one file, or if a UUID appears in several files. The first name encountered by **lb_admin** when it starts up is treated as the "primary name" for the UUID and all subsequent names are treated as "aliases." Any primary names or aliases can be entered as input to **lb_admin**, but the tool always uses primary names for output.

Note that this mechanism for associating names with UUIDs may be superseded by a naming service in a future software release.

2.11.1.2 Garbage Collection of Databases

An interactive **clean** command in **lb_admin** now allows you to find and delete obsolete entries in the Location Broker database you are administering (the "current database").

When you issue the **clean** command, **lb_admin** attempts to contact each server registered in the current database. If the server responds, the entry for its registration is left intact in the database. If the server does not respond, **lb_admin** tries to look up its registration in the Local Location Broker database at the host where the server is located, tells you the result of this lookup, and asks whether you want to delete the entry. If a server responds, but its UUIDs do not match the entry in the current database, **lb_admin** tells you this result and asks whether you want to delete the entry.

There are two situations in which it is likely that a database entry should be deleted:

1. The server does not respond, **lb_admin** succeeds in contacting the **llbd** at the host where the server is located, and the server is not registered with that **llbd**.
2. A server responds, but its UUIDs do not match the entry in the current database.

Entries that meet either of these conditions are probably safe to delete and are considered eligible for "automatic deletion" (see the next paragraph). In other situations, it is best not to delete the entry unless you can verify directly that the server is not running (for example, by listing the processes running on its host).

When the **clean** command ask whether you want to delete an entry, you have four ways to respond. A "y" response deletes the entry. An "n" response leaves the entry intact in the database. After either a "y" or an "n," **lb_admin** proceeds to check the next entry in the current database. A "g" response invokes automatic deletion, in which all eligible entries (see the previous paragraph) are deleted and all ineligible entries are left intact, without your being queried, until all entries have been checked. A "q" response terminates the **clean** operation.

2.11.1.3 New set_timeout Command

A new **set_timeout** command allows you to adjust the timeout period used by **lb_admin** in all of its operations. With an argument of **short** or **long**, the command sets the timeout accordingly. With no argument, the command displays the current timeout value.

2.11.1.4 Modification to the use_broker Command

The **use_broker** command in **lb_admin** has been modified so that when invoked without an argument, it tells whether the current Location Broker is local or global.

2.11.2 Changes to the Global Location Broker

The Global Location Broker (GLB) supports two distinct interfaces. The "application interface" services lookups and updates by clients. The "replication interface" is used by GLB replicas to communicate with each other and with the **drm_admin** administrative tool. The application interface can simultaneously support both the Internet IP and the Domain DDS network protocols. The replication interface, which formerly supported only DDS protocols, can now use either IP or DDS (but not both). All GLB replicas at a site should use the same protocol family for replication.

The ability to perform replication via IP protocols is useful at sites with internets that support only IP routing. This enhancement will also be useful at sites that plan to run GLB replicas on "foreign systems" (hosts other than Apollo systems) when a portable replicated GLB becomes available.

When you create the first GLB replica at your site, you can specify either DDS or IP as the address family; if you do not specify an address family, DDS is used by default. Subsequently created GLB replicas will use the same address family as the first replica. If your site requires GLB replication via IP and you have already created GLB replicas that use DDS, you can change the address families of the existing replicas from DDS to IP.

2.11.2.1 Changes to glbd and drm_admin

The GLB daemon, /etc/ncs/**glbd**, now supports two new options. **–family** allows you to specify the address family of the first GLB replica you create. (Replicas you create subsequently will use the same address family.) **–change_family** allows you to change the address family of an existing GLB replica.

The Data Replication Manager (DRM) administrative tool, /etc/ncs/**drm_admin**, now supports a **chrep** command that changes the address by which a replica appears in a replica list. This command supersedes the **reprep** command.

The online documentation for **glbd** and for **drm_admin** describes in detail the **–family** and **–change_family** options of **glbd** and the **chrep** command in **drm_admin**.

2.11.2.2 Changing the Address Families of GLB Replicas

Use this procedure to change all GLB replicas at a site from one address family to another:

1. For each GLB host, use the **stop** command in **drm_admin** to stop the GLB daemon. Restart the daemon, using the **–change_family** option of **glbd** to specify the new address family.

2. After all GLB replicas have been restarted, use **drm_admin** to check the replica list at one of the GLB hosts. If any replica is not listed with the new address family, use the **chrep** command to change its address. When all addresses in this replica list are correct, use the **merge_all** command to bring the replica lists at the other GLB hosts up to date.

2.11.2.3 Compatibility with SR10.0 and SR9.7 Systems

We do not recommend running the SR10.0 and SR10.1 versions of the GLB in the same network or internet. These versions do not interoperate correctly if the SR10.1 GLB replicas are using IP addresses. If you have GLB hosts at SR10.0 and at SR10.1, you should update the SR10.0 hosts to SR10.1.

If you have GLB hosts at SR9.7, you should replace the NCK software at those hosts with the software we provide in the /**sr9.7_compatibility** tree. Note that the SR10.1 version of this tree supersedes the SR10.0 version.

2.12 Support for Cross Development

SR10.1 contains support for those developing for both architectures now offered by Apollo: the M680X0 and the Parallel Reduced Instruction Set Multiprocessor (**PRISM**™) architectures. This support is manifest in a new object type, the *cmpexe*, which contains executables for both architectures; a new utility, **xar**, for creating objects of this type; and changes in the core development tools that enable developers to create executables for either architecture.

2.12.1 The cmpexe Type

The **cmpexe** (compound executable) is a new type of object that contains multiple executable versions (components) of the same program: one version that runs on 680X0-based workstations, and another that runs on the Series 10000. The loader determines which component of the cmpexe to load, depending on the machine on which the loader is running. The cmpexe object type is especially useful if you are creating a released product that can execute on either machine type.

The 680X0 executable in a cmpexe may be either **obj** or **COFF** format. Series 10000 executables are only COFF format.

There are three levels of interface to the cmpexe type. The lowest level is used by the loader, the type manager, and the **xar** utility. The next level is the streams level (**ios_$open**). The highest level is the **ldfcn**(4) (common object file access routine) calls.

The default behavior of the cmpexe stream manager is to return all the bytes in a cmpexe file. Programs that copy files or use **stat( )** to determine file size will run correctly on cmpexe files without modification (although they will ignore sparseness).

Coupled with the cmpexe streams manager is support for extended naming. Extended naming lets you access all or part of a cmpexe. Following are the extended names that can be used with a cmpexe file.

Extended Name	Results
m68k	680X0 executable
a88k	10000 executable
isp	native executable (depends on which machine command is run)
all	entire file (same as default (no extended name))

The **ldfcn** calls are a collection of functions for reading COFF files and archives that contain coff files. **ldopen** was modified so that if it attempts to open a cmpexe file (when no extended name is supplied), it will open a stream to the native executable (as if the **isp** extended name had been given). All tools that read coff files should use the **ldfcn** calls so they can handle cmpexe objects transparently. The following coff tools use **ldopen** and work with cmpexe files:

dump **list** **prof** **strings**
gprof **nm** **size** **systype**

You can use extended names with these tools if you are doing cross development (that is,

```
dump –vf cmpexe_file/m68k
```

will access the 68k executable in the cmpexe file regardless of the machine on which the command is being run). The command **strip** does not work on cmpexe files because it writes, and the cmpexe streams manager supports read access only. **strip** must be run on the component COFF files before they are joined in a cmpexe file. The modification to **ldopen** is part of SR10.1. SR10.1 also includes the COFF tools previously listed, built with the modified **ldopen**.

The linkers **/bin/ld** and **/com/bind** do not operate on cmpexe files, nor do the archivers, **/bin/ar** and **/com/lbr**. There is no support for archives of the cmpexe type. Compiling and linking be done before creating the cmpexe. If the type information for a cmpexe file is lost, (for example, it is copied back and forth with NFS), the loader operates properly, but tools that use the cmpexe streams manager will not.

2.12.2 Compound Executable Archiver

We provide a new utility, **/usr/apollo/bin/xar**, that allows you to create cmpexe objects and separate them into their component executables. The xar utility is the only one that writes cmpexe files. See the *Series 10000 Programmer's Handbook* (011404-A00) for a description of xar.

2.12.3 Changes in the Core Tools

The core tools (**/com/bind**, **/usr/bin/f77**, and **/bin/ld**) now enable you to create binaries for either architecture. This support comes primarily in the form of new options that allow you to specify the processor type. In order to support these options, bindable libraries and start-up routines that had resided in **/usr/lib** at SR10 have been moved to a subdirectory, **/usr/lib/isp_m68k**. A sister directory, **/usr/lib/isp_a88k**, has been added, providing the complement of bindable libraries and start-up routines needed for the Series 10000.

The bindable libraries and start-up routines that had been in **/usr/lib** have been replaced by symbolic links that the core tools resolve depending upon the argument given to the **–cpu** switch: **–cpu m68k** causes the core tools to resolve links in **/usr/lib** to **/usr/lib/isp_m68k** (containing bindable libraries and C initialization routines for M680X0-based machines); **–cpu a88k** causes the core tools to resolve links in **/usr/lib** to **/usr/lib/isp_a88k** (containing bindable libraries and C initialization routines for the Series 10000). (**/usr/lib** itself is a link to **../$(SYSTYPE)/usr/lib**, so the file ultimately resolved is appropriate not only for the machine for which you are developing, but for the environment (**BSD** or SysV) within which you are working).

Within the shell, the setting of the environment variable **ISP** determines the resolution of these symbolic links: when **ISP=m68k**, links in **/usr/lib** resolve to **/usr/lib/isp_m68k**; when **ISP=a88k**, links in **/usr/lib** resolve to **/usr/lib/isp_a88k**.

In general, the links in **/usr/lib** are named as follows:

linkname → **isp_$(ISP)/***filename*

For example, with **ISP=a88k** and **SYSTYPE=sys5.3**, **/usr/lib/crt0.o** resolves to **/sys5.3/usr/lib/isp_a88k/crt0.o**. We provide the link

isp_ → isp_m68k

in **/usr/lib** so that when **ISP** is null or nonexistent, all ISP-dependent links resolve to files in **/usr/lib/isp_m68k**.

The succeeding sections provide details about these and other changes in the core tools.

2.12.3.1 /com/bind

The following option is new:

−**cpu** *processor_type* Makes the output file specific to *processor_type* by setting the environment variable **ISP** to *processor_type*. This enables the **bind** to find the appropriate library versions in directories specified with **ISP**.

processor_type may have the value **a88k** for Series 10000 workstations or **m68k** for M680X0-based workstations.

2.12.3.2 /usr/bin/f77

We have added the following options to /usr/**bin**/f77. These options are identical to their counterparts for /**bin**/cc, documented in the *SysV Command Reference* and *BSD Command Reference*.

−A **cpu,***id* Generates code for a particular class of processor. Legal values for *id* are as follows:

id	**Description**
any	Standard code for the host processor
160	DSP160 code
460	DN460 code
660	660 code
90	DSP90 code
330	DN330 code
560	DN560 code
570	DN570 code
580	DN580 code
3000	DN3000 code
4000	DN4000 code
FPX	Floating-Point Accelerator Board available with DN580 and DN590
FPA1	Floating-Point Accelerator Board available with DN3000 and DN4000
PEB	Performance Enhancement Board
a88k	On a Series 10000 workstation, use the native **f77** compiler, FORTRAN Version 10.5.p; on an M680X0-based workstation, cross compile to the Series 10000 using FORTRAN Version 10.5.mpx (released at the same time as Series 10000 Domain System Software Release 10.0.p).
m68k	Not defined for the Series 10000; on an M680X0-based workstation, use the native **f77** compiler.

If you run **f77** on a 680X0-based workstation with the **–A cpu,m68k** option, **f77** calls the 680X0 native compiler. **f77** with **–A cpu,a88k** calls the cross-compiler (FORTRAN Version 10.5.mpx, released at the same time as Series 10000 Domain System Software Release 10.0.p). If you do not have the cross-compiler installed, the command returns an error message. If you run **f77** on a Series 10000 workstation, **–A cpu,a88k** calls the Series 10000 native compiler (FORTRAN Version 10.5.p, released at the same time as the Series 10000 operating system, SR10.0.p). If you do not have the compiler installed, the command returns an error message. When we release a cross-compiler for the M680X0 architecture that runs on the Series 10000 workstation, **–A cpu,m68k** will call that cross-compiler. Until then, running **f77** on a Series 10000 workstation with the **–A cpu,m68k** option is not legal.

–A nansi For .c files. Does not compile with ANSI rules. This option passes **–ntype** to the compiler and does not define the preprocessor symbol __STDC__.

–A systype,*type* Defines the target system **systype** for the compiled object. Values for type are as follows:

type	**Description**
any	Version independent
bsd4.2	4.2BSD
bsd4.3	4.3BSD
sys5	System V
sys5.3	System V Release 3

–A runtype,*type* Passes type information to the compiler and the linker.

2.12.3.3 /bin/ld

We've added the following option:

–cpu *processor_type* Makes the output file specific to *processor_type* by setting the environment variable **ISP** to *processor_type*. This enables the linker to find the appropriate library versions in directories specified with **ISP**.

processor_type may have the value **a88k** for Series 10000 workstations or **m68k** for M680X0 processors.

You can now use the **–r** and **–s** options together.

2.12.4 Changes in config

The installation tool **config** has been changed to allow you to install cross-development support. Please see Chapter 7, "Installing SR10.1."

2.13 Changes in /usr/apollo/bin

The following new functionality has been added in the form of new utilities and changes in existing ones residing in **/usr/apollo/bin**.

2.13.1 Changes in Existing Utilities

crty The following options are new:

–t[rait] Create a named trait instead of a type.

–f[ormat] Reformat the types file.

Systems before SR10.1 did not support named traits. They need to have the types file reformatted when you first create a named trait. Use this option to reformat the types file.

–format has no effect if the types file already has the new format.

Reformatted types files are upwardly compatible with older types files, and should not cause interoperability problems between nodes. Once reformatted, however, they can not be converted back to the older format. Older nodes will see named traits as types.

dlty The following option is new:

–t[rait] Delete a trait instead of a type.

intm The following option is new:

–t[rait] *trait_name* Install the type manager for *trait_name*. Without the **–trait** option, **intm** installs the default type manager.

inty The following options are new:

–t[rait] Install a trait instead of a type.

–f[ormat] Systems before SR10.1 did not support named traits. The first type you install a named trait on your system, you may see an error message showing that the types file has an older format. If you see that message, you must use **–format** option to switch the types file to a newer version.

The new (SR10.1) format is upwards compatible with the old format, and should not interfere with older nodes. Note, however, that older nodes may mistakenly see named traits as types.

2.13.2 New Utilities

The following utilities are new at SR10.1:

trty Test whether some type supports a given trait. **trty** prints a message and returns a status that a shell can test.

2.14 Support for Diskless Series 10000 Workstations

SR10.1 contains support needed to load and run the boot shell on a diskless Series 10000 workstation from an M680X0-based partner. The M680X0-based partner must have a full **/sau10** directory and a copy of **`node_data/boot_shell.a88k**; copy these from a Series 10000 workstation. Altogether, you will need about 25MB from the M680X0-based partner.

2.15 Changes in /com Utilities

The following new functionality has been added in the form of new utilities and changes in existing ones residing in /com.

2.15.1 ACL Operations on Links

In order to set or edit the ACL of a link, you must specify the **–links** option to **acl** and **edacl**.

2.15.2 Command Line Parsing

sigp and **ppri** no longer accept **/com/sh** wildcards.

2.15.3 Conversion of SR9.7 ACLs

cpf, **cpt**, and **acl** now accept the following option:

–conv[ert] This option sets the owner entry in the target from the first *person.%.%* entry in the source with P rights. The first *%.group.%* entry in the source becomes the group entry in the target, and the first *%.%.org* entry in the source becomes the organization entry in the target.

Used with the **–conv** option, **cpf**, **cpt**, and **acl** duplicate Domain/IX behavior. Their default behavior is to turn SR9.7 ACLs into SR10 extended entries. For more information on SR10 behavior, see *Managing Aegis System Software*.

2.16 Binaries Stored on NFS Volumes

Apollo workstations running SR10.1 in an NFS network can now execute Apollo binaries stored on NFS volumes. Such binaries must be COFF objects. Traceback information from processes created from such binaries is incomplete.

2.17 Delete Paste Buffers Command

A new command, **dpb**, tells the Display Manager whether it should delete **`node_data/paste_buffers** when the user logs out. The **dpb** command is valid only when run from a **`node_data/startup.?*** file; you cannot type it from the keyboard. See *Managing Aegis System Software* for information on startup files.

dpb accepts the following options:

–on Delete **`node_data/paste_buffers** when any user logs out.

–off Do not delete **`node_data/paste_buffers** when any user logs out.

By default, the Display Manger will not delete the contents of **`node_data/paste_buffers** at logout. If you do not specify an option, **dpb** toggles the current mode.

2.18 Domain Distributed Debugging Environment

The performance of **dde** has been improved substantially at SR10.1. You can obtain the best performance by using **dde** with these compilers:

Language	Version	Availability
FORTRAN	10.6	(FCS*)
Pascal	8.6	(Soon after FCS)
C	6.6	(Soon after FCS)

* First Customer Shipment

Chapter 3: 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.

3.1 Hardware Configuration Considerations

The following hardware configuration restrictions and recommendations apply to Domain/OS SR10.

- SR10 does not support DN100, DN400, DN420, or DN600 (SAU1) 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. 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 1, 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. All systems should have about four to five megabytes free disk space to boot and run a small number of server and user processes. As a general rule, the operating system requires at least 500K bytes per additional 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 50 MB or less of disk space may require additional storage, depending on your computing needs. Chapter 7 describes the minimum sizes of various software configurations.

Apollo offers many upgrades and add-on offerings to provide additional memory and storage space, including a program for replacing obsoleted SAU1 systems. 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 programs.

3.2 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 (that is, 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 that 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 (that is, 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.

3.3 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.

3.4 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.

3.5 Optional Product Versions Compatible with SR10

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

The tables are *not* intended to help you determine which optional products are available for purchase. For specific information about purchasing layered products, see your local sales representative. You should also see the "Installation of Pre-SR10 Optional Products" section next in this chapter for information on how to install optional products that are not being revised at SR10 onto SR10 systems.

TABLE 3-1. Compatibility of Layered Products: ALIS to Domain/CommonLISP

Product	Version		Comment
	SR9.7	SR10	
ALIS*	1.10DL/2.0	Not available	Not available at SR10 FCS**
D3M™	5.5	6.0	Version 5.5 must run on 9.7 nodes in a mixed network of SR9.7 and SR10 systems
DOCUMENTER'S WORKBENCH	N/A**	2.0	Newly supported at SR10
Domain/4014 Emulator	2.0/2.1	2.1	Older versions do not work on SR10
Domain 5080 Emulator	2.0/3.0	3.0	DOWNCASE must be set to true at SR10
Domain/ Access™	2.0	3.0	Older versions do not work on SR10
Domain/Ada™	1.1/2.0	2.0	Older versions do not work on SR10; V2.0 will be available shortly after SR10 FCS
Domain/C™	4.89	6.0	SR10 release includes a bug fix version of the SR9.7 compiler that runs on SR10 nodes and generates code for pre-SR10 systems
Domain/ CommonLISP™	2.1/2.2	2.2	Older versions do not work on SR10; V2.2 will be available shortly after SR10 FCS

* ALIS is a trademark of Applix, Inc.

** FCS means "first customer shipment." N/A means "not applicable."

TABLE 3-2. Compatibility of Layered Products: Domain/CORE to Domain/PCI

Product	Version		Comment
	SR9.7	SR10	
Domain/CORE™	9.5	9.5	See "Installation of Pre-SR10 Optional Products"
Domain/Delphi	1.1		SR10 books are now available
Domain/ Dialogue™	2.0	2.01	Versions before V2.01 do not work on SR10
Domain/GKS	1.0	1.0	See "Installation of Pre-SR10 Optional Products"
Domain/IX	9.5	N/A	In base software at SR10
Domain/ Laser-26™ Driver	2.2	N/A	In base software at SR10
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
Domain Pascal	7.54	8	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	3.0	
Domain/PCC Remote	1.1/3.0	N/A	Included in Domain/PCC V3.0
Domain/PCE	1.0/3.0	3.0	
Domain/PCI™-1	3.0/4.0	4.0	See the V4.0 release notes for restrictions on its use on SR10

TABLE 3-3. Compatibility of Layered Products: Domain/PCI-ENET to EtherController-AT

Product	Version		Comment
	SR9.7	SR10	
Domain/PCI-ENET	4.0	4.0	See the V4.0 release notes for restrictions on its use on SR10
Domain/PCI-RING	1.0/4.0	4.0	See the V4.0 release notes for restrictions on its use on SR10
Domain/PHIGS	1.0	1.0	See "Installation of Pre-SR10 Optional Products"
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	See "Installation of Pre-SR10 Optional Products"
DPSS/Mail	1.1	2.0	Older versions do not work on SR10
DSEE™	3.2.1	3.3.1	Version 3.2 must run on 9.7 nodes in a mixed network of SR9.7 and SR10 systems
Ethercontroller -MB	7.2	7.2	Cannot be used for Domain internet routing at SR10; Also, see "Installation of Pre-SR10 Optional Products"
EtherBridge	1.4	N/A	In base software at SR10
EtherController -AT	2.0	N/A	In base software at SR10

TABLE 3-4. Compatibility of Layered Products: FORTRAN 77 to Network License Server

Product	Version		Comment
	SR9.7	SR10	
FORTRAN 77	9.66	10.00	SR10 release includes a bug fix version of the SR9.7 compiler that runs on SR10 nodes and generates code for pre-SR10 systems
2DGMR	1.0/2.0/2.1	2.1	2DGMR library is not part of base software at SR10
3DGMR™	2.5/2.6	2.5/2.6	See "Installation of Pre-SR10 Optional Products"
GPIO	9.6.1	10.0	Older versions do not work on SR10
GSR	2.0	2.1	Older versions do not work on SR10
Interleaf*	2.5.30	Not available	Not available at FCS
LaserPrint 800 Driver	1.0	N/A	In base software at SR10
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"
Network License Server™	1.1	1.1	See "Installation of Pre-SR10 Optional Products"

* Interleaf is a trademark of Interleaf Inc. NFS is a registered trademark of Sun Microsystems, Inc.

TABLE 3-5. Compatibility of Layered Products: PostScript Driver for MMP to Versatec V80 Driver

Product	Version		Comment
	SR9.7	SR10	
PostScript* Driver for MMP	1.0	Not available	Not available at FCS
PostScript Driver for Versatec* V80	1.0	Not available	Not available at FCS
RJE	2.1	Not available	Not available at FCS
SPE Board Driver	1.3	2.0	Older versions do not work on SR10
TCP/IP	3.1	N/A	In base software at SR10
Tektronix* 4692 Driver	1.1	N/A	In base software at SR10
TranScript*	Not available	2.0	Newly supported at SR10
Versatec V80 Driver	2.1	2.1	See "Installation of Pre-SR10 Optional Products"

3.6 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 **install/sr9.7_compat** installation tools (located only in the Authorized Area) 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 re-released at SR10 onto an SR10 node, and you are *not* using the tools from /**install/sr9.7_compat**, you must run the scripts from an Aegis shell (**/com/sh**) with the INPROCESS and DOWNCASE environment variables set to "true."

* PostScript is a registered trademark and TranScript is a trademark of Adobe Systems, Inc. Versatec is a registered trademark of Versatec. Tektronix is a registered trademark of Tektronix, Inc.

3.6.1 Products that Require /com

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.

3.6.2 Versatec Printer Driver

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**.

3.6.3 EtherController-MB

Before installing SR10.1, be sure to archive a copy of the SFW-ETH-MB software that you received with your COM-ECMB ETHERNET controller. Restore this software after installing SR10.1.

The EtherController-MB product (COM-ECMB) documentation, *Unpacking and Installing the EtherController-MB* (Order No. 008265, Rev. 01, p. 5–3) instructs you to build a device descriptor file for the controller by executing the following Aegis commands:

```
$ wd /sys/drivers/ecmb
$ build_ddf.sh
```

Do not follow these instructions if you install EtherController-MB on an SR10 system. Instead, use the following command to create a device descriptor file for the EtherController-MB. In the DM command window, type the following (we have split it into two lines for printing purposes):

```
cp authorized_area/install/sr9.7_compat/install_compat/com/sh –c
   /sys/drivers/ecmb/build_ddf.sh
```

You might need to ask a system administrator for the *//node_name* of the Authorized Area that contains the **sr9.7_compat** directory. If you want more information about authorized software installation areas refer to *Installing Software with Apollo's Release and Installation Tools.*

3.7 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.*

3.8 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 at SR10. 2DGMR Version 1.0 was included in previous releases of software.

3.9 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, **sr9.7_executables** and **compat_with_sr9.7**, which are described briefly in the following subsections. All three of the directories mentioned in this paragraph have **README** files that describe their contents in greater detail. Also, *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.

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. Also, compatibility tools that should be used during installation procedures are located in the /**install/tools** directory.

3.9.1 The sr9.7_executables Directory

The **sr9.7_executables** directory contains programs that you should run on SR9.7 systems 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 HoneyDanBer UUCP that will work with SR10 HoneyDanBer UUCP, software for printing in a mixed SR10/SR9.7 environment, NCS support, and versions of **rbak**, **emt**, and **lusr**. It also provides libraries that allow you to run SR10 compilers on SR9.7 nodes.

We also ship an SR9.7 version of streams in the library file /**sr9.7_compatibility/sr9.7_executables/lib/streams** *that is not mentioned in Making the Transition to SR10 Operating System Releases.* This library is described in detail in Section 5.2.1 of these notes.

3.9.2 The compat_with_sr9.7 Directory

The **compat_with_sr9.7** directory contains software to be run on SR10 nodes which is needed for compatibility reasons. In particular, it contains software necessary for building SR9.7 obj format object files on an SR10 node, and for otherwise dealing with mixed object module types. These include versions of the librarian and various UNIX program development tools, SR9.7-compatible versions of the binder and librarian, a tool for properly marking COFF objects that were generated on or by SR9.7 systems, and a version of the **/lib/swtlib** that should be used when generating SR9.7-compatible objects. (Note that **com/bind** in this directory is actually a link to **/com/bind_sr9.7**, which is only installed if you install an Aegis environment.) SR9.7-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.

3.10 Superseded Software

At SR10 we provide several superseded commands, interfaces, and calls; these support features and capabilities that will not be provided in a future major release. The superseded commands include **aqdev**, **fst**, **lkob**, and **lopstr**. These commands only work in environments where programs are invoked within the same process; in-process invocation of programs will not be supported in a future release. For information on superseded Domain/OS calls and interfaces, see Section A.2.3 in Appendix A. Appendix A also lists the commands that are obsolete and are not provided in SR10.

Chapter 4: Bugs Fixed by SR10

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.

4.1 Bugs Fixed that were Reported in APRs or UCRs

This is a summary of the standard software APRs that have been fixed or resolved in this release. In some cases, the APR was resolved in a previous release, but this was not reported.

These reports are listed by the affected software component. If a report was submitted as a UCR, the APR number is the same as the UCR number. Note that these APRs are generally listed according to the way in which they were assigned. As a result, some bugs may be listed under a seemingly incongruous heading. For example, some **/com** command APRs are listed under the Aegis Kernel and documentation errors may be listed under the software component that was documented.

Aegis Backup

000DC1F5 Doing an **rbak** with the **-sacl** option of a file larger than 160 MB results in the error message "Heap Table Full."

0D881816 **wbak** loses files from a large directory.

0D883564 Timeouts during **rbak**.

0D886764 No **–rewind** option in **rbak** for reel-to-reel tapes.

0D886849 Documentation incomplete in coverage of the way **wbak** handles links.

000DBC97 Help file states that **mvf** should act as **cpf** followed by **dlf**.

Aegis Kernel

0D880675 Some objects are left locked after logout following disk full program fault.

0D882513 If a DSP80 is attached to a device that sends an XOFF to **/dev/sio.spm** during the startup sequence, the Server Process Manager hangs until a carriage return is entered on the dumb terminal.

0D882614 Mapping failure after multiple maps and unmaps.

0D882672 Problem when deleting large remote files.

0D882790 Inconsistent reporting of guard faults.

0D883184 Cannot **fork** and **exec** a process which acquires the display in direct mode and interact with that process through the display and mouse.

0D883326 A confusing error message is generated when trying to run DN460-specific object code on a DN560.

0D883572 Use of **acl** with wildcards confuses user.

0D883596 **chpass** is broken on SIO lines at SR9.0.

0D883653 A problem with asynchronous flow control on the DN550 (hex FF).

0D883666 **rws.ins.ftn** does not include **rws_$alloc_heap**, **rws_$release_heap**, or **rws_$alloc_rw**.

0D883793 The use of colons in file names is inconsistent within the SR9.2.3 release and inconsistent with all other releases.

0D883862 Questions on behavior of boot shell environment.

0D884062 Problem deleting UNIX symbolic links by using the Aegis **dll** command with wildcards.

0D884167 If a directory has the maximum number of links, you cannot always delete one and create a new one.

0D884191 Problem handling xon/xoff.

0D884451 **ios_$set_rec_type** not working as documented.

0D884546 Documentation of **name_$read_dir** is incorrect.

0D884583 Aegis becomes case sensitive (DOWNCASE false) in some circumstances.

0D884600 **dlf** and **dlt** do not delete objects if their names include explicit colon (:) or dash (-) characters.

0D885078 Remote **find_orphans** does not recognize the **os_paging_file**.

0D885232 Directories sometimes get lost and must be fixed with **sald**.

0D885264 Passwords limited to lowercase; cannot limit use of **login –p, –h**.

0D885323 Directories are limited to 1300 entries.

0D885406 Various password enhancement requests.

0D885412 **edacct** does not warn you when it overwrites an old account.

0D885579 It can take 20 or more minutes to log in to a node that has been disconnected from the network and runs stand-alone.

0D885584 On log-in over FTP, the password to the remote node remains displayed on the transcript pad.

0D885746 Using DOMAIN/IX SysV, the **assert** macro with a false conditional fails.

0D885776 It can take an excessive amount of time to log in on an internet.

0D885788 Request to have /**com**/**acl** and /**com**/**edacl** offer a **-pdt** option.

0D885798 Program output gets lost when run from a Bourne script with redirected stdout.

0D885826 **ptrace** in SR9.5 will not single step through code.

0D885876 Request to eliminate requirement to **chuvol** when moving disks between systems.

0D885956 Request for ability to monitor password changes.

0D886059 Problem deleting objects with long names that include uppercase characters.

0D886064 Request to correct problems related to **edppo -d** option.

0D886169 Cannot compile programs that use the include file <sys/**un.h**>.

0D886218 The only way to exit from **invol** in the MD is to give an incorrect controller option or to RESET.

0D886314 Problems with serial communications.

0D886404 Failure to log in correctly on a dial-up line prevents further logins over the port.

0D886406 **find_orphans** on a node that is booted diskless lists the **os_paging_file** as an orphaned file.

0D886481 Request for a utility that deletes unnecessary system files from the disk.

0D886598 Problems starting large numbers of processes in background by using the ampersand (&).

0D886608 Problems using Aegis utilities with **DOWNCASE** environment variable set false.

0D886643 Request to put /**systest**/**lst** command in /**com**.

0D886656 Error status returned incorrectly when using **select**(2) on pipes.

0D886683 **tb** command can't get terminating faults when a program is not run in-process.

0D886713 Problem with two remote file servers trying to access same directory simultaneously (in multiple remote DSEE builds).

0D886734 Request for accounting support.

0D886753 **chpass** ignores the login password minimum set by **edacct**.

0D886853 **chpass** allows you to create passwords with illegal embedded spaces.

0D886890 Problems related to **spm** printing diagnostic information over an SIO line.

0D886900 Request to address the ease of aborting processes that are owned by others.

0D886950 Request for the ability to create processes that can only be signaled by the owner or the operating system.

000DBBE5 Intermittent reference to illegal address while processing events (raw mode input).

000DBC3B Request for a way to increase the stack size.

000DBC50 Nodes with a PEB board crash with a PEB interrupt on power-up.

000DBD94 /sys/**dm/startup_login.191** contains case-incorrect file name.

000DBEC4 **find_orphans** cannot put more than 1300 files in the orphans directory, and generates illegal file names if there is a large number of orphans.

000DBF3E The IRQ3 bus signal should not be hard-coded on the DN3000 as the token ring controller.

000DBF86 Disk block header errors at daddr 1.

000DBF90 Free disk space is lost on process creation.

000DC0B6 Getting disk block header errors.

000DC1E2 NIL process problem (get "No ASID available" error, with many suspended nil processes).

000DC20A DM **shut** command signals processes with an Apollo-specific signal.

000DC232 Protected subsystems always run in a new process.

000DC24B When writing to a bitmap in frame mode, the program terminates an "invalid data in write to pad (stream manager/IOS)" error.

Aegis Shell

0D882709 DM is unable to read files with names that include special characters.

0D883640 Lack of proper notification of newly defined Apollo environment variables.

0D886704 Dot-dot (..) is not treated like backslash (\) in Aegis wildcard expansion.

0D886737 Dot-dot (..) is not treated like backslash (\) in Aegis wildcard expansion.

Alarm Server

0D883065 Alarm Server crashed when many alarm windows were on the screen.

Bourne Shell

0D885443 The Bourne shell returns a "no space" error message at random intervals.

Shell Startup

0D884433 Request to provide a way to start the shell specified in the /etc/**passwd** file on login.

/com Commands

0D884871 The character combination :*x*, where *x* is a number, cannot be explicitly used in a file name.

0D885333 UNIX symbolic links are not fully supported by Aegis commands.

0D885543 If **existf** is not supplied an argument, it gives a usage error that does not come out on either STDOUT or ERROUT.

0D885690 The **-code** option to **sigp** is not documented.

0D885729 There is no method of using Aegis commands to get UNIX symbolic links listed.

0D886453 **edacl –file** bug: failure to check if the type of the object matches the type that was specified results in incorrect access rights being supplied.

/com/acl

0D884879 Request to be able to prevent **dlt** from deleting contents of a directory if you do not have delete (D) rights on the directory.

000DC095 Problem with **acl –all** and Domain/IX permissions.

000DC1A2 /**salacl** the date/time-used stamps on affected files on the disk; request for option to preserve the stamp.

/com/bind

0D886189 **/com/bind** does not resolve UNIX symbolic links.

0D886292 **/com/bind** has problems resolving symbolic links with uppercase letters.

000DBF7B Using the same name for global data and procedures results in errors.

000DBFD6 Fatal binder error "get or put outside file" when binding object modules greater than 15000 blocks.

000DC13A The **/bin/cc** compilers do not always receive and/or return an error completion code to the shell when they can't do a job. (Exit value of **bind** is not passed to **/bin/cc.**)

/com/cmf

0D886970 **cmf** outputs extra blank lines when run with the **-tb** option.

/com/cpt

000DC086 **cpt** will not copy files when the file name has an embedded blank.

/com/crp

0D881287 When running ETHERNET* on a DSP80, the password is echoed to the screen during login to a remote host.

0D884599 When using **crp** a second time (**crp –on** *//node1* then, from there, **crp –on** *//node2*), the password is visible in the window.

0D886996 Problems with implementation of pipes for **crp**; raw mode and type-ahead are not fully supported.

000DC0A1 If you have a long SID and **crp –on –me** to your own node, **lusr –me** reports the node ID as the organization ID. **spm/crp** protocol incorrectly limited SID text to 32 characters.

/com/crucr

000DBD25 If you put an "&" sign in a template, the Display Manager is fouled up when **crucr** attempts to set up the editing window.

/com/dlf Using dot-dot (..) for the parent directory in **dlf** results in unexpected format in the transcript.

/com/lbr

0D881198 Request to modify **lbr** options.

0D883475 Request to modify **lbr** options.

/com/ld

0D882092 Request to speed up **ld**.

0D884170 When a pathname with wildcards is specified to **ld**, the output is always lowercase; with no wildcards, case is preserved.

0D886994 Dot-dot (..) is not treated like backslash (\) in Aegis wildcard expansion.

/com/mvf Dot-dot (..) does not work when specified in a **mvf** command.

com/ppri

* ETHERNET is a registered trademark of Xerox Corp.

000DBD0A The **ppri -hi** and **-lo** switches don't work correctly.

/com/sald

000DBCE3 **sald** fails on pathnames that exceed 128 characters.

debug Note: These bugs were reported in Domain/Debug and are fixed in the new Domain Distributed Debugging Environment debugger.

0D880888 The call "SYSTEM" works fine in an application except when running the debugger.

0D881704 Request to allow parameters in debugger macro specifications.

0D881739 Request for support of C pointer de-referencing notation.

0D882613 Request for a watch command to inform when specified variables are changed.

0D883132 Request for a switch that makes the debugger go into include files.

0D883178 Request for a watch command to inform when specified variables are changed.

0D883367 Erroneous array indexing crashes debugger and the system; enhancement requests.

0D883684 The debugger cannot be used with Pascal %INCLUDE.

0D883686 The debugger loses the current line when scrolling in the source code window.

0D883874 The debugger cannot step through include files in which macros and small subroutines are defined.

0D883966 **examine** and go problems.

0D885401 Request for a watch option and the ability to pause the debugger during execution of one piece of code and run a different piece of code while still in the debugger.

0D885796 The definition of a label used in a macro is not always valid.

0D886592 Cross-process debugger cannot be used to debug device drivers due to memory access restrictions.

0D886690 Request to have source line numbers scroll with the code.

000DBF08 Debugger **sh** command should invoke a shell of same type as that in which the debugger runs.

000DBF2D Request for functionality provided by VAX* "SET WATCH" and "SETSCOPR" facilities.

Display Manager

0D886004 The DM has a 256-character line length limit.

0D886076 DM **xp –f** command cannot handle file names with uppercase characters.

0D886246 The DM is limited to 256 character lines.

0D886805 DM does not inform Domain/Dialogue when working directory changes due to moving the mouse into the Domain/Dialogue window from another window.

000DBBBC Small fonts have problems with descenders getting detached; need multiple **au** operations to move from input pad to transcript pad.

000DBC94 Syntax error in **pad.ins.ftn: logical** type identifier starts in wrong column.

000DBCFF A **pgm_$invoke** problem caused by the DM **lo** command being processed once the user is logged out.

0D886241 The cursor often jumps after being placed by the mouse.

0D886311 If you log in to Domain/IX and log out before the initial shell is complete, logout does not kill the initial shell.

0D886439 Problems with commands finding files whose names include uppercase letters.

0D886526 DM crashed on doing **dr**.

0D886935 **dr;echo –r** also selects the line following the end of the marked region for substitution.

0D886941 The window system may lock up when several programs close direct mode windows simultaneously.

0D886961 The definition of **tpad_$inq_dtype** in /sys/ins/**tpad.ins.c** is incorrect.

000DBBE0 DN4000 could hang while sending characters to the DM in some cases.

000DBBF6 Using two input devices simultaneously causes problems in tpm absolute mode.

000DBD9A DM **cps** command converts arguments to uppercase.

000DC05E SR10 beta1 international characters problem.

000DC2F0 The **case** command is limited to files not exceeding 32768 lines long.

* VAX is a registered trademark of Digital Equipment Corporation.

Documentation

0D882426 Corrections to **edstr** command documentation.

0D885307 **ios_$open** documentation does not include read-only open option.

0D885382 **ios_$open** documentation does not state that open options is a set.

0D885846 **ms_$map** documentation does not describe **status_$ok** correctly.

0D885934 **pgm_$** call documentation error section typo.

0D885952 4.2BSD **col** command does not support the –x option.

0D886135 4.2BSD **rlogin –8** option not documented.

0D886219 **ec2_$wait** never returns a 0 value; the consequences for C users should be better documented.

0D886420 The documentation for **ios_$get_dir** is wrong; the parameters are out of order.

0D886519 The *Domain System Command Reference* does not specify the maximum number of characters that can be sent by using **/com/send_alarm**.

0D886575 **ioctl**(2) documentation refers to **<sys/toctl.h>**; this file is empty.

0D886838 Figure 11–1 in *Domain System User's Guide* will not work in Aegis shell.

000DBBF0 Several UNIX SysV manual page errors.

000DBC0D Request for a clearer definition of the **time_$wait** call's **clock** parameter.

000DBC19 **userlib.private** documentation in the *Domain Binder and Librarian Reference* is inaccurate.

000DBD60 The DPSS/Mail **forward** command does not have a **-verify** option.

000DBD71 **help** used from anything but DM types file and requires write access.

000DBDFD The documentation for **/com/fmc** should reflect the inherent limits to this command.

000DBE6A **help** command when in a **crp** window fails unless have at least read and write access.

000DBE63 The **login** shell command should invoke **~user_data/sh/startup**.

000DBEFA The **fpat** command won't accept the maximum token length specified in the help page.

000DBF88 The **col** documentation documents unsupported **-x** option.

000DBFC1 There is no documentation that the UCR system exists.

000DBFDE *Getting Started with Your DOMAIN System* states that file names can start with a space.

000DC094 The **fmc** command encountered an undocumented limit on number of lines and columns and returned an “insufficient buffer space” error message.

Domain/IX

0D881106 Running **diff** on files produces the wrong message when the files are the same.

0D881379 Problems with mapping ACLs to UNIX permissions.

0D881401 If you are in a **crp** window accessing a remote host through an ETHERNET connection, your password will be echoed to the screen.

0D882159 Cannot invoke a shell script from an Aegis shell if the pathname contains a colon.

0D882183 **fopen** fails when trying to open a file for writing via a link and the file does not exist.

0D882647 If you are in a **crp** window accessing a remote host through an ETHERNET connection, your password will be echoed to the screen.

0D882736 The DM cannot properly close a window in which a process has been forked.

0D882935 The wrong error message is displayed when trying to redirect into a locked file.

0D882964 **cron** keeps /etc/**passwd.map** locked.

0D883031 Call to **drand** from FORTRAN fails due to uninitialized static storage.

0D883038 The loader takes too long to load large local static structures.

0D883200 Domain/IX does not give any indication that programs fail in certain cases.

0D883237 On exiting from **tip**, the lock file in /usr/spool/**uucp** is not deleted unless the user is root.

0D883239 **date** does not report time in the proper time zone.

0D883370 **stty –tabs** fails when logging in via **rlogin** or **telnet**.

0D883452 Several C include files are missing from the Domain C Product.

0D883544 BSD **mail** forgets the destination person if it is specified in the command line.

0D883606 **crpasswd** could not change rights on /etc/**passwd**.

0D883687 **ls** and **ex** are slower than Aegis counterparts.

0D883699 **scanf** error; **memory.h** is missing; **fopen** question.

0D883747 In 4.2BSD Bourne shell, when using the **test** command to verify write rights on a directory, the incorrect values are returned.

0D883792 **fwrite** does not return error code for read-only file.

0D883818 SysV UUCP fails because **uucico.real** is not working.

0D883858 Some C Compiler features require Domain/IX.

0D883875 /**bin**/cc does not handle case mapping when uppercase text is embedded in symbolic link.

0D883930 /**user** directory can only handle 315 links.

0D884059 **fmod** library function does not work as documented.

0D884067 **chmod** changes modification dates of files modified.

0D884144 The **cp** command incompletely copies unformatted files (file sizes change, record information lost when copying a rec type file).

0D884171 Values for HUGE and MAXFLOAT are too large for C compiler.

0D884181 **crpasswd** could not change rights on /etc/**passwd** etc.

0D884193 SysV **uucico.real** does not work in MASTER mode (when the L-device is of type DIR).

0D884253 Terminal I/O does not support tab expansion.

0D884282 **dup** exceeds documented maximum number of open files.

0D884363 SysV **mkdir**(1) does not complain if the specified directory exists.

0D884439 Case sensitivity problem with **emacs** invoked by using the VT100 emulator.

0D884447 SysV **ps –u** command does not take a log-in name.

0D884475 User on multiple projects gets wrong rights for a project.

0D884481 UNIX soft links with uppercase characters resolve differently in Aegis and UNIX shells.

0D885031 A program that calls **clock**() dies with a memory fault error.

0D885096 Request for more pseudoterminals and changes to **pty** allocation and releasing.

0D885097 Bourne shell process numbers do not change.

0D885160 BSD **select** call returns incorrect error status in some cases.

0D885152 Performance of **tar** with a cartridge tape is poor.

0D885156 SysV **basename** fails intermittently.

0D885173 When a file is opened with the UNIX **open( )** call with the **truncate** option, the file size returned by **stat** calls is not updated until the file is closed.

0D885221 The **j0** function fails on input values larger that 8.

0D885290 **kill –1 –1** does not work correctly.

0D885273 **troff** does not recognize some font names.

0D885265 Problems with NAMECHARS and shell interpreting `**node_data**.

0D885305 Cannot kill a looping background shell script.

0D885367 Cannot kill a looping background shell script.

0D885392 BSD **gethostid** does not report host ID.

0D885420 Request to remove ability to edit HDRU and **pty_slave** files.

0D885462 **ptys** don't erase control characters properly, don't support TIOCGWINSZ; **signal(3)** should support SIGWINCH.

0D885641 Request for **lockf** call.

0D885760 The C shell cannot be set in-process when logged in as root.

0D885780 It is possible for a process to receive a SIGCLD before the child process is completely dead, causing timing problems.

0D885785 Heap corruption fault while running 4.2BSD **diff**.

0D885872 Sample **awk** test case from the *Domain/IX Support Tools Guide* returns a memory fault in SysV.

0D885899 BSD **troff** fails with SIGSEGV when using the ms **.DS** and **.DE** macros.

0D885902 **talk** to you will not work if your name is longer than 8 characters.

0D885920 Suggestions for reconciling the Aegis and UNIX systems.

0D885949 **tar** has trouble opening tape file.

0D886013 Domain/IX shells run out of space (following multiple access violations).

0D886271 Request for System V **lockf**.

0D886834 The Apollo FORTRAN compiler forces file names to lowercase.

0D886055 Request for UNIX domain socket support.

0D886080 **dq –i** is ignored by a 4.2BSD Bourne shell started by **start_sh**.

0D886121 4.2BSD **ls –a** does not show dot (.) and dot-dot (..) entries.

0D886178 Not all of our files are arrays of bytes.

0D886211 The C shell cannot be set in-process when logged in as root.

0D886220 **du** does not handle case-sensitive file names correctly when invoked from **/com/sh**.

0D886227 **strip** will not work on a file that has uppercase letters in its name.

0D886278 The SysV C shell does not understand the #! convention.

0D886291 4.2BSD **ar** cannot distinguish case.

0D886325 SysV **regcmp** (both command and function) produce incorrect results.

0D886339 Request for information on **freopen** of **/dev/tty**.

0D886348 BSD **troff** suffers an access violation.

0D886391 Erasing the ICANON bit together with setting the ECHO bit does not work on standard windows; changes within a window made with **ioctl** or **stty** do not persist after leaving the specific application.

0D886392 BSD **diff** fails with a heap corruption.

0D886440 **tty** command always returns **/dev/tty**.

0D886470 The **printf, fprintf** and **sprintf** routines have rounding problems with extended precision numbers.

0D886536 SysV protections are not correct in System V for users in multiple projects.

0D886573 BSD **cp** and **tar** change record length of file header.

0D886733 The **who** and **lusr** commands do not properly list all users on the network.

0D886770 Problem with matching regular expressions that use number ranges.

0D886816 SysV print spooler **disable** command memory faults.

0D886862 SysV **/etc/mvdir** fails when run in a C shell.

0D886877 File name wildcard expansions above a certain length cause the shell to hang.

0D886879 **~/.logout** is not executed on DM logout.

0D886884 4.2BSD **find** does not work with a slash (/) appended to the end of a pathname.

000DBBCB When you use **/SysV/usr/games/quiz**, and you want to be quizzed on Shakespeare, you get a memory fault error.

000DBBE6 Child processes don't inherit the full (large) environment of the parent.

000DBBE8 Some Bourne scripts that use **awk** fail.

000DBBE9 **make** scripts fail with wrong environment variable or input text.

000DBC4A Performance of (some) Domain/IX commands is limited (by **stat**).

000DBD5F Keyboard-generated SIGINT signals have no effect in shells created by **/bin/start_sh**.

000DBE23 **awk** fails with a memory fault when run by a SysV shell script.

000DEB9B **/4.2BSD/usr/ucb/vi** causes problems when invoked on a file that is currently being written by some other process.

000DBEED If a compiler is invoked from a UNIX shell, all include files must be case correct.

000DBF2E **touch** without any file names causes a segmentation fault.

Domain/IX clib

0D884223 At SR9.2, I/O buffers are not initialized when C subroutines are called by FORTRAN main routines.

0D886585 The **bcopy()** function copies only up to 65536 bytes. This limit is not described.

0D886746 The 4.2BSD **log()** call returns 0 with no error when called on 0.

0D886769 **sigp** and **kill** let a user send signals to processes the user does not own.

0D886808 Request for support of **lockf**.

0D886856 **chown**, mkdir, and **cp** do not use the ID and IF ACL settings.

000DBCFB **gcvft** call has precision problems.

0DC259 The byte copying routine **bcopy** is incorrect when copying greater than 64K bytes from an odd address to an even address.

Domain/IX Programming Tools

0D886270 **cpp** does not recognize the **defined** operator.

0D886548 Domain/IX **/bin/cc** does not seem to pass the **-r** to the linker (**/bin/cc** always checks after the link step for a start address).

0D886593 **strip** cannot handle files with uppercase characters in their names.

0D886855 **strip** cannot handle files with uppercase characters in their names.

0D886903 **/com/bind** does not recognize 4.2BSD symbolic links.

0D886989 **/bin/ld** does not preserve marks.

000DBBBF **make** scripts have problems with cc continuing on a linker error.

000DBD1B **cb** gets a segmentation fault with a particular file.

000DBD7F sccs appears to be unable to handle a delta for a very large file under SR9.5.

000DBDF0 **ar** command has a 16-character file name limit.

000DBE7D 4.2BSD **ar** has a 15-character file name limit.

000DBECB **/usr/bin/cflow** doesn't iniiialize the o shell environment variable.

emt

0D883473 **emt** sends spurious characters to a host that uses only RS-232 flow control.

0D885483 **emt** does not allow RAW mode when accessing a node using **telnet**.

0D885677 **emt** has a problem when input is redirected.

0D885723 If you shut of a modem on an SIO line, the quit signal is sent to the wrong process.

0D885932 **emt** will not support communications at 9600 baud.

000DBDFB **emt** loses characters when talking to the SIO port.

GPR

0D882579 **gpr_$set_draw_value** in frame mode does not set to -2 properly.

0D884455 Text is positioned wrong on DN570s if the text path is not to the right.

0D885342 A conflict exists between variables defined in **<types.h>** and **gpr.ins.c**.

0D885961 Maximum size of hidden display memory bitmap on DN570 series is 224 by 214.

0D886051 On a DN580, a color set after a **gpr_$set_plane_mask** call is given the wrong draw value.

0D886146 Problem with nonzero status when GPR is initialized in no-display mode.

0D886228 **gpr_$borrow-nc** mode with inverse video on the background color off does not work correctly.

0D886263 Allocation of hidden display memory bitmaps causes some to occupy the same area of memory.

0D886328 **gpr_$init** with a **unit** parameter of 2 fails.

0D886605 Lines created from right to left or from top to bottom using **gm_$polyline_2dreal** (mu<>1) are not correct if the origin and end of the lines are outside the defined window.

0D886735 Resizing an invisible window with **pad_$set_full_window** and displaying it by using **pad_$select_window** truncates the window to the original size.

000DBBB4 Problems with the cursor pattern display occur when using HDM. GPR queue overflows and input is lost.

000DBD0D **gpr_$close_fill_pgon** crashes on some data (**gpr_$fast_traps** does not support filling of complex polygons with coincident edges).

000DBD2C The SR9.7 **gpr_$init** call fails with a "wrong display hardware" error when an attempt is made to open a main memory bitmap on a DSP node.

000DBD57 Bug in detecting down button events causes events to be lost after the first pass.

000DBD9B DN570 hidden display is limited to 214 x 214 pixels (documentation error).

000DBDC2 In frame mode, attempts to display white-filled rectangles over a black-filled rectangle do not work.

000DBE10 **gpr_$init(gpr_$direct_rgb)** will crash a DN570.

000DBE27 **gpr_$init(gpr_$no_display** ...) on a DSP90 generates the error "Wrong display hardware."

000DBE3E Raster operations are ignored for wide lines on DN570 and DN570-T.

000DBEE7 **gpr_$make_bitmap_from_array** causes an access violation error at SR9.7.

000DBFEF There is a problem with getting the correct colors filled when trying to display filled rectangles in frame mode.

000DBFF1 **gpr_$no_display** does not initialize the bitmap plane number.

000DC03F Running a true color (**direct_rdb**) program on a non-24-plane (DN3000) node, causes the node to hang.

000DC0E3 C insert file declares both **gpr_$inq_background** and **gpr_$inq_foreground** as void.

Help Files

0D886649 The **sigp** help file does not include the **-code** option on the usage line.

0D886913 **help syscalls/ios** does not work.

Install and crucr

0D882884 Request for more consistent prompts, speedier interactive use.

0D883151 Problem with ACLs in add mode; /sys/**node_data** can get ACL'ed too tightly.

0D883308 SR8 install problem with DPSS™.

0D883508 Request for checker to make sure install was done correctly.

0D883535 Request for changes to **crucr** form format, **crucr** to work over SIO.

0D883719 Request for global install.

0D883808 Request for checker to make sure install was done correctly.

0D883896 /sys/**alarm**, Domain/Dialogue, **mgrs**, **ns**, **sf**, and **sysdef** are not protected by the secure templates.

0D883899 Install procedure creates an unprotected install directory.

0D884089 Files only used by other node types are automatically installed.

0D884668 You get UUCP manual pages when you answer "no" to the manuals/documentation prompt.

0D884670 If you install both UNIX environments on the source node, there is no way to selectively install one version to user nodes.

0D884761 Domain/IX installation results in ACLs that are too open for %.

0D884762 Domain/IX installs /**dev** in `**node_data**; diskless nodes should get the partner's devices.

0D884786 Problems with /**install/acl_SysV** and SysV /**usr/mail** installation.

0D884925 UCR template needs more reply address lines.

0D884847 Install does not check for missing files, but dies when it encounters one.

0D885094 **crucr** has trouble formatting the heading of the UCR if the customer's name has a ' character.

0D885353 When installing mail over the network, the script uses the wrong file name for the DPSS/Mail release document, resulting in an error returned during the install.

0D885380 The DPSS/Mail release documentation incorrectly tells you to answer "no" to a modify rights question.

0D885784 **crucr** creates an incorrect header.

0D885917 /sys5/**usr/lib/graf** is missing in the Domain/IX **acl_both** script.

0D885919 Domain/IX installs all manuals, games, and release notes, filling the disk.

0D885975 Domain/IX install does not replace the **ptys** in **/dev**.

0D885996 Domain/IX install does not replace the **ptys** in **/dev**.

0D886050 **crucr** should not use the American date format.

0D886012 The Domain/IX installation sets the **/usr/lib/send_mail** ACLs incorrectly; as a result **sendmail** seems to go into an infinite loop.

0D886057 The Domain/IX install does not preserve several user-modified files.

0D886235 /etc/net/**uname** should be a link to **'node_data/thishost**.

0D886841 The Domain/IX **/usr/preserve** link to **`node_data/preserve.xxx** is not created.

0D886982 Various comments on the installation procedure; the install crashes if you remove unwanted files from the source.

0D886893 The Domain/IX system administrator install replaces some networking configuration files; the installation procedure should be changed so you can install to and from the source area.

000DBBE3 The **protection** program does not set the registry ACLs.

000DBC59 Several installation problems, requests.

000DBCBC Several installation problems, requests.

000DBD76 **/install**/acl_SysV doesn't define the path for 'UMAIL'.

000DBDE0 Errors on installing SR9.6 (Same as APR dbcbc).

000DBE33 The Domain/IX system administrator install deletes any existing **/usr** directory without a warning.

netmain

0D886534 A probe for CPU time returns error message "not a level two process" from all nodes that are running SR9.5 or higher when probed from an SR9.5 node.

000DBCC4 **netmain** does not report CPU time at SR9.5.

000DBD70 **netmain_srvr cpu_time** probe fails at SR9.6.

Registry

000DBCF1 It is cumbersome to change users' default projects.

000DBD6E The **chpass** command ignores the registry minimum password-length setting.

Print Server

0D884598 A problem printing from a DSP90 parallel line printer port.

0D885351 Several print server and documentation problems and requests.

0D885461 The printer dies when it encounters the wrong type of file.

0D886657 **prf.ins.c** incorrectly declares the **prf_$get_opt** call.

0D886676 PostScript code can exceed the virtual memory space of the APPLE* LaserWriter*.

0D886761 The print server filters out **prf** options, depending on the DEVICE specification.

0D886766 Aborting the print server when it is plotting on a GENICOM* 3404 leaves the printer in plot mode.

0D886795 The **prf** bottom margin option does not work on the APPLE LaserWriter.

0D886815 The first line in the first file printed using the **prf** left margin option gets the wrong margin.

000DBBF5 An image created with GPR and stored in a GMF file does not print correctly on an APPLE LaserWriter.

000DBC44 The print server does not interpret GMF files created by using **xi**.

000DBC4B The print server does not handle non-printable characters correctly; it can hang.

000DBCA4 The **prf** option **-bot** is ignored at SR9.6 and SR9.7 on an APPLE LaserWriter.

000DBD31 The print server cannot print a file if its name has uppercase characters.

000DBDA2 The **prfd** READ option displays its output on a single line.

000DBE18 It is a complicated process to kill a print job that has started printing.

000DBE5D Request for IMAGEN* printer support.

000DBE64 Request for a **print_config** parameter for user-selectable print queue scan intervals.

000DBFA5 Request for change in default left margin.

* APPLE and LaserWriter are registered trademarks of Apple Computer, Inc.

* GENICOM is a trademark of GENICOM Corp. IMAGEN is a registered trademark of IMAGEN Corp.

000DBFB2 Request for extensions to printing capabilities.

000DBFF8 The print server cannot print a file if its name has uppercase characters

000DC068 The print server cannot print a file if its name begins with a period (.).

salvol

000DBFAA **salvol** needs a progress monitor.

SIO and siologin

0D881754 When DTR (Date Terminal Ready) goes down, the SIO line should act as if XOFF (Transmit off) was seen.

0D882326 It is not possible to time out a dial-in log-in process if someone calls in and doesn't type anything for an infinite period of time.

0D882432 The first characters are often garbage when a dial-in SIO connection is made.

0D882599 **siologin** sends a message to the SIO line when it starts up, causing problems if you have an intelligent modem attached.

0D882782 Intermittent failure to see a change in the CTS line, resulting in a deadlock.

0D882873 The **siotf/siorf** combination will not allow transfer of files of type rec.

0D884164 <BREAK> results in garbage characters on SIO lines when using **emt**.

0D884399 SIO line hangs during login over a modem.

0D884634 The SIO monitor should not try to create a new process for a port that already has a process running on it.

0D884694 Rapid cursor commands to an SIO port hangs **vi**.

0D885336 **tctl** does not work correctly in a Phase II shell.

0D885362 Problems logging in over an SIO line: Aegis **~user_data/sh/startup** is not run, and Domain/IX BSD **stat_csh** gives a "process stop" error.

0D885477 **siologin** survives unexpected terminations when running a 4.2BSD shell.

0D885505 Files of type pty and pty_slave will cause a DM hang when a read is attempted from the DM.

0D885733 **siologin** shell ignores DCD, hangs.

000DC0D7 Multiple FORTRAN BACKSPACEs and FORMATs result in reading the same line.

000DC1AD Interrupt characters are not received over an SIO line in CBREAK mode.

000DC1E4 Request to use CBREAK mode and have interrupt/signal handling over the SIO line.

Standard Streams

0D883853 FORTRAN cannot access DSEE elements as if they were regular files.

0D884001 Creating view on file of type pipe hangs the node.

0D884183 The operator “>” changes an SIO line that was set to "raw" by **tctl** to a non-raw state.

0D885310 /sys/ins/**type_uids.ftn** does not include the values for the type UIDs.

0D886360 It is not possible use **ios_$inquire_only_opt** to open a file that is on another node and is locked.

TCP/IP

0D882975 /**com**/**ftp** goes out of phase with the remote FTP daemon when the remote server sends more than one line in response to a command.

0D884010 The TCP/IP release Version 2.1 does not include help files for **ftp** or **telnet**, and doesn’t inform you of the escape character on startup.

0D884102 There is no way to throttle data when doing a list using both the VT100 emulator and **telnet** to connect to a foreign host.

0D884824 **telnet** to an Apollo system running **telnet_server** gives you an Aegis shell; you can’t start a C shell.

0D885313 Request for Domain/IX **ftp structure** command to support the record structure.

0D885437 **mput** and **mget** options when transferring large number of files result in random files not being transferred.

0D885560 The shell script /sys/**drivers**/**ecmb**/**build_ddf.sh** fails in any UNIX shell.

0D885657 When using Domain/IX 4.2BSD **telnet** and logging in to a Domain/IX node from a terminal server, the terminal will lock up from time to time.

0D885907 **rwho** doesn’t report users logged o nto nodes on the ring as it used to at the SR9.2 revision. It does report those users logged onto ETHERNET hosts.

0D885976 **makehost.sh** corrupts the /etc/**networks** file if any of the network names exceeds 16 characters.

0D886417 The **crp** command followed by **telnet** results in an incorrect end-of-line sequence being transmitted by **telnet**.

0D886446 Request for a configuration file for **telnet_server**.

0D886703 Request for SysV to support 4.2BSD socket routines.

0D886820 Request to support **named**.

0D886914 Request for TCP/IP to support the **named** server.

000DBC2E The hash mode in **ftp** prints all the hash marks when the transfer completes; it should print a hash mark (#) after every 1K bytes are transmitted.

000DBD89 The Aegis **telnet** program does not recognize names in the form "host.site.edu" even though they are listed as one of the aliases for the machine in the **/sys/tcp/hostmap/local.txt** file.

000DBD98 **/com/telnet** does not handle control characters correctly.

000DBE6F If you log in to a machine via **telnet**, the created process does not have a name.

000DBED3 **telnet** does not properly interpret CTRL-Z; the window locks up in a "data" mode.

000DBF96 **rcp** hangs.

000DC135 The Aegis **telnet_server** does not time out at the log-in prompt; CTRL-H and CTRL-U do not work.

UUCP

0D883157 The 4.2BSD UUCP (**uucico.real**) command directs a Hayes-compatible modem to pulse dial the phone number.

0D886819 Request for support of 4.3BSD UUCP software in order to support different grades of traffic.

0D886882 Request for **tip** to support Hayes-compatible modems.

000DBC65 UUCP truncates the pathname of the destination to eight characters.

000DBD8B The sample **`node_data/siomonit_file** provided with the UUCP software is wrong; it should run **user_data/sh/startup**, not **user_data/startup_sh**.

000DBEA6 Several 4.2BSD **tip** commands to transfer files do not work; the process hangs after the **~>**, **~<**, **~p**, or **~t** command.

vfmt

0D883735 **vmft_$decode** returns an incorrect result but no error status if the string to be decoded represents an integer that is larger than the maximum allowable integer.

0D886858 **vfmt with the %D** directive expects a 32-bit integer.

4.2 Additional 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. These bugs are either additional to the bugs listed in Section 3.1, or provide additional descriptions of particular bug fixes of note.

4.2.1 The fork Call

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

4.2.2 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**.

4.2.3 TCP/IP

The following bugs have been fixed in TCP/IP since TCP/IP Version 3.0:

- Data corruption problems caused by a faulty repacketization algorithm have been corrected. These problems occurred infrequently during the transfer of large files over noisy networks to some other vendors' equipment.
- The TCP server process, **/etc/tcpd**, is more robust.
- We now have Maximum Time-To-Live (MAXTTL) and ping interval parameters for **/etc/tcpd** that can be specified by the user.
- We now correctly handle Internet Type A addresses for networks 3.0.0.0 and 10.0.0.0.
- **routed** now correctly handles broadcast packets. TCP/IP now accepts all valid broadcast addresses.
- Routing code that caused incorrect routing table entries for gateways between networks with different address masks (for example, Class A and Class B) or between subnetted and non-subnetted networks has been fixed.
- The routing table now handles up to 512 routes.

- We no longer time out a connection when the remote host is sending 0-windows. We now ACK 0-window probes.
- We now send one-character data packets, known as 0-window probes, at intervals that range from 2 to 30 seconds.
- The problem of user applications (such as **ftp** and **telnet**) hanging or crashing when /etc/**tcpd** needed to manage large numbers of sockets quickly has been fixed.
- Socket connections no longer hang on client nodes.
- A bug in TCP/IP Version 3.0 that affects error status returns from **socket**(2) calls has been fixed. (This bug fix first appeared in SR9.6.) Some transient errors were returned more than once in Version 3.0. Now, transient errors are returned by the socket interface only once. Permanent error conditions are reported on every call.
- TCP/IP now allows a program to reconnect to a connected datagram socket. The socket is reconnected to the new target socket address (sockaddr). However, a second **connect**(2) to a stream socket returns an error. (This feature first appeared in SR9.6)
- A bug in Version 3.0 that affects IP User Data Protocol (UDP) based server processes (such as **routed** and **rwhod**) has been fixed. This bug caused some incoming UDP datagrams to be discarded, even though a socket was open on the port to which they were addressed, if the process owning the socket was calling **sendto**() at the time the incoming datagrams arrived.
- UDP reads and writes up to 8K bytes are now allowed.
- Multiple bugs that caused **ftp** sessions to hang have been fixed.
- Hash marks now are printed during data transfer when requested with the **ftp** hash option. The output is displayed correctly in **crp** windows, **vt100** windows, over **rlogin** and **telnet** connections, and over an SIO line. However, the output is still not displayed in standard pads until the entire file is transferred.
- The socket **ioctls** for SIOCGPGRP (get process group) and SIOCSPGRP (set process group) did not work at TCP/IP Version 3.0. The process group was not set or returned correctly. Use of the SIOCGPGRP **ioctl** returned 8 bytes of information to the caller, which improperly overwrote part of the caller's data area. These **ioctls** are now fixed.
- **rwho** now reports all users logged on to the ring.
- /sys/tcp/**setroute** supports **gateway.g_flags** GWSTATIC and GWPRIMARY.
- Large ICMP packets (for example, ICMP echo requests) now are fragmented as required.
- TCP/IP now recognizes the dr1 physical interface type, which allows TCP/IP to operate in a Domain internet environment.
- The **inetd** process no longer hangs if the **tcp_server** process is killed. Previously, the **kill -9** command caused **inetd** to hang. (This fix first appeared in SR9.6.)

- Multiple problems that caused The X Window System* to hang have been fixed.
- A bug that caused **tcpreset** to hang has been fixed.
- **htable** now handles the new, larger NIC tables.

4.2.4 Berkeley Distribution Bug Fixed

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

The 4.2BSD and 4.3BSD **/usr/include/utmp.h** files 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**.

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

Chapter 5: Bugs Fixed by SR10.1

This chapter describes bugs fixed and limitations removed by SR10.1. We distinguish those reported through the Apollo Problem Reporting (APR) system from all others.

5.1 Bugs Fixed That Were Reported in APRs

This section describes fixes to bugs received as Apollo Problem Reports.

5.1.1 Color Generator

000DC1DE If the program is compiled with the –DBS option or if the comment (C) is removed from the WRITE(*,*) T statement the correct results are obtained.

000DC29F Compilation with **–nopt** AND **–cpu 330** results in a fatal incorrect results of a “constant”. Without **–cpu 330** or **–dba** the code works OK.

000DC2EA Getting erroneous output when compiling with the **-cpu fpx** switch.

5.1.2 Apollo Token Ring

000DC2D4 A DN400-series CPU problem caused the node to transmit a corrupted internal length onto the network.

5.1.3 ETHERNET

000DC455 Short ethernet packets cause a 0002 abort.

5.1.4 Aegis Kernel

000DBE16 When running **/install/acl_dir/enable_prot_subs**, gets an error message when **/com/login** is already a manager in the login protected subsystem.

5.1.5 Aegis Shell

0D885583 Code which works on two lines does not work on one.

0D882259 Shell script line gives syntax error, but if you use 2 lines it works fine.

5.1.6 CC

000DC1F2 When compiling with the **–cpu 3000** switch, the cc fails.

000DC203 When assigning a **float** to an **unsigned int**, if the **float** is larger than an **int** but small enough to fit in **unsigned int**, gets a floating point operand error.

000DC371 Given **unsigned int u = 1**; and **short s = -1**;, u*s=0x0000ffff.

000DC397 Given **unsigned int u = 1**; and **short** s = **-1**;, u*s=0x0000ffff.

5.1.7 /com/lbr

0D881690 There should be a way to group several input modules after a switch.

5.1.8 Documentation

000DC340 Formula as written does not compile; values of **time**(*i*) can become negative if the high order bit is turned on in a 2-byte integer.

5.1.9 Domain/IX

000DC233 Problems with sys5 **cp** trying to copy a **dp_dpd** file.

000DC5C2 /sys5/**bin/passwd** changes the password to some unknown value.

5.1.10 Domain/IX clib

000DC4BC There may be a bug in BSD **connect**( ).

5.1.11 Domain/IX Language Tools

000DC0BF Bus error when stepping in **dbx**.

000DC496 Can't pass **–cpu** switches to **f**77.

5.1.12 FORTRAN

000DBE3D Modification of loop variable within loop does not create warning.

000DC119 Program crashes with memory fault when compiler should report an error due to missing *.

000DC286 Compiler bug involves a complex number raised to a real power and compiler options.

000DC378 Formatted write incorrect.

5.1.13 GPR

000DC541 Functions **gpr_$inq_background** and **gpr_$inq_foreground** are missing from /sys/ins/gpr.ins.ftn.

5.1.14 Installation

000DC464 **media_install** and **media_install_opt** do not permit you to load an authorized area which is not at the root level of the current node.

000DC468 RAI does not handle updating an OS of a large AA on a 170 MB disk.

5.1.15 Naming Server

0D886758 The symbols **name_$pnamlen_max** and **name_$complen_max**, which are defined in **base.ins.pas** and **base.ins.c**, are not defined in any of the FORTRAN include files.

5.1.16 NCS

000DC38E Line following error message contains the name of an Apollo node.

5.1.17 netmain

000DC386 **netmain_note** always fails with non-empty string.

5.1.18 netstat

000DC026 On the DN4000 configured with 380fa disk drives, **netstat -config** reports only 380 MB.

5.1.19 Pascal

000DC05D Crashes with precompiled include files.

000DC2BB Related functions are not able to access one another.

000DC308 Compiling program using **–DBA** and **–SUBCHK** produces compiler failure, "Error 143 Lost Value of Node".

000DC3EF Program causes a compiler failure when compiled with **–SUBCHK**.

000DC447 Different behaviors of **%slib** directive in **pas** 7.3808 and 7.52

000DC4B8 Program elicits error

```
******** Line 654: [Error 143]  Compiler failure, lost value of node.
```

when compiled with the **–opt 3** switch.

5.1.20 Print Server

000DBF20 Problem with Imagen printer and files created with **XI**.

000DC1E9 Laser-26 changes print trays after 250 pages.

000DC438 LaserWriter cuts off left hand side.

000DC515 Please make allowances for tabs set at widths other than 5.

5.1.21 vt100

000DC3B6 Command sequence leads to crash.

5.2 SR10 Bugs and Limitations Fixed by SR10.1

5.2.1 uucp

The SR10.1 version of **uucico.real** no longer gets an instruction fault when attempting to compute transfer rate statitistics with a transfer time of zero. (The SR9.7 compatibility versions of **uucico.real** (**/sr9.7_compatibility/sr9.7_executables/bsd4.2/usr/lib/uucp/uucico.real** and **/sr9.7_compatibility/sr9.7_executables/sys5/usr/lib/uucp/uucico.real**) still contain this bug.) The **/usr/lib/uucp/Uutry** and **/usr/lib/uucp/uusched** commands now execute the correct **uucico** program.

5.2.2 TCP/IP

- **/etc/tcpd** no longer crashes on nodes connected to the Arpanet.
- Sockets created in the local case are now done correctly.
- **ftp** prints hash marks correctly.

5.2.3 loader

Two errors in the C include file for the loader interface have been fixed:

```
A88               #define loader_$kg_text_only
```

has been changed to

```
B88               #define loader_$kg_function_only
```

and

```
A110              #define loader_$sym_text
```

has been changed to

```
B110              #define loader_$sym_function
```

These changes reconcile **/usr/include/apollo/loader.h** with **/sys/ins/loader.ins.pas.**

5.2.4 Display Manager

- The shape of the next window created without specifying a window size no longer affected by a previous abort of the **echo –r** command.
- Highlighting is now erased completely if you mark a text region in a pad, move the cursor out of the pad, and then abort the operation.
- The DM now allows you to edit **/dev/console.**

5.2.5 Aegis Shell

- When the DOWNCASE environment variable is set to "true," **/com/sh** now interprets a leading dot (".") in a pathname as the current working directory, not as a character in the leafname.
- An equal sign ("=") can now be escaped with an "at" sign ("@").

5.2.6 C Shell

The **limit** command now lets you set the stack size of the stack regions of child processes.

5.2.7 Aegis Commands

Aegis commands are now compatible with UNIX pipelines. The following describes bugs in specific commands that have been fixed for SR10.1:

ld — **ld** now matches uppercase letters in patterns to the corresponding uppercase and lowercase letters in filenames.

arcf — **arcf** is now case sensitive.

cpf — **cpf** now handles pathnames of the maximum size.

fpat — **fpat** now recogizes "–p" as a search pattern. It now handles pathnames of the maximum size.

fserr — **fserr** performance has been improved.

lbr — **–replace** now accepts wildcards. It no longer adds some files to the archive twice.

ld — **ld** now handles pathnames of the maximum size.

lvolfs — **lvolfs –n** *//node_name* now always returns the the network ID of *node_name*.

pagf — **pagf** now handles pathnames of the maximum size.

prf — **prf** now handles pathnames of the maximum size.

–version — Inconsistencies of results when using the **–version** options to commands have been eliminated.

5.2.8 The crp Utility

- It's now possible to use **crp –me** to access a pre-SR9.7 workstation.
- The UNIX command **cat** with no arguments now works in a **crp** window.
- The Display Manager no longers changes the working directory to **/usr/man** or **/usr/catman** after running the UNIX **man** command within **crp**.
- Pathnames longer than 256 characters are no longer truncated from within **crp**.
- Extensible naming is now supported over **crp**.

- Programs that use **TIOCGPGRP** with **ioctl( )** no longer fail with a "Bad file number" error if they are run in a crp window on a pre-SR10 node and the remote process is on an SR10 node.

5.2.9 UNIX Commands

- **cron** now sets the organization ID for programs that it runs.
- SR10.1 supports **fingerd**.
- **dbx** can now print or assign the value of a variable that is local to an inner block.
- **passwd**, **chfn**, and **chsh** now work from within **crp** and **rlogin** processes.

5.2.10 BSD Commands and Utilities

- **vi** now handles large paste buffers successfully.
- You no longer get a "csh: permission denied" error if **/bsd4.3/usr/ucb/mail** has the group ID set and your **~/.mailrc** file has set crt=n (where *n* → any number) and received mail exceeds the length specified by *n* (so that it goes through **/usr/ucb/more**).

5.2.11 SysV glossary, locate, starter, and usage Commands

At SR10, the SysV **glossary**, **help**, **locate**, **starter**, and **usage** commands all were links to the executable file for **help**(1). These commands were not supported at SR10 and are not installed at SR10.1.

5.2.12 sendmail

sendmail now supports frozen configuration files.

5.2.13 SR9.X File Ownership and setuid

When copying from an SR9.7 workstation to an SR10.1 workstation, the UNIX command **/bin/cp** converts ACLs to UNIX protections using the SR9.7 Domain/IX model. This enables SR10.1 programs to manipulate these protections. The Aegis commands **/com/cpf**, **/com/cpt**, and **/com/acl** now accept a switch, **–convert**, that provides them the same capability.

5.2.14 Printing

- The print manager configuration file now accepts more spaces at the beginning of each line than shown in **/domain_examples/hardcopy**.
- FORTRAN insert files have been added.
- **suspend** and **continue** operations now work.
- The **update_$status** call has been added to **prsvr**.
- The Aegis printing system now prints headers and footers on all printers.

5.2.15 Tasking

Concurrent Programming Support, also known as tasking, now supports waiting on SIO lines. CPS also now supports the UNIX system calls.

5.2.16 UNIX Calls

The **open( )** system call in SysV now accepts the O_SYNC flag. The **stat( )** system call now determines the size of "rec" type files accurately.

5.2.17 Registry Server

The memory allocation problems that were causing **rgyd** to crash with an out-of-memory error have been fixed.

5.2.18 Common Commands

dspst The bug that caused **dspst** command to not work and return a "unable to get information - device is offline" if any communications controller port was not open has been fixed.

vt100 The **–std**, **–lines**, and **–columns** options to the **vt100** command now work.

5.2.19 /etc Commands and Files

mkdevno The **mkdevno** command no longer hands out duplicate major device numbers.

reboot /etc/**reboot** now reboots the workstation if the **init** process has died.

uctob The **–br** (brief) option to /etc/**uctob** now works correctly.

5.2.20 Creating Directories

At SR10, a "create directory" operation would fail if all of these conditions were true:

- Your naming directory is different from your working directory.
- You try to perform the "create directory" operation in a third directory (not naming directory or working directory).
- The third directory has an extended ACL for the initial default ACL for directories.

This behavior has been fixed in SR10.1.

5.2.21 Alarm Server

The alarm server now leaves tracebacks.

5.2.22 Debugger

The debugger now prints 8-bit characters.

5.2.23 GPR Header File

The ampersands have been removed from in front of references to **gpr_$line_pattern_t** arguments in the **/usr/apollo/include/gpr.h** header file. This declares line pattern arrays to be passed by reference.

5.2.24 GPIO

GPIO no longer tries to shut down global device drivers that it has already shut down in response to a system shutdown or exit from the Display Manager or Server Process Manager.

5.2.25 chacl

If an initial file or directory ACL is set up so that the required entry person, group, or organization ID is inherited from the process and you then use **chacl** to change the entry in any way that results in newly created files or directories getting a specific person, group, or organization ID, rather than inheriting it from the process, files or directories created after this change now will have the UNIX ID that corresponds to the person, group, or organization specified by the initial ACL.

5.2.26 cvtrgy

The **cvtrgy** utility no longer reports the failure to set protection rights in an ACL as a fatal error.

5.2.27 FORTRAN Compiler

The –O option now works.

5.2.28 DSEE Bugs

The DSEE **DELETE BRANCH, DELETE VERSION, RENAME BRANCH** and **RENAME ELEMENT** commands no longer corrupt the values of the sizes of each version of a DSEE element cached in the history file.

5.2.29 Installation

install now correctly creates **/usr/preserve** as a link to **`node_data/preserve**.

Chapter 6: Limitations and Known Bugs

This Chapter discusses limitations and known bugs in Domain System Software at SR10.1

6.1 Cartridge Tape Devices

The no-rewind cartridge tape device, **/dev/rct12**, rewinds on file close. This and other bugs in the cartridge tape interface create the potential for loss of data when using the UNIX tape handling utility **tar** to write to and extract from the cartridge tape. To work around these limitations, use the UNIX tape handling utility **mt** to position the tape before transferring data using **tar**.

You should retension the cartridge tape before each use. Use **mt** like this:

mt –f *device* **retension**

where *device* is one of either **/dev/rct8** or **/dev/rct12**.

You must always rewind the tape after inserting it into the drive and before using **tar**. This is true *even if you rewound the tape before inserting it into the drive.* Use the **mt** command

mt –f *device* **rewind**

where *device* is one of either **/dev/rct8** or **/dev/rct12**.

6.1.1 Accessing the First File

If you want to access the first file on the tape, you must first rewind the tape using **mt**, as shown above. Then, use **tar** to write or extract the first file from the cartridge tape.

6.1.2 Accessing Files Other than the First

To access any file other than the first, you must, in addition to rewinding the tape, use **mt** to position the tape before transferring data using **tar**, or you risk losing data.

6.1.2.1 Extracting Files Other than the First

To extract files other than the first using **tar**:

- Use **mt** to rewind the cartridge tape.
- Use **mt** to forward space to the beginning of the file you want to extract.
- Extract the file using **tar**.

For example, to extract the *n*th file from the cartridge tape, issue the following commands:

```
mt –f /dev/rct12 rewind
mt –f /dev/rct12 fsf n-1
tar xvf /dev/rct12
```

6.1.2.2 Writing Files Other than the First

To write files other than the first using **tar**:

- Use **mt** to rewind the cartridge tape.
- Use **mt** to forward space just beyond the file mark for the last file on the tape you want to keep.
- Write the data to the tape using **tar**.

Forward spacing just beyond the file mark for the last file on the tape you want to keep requires forward spacing just beyond the data for that file and then forward spacing over the mark. The only way to forward space over the mark is to attempt to forward space to a file beyond the current position. This operation results in a harmless "no data" error. For example, to write the *n*th and file to **/dev/rmt12**, issue the following commands:

```
mt –f /dev/rct12 rewind
mt –f /dev/rct12 fsf n-1
mt –f /dev/rct12 fsf 1    ← results in harmless error
tar cvf /dev/rct12
```

6.2 Magnetic Tape Devices

There are restrictions in the use of the SCSI magnetic tape devices **/dev/rmt8** and **/dev/rmt12** when accessed from the UNIX tape handling utilities **tar** and **mt**:

- To use **/dev/rmt8** with the SCSI interface, the device descriptor file must have "reopen" mode turned off (see **edmtdesc(8)** (BSD) or **edmtdesc(1M)** (SysV)). Use the following command:

  ```
  edmtdesc /dev/rmt8 –s reo no
  ```

- **mt** is capable only of rewinding devices attached to the SCSI interface. You must use the **rewind** argument to the **–scsi** option of **mt**. Use **tar** to position the tape before transferring data using **mt**. For example, to extract the second file, issue the following commands:

  ```
  mt –f /dev/rmt12 –scsi rewind
  tar tvf  /dev/rmt12          ← position tape beyond the first file
  tar xvf  /dev/rmt12          ← restore data from the second file
  ```

6.3 Other Limitations

This section describes known limitations in Domain System Software at SR10.1.

6.3.1 Writing to Tape

At SR10.1, you cannot write to tape records that are not an integer multiple of the blocking size. For example, if you use the **open** routine to open the tape drive and then try to write a record (buffer) that is less than 512 bytes, you will get the following error:

```
tfp_$write_record record length less than ctcb record length.
write number written = -1, errno 5
```

6.3.2 The settimeofday( ) Call, the time_$clock Call, and Event Counts

To compute the value to wait for on the event count returned by **time_$get_ec**, use the value returned by **ec2_$read**, and not that returned by **time_$clock**. This method is preferred because the BSD system call **settimeofday( )** (see **gettimeofday**(2)) affects the value returned by **time_$clock** without a corresponding adjustment of event counts returned by **time_$get_ec**.

6.3.3 Booting Diskless from Workstations Running TECHnet

Workstations running TECHnet cannot serve as partners to diskless workstations trying to boot over the ETHERNET. This is because the address of the ETHERNET controller required by TECHnet differs from the one of the workstation in the standard network. In order to boot workstations diskless from such a partner, TECHnet must be brought down temporarily, changing the partner's ETHERNET controller address to the one for the workstation in the standard network. Once all the diskless workstations are booted, the address of the partner's ETHERNET controller can be changed back to the one required by TECHnet.

6.3.4 inlib

Use the **inlib** command from an Aegis shell only if the **INPROCESS** environment variable is not set or set to "false."

6.3.5 Display Manager

- Processes trying to read from standard input but running unattached to standard input have no way of waiting on the Display Manager.
- Attempts to read excessively large files (on the order of many megabytes) may hang the Display Manager.
- When in overstrike mode, the DM command **en** replaces the character under the cursor with a newline, unless that character is an underscore ("_").
- If you use the DM ce command (or EDIT key) on an SR10 node to create a file on a pre-SR10 node, the file does not simply inherit the ACL rights from the initial file ACL of the directory that contains it. Instead, it will get a UNIX ACL that does not include execute rights.

- C programs that create panes in frame mode must now explicitly close the frame by using **pad_$close_frame, pad_$delete_frame** or **gpr_$terminate**. If the frame is not explicitly closed by the program, then the pane is left in frame mode when the program exits. This is only true for C programs and is a result of a change to the UNIX exit routines.

6.3.6 sendmail

The **sendmail** utility's support of frozen configuration files (the **–bz** option) is not robust across different machine types or across changes in system software on the same machine type. Under these conditions, **sendmail** may fail in one of the two most likely modes:

- Clean exit with an entry in the system log.
- Sudden program failure with debugging information accessible through **tb**(1).

6.3.7 Turbo Systems with FPX

With an MD (boot PROM) revision earlier than 5.7, you cannot boot SR10 on a Turbo (DN570-T, DN580-T, DN590-T, or DSP500-T) with an FPX board by typing EX **DOMAIN_OS** at the MD; the FPX microcode will not be loaded. With an MD revision earlier than 5.7, you also cannot boot by loading the microcode manually and then typing **EX DOMAIN_OS**. Regardless of the MD revision level, you can manually boot a turbo with an FPX by entering EX AEGIS at the MD. Also, putting the node in normal mode and pressing the reset button does work correctly.

6.3.8 Booting from Floppy Disks

Booting from floppy disks is supported only on systems that have both floppy and winchester disks. You cannot boot from floppy disk on a node that has only a floppy disk drive and a storage module device (SMD) drive.

6.3.9 UNIX Media Installs and /com

The procedures for installing from distribution media always leave a **/com** directory on the node. If you are installing only UNIX environments, you can delete the /com directory tree after you have installed Domain/OS onto the node.

6.3.10 Debugger and DSEE Memory Requirements

The Domain Distributed Debugging Environment debugger and Domain Software Engineering Environment (DSEE) are only supported on systems with a minimum of three megabytes of physical memory.

6.3.11 The invol Utility

The invol utility gets a drive timeout if there is a badspot in the badspot cylinder. This is a non-fatal error.

Running **invol** with either of the UNIX protections specified (**–b** or **–5** option) sets Aegis protections on the /sys directory and **sysboot** file; Aegis-style protection inheritance is required for these objects for the proper operation of some software.

6.3.12 Crash Logging Systems with Storage Modules

Normally when a machine crashes and reboots, a crash record is added to the system error log. This is shown by the /systest/ssr_util/lsyserr command. However, SAU3 machines (DSP80 and DSP90) booted from a storage module have a bug in their PROMs that defeat this mechanism. That is, if they crash and are then rebooted, nothing shows up in the error log. This bug may apply to all machines that boot from a storage module, and is not new at SR10.

6.3.13 Diskless Nodes and /dev

If a diskless node's partner has a default Aegis environment, even if the UNIX operating system is installed on the partner, **netman** always runs the **netman_com.sh** script. As a result the UNIX devices, including the pad devices, are not created in the diskless node's **`node_data/dev** directory. If the diskless nodes with an Aegis partner will be running UNIX environments, you should change the **/sys/net/netman.rc** link to point to **netman_bin.sh** instead of **netman_com.sh**.

6.3.14 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 affect 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 always set the execute bit. Therefore, if you use Aegis style protection inheritance and the initial file ACL of the containing directory does not specify execution rights, or if you use UNIX style protection inheritance, you must use the **edacl**, **chacl**, or **chmod** command to set the execute 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 (> file_name >? filename). 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** built-in command 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
```

6.3.15 Bourne Shell

If you send a **SIGTSTP** signal (which is generated by a UNIX CTRL/Z key combination) to a Bourne shell that was started by the DM, a **SIGKILL** signal will be delivered instead.

6.3.16 C Shell

The C shell "filec" feature does not work for Display Manager pads.

6.3.17 Aegis Commands

The **dlt** command cannot traverse more than approximately 125 tree levels.

6.3.18 The crp Command

The following restrictions are known for the crp command:

- **crp** does not pass environment variables to the remote process.
- When you **crp** from an SR10 node to a pre-SR10 node and log in by using **/com/login**, the password will appear in the transcript pad.

6.3.19 Common Commands

The following restrictions apply to commands (other than **crp**) 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.

edfont You cannot use the **edfont** utility to view a font if you do not have write rights on the file. You cannot edit a font that is in one format and save it in a different format. Similarly, you cannot edit a font and then save the results by using a new file name.

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.

wbak The **wbak** command cannot traverse more than approximately 125 tree levels.

6.3.20 Commands and Files in /etc

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.

6.3.21 UNIX Commands

The following restrictions apply in both the BSD and SysV environments:

cpp The **cpp** preprocessors do not know about ANSI C syntax.

df The **df** command will not report on devices that are mounted via NFS.

ls The **ls** command always reports that files on SR9.7 systems have extended ACLs.

6.3.22 BSD Commands

chfn The **chfn** utility does not check the length of each field you enter. The **fullname** is limited to 32 characters. Remaining account **misc_info** is limited to 256 characters. Exceeding either limit results in a segmentation fault. You cannot use **chfn** to change your **fullname**; only the registry administrator can change this field.

mail **~h**, when used in a **vt100** window and via **rlogin** does not work properly. Also, when used directly in a DM window it does not output the old header value.

6.3.23 sendmail

If you use the **sendmail –C***path* command and are not logged in as root, **sendmail** may loop a long time while attempting to queue a message.

6.3.24 /usr/adm/lastlog

We do not support **/usr/adm/lastlog**.

6.3.25 Dates on Device Files

We do not update the date/time-modified or used on **/dev** device files when they are opened or written to.

6.3.26 Reading Sparse Files

If a program creates a sparse file, for example by using **lseek** followed by **write**, the holes will be filled when the file is read. This behavior differs from that of most UNIX systems, where the file is not filled simply by reading.

6.3.27 Eight-Bit Font Support

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

- Domain/Dialogue supports the new 8-bit font format. However, it does not support 8-bit characters. Therefore, it is currently still limited to the 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 by using the SR9.7 version of edfont. You can then convert these fonts to 8-bit fomat and have fonts with identical appearances in both formats. In this case, you may have to resolve conflicting font names.

6.3.28 Eight-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.

- User log-in 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 an 8-bit 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 file names that include 8-bit characters.
- Eight-bit support is included in the SR10 print server. However, only PostScript printers support the full 8-bit character set. 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.
- The following Latin-1 characters do not print (you get a blank) when using a PostScript printer:

  copyright sign
  soft hyphen
  registered trademark
  macroncapital
  ring above, degree sign
  plus/minus
  superscript 3
  superscript 2
  superscript 1
  fraction one-quarter
  fraction one-half
  fraction three-quarters
  Icelandic ETH
  capital Y with acute accent
  capital Icelandic THORN
  lowercase Icelandic ETH
  lowercase Y with acute accent

- Additionally, several Latin-1 characters do not print correctly when using a PostScript printer. The following characters print as lowercase "i" characters, without the diacritical marks:

  "i" with grave accent
  "i" with acute accent
  "i" with circumflex
  "i" with diaresis

- 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.

6.3.29 Quit and SIGINT Signaling

The Aegis CTRL/Q and UNIX CTRL/C default key combinations, DM **dq** and **dq –i** (which is identical to **dq**) commands, the UNIX **kill –INT** command, and any call that generates an Aegis quit fault or UNIX **SIGINT** signal result in a traceback entry being added to **`node_data/system_logs/proc_dump** for the signaled process. Also, whenever a process dies due to a "starred" UNIX signal (that is, a signal that is listed on the **signal**(2) manual page with an asterisk (*) following the signal number), the a traceback is left in **`node_data/system_logs/proc_dump**.

6.3.30 Blasting Processes

Blasting a process (by using the Aegis **sigp –b** or DM **dq –b** command, or by responding "yes" 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 as soon as possible after blasting and reboot. This will restore the system resource state. On systems with a UNIX environment, you should first try the **kill –9** command before using **sigp –blast**. The **kill** command will not leave indeterminate system resources.

6.3.31 Tracebacks

Due to the changes required to support program-per-process execution and the way traceback works at SR10, there are now some circumstances in which traceback information is no longer available. These circumstances include the following:

- A traceback is now written out only at process exit. If a program runs in-process, the process does not exit and there is no traceback written.
- If a program does not resignal faults from its cleanup handlers, a traceback is saved only if the exit status is a fault status. Any call to **pgm_$exit(0)** or **exit( )** will exit with a status that does not trigger the saving of a traceback.

6.3.32 getty on an Aegis-Only System

Although the **getty** utility is in the common /etc directory, you cannot use it to control SIO lines on a node where only Aegis is installed; you must use the Aegis **siologin** utility.

6.3.33 Special Devices

You cannot open a storage module block special device if the device is such that **invol** or **salvol** would require additional user input.

6.3.34 UUCP

UUCP does not currently support communication over TCP, TLI, or X.25. SR10.1 does not support UUCP dialout on a line owned by **getty**.

6.3.35 Printing

The print software does not tell the user if a specific **prf** option is unsupported by the specified printer; the option is simply ignored.

6.3.36 Standard Streams and obj Files

If you run an obj format C program on an SR10 node from a program that has separate **errin** and **errout** streams, both **ios_$errin** and **ios_$errout** must be redirected to the same stream/file.

6.3.37 Domain/OS Calls

The following restriction is known in Domain/OS calls:

stream_$redefine — You can execute a stream_$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.

You should not intermix Domain/OS and functionally similar UNIX calls. For example you should not use both **ios_$** calls and UNIX I/O calls in the same context.

6.3.38 Partial-Line Output to the DM

Partial-line output to a transcript pad (that is, output without a newline character) is only forced out when the program attempts an input operation on the related input pad or when the program explicitly calls **pad_$force_prompt**.

6.3.39 Naming Server

At SR10, we cannot guarantee atomicity for certain naming server operations on remote files. As a result, we do not support using the creation of remote file names (especially hard or soft links) as semaphores or for concurrency management. Only if the operation includes the creation of a new object is atomicity provided. For UNIX users, this atomicity is achieved by specifying **ios_$no_pre_exist_mode** in the **ios_$create** call. Here are the particulars:

- For name-adding operations that do not involve the creation of an object (for example, **link**(2), **name_$add_link**), we cannot guarantee that a successful return means the operation was atomically performed for this process. We guarantee that successful status means the operation is completed, but not that it was necessarily completed by this call. We do check for the existence of the name before the remote request is sent (and return "name already exists" if it does at that point), but there is a window in which another process, on the same or a different node, could intervene and cause the name adding operation to occur first. We do not return "name already exists" in that case.

- For name deletion operations (for example, **unlink**(2), **name_$delete_file**), we cannot guarantee that a successful return means the operation was atomically performed for this process. We guarantee that the operation is completed, but not that it was necessarily completed by this caller. We check for the existence of the name before sending the request to delete to the remote system (and return "name not found" if appropriate at that point); but there is a window when the request is sent in which another process, on the same or a different node, could intervene and cause the deletion operation to occur first. We do not return "name not found" in that case. Thus we do not return "name already exists" or "name not found" reliably in some cases where it may be appropriate. We return successful status instead; when we do return either status, it is correct. In both cases, the resulting expected condition exists.

6.3.40 Registry Creation

You can now create a registry for the first time only by using the new **rgy_create** command. This command is shipped only in the **/install/tools** 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 log in using the full text **user.none.none** and the registry is unavailable, you do not need a password.

If you want to use your existing SR9.7 registry, you must use **/install/tools/cvtrgy** to create an SR10 registry from your SR9.7 registry. This program runs on SR9.7 nodes only. You may want to make a backup copy of your SR9.7 registry first; then run cvtrgy against the original registry to create your SR10 registry. You must also replace the pre-SR10 versions of crpasswd with the version that is shipped in **/install/tools/crpasswd**.

6.3.41 Registry Modification

Any command that changes the registry may successfully update the registry, yet return with a "registry unavailable" error message. Therefore, if you get this error message after using a **chfn**, **chsh**, **edrgy**, or **passwd** command, you should check the registry to make sure that the change actually was made.

6.3.42 Debugger

In Domain C, when referencing an element in a union which is an element of a structure, it is not necessary to give the union field's name. The debugger currently requires it.

6.3.43 SAX

SAX may show intermittent problems on nodes with 2 megabytes of memory. These problems are due to the heavy load SAX places on the system while being run as a full system acceptance. To alleviate the problem, attempt to run hardware specific tests, such as

```
/systest/sax –only cpu
```

or remove one or more tests by using the

```
script_id> [yes|no]
```

command, for example:

```
/systest/sax –cpu no
```

For more expert mode options please refer to *Using DOMAIN Diagnostics*.

6.3.44 GPR

On DN570 and DN570-T systems, the maximum size of a hidden bitmap is restricted to 224 × 214 pixels and not the documented 224 × 224 pixels.

6.3.45 C Compilers

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

6.3.46 Open System 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**.

6.3.47 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.

6.3.48 Ethercontroller-MB and Apollo Routing

The Ethercontroller-MB product, also known as COM-ECMB, cannot be used for routing Domain messages between Domain networks. It can be used for routing TCP/IP however. As a result, you should either use a non-MULTIBUS* node as your Domain internet routing node, or keep any routing node(s) that uses the EtherController-MB at SR9.7.

* MULTIBUS is a registered trademark of Intel Corp.

6.4 SR10.1-Related SR9.7 Restrictions

The following restrictions apply to operations done on an SR9.7 (or earlier release) node in a network with SR10 systems, or apply to the interoperability of SR10 and earlier releases. Also see Chapter 2 for additional information on hardware and software compatibility.

6.4.1 Shell Scripts

SR9.7 systems cannot execute UNIX Bourne or C shell scripts that are in unstruct format files (as opposed to uasc files).

6.4.2 Aegis Commands

The following restrictions exist in Aegis Commands:

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** *does* work from SR9.7 nodes to SR10 nodes.

You cannot use **crp** from an SR9.7 node to an SR10 node and then use the BSD **man** command; it fails with a "TIOCGPGRP: Bad file number" message.

existf — The **existf** command returns false if it receives any error. As one result, **existf /dev/sio** will always return false (however, **existf /dev/sio1** will not).

ld — 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.

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

xsubs — The SR9.7 version of the **xsubs** command cannot execute scripts with the "unstruct" type, that is, scripts generated on SR10 systems.

6.4.3 UNIX ls Command

An attempt to use **ls** on an SR9.7 node to list an SR10 directory that contains names longer than 32 characters fails with a "*directoryname* unreadable" error message.

6.4.4 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 file name is *not* truncated if a program on an SR10 node makes the call.

6.4.5 COFF UID

An 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
```

6.4.6 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

If the file is kept on an SR10 node, you can use the **/sr9.7_compatibility/compat_with_sr9.7/com/make_bin** program to convert the file so that an SR10 DM will not open a pad on it. The **make_bin** program can only run on SR10 nodes and will not prevent any SR9.7 DM from accessing the file.

6.4.7 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 type managers.

Normally, if you create a new type 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.

6.4.8 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
Tdebug_* (eight types)
dev_tty
display
group
log
nuls
org
osio
passwd
spe
spmio
spmio_9x
uds

6.4.9 Eight-Bit Font Support

The fonts listed in Subsection 1.3.7 are supplied only in 8-bit format and cannot be used on SR9.7 systems.

6.4.10 Eight-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 an SR9.7 node, you must change the name to one that only contains ASCII characters.
- Pre-SR10 nodes cannot correctly read 8-bit characters.

6.4.11 Printing

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.

6.4.12 TCP/IP

At SR10, TCP/IP administration is done in a manner that is different from SR9.7 Aegis TCP/IP. As a result, if you are running a mixed environment of SR9.7 and SR10 TCP/IP nodes, you should use the following rules to make your decision about what nodes should be TCP/IP administrative nodes based on the type of SR9.7 nodes running on the network.

- If the SR9.7 nodes are running Domain/IX, they can use the same administrative node as SR10 nodes, since both types of nodes can use the same /etc/**hosts** and /etc/**networks** files for name-address mapping. Locate the /etc/**hosts** and /etc/**networks** files on either an SR9.7 or SR10 node and set up the links so that all machines find the same physical copy of the files. We recommend that you use an SR9.7 node to allow pre-SR9.7 nodes to use TCP/IP services.
- If the SR9.7 nodes are Aegis-only TCP/IP users, they use name-address mapping information in the /sys/**tcp directory**, which is no longer present on SR10 systems. In this case, we recommend that you use two TCP/IP administrative nodes—one for the SR9.7 Aegis-only nodes and another for the SR10 nodes.

If you maintain two TCP/IP administrative nodes on one network, you must keep their name-address mapping information consistent. We suggest that you maintain the SR9.7 /sys/**tcp/hostmap/local.txt** file as the master host database. Make changes only to the **local.txt file**, then regenerate the SR9.7 database by using the /sys/**tcp/hostmap/makehost.sh** shell script, and then generate an /etc/**hosts** file for the SR10 administrative node by using the SR10 /etc/**htable** utility.

Do not edit /etc/**hosts** to make further changes. Always modify the SR9.7 **local.txt** file and then regenerate the SR10 /etc/**hosts** file from the **local.txt** file. If you follow this process, the two administrative files will always contain the same name-address mappings.

6.4.13 UUCP

Domain/IX (pre-SR10) UUCP cannot share spool queues with the SR10 UUCP. The /**sr9.7_compatibility/sr9.7_executables** directory contains versions of the HoneyDanBer 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.

6.5 Other Known Bugs

This section documents bugs known to exist in Domain System Software at SR10.1.

6.5.1 distaa

distaa is unable to distribute an Authorized Area according to instructions in a file whose name you pass to **distaa** as an argument. **distaa** tells you the file is unusable.

6.5.2 Display Manager

The following DM bugs are known:

- You cannot boot the DM over an SIO line.
- The DM converts a tab character into eight times the width of a space character; as a result the spacing of text with tabs is incorrect.
- If you open a file just as another is closing it, you may get a lock on the **.bak** file rather than the file itself, even though your header says you have the file itself. Then someone else can come along and open the actual file. In this case, you will get a "name not found" error message when you try to close the file. You can use **llkob** to see which object you actually do have locked if you suspect this situation.
- You cannot undo a paste operation when the paste buffer is longer than 64 lines for an input pad, or 512 lines for an edit pad. The undo buffer overflows, you get a "nothing to undo" error message, and you cannot undo previous commands.
- The DM does not properly handle command input lines that end with an equal sign ("="). You can use such lines if you enclose the argument in single quotes.
- The SHIFT/READ and M3 commands do not work when the pathname is at the beginning of the first line of a file.
- If you delete the newline between two lines, and the sum of the two line lengths exceed 1024 characters (the maximum DM line length), the excess characters at the end of the second line are truncated and lost.
- The Display Manager may declare a "pattern too long" for the command following the offending pattern.

6.5.3 Installation

When the the disk space needed for an installation exceeds the amount available, the message delivered by **install** reports the available disk space incorrectly.

6.5.4 ios_$change_pathname

Using **ios_$change_pathname** to rename "dot" (".") creates a circular naming tree.

6.5.5 BSD mail

The message header fields in BSD **mail** are not editable by way of **~h**.

6.5.6 /usr/bin/bc

bc doesn't work over **crp**.

6.5.7 SysV /usr/bin/remsh

/sys5.3/usr/**bin/remsh** fails if you do not specify a command to execute on the remote host. It does not connect you to the remote host using **rlogin**. If you want to connect to the remote host using **rlogin** from SysV, you must specify /sys5.3/usr/**bin/rlogin**.

6.5.8 rgyd

Disk full errors at a replica site can place the master server in an abnormal state.

6.5.9 sendmail

If one of the targets fails when calling a SMTP mailer, all targets are requeued. As a result, some targets get duplicate mail. When running in SMTP mode with F=1, a reply of 050 will make **sendmail** hang if the target is an alias.

6.5.10 Print Server

The optional fonts must be installed on any workstation on which you plan to run a print server. Failure to do this will result in text jobs not being printed.

The print server may occasionally hang when it is started up if there is excessive network traffic or instability. If this occurs use the **prf** as follows:

```
prf –sig_printer <printer_name> continue
```

6.5.11 DSEE

A known bug in the interaction of the Domain Software Engineering Environment (DSEE) facility with Domain/OS SR10.1 causes a problem in the transcripts of serial remote builds. Specifically, if one of the tools in the build writes transcript data in chunks greater than 1 KB, then **DSEE** loses part of the transcript.

A workaround is to execute the build on more than one node, or on the local node. Here is an example of using more than one node:

```
DSEE> set builder //node1 //node2
```

Here is an example of using the local node:

```
DSEE> set builder –none
```

6.5.12 SysV Graphics Utilities

/sys5.3/usr/**bin/graf/ad** does not work on color workstations.

6.5.13 SAU6 and SAU8 Disk Diagnostics

The disk diagnostics **smd.dex** in the /**sau6** and /**sau8** directories do not catch attempts to write to the bad block list cylinder before doing any testing. They do not write to the bad block tracks or cause any damage, but do terminate with an error message when they reach the tracks.

6.5.14 MD MX Command

The Mnemonic Debugger MX command (which is only valid on the DN460, DN660 and DSP160) cannot load any /**sau4** volume utilities except **fbs**.

6.5.15 The init Program

When running on a DSP server with a terminal connected to the SIO line, the output messages from /**etc**/**rc.user** are lost. The workaround is to use DM startup files in place of /**etc**/**rc.usr**.

6.5.16 DN5X0 Initialization Messages

DN5X0 Systems may not display the Apollo logo and may clip the first character of the startup banner at boot time. This bug does not affect DN5X0-T (turbo) systems.

6.5.17 Node Shutdown

If a process has wired pages required by GPIO and has not been killed before the node is shut down, the system may crash during the shutdown procedure with a 60008 (inconsistent mmape) error.

6.5.18 The /dev/display Device

The following bugs are known in /**dev**/**display**, the UNIX device used to write messages to the Display Manager in a new pad:

- The owner of /**dev**/**display** is not set when you log in to the DM. If you use the /**bsd4.3**/**bin**/**echo** command with multiple arguments and redirect the output to /dev/display, the first argument is omitted. The workaround is to enclose the entire message in single quotes.
- /**dev**/**display** is only writable by root.

6.5.19 The Registry and DPSS/Mail

If a user has only one account and it has a minimum abbreviation of the form *person.group* or *person.group.organization*, then DPSS/Mail won't be able to deliver mail to that user. Resolve this problem by changing such a user's minimum abbreviation to be person (without *.group* or *.organization*). You can check abbreviations by reading the /etc/**passwd**file, or by using the view account_name -f subcommand of **edrgy**.

6.5.20 Eight-Bit Fonts

Some 8-bit characters do not have their margins or width defined correctly. As a result you may see behavior such as an erase operation not removing all of a character.

6.5.21 The crp Utility

The following bugs apply if you use the crp command to create a process on a remote node.

- Programs that use **termcap** mistakenly believe that they are running in a VT100 emulator when they are executed in a process that was started by using **crp**. As a result, commands and utilities that require and normally start the VT100 emulator, such as **vi** and **man**, do not work correctly in a crp pad; they behave as if an emulator is available, even though one is not. You can use these utilities in a **crp** pad by first starting the emulator (by using the **vt100** command) and then running the utility.
- Most remoted pad and tty operations do not return a completion status. The operations affected are those that return only a status code to the caller (for example, **pad_$raw**).
- A timing condition exists wherein incorrect results may occur for remoted operations and invalid data may be transmitted to the remote processes. This condition is more likely to occur when background processes are sharing remote pad streams.
- Programs that wait on an **ios_$get_ec_key** eventcount (or the streams equivalent) for stream input never receive notification that input is available when they are executed in a process that was created by using **crp**.
- Programs that issue the conditional form of **ios_$get** (or the streams equivalent) will cause pad I/O to be confused thereafter; user input will appear in the transcript pad mixed with program output.

6.5.22 Common Commands

The following bugs exist in commands (except for **crp**) 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.

bldt The **bldt –n** *//node_name* command does not always return the network ID of *node_name*.

edfont If you use **edfont** to add glyphs (i.e., characters) to a font, close the font file, use the DM to view it, load another font, and then go back and add more glyphs that come earlier in the font, **edfont** may lose part of the first set of glyphs you added. For example, we have 256-character fonts. If you add glyphs at the end of the font (say, positions 240 through 255), and then go back and add glyphs that come earlier (say, 192 through 198), **edfont** may lose some of the characters in positions 240 through 255. The workaround is to add glyphs in numeric order.

inty Using **inty** on an SR10 node to install a type on an SR9.7 node deletes the type if it already exists; in this case, you must use the command again to install the type. **inty –r** works correctly.

6.5.23 /etc Commands and Files

/etc/**reboot** does not work on DSP4000 server systems.

6.5.24 BSD and SysV Commands and Utilities

The **passwd** command bypasses any policy restrictions that the system administrator may have set on password contents (for example, minimum length, contents), whether for the organization or for the registry as a whole.

The SCCS utilities cannot accept commands with filenames that exceed 32 characters.

6.5.25 BSD Commands and Utilities

The **size** command cannot deal correctly with large COFF objects; it can fail with an "has too many fields" error message fron **awk**.

6.5.26 Printing

/sys/**hardcopy**/**prmonit** is not documented. It is a demonstration program for monitoring print resources and activity.

6.5.27 ETHERNET Drivers

If you have not started any service on an ETHERNET controller which is physically present and you close that unit, Domain/OS will crash with a 00120011 (access violation) status code.

6.5.28 TCP/IP

Urgent data is not handled correctly on **LOCAL** connections; that is, on connections to an address assigned to one of the local interfaces. Urgent data to address 127.0.0.1, **LOCALHOST**, is handled correctly. This will be fixed in an upcoming release.

/etc/**mkhosts** normally takes an argument on the command line which is the pathname of the input file to use. The output files are generated as that pathname with **.pag** and **.dir** appended. At SR10, the input file will be /etc/hosts, regardless of the pathname on the command line. The output files will be generated with the specified pathname. As a result, we recommend that you only specify /etc/hosts when you run /etc/**mkhosts**; this will generate /etc/**hosts.dir** and /etc/**hosts.pag**.

6.5.29 Debugger

The following bugs are known in the Domain Distributed Debugging Environment:

- **STEP –SIGNAL** does not deliver the given signal to the target.
- FORTRAN alternate entry points are not supported.
- Argument information is sometimes unavailable in FORTRAN.

- If you are debugging an obj format module and set your environment to a module, no source file will be displayed and line number references will not be recognized. Set your environment to a routine instead.
- Variables declared in unnamed C inner scopes cannot be referenced because the compiler does not provide information about the address range associated with the scope. In COFF, there is a workaround; you can access the variables by using their fully qualified name (for example, //file/routine/ub.1/varname).
- The debugger cannot find correct address for variables in registers in code ranges that have been optimized away.
- The debugger cannot step through a C statement that includes auto-initializations.
- A simple **go** command in the middle of an action list is not handled correctly.
- You cannot use property **fork –parent –child** to follow the new process as well as the original process, if the new process is created by using **pgm_$invoke**. This is an operating system limitation.
- You cannot use the **step –over** command to step through code that was loaded read-only, for example, when you attach to a running program, or when you step into a shared library.
- If you FREE a process from the debugger, when it terminates it turns into a "zombie" process, and thus occupies a process slot until you exit the debugger.

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, key-bindings, 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/dde_error_log**)
    - A small test program, if possible

6.5.30 Online Examples

The **/domain_examples/cc_examples/mbx_server** and **/domain_examples/cc_examples/mbx_client** fail when compiled with optimization on.

6.5.31 Domain/Delphi

Domain/Delphi version 1.0 electronic document retrieval software does not display certain manual pages correctly, and may cause nodes to hang briefly. This affects only SR10 nodes attempting to display pages from SR9.7 Domain/Delphi-compatible books. A fix is shipped with the retrieval software that accompanies SR10 versions of Domain/Delphi-compatible books.

6.5.32 GPR

The GPR event types **gpr_$latin_1** and **gpr_$function_key** are no longer supported. They are incorrectly included in the /sys/**ins/gpr.ins.language_name** and /**usr/apollo/include/gpr.h** files.

6.5.33 GPIO Include Files

If you compile with /**bin/cc**, the GPIO include files will not compile because of a missing character in a **#define** statement. You should change "#defin"" to "#define" at the start of line line 102 of /sys/**ins/pbu.ins.c** and at the start of line 106 of /**usr/apollo/include/pbu.h**. (In both files, this line specifies the value of **pbu_$max_virtual_address**.)

In the declaration of pbu2_$allocate_map, then length parameter should be

```
linteger          &length, /* length of area needed (bytes) */
```

instead of

```
pinteger          &length, /* length of area needed (bytes) */
```

In the declaration of pbu_$wire_special, the pa_list parameter should be

```
pbu_$pa_list_t  pa_list,   /* returned list of physical addresses */
```

instead of

```
pbu_$pa_list_t  *pa_list,  /* returned list of physical addresses */
```

All functions returning pointers should have the **#options(a0_return)** compiler directive added to the routine declaration. The declarations that need this are:

```
pbu_$allocate_ec
pbu_$map_controller
pbu2_$map_controller
pbu_$mem_ptr
```

Here's an example:

```
extern char *pbu_$mem_ptr(                       /* return pointer to shared ctlr */
            name_$long_pname_t  &pathname,  /* pathname of ddf */
            short               &namelen,   /* length of pathname */
            linteger            *mem_len,   /* length of area mapped */
            status_$t           *status     /* returned status */
);

extern char *pbu_$mem_ptr(                       /* return pointer to shared ctlr */
            name_$long_pname_t  &pathname,  /* pathname of ddf */
            short               &namelen,   /* length of pathname */
            linteger            *mem_len,   /* length of area mapped */
            status_$t           *status     /* returned status */
) #options(a0_return);
```

Chapter 7: Installing SR10.1

In order to install SR10.1, you must read both *Installing Software with Apollo's Release and Installation Tools* (008860-A00) and this Chapter. *Installing Software with Apollo's Release and Installation Tools* provides detailed instructions for using the tools introduced at SR10 to install standard system software on Domain workstations and servers. It also discusses in detail some key concepts related to installation, for example, Authorized Areas, interactive vs. default configuration of software, conservation of space through "hard link" installations, "open" and "closed" protection models, and the installation tools **install++** and **distaa.** This Chapter provides not only information specific to SR10.1, but important information about new and changed functionality in the installation procedures.

7.1 Optional Products

Use the procedures described in this Chapter and in *Installing Software with Apollo's Release and Installation Tools* (008860-A00) to install optional products. You can also use these these procedures to create an Authorized Area (AA) for the Domain/OS product.

When you have prepared for the installation by loading software into the appropriate Authorized Area(s) and merging product release indexes as appropriate, install the software that you want to run on this workstation by using the procedures described in Chapters 4 and 5 of *Installing Software with Apollo's Release and Installation Tools.*

7.2 Workaround for install Bug

A bug in the **install** program, may cause problems installing optional product software on networks which were previously running SR10.0 of Domain/OS. To avoid this problem, follow the steps below. If this is the first time you are installing SR10.1, this procedure is not necessary.

1. Save the file *<authorized_area>***/install/ri.apollo.os.v.10.0//ri.apollo.os.v.10.0** in a safe place on your network. Do this before you begin to load and install any system software from media. This file is the **release index** for SR10.0 of Domain/OS.
2. During the media installation, you are asked if you want to delete the Domain/OS product directory in your Authorized Area (see below). Answer "yes."
3. After completing the media loading and installation (including shutdown/reboot) of SR10.1 Domain/OS, but *before* beginning to load and install optional product software, create the subdirectory **ri.apollo.os.v.10.0** in your optional software *<authorized_area>***/install** directory. Then copy the release index for SR10.0 of Domain/OS, which you saved earlier, into this directory as the file

ri.apollo.os.v.10.0.

4. Now you can load and install optional product software.

7.3 Preparing a Pre-SR10 Network to Load SR10.1

If you are installing SR10.1 of Domain/OS for the first time on a node, and that node is part of a network containing only pre-SR10 nodes, first boot the node from the SR10.0 boot media supplied with this beta release. Then **invol** the node's disk by using the SR10.1 version of **invol**. To boot the disk from media and **invol** the disk, follow the directions in Appendix A of *Installing Software with Apollo's Release and Installation Tools* up to the point on page A-19 where you log in. Use **minst** to complete the loading of SR10.1 into the AA and its installation on the node, as explained in the next section.

7.4 The minst Program

At SR10.1, the scripts **media_install**, **media_install_opt**, **media_install_opt_sr9**, and **dsp_media_install**, as well as the **update** program, are replaced by the **minst** (*m*edia *inst*all) program, located in the *<authorized_area>*/**install/tools** directory. This program provides the combined functionality of all four scripts. **minst** loads software into an Authorized Area and (optionally) installs the software to run on one or more nodes. The Authorized Area can be either an entry-level directory or a subdirectory. Therefore, you can use **minst** to load software into an Authorized Area on a volume mounted on a Domain Server Processor (DSP). Note that **minst** is invoked automatically when you boot from SR10.1 boot media.

7.4.1 Running minst

Before you run the **minst** program, restore the latest version of it and the other installation tools from file 1 of the standard system software distribution media, using the following procedure. First, change your working directory to that of the Authorized Area, using **wd** (in an Aegis environment) or **cd** (in a UNIX environment). Then, issue this command:

```
rbak –dev <media> –f 1 –ms –sacl –pdt –l –force –all
```

(substitute the appropriate media type (**ct**, **f0**, or **mt** for *<media>*).

Invoke **minst** with this command:

```
/install/tools/minst <source_node>
```

where *<source_node>* provides the name of the workstation containing your Authorized Area.

For example, if your Authorized Area is on the node **//pickle**, issue this command:

```
/install/tools/minst //pickle
```

If this is the first time you have run **minst** on this node, you may be asked if the node's disk was just initialized (with **invol**).

```
Have you just run invol on this node's disk? : [ yes no ]
```

If the node's disk was just initialized, answer "yes' to this question.

You may also be asked if the “newly invol'd node” is the first SR10 node on this network. Answer “yes” to this question only if there are no other nodes on this network running an SR10.x version of Domain/OS. In this situation, you may only load and install Domain/OS during this **minst** session.

```
Is this newly invol'd node the first sr10.x node on the network? If the answer
to this question is 'yes', only the Domain/OS product is to be loaded and
installed during this MINST session. After loading and installing Domain/OS,
and exiting MINST, shut down your node and reboot it from its own disk
before loading optional software products. : [ yes no ]
```

If the node's disk was just initialized, you may be asked if you want to recatalog the target node. Answer “yes.”

```
The target node is currently: //first Do you wish to recatalog the target node?
: [ yes no ]
```

Then enter the new node name (including leading slashes) at the prompt.

Note that if you omit the Authorized Area path argument, or if you have just booted from SR10.1 boot media, **minst** prompts you for the name of the workstation to contain the Authorized Area.

7.4.2 Deleting SR10.0 Domain/OS on an Authrozied Area Node

If you are loading SR10.1 of Domain/OS into an Aa that contains SR10.0 of Domain/OS, you are given a chance to delete the SR10.0 Domain/OS release directory from your AA node. Note that if you installed software on this node using hard links, deleting the SR10.0 release directory causes the system software running on the node to be replaced by the SR10.1 release.

Type yes at the following prompt if you want to delete the SR10.0 Domain/OS release directory from your AA node before installing the SR10.1 Domain/OS release directory: If the system files on this node are hard-linked to the files in the AA, they are deleted also.

```
As a space-saving option, you may delete the following OS branches from
the current Authorized Area.

//source_node/sr10/install/ri.apollo.os.v.10.0

Do you want to delete these branches? (yes or no) : yes
```

If you do not want to delete the SR10.0 release directory (that is, if the disk on the AA node is big enough to hold the release directories for both the SR10.0 and the SR10.1 versions of Domain/OS concurrently), type **no** at the above prompt.

7.4.3 The Questions

minst then displays a set of questions. The set of questions that you see depends on the answers you supply. Default answers to the questions are indicated by a parenthesized capital letter ("(D)"); choose them by simply pressing [RETURN]. The questions about **/sau** directories and about "open" and "closed" protection models are asked (and *must* be answered) for all products, but the answers are only used when installing Domain/OS.

The entire set of questions follows.

```
Please enter your media type for this installation.
                Media types : ct  (cartridge tape)
                              mt (magnetic tape)
                              f0  (floppy)

: [ ct(D) mt f0 ]

Do you want to see all listings produced by the MINST program?

: [ yes no(D) ]

Do you wish to load all available products into your Authorized Area, or
select the products to be loaded from a list of available products?

: [ all(D) select ]

After software has been loaded into your Authorized Area, do you want to
install software to run on this node/group of nodes)?

: [ yes(D) no ]
```

Figure 7-1. minst Prompts (1 of 4).

Note that if this node is the first SR10 node on this network, the above question is not asked. In this situation, you must install Domain/OS as soon as it is loaded into the Authorized Area.

```
This program gives you the option of either (1) installing products from each
media volume as soon as they are loaded into your Authorized Area, or (2)
postponing the installation until products from an entire set of media volumes
has been loaded into the Authorized Area. Do you want to install products
separately (after each product/tape) or together (once at the end of software
loading phase)?

: [ separate together(D) ]
```

Figure 7-2. minst Prompts (2 of 4).

Note that unless you have a special reason to vary the installation options for the products you are about to install, we strongly recommend that you answer "together" to the above question.

If you are installing software to this node/group of nodes after all products have been loaded into your Authorized Area, you can:

[a] install all the products loaded into the Authorized Area by the program during this session.

[b] install a subset of the products loaded into the Authorized Area by the program during this session.

[c] install a selection of all the products available in the Authorized Area, including products previously loaded as well as products loaded by the program during this session.

Enter the letter corresponding to your choice.

: [a(D) b c]

Do you want to select options for the installation tool INSTALL++ once for the entire set of installations, or do you want to be prompted for INSTALL++ options for each media set. (A media set is either a single piece of media containing an entire product, multiple pieces of media that all together comprise a single product, or a single piece of media containing multiple products.)

: [once(D) per_media_set]

Do you want to run INSTALL++ configuring software interactively or using the default configuration as shipped by Apollo? : [defaults(D) interactively]

If you are installing software to run on your Authorized Area node, do you want to conserve space by installing objects as hard links to the Authorized Area?

: [yes no(D)]

Figure 7-3. minst Prompts (3 of 4).

Note that the Selection Component Tables at the end of this chapter provide information on sizes of various configurations for Domain/OS.

```
If you are installing Domain/OS, do you want to install software using an
(open) or (closed) protection model? If you are not installing Domain/OS,
simply press <RETURN>.

: [ open(D) closed ]

Enter a list of the /sau directories you want installed. (For example: sau2 sau3
sau4 sau5 sau6 sau7 sau8 sau10) Separate /saus by commas or spaces. You
MUST enter at least one /sau directory name, even if you are only installing
optional products. Neither NONE nor ALL are available as options.

[]

If you are updating software (rather than installing a product that has not pre-
viously been installed on the node), you can overwrite any earlier 'customiza-
tion' of the product on this node? If you do so, any product files modified or
deleted by the node owner will be replaced in this installation. Do you want
to overwrite earlier customization?

[ yes no(D) ]

Enter the name(s) of the nodes on which you want to install software. (For
example: //node) Separate names by commas or carriage returns. When
you're done entering node names, type 'exit'. Note that to install SR10.x of
Domain/OS on a node running a pre-SR10 version of system software, you
must first invol the node using a SR10.x version of the invol program.

[]
```

Figure 7-4. minst Prompts (4 of 4).

When you enter the names of nodes on which to install software, **minst** echoes your choices and prompts you to confirm them; it also prompts you to confirm your answers to the above questions. Note that if you are installing to multiple nodes, the same software configuration is installed on all of them. Then, it lets you save your answers to a personalized answer file. The file is named **answers.***<account>*, where *<account>* is your user account name, and is stored in the directory **/install/minst/answers**. If you run **minst** again, you can use the information saved in your personal answer file instead of having to reanswer the questions.

7.4.4 Selecting Products

When you have finished answering the preceding questions, the program **may** display a list of products. It displays the list if you indicated in answers to the earlier questions that you want to "select" products to be loaded into your Authorized Area rather than load "all" available products. You are then prompted to select the products you want to load into the Authorized Area, and given a chance to open a file containing general information on SR10.x installations. The file is especially useful if this is the first time you are loading SR10.x of Domain/OS.

7.4.5 Loading and Installing Products

minst now proceeds to the AA loading step. and loads the first product into the AA. If the product is distributed over more than one media volume, you are prompted to insert new volumes as necessary.

if you insert media volume 2 instead of 1, or volume 3 instead of 2) the **minst** program may fail, forcing you to start it over again from the beginning.

If the first product loaded into your AA is SR10.1 of Domain/OS, and this is the first SR10 node in this network, **minst** now installs the software and exits. If you chose to configure software interactively, you are now put into the configuration phase of the installation, described in Chapter 5 of *Installing Software with Apollo's Release and Installation Tools*. Otherwise, the default configuration is used. After the directions in the sections "Shutting Down and Rebooting Your Node(s)" and "Setting Up a Registry Site" below to complete the installation. Then invoke **minst** again to load and install optional software products.

7.4.5.1 Completing Configuration and Product Loading/Installation

If you were **not** loading the Domain/OS product for the first time on the first SR10 node in the network, then after the first product is completely loaded into the AA, one of the following three scenarios take place. The set of actions you must perform depends upon your answers to the questions described above,:

1. If you just loaded SR10.1 of Domain/OS, and this is the first SR10 node in this network, **minst** now installs the software and exits. Follow the directions in the sections "Shutting Down and Rebooting Your Node(s)" and "Setting Up a Registry Site" to complete the installation. Then invoke **minst** again to load and install optional software products.

2. If you chose to install each new product separately after loading it into the AA, and to configure products interactively, you are now put into the configuration phase of the installation, described in Chapter 5 of *Installing Software with Apollo's Release and Installation Tools*. The product is then installed on the target node(s) according to the configuration you select. After the product has been installed, **minst** asks of you want to "continue" loading products. If you want to continue, you are prompted to insert the media containing the next product or set of products.

3. If you chose to install each new product separately after loading it into the AA, but to configure products according to the defaults shipped from Apollo, the product is now installed on the target node(s) without any further action on your part. After the product has been installed, **minst** asks of you want to "continue" loading products. If you want to continue, you are prompted to insert the media containing the next product or set of products.

4. If you chose to install all products together after all desired products had been loaded into the AA, **minst** asks of you want to "continue" loading products. If you want to continue, you are prompted to insert the media containing the next product or set of products.

Note that when **minst** displays the following prompt

```
Please load volume 1 of the media for the next product into the drive
Enter <RETURN> when ready:
```

it refers to a media volume with a name like STD_STD_SFW_1 or STD_PAS_1, not a boot media volume; **do not** insert a volume of boot media at this prompt.

The single set of actions (from the three above) that applies to your situation is repeated as you continue to insert new media volumes and load products into the AA. If you chose to select available products from a list, rather than to load all available products into your Authorized Area, you are prompted at the beginning of each media volume to select products to be loaded from the products available on that volume.

When you have loaded the last product into the AA, answer "proceed" to the question presented by **minst** about "continuing" with the AA loading or "proceeding" to installation. If **minst** has already installed your products, you are now finished with the installation of SR10.1 system software and associated optional products.

If you chose to install all products together after all desired products had been loaded into the AA, and to configure products interactively, you are now put into the configuration phase of the installation, described in Chapter 5 of *Installing Software with Apollo's Release and Installation Tools*. After you have configured the products you are installing, **minst** installs those products on the target node(s) according to the configuration you selected.

If you chose to install all products together after all desired products had been loaded into the AA, but to use the default configurations shipped by Apollo for all products. **minst** now installs all selected products on the target node(s) according to the default configuration.

7.4.5.2 Shutting Down and Rebooting Your Node(s)

To complete the installation of software on one or more nodes, shut down each node and reboot the node from its own disk, as described in the section "Completing the System Software Installation" in Appendix A of *Installing Software with Apollo's Release and Installation Tools*.

7.4.5.3 Setting Up a Registry Site

If you are loading an SR10.x version of Domain/OS for the very first time on your network, follow the directions in the section "Setting Up an SR10 Registry Site," in Appendix A of *Installing Software with Apollo's Release and Installation Tools*, to set up your SR10 registry. Then perform the node administration tasks described in the section "Node Administration Tasks" of that appendix. If the target(s) of this installation were previously running SR10.0 system software, this step can be skipped.

7.4.6 Troubleshooting the minst Program

If the **minst** program fails before it has completed loading any product software into an AA, either

1. Restart the program from the beginning, or
2. Use the **distaa** program described in Chapter 3 of *Installing Software with Apollo's Release and Installation Tools* to continue loading software into the AA. Find the correct **distaa** command line to use by searching backward at the DM command line for the string "distaa command line."

   Command:
   \distaa command line

   Use the same **distaa** command line that the **minst** program used.

If the **minst** program fails after software has been loaded into an AA, but before the installation from the AA is complete, use the **install++** program described in Chapter 4 of *Installing Software with Apollo's Release and Installation Tools* to install the software contained in the AA.

7.5 Changes to distaa Since SR10.0

The **distaa** program, which is used to load software into an Authorized Area, but which, unlike **minst** does not install software to run on a node or nodes, has been modified since its release at SR10.0. The options **–c** and **–r** are no longer supported. However, a new option, **–m** <*media*> has been added. The value for <*media*> can be either **floppy**, **magtape**, or **cartridge**. The default value is **cartridge**.

In addition, the program now requires that exactly one of the options **–a** and *filename* be used on the command line. The earlier version of the program allowed the two options to be used together. The new syntax is

distaa [–a] [–e] [–f] [–v] [–m <*media*> **]** *pathname* [*filename*]

where:

–a Loads into the Authorized Area all the software available from the release media, in their entirety.

–e *max_error* Sets a limit on the number of restoration errors distaa will detect before quitting. By default, there is no error limit.

–f Forces copying of every object loaded; the default is to copy only objects which are not already present on the disk.

–m *media* Allows you to specify the type of media to be used (where *media* is one of **floppy**, **magtape**, or **cartridge**). **distaa** defaults to **cartridge**.

–v Displays all listings produced by the tool; the default is to show only error messages produced by the tool.

filename Pathname to a file that contains product-specific instructions for determining which objects to load and where on the network to load them.

Remember that you must specify one and only one of the arguments *filename* or **–a**.

The **distaa** program no longer produces baseline files; therefore, the install tree within an Authorized Area no longer contains the **distaa** subdirectory to contain those files.

7.6 Changes to config Since SR10.0

The installation tool **config** has been changed to accommodate the two architectures now available from Apollo: M680X0 and PRISM™. The following figures illustrate areas where the **config** interactive sequence has changed.

```
MACHINE-SPECIFIC SUPPORT
    The sauN directories provide support for Apollo machine types:
        sau2  – dn300, dn320, dn330
        sau3  – dsp80, dsp90
        sau4  – dn460, dn660, dsp160
        sau5  – dn550, dn560, dn570, dn580
        sau6  – dn570-T, dn580-T, dsp590-T
        sau7  – dn4000, dsp4000
        sau8  – dn3000, dsp3000
        sau10 – dn10000, dsp10000
    You must select the sau(s) for all machine types using this
    installation configuration, including the saus for any
    machines that will boot diskless from the target(s).

    Note: Unless this is a compexe'd release, only ISP compatible
          saus are available.
```

Figure 7-5. Changes to Machine-Specific Support Region of config Interactive Sequence to Support PRISM™ Architecture (*Italics* Indicate New Material).

```
HARDWARE DIAGNOSTICS
    Offline hardware diagnostics (in /sauN) are used for troubleshooting
    hardware problems. They do not need to be local to every node,
    but they should be available somewhere on the network. Disk space
    requirements differ for different sau's.

    Note: Unless this is a compexe'd release, diagnostics
          are only available for ISP compatible saus.
```

Figure 7-6. Changes to Hardware Diagnostics Region of config Interactive Sequence to Support PRISM™ Architecture (*Italics* Indicate New Material).

```
CROSS-ISP DEVELOPMENT
        We support code generation tools for two types of target processors:
        Motorola 680x0 (m68000) and Apollo PRISM (prism).  Each processor type
        requires a different set of archive libraries in /usr/lib.  Please
        select the processor target(s) for which you wish to develop software.
        You should choose at least the type which is compatible with the
        target node.
Please select the ISP type of the machine you wish to develop software
for, or 'both' for both.
```

Figure 7-7. Additions to config Interactive Sequence to Support Cross-ISP Development.

7.7 Installation and Configuration Considerations

In a few cases, there are additional configuration steps or special installation considerations that are required for proper operation of certain software components. The following subsections describe these procedures.

7.7.1 Access to Target Workstations

Workstations seeking to install software from the Authorized Area must grant full privileges to the **install** program. This condition is the default. However, users may change it using **lprotect**. The command

```
lprotect –rmtroot none
```

grants no privileges to remote processes running as root; under this condition, **install** will fail.

To determine the privileges currently granted by a workstation to remote processes running as root, issue this command:

```
/etc/lprotect
```

In order for **install** to succeed, **lprotect** must return a message declaring

```
All remote root requests are honored. (–rmtroot all)
```

If **lprotect** returns any message but this, **install** will fail. To provide the access necessary for **install** to succeed, issue this command from a shell at the target workstation:

```
/etc/lprotect –rmtroot all
```

7.7.2 Accessing Tools

In step 3 of the "Accessing Tools" procedure, at the top of page 3-4, of *Installing Software with Apollo's Release and Installation Tools* the **rbak** command should include the –all option as follows:

```
./rbak dev media –f 1 –ms –sacl –pdt –l –force –all
```

7.7.3 Booting and Installing your First SR10 Authorized Area

The two situations below are not documented completely in *Installing Software with Apollo's Release and Installation Tools*. We expect that they apply to very few sites.

Situation A You want to create your first SR10 Authorized Area on a node that has a magnetic tape drive, but neither a cartridge tape nor a floppy disk drive. Your boot media are floppy disks.

Situation B You want to create your first SR10 Authorized Area on a node that has a magnetic tape drive, but neither a cartridge tape nor a floppy disk drive. Your boot media are cartridge tapes.

Modify the procedures described in Appendix A of the *Installing Software with Apollo's Release and Installation Tools* to perform the tasks required by the above situations as described in the following subsections.

7.7.3.1 Situation A

In this situation you want to create your first SR10 Authorized Area on a node that has a magnetic tape drive, but neither a cartridge tape nor a floppy disk drive. Your boot media are floppy disks. Here you must use two nodes to load the first SR10 Authorized Area on your network: Node1 has a floppy disk drive and, if possible, a monitor; Node2 has a magnetic tape drive. Use the following procedure to boot the nodes and install the Authorized Area.

1. Follow the directions in Appendix A of *Installing Software with Apollo's Release and Installation Tools* to boot Node1 from floppy disks. However, just before you complete the boot process by typing GO (page A-19), insert the floppy labeled **flp5_remote*n*** or **flp8_remote*n*** into the floppy drive. (Here *n* refers to the /**sau*n*** corresponding to Node2's machine type.)

2. Load the contents of the floppy to Node1's disk. For example, if Node2 is a DSP500, type

   ```
   ) cf /install/load_remote5
   ) go
   ```

3. The Display Manager (DM) or Server Process Manager (**spm**) process starts up. Log in as **user**.

4. Start the **netman** process on Node1. NOTE: If you are using a console, start **netman** as a background process.

5. Now go to Node2. If Node2 is a DSP, use an **emt** window or a terminal over an SIO line to access it. Boot Node2 diskless from Node1. When the system boots, it runs the DM or the **spm** process on Node2 as appropriate. Log in as **user**.

6. In a shell, run the **calendar** program on Node2.

7. Run the **invol** program on Node2.

8. Run the **calendar** program again on Node2.
9. Mount Node2's boot volume on Node1's file system.
10. Run the **minst** program to create an Authorized Area on Node2. The program takes as an argument the pathname of the Authorized Area to be created; enter the name of Node2's boot volume as you mounted it in step 11. For example:

    ```
    $ /install/tools/minst  //node1/node2
    ```

    Follow the directions for using **minst** (Section 7.4 above) to load software into an Authorized Area and install software on Node2.
11. After **minst** completes loading the Authorized Area and installing, software, unmount Node2's boot volume from Node1's file system. Then shut down and reboot Node2 from its own disk or storage module.
12. At this point, you may wish to re-catalog and rename Node1 and Node2.
13. Now go to the section "Setting Up an SR10 Registry Site" in Appendix A of the installation manual and continue from there.

7.7.3.2 Situation B

In this case, you want to create your first SR10 Authorized Area on a node that has a magnetic tape drive, but has neither a cartridge tape nor a floppy disk drive. Your boot media are cartridge tapes. Here, you must use two nodes to load the first SR10 Authorized Area on your network; Node1 has a cartridge tape drive and, if possible, a monitor, Node2 has a magnetic tape drive. Use the following procedure to boot the nodes and install the Authorized Area.

1. Follow the directions in Appendix A of *Installing Software with Apollo's Release and Installation Tools* to boot Node1 from cartridge tape.
2. Load the "aegis_small" Authorized Area onto Node1; be sure to select the /sau directories for both Node1 and Node2. Again, use the procedures in Appendix A.
3. Install all the software from that Authorized Area to run on Node1. Again, use the procedures in Appendix A.
4. Start the **netman** process on Node1. NOTE: If you are using a console, start **netman** as a background process.
5. Now go to Node2. If Node2 is a DSP, use an **emt** window or a terminal over an SIO line to access it. Boot Node2 diskless from Node1.
6. In a shell, run the **calendar** program on Node2.
7. Run the **invol** program on Node2.
8. Run the **calendar** program again on Node2.
9. Mount Node2's boot volume on Node1's file system.

10. Run the program **minst** to create an Authorized Area on Node2. The program takes as an argument the pathname of the Authorized Area to be created; enter the name of Node2's boot volume as you mounted it in step 11.

    ```
    $ /install/tools/minst //node1/node2
    ```

    Follow the directions for using **minst** (Section 7.4 above) to load software into an Authorized Area and install software on Node2.

11. At this point, you may wish to re-catalog and rename Node1 and Node2.
12. Unmount Node2's boot volume from Node1's file system. Then shut down and reboot Node2 from its own disk or storage module.
13. Now go to the section "Setting Up an SR10 Registry Site" in Appendix A of the installation manual and continue from there.
14. When you've finished work on Node2, you may want to delete the Authorized Area you originally created on Node1.

7.7.4 Media Installs to Small Disks

Installing from media (cartridge tape or floppies) is supported *only* on nodes with disks of at least 50 megabytes. You cannot install SR10.1 from media on a 30-megabyte disk. You can install a small Aegis environment version of SR10.1 over the network on a 30-megabyte disk. There are notes in the installation guide specifically directed at dealing with a 50-megabyte disk installation. If your network only has 30-megabyte disks, contact your service representative to install SR10.1 onto your network.

7.7.5 Media Boot

If you boot a node from media, you must complete the installation of software onto the node before any other nodes can boot diskless, using this node as a partner.

If booting from cartridge tape, retensioning the tape before attempting to boot. Before shutting down the node, write-protect the cartridge, insert it in the drive, and issue this command:

```
rbak –dev ct –reten
```

If booting off the tape fails, retension it at another workstation and try again. Retensioning takes about 90 seconds.

7.7.6 Installing from a Limited Authorized Area

If you use the **config** or **install++** program to configure Domain/OS and the Authorized Area contains only a subset of the /sau directories, the program may display the following message when you exit from configuration:

```
All queries answered for product: os  10.1

CONFIG> exit
***WARNING*** The following products may not be installed due to missing
              objects in the authorized area.  Use the 'validate' command for
              further information, or deselect the products in question.
1. os 10.0
Do you still want to exit? (y/n):
```

If you see the above message, respond with **n** and enter the following command to ensure that there are no errors:

```
CONFIG> val os

The installation of os 10.1 should succeed
```

You can then exit the configuration and install the software.

7.7.7 Override Files

If you are making a set of overrides for Domain/OS, as described in the "Managing Subsequent User Installations" section of Chapter 3 of *Installing Software with Apollo's Release and Installation Tools*, pre-answer the following question:

```
Do you want detailed explanatory information during this configuration?
```

with "no". If you do not do this, extra messages will sometimes be displayed on the screen as users configure the operating system. These extra messages do not affect the installation of the software, but might be confusing. The override files shipped by Apollo have this question pre-answered with "no".

7.7.8 Creating an AA on a Pre-SR10 Node

The "Loading Software into an Authorized Area" section in Chapter 3 of *Installing Software with Apollo's Release and Installation Tools* describes how to use the **media_install_opt** script to load multiple products into an Authorized Area (AA). If you are creating an AA on a pre-SR10 node, you must use **minst**.

7.7.9 Adding Subcomponents to the AA

We do not recommend that you use **rbak** to pull software directly from the distribution media because installation of a function is seldom as simple as just copying one file. If you need to add functionality that can be installed from your Authorized Area, you should use the **cfgsa** and **distaa** tools to achieve this task. To update an AA with another subcomponent:

1. Use the **cfgsa** tool to create override and AA selection files as described in Chapter 3 of *Installing Software with Apollo's Release and Installation Tools*. This new configuration should be a superset of the old product in the AA. If it is a subset of the product in the AA, users will not be able to install all of the previously available functionality.

2. Use the **mv** or **mvf** command to move the ov.*new_file* to to the **/install/overrides/ri.apollo.***product_name._version* directory.

3. Run **distaa** with the new selection file (aa.*new_file*) you just created.

7.7.10 Inability to Restore Release Index

If for any reason the release index for a product cannot be restored from media, then the remainder of that product will not be restored even if the **–f** option to **distaa** was used. **distaa** will inform you of this condition with the following message:

```
WARNING: Unable to restore release index for product version. None of the
         remainder of this product will be restored from media.
```

7.7.11 Removing the AA /install Directory

After completing installation of the SR10.1 version of Domain/OS, you may decide to delete the install directory within your AA to recover disk space. Information about deleting the directory is in the section "Recommendations for Small Networks" in Appendix A of *Installing Software with Apollo's Release and Installation Tools.*

If you do decide to take this step, first make a copy of the Domain/OS release index file. This file is located at *authorized_area***/install/ri.apollo.os.v.10.1/ri.apollo.os.v.10.1**, where *authorized_area* should be replaced by the pathname of your AA (if it is not at the entry (/) level). Save the copy of the Domain/OS release index somewhere else on the system. After you delete the install tree, create an empty *authorized_area***/install/ri.apollo.os.v.10.1** directory and move the release index into it.

You must preserve this file in order to be able to install optional products on SR10.1 nodes. The AA holding the optional product(s) must contain a Domain/OS release index in *authorized_area***/install/ri.apollo.os.v.10.1/ri.apollo.os.v.10.1** or a link to such a file on another node. You do not need the file to install products on pre-SR10 nodes, even if they are compatible with SR10.1 as well as pre-SR10 system software.

Note that if you don't save the release index file somewhere on your system, you must restore it from file #2 of the standard software release media in order to install optional software products on SR10.1 nodes.

7.7.12 The uucp Account Log-in Shell

You must change the log-in shell for the **uucp** account from **/usr/lib/uucp/uucico** to **/usr/lib/uucp/uucico.real**. Otherwise, UUCP programs will not function correctly. Use **/etc/edrgy** to modify the registry account for **uucp**.

7.7.13 BSD /usr/lib/find

The BSD **/usr/lib/find** directory is installed directly in the **/bsd4.3/usr/lib** directory tree. If several systems use a single node for their UNIX software, you may want to copy this directory to each disked node's **/sys/node_data** directory and then make **/bsd4.3/usr/lib/find** a link to **/sys/node_data/find**. This procedure will make sure that each disked node has its own find database.

7.7.14 Optional Product Installations

The following subsections describe considerations and bugs that exist in the installation procedures for optional products.

7.7.14.1 Domain/Dialogue and Domain/PAK Installation

The installation procedures for Domain/Dialogue and Domain/PAK do not ask you if you want help files or online examples. They will install the files *only* if the /sys/**help** or /**domain_examples** directories already exist. If wish to get the help files or online examples when you install either of these products, create the directories, if they do not exist, before starting the installation procedure.

7.7.14.2 NFS

The NFS installation procedure does not abort if the /**bsd4.3** or /sys**5.3** directory does not exist; instead it continues, and produces multiple error messages, but does not install anything.

7.7.14.3 Domain/Dialogue, Domain/Pascal, and D3M

There is a bug in the release indexes for Domain/Dialogue™, Domain/Pascal, and D3M™ that causes an error on exiting the CONFIG phase of the installation. After you configure one of these products and attempt to exit, you get the message:

```
CONFIG> exit
***WARNING*** The following products may not be installed due to missing
              objects in the authorized area. Use the 'validate' command for
              further information, or deselect the products in question.
1. product name
```

You get this message whether or not you have validated the product during the configuration process. You can safely ignore this message and answer 'y' to the prompt:

```
Do you still want to exit
```

For example:

```
CONFIG> val d3m

        The installation of d3m 6.0 should succeed

CONFIG> exit

***WARNING*** The following products may not be installed due to missing
              objects in the authorized area. Use the 'validate' command for
              further information, or deselect the products in question.
1. d3m 6.0
Do you still want to exit? (y/n): y
New configuration file saved in /conf
```

7.7.14.4 SPE Installation

The following two problems exist with the SPE installation:

- If the **/domain_examples** and /sys/**ins** directories *do not* exist on your node and you respond that examples and insert files for SPE *should* be installed (copy) you get these warnings:

  ```
  WARNING: no /domain_examples directory - not installing SPE examples
  WARNING: no /sys/ins directory - not installing SPE insert files
  ```

  Then the program creates the directories and copies the files as you requested.

- If the **/domain_examples** and /sys/**ins** directories *do* exist on your node and you respond that examples and insert files for SPE *should not* be installed (none), then the program copies the files despite your request that it not install them.

7.7.14.5 TranScript Installation

If the /**domain_examples**, /bsd4.3/usr/**man**, and /sys5.3/usr/**catman** directories do not exist and are not links, the TranScript* installation procedure displays the following warning messages:

```
WARNING: /sys5.3/usr/catman is a link - not installing sys5.3 TranScript man pages
WARNING: /bsd4.3/usr/man is a link - not installing bsd4.3 TranScript man pages
WARNING: no /domain_examples directory - not installing TranScript examples
```

However, the procedure will create the required directories and install the **man** pages and examples.

7.8 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.

* TranScript is a trademark of Adobe Systems, Inc.

- 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, the SR10.1 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 SR10.1.
- A **preserve.list** file (optional, created by the user) 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 AA — that is, a source area for installations over the network — its **/install** directory contains the following objects in addition to the files and directories listed above:

- 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 SR10.1 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 SR10.1 nodes but are not being re-released with SR10.1.
- An **/install/<site>** directory created by the site system administrator and containing site-specific configuration, override, and Authorized Area selection files.

The following directories and files have been added to the Authorized Area tree as a result of the replacement of media installation scripts by the **minst** program:

/install/templates/apollo/minst	(directory)
/install/minst/answers	(directory created by MINST at user site)
/install/templates/apollo/minst/questions	(file)
/install/tools_sr9	(directory)

The **/install/tools_sr9** directory contains **obj** versions of all the tools shipped in the **/install/tools** directory. (Tools in the **/install/tools** directory are in COFF format.) Users can run the tools in the **/install/tools_sr9** directory from nodes running the SR9.7 version of the operating system.

7.9 Installation Tools for Solution Suppliers

All systems, including nodes that do not have a SysV environment installed, have a /sys5.3/**bin** directory. On systems that have only BSD or Aegis, this directory contains the following commands: **cat, chgrp, chmod, chown, cmp, cp, cpio, diff, expr, find, grep, id, ln, ls, mkdir, mv, rm, rmdir, sed, sort, sum, tar, uniq,** and **wc**. In addition, we always supply a SysV Bourne shell in /etc/sys_sh. These files enable solution suppliers to provide a single (SysV) script that will correctly install software on all nodes, independent of the installed environments.

We also provide a new script, /etc/**invoke_script**, that invokes either an Aegis shell script or a Bourne shell script, depending upon the environment or environments installed on the system.

7.10 Media Types

We distribute SR10.1 on magnetic tapes, ¼ in. streaming cartridge tapes, 5-¼ in. floppy disks, or 8 in. floppy disks as indicated in the following subsections.

7.10.1 Streaming Cartridge Tapes

Streaming cartridge tape distributions of SR10.1 consist of the following tapes:

CRTG_STD_SFW_1
CRTG_STD_SFW_2
CRTG_STD_SFW_3
CRTG_STD_SFW_BOOT

7.10.2 Magnetic Tapes

Magnetic tape distributions of SR10.1 consist of the following tapes:

MT_STD_SFW_1
MT_STD_SFW_2
MT_STD_SFW_3

The magnetic tape distribution also includes the following floppy boot disks:

FLP8_PREP2
FLP8_BOOT2
FLP8_BASIC_1
FLP8_BASIC_2
FLP8_TOOLS_1
FLP8_TOOLS_2
FLP8_LIBRARY_1
FLP8_LIBRARY_2
FLP8_LIBRARY_3
FLP8_LIBRARY_4
FLP8_LIBRARY_5
FLP8_PREP4
FLP8_BOOT4
FLP8_REMOTE2
FLP8_REMOTE3
FLP8_REMOTE4
FLP8_REMOTE5
FLP8_REMOTE6
FLP8_REMOTE7
FLP8_REMOTE8
FLP8_UCODE4

7.10.3 Floppy Disks

SR10.1 on floppy disks will be available only after FCS. You should consult your sales representative regarding availability.

The floppy disk distribution consists of eight sets of disks, available in either 5-¼ or 8 in. sizes. The following disks are the first in each set of the 5-¼ in. distribution:

FLP5_STD_SFW DISK_1.01
FLP5_STD_SFW DISK_2.01
FLP5_STD_SFW DISK_3.01
FLP5_STD_SFW DISK_4.01
FLP5_STD_SFW DISK_5.01
FLP5_STD_SFW DISK_6.01
FLP5_STD_SFW DISK_7.01
FLP5_STD_SFW DISK_8.01

The following disks are the first in each set of the 8 in. distribution:

FLP5_STD_SFW DISK_1.01
FLP5_STD_SFW DISK_2.01
FLP5_STD_SFW DISK_3.01
FLP5_STD_SFW DISK_4.01
FLP5_STD_SFW DISK_5.01
FLP5_STD_SFW DISK_6.01
FLP5_STD_SFW DISK_7.01
FLP5_STD_SFW DISK_8.01

The number of disks in each set is not known at the time of this writing.

7.11 Canned Selection Files and Configurations

SR10.1 is shipped with 13 pairs of selection and override files, plus one configuration file that works with any pair of selection/override files, that you can use to install your software. *Installing Software with Apollo's Release and Installation Tools* describes selection, override, and configuration files and their purposes in detail. The following subsection describes the components that are specified by the selection files. These descriptions are followed by tables listing the selection components and their sizes. The last subsection describes what software is installed on user nodes with each of these canned configurations.

7.11.1 Selection Component Descriptions

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

sysboot — The program that loads Domain/OS operating system into the computer's memory.

install utilities — Domain/OS commands used by one or more of the installation programs.

bscom

Boot shell utilities for use from the Phase II boot shell.

domain_examples

A directory containing online programming examples.

base etc

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

etc/tcp

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

lib

The Domain/OS library routines.

saus

Directories containing the stand-alone utilities for all Apollo machine types supported by version 10 of Domain/OS, including sau2, sau3, sau4, sau5, sau6, sau7, and sau8.

sau diagnostics

Offline hardware diagnostics used for troubleshooting hardware problems.

sau_sys help

Help files for offline hardware diagnostics.

sr9.7_compatibility

A directory containing files to be used in a network where some nodes are running SR10.1 and some nodes are running SR9.7.

sys

The top-level Domain/OS system directory.

optional fonts

All fonts *not* considered "standard." Non-standard fonts include foreign-language fonts in all sizes, some graphics fonts (chess, symbol), courier-boldoblique, courier-oblique, times-bolditalic, and times-italic fonts in all sizes, an Old-English font, and unusual point sizes (7, 9, 14, 18, 24) for some of the standard fonts.

The standard system fonts (installed on every node) are:

/sys/dm/fonts/b.12
/sys/dm/fonts/blank
/sys/dm/fonts/brush_font
/sys/dm/fonts/courier-bold10
/sys/dm/fonts/courier-bold12
/sys/dm/fonts/courier10
/sys/dm/fonts/courier12
/sys/dm/fonts/din_f7x11
/sys/dm/fonts/f16.b
/sys/dm/fonts/f16.i
/sys/dm/fonts/f5x7
/sys/dm/fonts/f5x7.iv
/sys/dm/fonts/f5x9
/sys/dm/fonts/f5x9.b
/sys/dm/fonts/f5x9.iv
/sys/dm/fonts/f7x13
/sys/dm/fonts/f7x13.b
/sys/dm/fonts/f7x13.b.iv
/sys/dm/fonts/f7x13.i
/sys/dm/fonts/f9x15
/sys/dm/fonts/f9x15.iv
/sys/dm/fonts/gray64.5x10
/sys/dm/fonts/helvetica-bold10
/sys/dm/fonts/helvetica-bold12
/sys/dm/fonts/helvetica-boldoblique10
/sys/dm/fonts/helvetica-boldoblique12
/sys/dm/fonts/helvetica-oblique10
/sys/dm/fonts/helvetica-oblique12
/sys/dm/fonts/helvetica10
/sys/dm/fonts/helvetica12
/sys/dm/fonts/i.12
/sys/dm/fonts/icons
/sys/dm/fonts/legend
/sys/dm/fonts/legend.1280bw
/sys/dm/fonts/legend.19l
/sys/dm/fonts/nonie.r.16
/sys/dm/fonts/nonie.r.8
/sys/dm/fonts/r.12
/sys/dm/fonts/scvc5x10.f.r
/sys/dm/fonts/scvc5x10.r.b
/sys/dm/fonts/scvc5x10.r.r
/sys/dm/fonts/scvc8x16.i.r
/sys/dm/fonts/scvc8x16.r.r
/sys/dm/fonts/sideways8x5.r.r
/sys/dm/fonts/std
/sys/dm/fonts/std.1280bw
/sys/dm/fonts/std.1280color
/sys/dm/fonts/std.19l
/sys/dm/fonts/std.color
/sys/dm/fonts/times-bold10
/sys/dm/fonts/times-bold12
/sys/dm/fonts/times-roman10
/sys/dm/fonts/times-roman12
/sys/dm/fonts/vt100l
/sys/dm/fonts/vt100l.b
/sys/dm/fonts/vt100l.dhb
/sys/dm/fonts/vt100l.dht
/sys/dm/fonts/vt100l.dw
/sys/dm/fonts/vt100s
/sys/dm/fonts/vt100s.b
/sys/dm/fonts/vt100s.dhb
/sys/dm/fonts/vt100s.dht
/sys/dm/fonts/vt100s.dw

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

systest
A directory containing various online system tests and exercisers.

systest/ssr_util
A directory containing field service utilities.

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

com
A directory containing a large set of Aegis environment commands.

sys/help
A directory containing help files for the Aegis environment.

sys/ins
Apollo-specific include files for Aegis software development.

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

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

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

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

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

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

sys5.3 usr
SysV environment commands that reside in **/etc** and either have non-identical counterparts or no counterparts in a BSD environment.

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

usr/games
A directory containing a collection of games including games from the SysV distribution, the BSD distribution, and some developed by Apollo.

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

usr/apollo/man
A directory containing manual pages with detailed descriptions of the Domain/OS system calls. These manual pages contain the same information as the Help files for these calls.

7.11.2 Selection Component Tables

The following tables list the software components that are installed into your Authorized Area if you use the predefined selection files. They also specify the sizes of each component of all the software that is installed. They should help you determine the particular selection file that is most appropriate for your use and disk sizes. The first table covers the small and medium sized configurations; the second table covers the large configurations. The third table lists the sizes of the SAU software for each machine type.

Note that these tables give the size of the Authorized Area and *not* the size of the software that will be installed on user nodes. You can **reduce** the size of the software that is installed on the user nodes by configuring a subset of the installed software. For example, you need only configure a single **/sau***n* directory for each node (that does not serve as a partner to diskless node of a different machine type).

Each row in the first two tables corresponds to a selection component that is determined by the release index. As a general rule, the row identifies the directory that contains the software to be installed. However, some directories, such as **/etc**, are split among several selections, and some selections determine the software that is installed in several directories. We have also collapsed all the SAU selections into a single row.

Each Configuration Code column corresponds to a particular predefined configuration. For example, the AVM column defines the contents of the **aa.aegis_sys5.3_medium** selection file. The key that follows the tables lists the meanings of the one-character configuration code components.

NOTE: The disk where the Authorized Area (AA) will be located must have a minimum of 25 megabytes of free space for use during the installation from media *in addition to the disk space listed in the following tables.*

Table Key:

A	=	aegis
B	=	bsd4.3
V	=	sys5.3
S	=	small
M	=	medium
L	=	large
P	=	prog (for programmers)
*	=	See following table for individual SAU sizes
#	=	See following table for individual SAU diagnostic sizes

TABLE 7-1. AA Size for Small and Medium Selections

Component	Size (MB)	Configuration Code						
		AS	*ASP*	*AM*	*BM*	*VM*	*ABM*	*AVM*
sysboot	0.01	X	X	X	X	X	X	X
install utilities	0.07	X	X	X	X	X	X	X
bscom	0.5							
domain_examples	1.3							
base etc	2.5	X	X	X	X	X	X	X
etc/tcp	0.3			X	X	X	X	X
lib	4.8	X	X	X	X	X	X	X
saus (2-8)*	11.4	X	X	X	X	X	X	X
sau diagnostics (2-8)#	12.1							
sau_sys help	0.05							
sr9.7_compatibility	3.0							
sys	7.5	X	X	X	X	X	X	X
optional fonts	0.4			X	X	X	X	X
base sys5.3 commands	0.3	X	X	X	X	X	X	X
systest	3.1							
systest/ssr_util	1.5							
base usr	1.7	X	X	X	X	X	X	X
com	1.7	X	X	X			X	X
sys/help	1.4							
sys/ins	1.1		X	X			X	X
sys/source	0.3							
usr/apollo/include	0.7		X	X	X	X	X	X
usr/apollo/man	1.6							
usr/games	2.8							
usr/new	4.5							
bsd4.3	10.7				X		X	
bsd4.3 etc	0.3				X		X	
bsd4.3 usr	0.3				X		X	
sys5.3	13.2					X		X
sys5.3 etc	0.2					X		X
sys5.3 usr	0.3					X		X
Total (Approximate)	90	30	32	33	41	43	44	46

NOTE: An additional minimum of 25 megabytes of free space must be available during the installation from media.

TABLE 7-2. AA Size for Large Selections

Component	Size (MB)	Configuration Code					
		AL	*BL*	*VL*	*ABL*	*AVL*	*ABVL*
sysboot	0.01	X	X	X	X	X	X
install utilities	0.07	X	X	X	X	X	X
bscom	0.5	X	X	X	X	X	X
domain_examples	1.3	X	X	X	X	X	X
base etc	2.5	X	X	X	X	X	X
etc/tcp	0.3	X	X	X	X	X	X
lib	4.8	X	X	X	X	X	X
saus (2-8)*	11.4	X	X	X	X	X	X
sau diagnostics (2-8)#	12.1	X	X	X	X	X	X
sau_sys help	0.05	X	X	X	X	X	X
sr9.7_compatibility	3.0	X	X	X	X	X	X
sys	7.5	X	X	X	X	X	X
optional fonts	0.4	X	X	X	X	X	X
base sys5.3 commands	0.3	X	X	X	X	X	X
systest	3.1	X	X	X	X	X	X
systest/ssr_util	1.5	X	X	X	X	X	X
base usr	1.7	X	X	X	X	X	X
com	1.7	X			X	X	X
sys/help	1.4	X			X	X	X
sys/ins	1.1	X			X	X	X
sys/source	0.3	X			X	X	X
usr/apollo/include	0.7	X	X	X	X	X	X
usr/apollo/man	1.6		X	X	X	X	X
usr/games	2.8		X	X	X	X	X
usr/new	4.5		X	X	X	X	X
bsd4.3	10.7		X		X		X
bsd4.3 etc	0.3		X		X		X
bsd4.3 usr	0.3		X		X		X
sys5.3	13.2			X		X	X
sys5.3 etc	0.2			X		X	X
sys5.3 usr	0.3			X		X	X
Total(Approximate)	90	65	71	74	76	78	90

NOTE: An additional minimum of 25 megabytes of free space must be available during the installation from media.

TABLE 7-3. SAU and SAU Diagnostic Sizes

Machine Type	SAU Size (MB)	Diagnostic Size (MB)
sau2 (DN3x0)	1.4	0.6
sau3 (DSPx0)	1.4	0.6
sau4 (DN/DSPx60)	1.7	3.0
sau5 (DN/DSP5x0)	1.5	2.5
sau6 (DN/DSP5x0-T)	1.8	2.9
sau7 (DN/DSP4000)	1.8	1.1
sau8 (DN/DSP3000)	1.8	1.4
Total	11.4	12.1

7.11.3 Software Installed on User Nodes

The following subsections describe what is installed (or not installed) for each of the canned configurations for base software that we ship, and provides information on the size of the software that is installed. Please refer to section 4.5.1 for information about these high-level components.

7.11.3.1 Small Aegis

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

- Any or all /**sau** directories you need (you specify which to install), but you do not get hardware diagnostics.
- The /sys5.3/**bin** "guaranteed commands" used to install third-party applications.
- The /**usr/apollo/bin** commands.
- The Apollo network administration utilities: **cpboot, edns, lcnet, netmain, probenet**, routing tools and registry tools.
- Support for printing but not in a mixed network (SR9.7 and SR10 and SR10.1).

You do *not* get the following things:

- /**bscom** (boot shell commands)
- /**domain_examples**
- /**sr9.7_compatibility trees**
- /systest or /systest/**ssr_util**
- /**sys/help**
- /**sys/source**
- A large set of optional fonts

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

The small Aegis configuration requires approximately 17 megabytes for one SAU or 28 megabytes for all SAUs.

7.11.3.2 Small Aegis for Programmers

This is a minimum Aegis environment with support for software development. You get everything as described in Small Aegis with these additions:

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

The small Aegis configuration for programmers requires approximately 22 megabytes for one SAU or 32 megabytes for all SAUs.

7.11.3.3 Medium Aegis

This is a more complete Aegis environment. You get everything as described in Small Aegis for Programmers, with these additions:

- Support for printing in a mixed (SR9.7, SR10, and SR10.1) network
- The large set of optional fonts
- TCP/IP administration utilities
- TCP/IP user utilities (**ftp**, **telnet**)
- The font editing utilities
- The full subsets of /**com**, including these:
  - Open System Toolkit utilities (**crty**, **crtyobj**)
  - Serial line communication commands (**em3270.xxx**, **siorf**,**siotf**)
  - Spelling checker software (**fserr**)

The Aegis medium configuration requires approximately 31 megabytes for one SAU or 41 megabytes for all SAUs.

7.11.3.4 Large Aegis

This configuration includes everything available in an Aegis environment. In addition to the things in Medium Aegis, it picks up the following:

- Hardware diagnostics
- **/bscom**
- /systest including /systest/ssr_util
- **/domain_examples**
- /sr9.7_compatibility
- /sys/help
- /sys/source

7.11.3.5 Medium BSD and Medium SysV

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

- To specify the **sau***n* directories you need, but you do not get hardware diagnostics.
- The /sys5.3/**bin** "guaranteed commands" used to install third-party applications. (These are part of standard SysV environment anyway).
- The Apollo network administration utilities: **cpboot, edns, lcnet, netmain, probenet**, routing tools and registry tools.
- All standard **bsd4.3** or sys5.3 trees except where noted below.
- **/usr/apollo/bin** commands
- The large set of optional fonts
- The high-level debugger (**dde**)
- TCP/IP administration files and utilities
- TCP/IP utilities (such as **ftp, rlogin**)
- **/usr/include** (***.h** files for BSD or SysV calls)
- **/usr/include/apollo** (***.h** files for Domain/OS calls)
- Support for UNIX mail
- Support for UNIX printing
- Support for UNIX program development (**ld, make, sccs**, etc.)

You do not get the following:

- **/bscom** (boot shell commands)
- **/domain_examples**
- **/sr9.7_compatibility** trees
- **/systest** or **/systest/ssr_util**
- Support for Domain hardcopy (printing)
- Font editing utilities
- **/usr/apollo/man** (manual pages for Domain/OS calls)
- **/usr/man**
- **/usr/games**
- **/usr/new**
- Support for UUCP
- Support for BSD or SysV graphics
- Support for BSD or SysV text processing (**nroff**, **troff**, etc.)

The BSD medium configuration requires approximately 27 megabytes for one SAU or 37 megabytes for all SAUs. The SysV medium configuration requires approximately 29 megabytes for one SAU or 39 megabytes for all SAUs.

7.11.3.6 Large BSD and Large SysV

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

7.11.3.7 Combination Medium Configurations

The medium combined configurations (Aegis and BSD, Aegis and SysV) are direct concatenations of the individual ones listed above, except they do not include the font utilities that medium Aegis includes.

The combined Aegis and BSD medium configuration requires approximately 32 megabytes for one SAU or 41 megabytes for all SAUs. The combined Aegis and SysV medium configuration requires approximately 33 megabytes for one SAU or 41 megabytes for all SAUs.

7.11.3.8 Combination Large Configurations

The large combined configurations (Aegis and BSD, Aegis and SysV, Aegis and BSD and SysV) include everything available in the respective environments.

The combined large configuration for all three environments requires approximately 75 megabytes for one SAU or 95 megabytes for all SAUs.

Chapter 8: Documentation

This chapter describes the new documentation at SR10 and SR10.1. 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 SR10 and SR10.1. The rest of the Chapter describes the changes in individual documents.

8.1 Introduction to SR10 Documentation

This introduction contains the following sections:

- Organization of the set at SR10 and SR10.1
- Changes in the documentation to help you find information more easily
- Changes to Documentation Order Numbers and Ordering
- SR10 books and their relation to earlier versions

8.1.1 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 following:

- Task-oriented user's guides, such as *Using Your Aegis Environment*
- Command references, such as the *BSD Command Reference*
- Programmer's or Calls references, such as the *SysV Programmer's Reference*
- System administration guides, such as *Managing Aegis System 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 like 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.) Novice 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.

8.1.2 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*
- Online manuals list available through **help** or in **/install/doc/apollo**
- 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 wraparound 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 pocket guide that also lists all the manuals in the documentation set, by category and icon. Use this guide 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. Because the help facility is only available when the Aegis environment is installed, we have also made this list available in the file **/install/doc/apollo/os.v.10.0__manuals** (note the two underscores). This file is installed on all systems.

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

At SR10, many 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 hardcopy book. You do not even need to have on hand hardcopy 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 First Customer Shipment; it is scheduled to be available in September.

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 online, via the **man** command, and summaries of the OS call interfaces were available through the **help** facility. Now, you can also use the **help** 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.

8.1.3 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.

8.1.3.1 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.

8.1.3.2 Ordering Manuals at SR10

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 enables us to maintain two versions of one manual at the same time. As a result, you can still order SR9.7 version manuals. You can order the SR9.7 version of a manual, the SR10 version, or simply order the latest version. To order the SR9.7 version, specify both the *Manual Number* and the *Revision Number*. For example, to order the SR9.7 version of the software installation manual *Installing Domain Software*, Specify Manual No. 008860 Rev. 02. To specifically order an SR10 manual, specify the six digit *Base Manual Number* plus the three digit *Version Suffix*. So, to order the SR10 version of the installation manual, *Installing Software with Apollo's Release and Installation Tools*, specify Manual No. 008860-A00. To order the latest version of the manual, simply specify the six-digit *Base Manual Number*. Thus, to order the latest version of the software installation manual, specify Manual No. 008860.

When we mention manual part numbers in our documentation, we usually only mention the base number and not the suffix. As with ordering manuals, 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.

8.1.3.3 SR10 and Pre-SR10 Manuals

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 full part number that support 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 by entering **help manuals** in an Aegis shell or by looking at the **/install/doc/apollo/os.v.10.0__manuals** file (note that there are *two* underscores before **manuals**).

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.

TABLE 8-1. Domain/OS Documentation Changes at SR10

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 02
Getting Started with Domain/OS	002348-A00	Getting Started with your Domain System Getting Started with your Domain/IX 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 Domain Documentation Quick Reference	011242-A00 002685-A00	Technical Publications Overview	002685, Rev 07
Using TCP/IP Network Applications	008867-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

TABLE 8-2. Domain/OS Documentation Changes at SR10 (Continued)

SR10 Manual	SR10 Part Number	Corresponding Pre-SR10 Manual	Pre-SR10 Part Number
Aegis Utilities Reference	009414-A00	Domain System Utilities Reference	009414, Rev 04
Using Your BSD Environment	011020-A00	Domain/IX System User's Guide	005803, Rev 00
BSD Command Reference	005800-A00	Domain/IX Command Reference for BSD4.2	005800, Rev 04
Using Your SysV Environment	011022-A00	Domain/IX System User's Guide	005803, Rev 00
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/OS and Domain Routing in an Internet	005694-A00	Managing the Domain Environment in an Internet	005694, Rev 01

TABLE 8-3. Domain/OS Documentation Changes at SR10 (Continued)

SR10 Manual	SR10 Part Number	Corresponding Pre-SR10 Manual	Pre-SR10 Part Number
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, Volume 1	007196-A00	Domain System Call Reference	007196, Rev 01
Domain/OS Call Reference, Volume 2	012888-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

TABLE 8-4. Domain/OS Documentation Changes at SR10 (Continued)

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

8.1.4 Manuals at SR10.1

The following manuals describing Personal Workstation hardware are new or revised at SR10.1:

- *Domain Hardware Site Planning Specifications* (009859-A02)
- *Unpacking and Installing Your Domain Series Personal Workstations and Servers* (007857-A01)
- *Operating the Domain Series Personal Workstations and Servers* (007858-A00)
- *Servicing the Domain Personal Workstations and Servers* (007859-A01)
- *Using Domain Diagnostics Volume 1* (009329-A01)
- *Using Domain Diagnostics Volume 2* (011775-A01)
- *Domain Series DN4500 Configuration Worksheet* (013797-A00)

8.2 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.

8.2.1 Domain System Software Release Notes

Appendix A of the *Domain System Software Release Notes* for SR10.0 (Order No. 005809-A01) erroneously claims that SR10.0 supports the 4.3BSD utilities **pdx**(1) (Pascal debugger) and **pix**(1) (a Pascal interpreter and executor).

8.2.2 Making the Transition to SR10 Operating System Releases

Please make the following additions and corrections to *Making the Transition to SR10 Operating System Releases*.

8.2.2.1 Libraries

Add the following paragraph to Section 4.1.7, "Libraries":

We are shipping an SR9.7 version of **/lib/streams** to correct an error you may encounter when running in a mixed network. If you run a program on an SR9.7 node that closes a temporary file that is resident on an SR10 node using the **ios_$close** call, the call will return the error status f0001, "object not found". The operation has correctly occurred on the SR10 node, but the return of that status code may cause the program to fail. Install the **/lib/streams** from **/SR9.7_compatibility/SR9.7_executables/lib/streams** on the SR9.7 node to correct this behavior. Please note that this version of the streams library supports only Version 3.1 of TCP/IP. If you are running Version 3.0 of TCP/IP, you should not install this version of **/lib/streams** on your SR9.7 node.

8.2.2.2 The cvtrgy Command

The hardcopy version of *Making the Transition to SR10 Operating System Releases* does not state that **cvtrgy** is located in the **/install/tools** directory. You should update the manual pages 3-6 through 3-13 on **cvtrgy** to indicate the command's location and to state that you must specify **/install/tools/cvtrgy** to run this program.

8.2.2.3 The Bind Utility

The section titled "The Bind Utility" in Chapter 4 of *Making the Transition to SR10 Operating System Releases* (p. 4–3 in the hardcopy version, p. 4–2 (section 4.1.3) of the online version) contains the following paragraph:

- At SR10, you must use the **-mergebss** bind option whenever you compile a C program with either **/bin/cc** or with **/com/cc** and the **-bss** option. The **-mergebss** option allocates space for globals, and is required, even if the C program consists of only one source file. When you bind with **-mergebss**, you cannot run the output file through the binder again.

The **–mergebss** option is obsolete and should not be used. Replace each occurrence of **–mergebss** in the above paragraph with **–allocbss**, so that it now reads:

- At SR10, you must use the **–allocbss** bind option whenever you compile a C program with either /**bin**/cc or with /**com**/cc and the –bss option. The **–allocbss** option allocates space for globals, and is required, even if the C program consists of only one source file. When you bind with **–allocbss**, you cannot run the output file through the binder again.

8.2.3 Managing BSD System Software

Delete the line on page 7-3 (Section 7.1.2) of *Managing BSD System Software* that states "Dialers attached to Local Area Networks (LANs)."

8.2.4 Managing SysV System Software

Page 3-14 of *Managing SysV System Software* lists the **`node_data/login_log.config** file. This should be **`node_data/etc/login_log.conf**.

8.2.5 Managing Aegis System Software

Add the following to Section 3.8.9 ("Log-Out Script Processing"):

> In order for log-out script processing to work, the INPROCESS environment variable must be set to "true." Use this command:
>
> **$ setvar –type env INPROCESS true**

8.2.6 Configuring and Managing TCP/IP

The following information should be added to *Configuring and Managing TCP/IP*:

The /etc/**ftpd** daemon uses the call **getusershell**(3) to validate the user log-in shell for the remote user logging in to a node through **ftpd**. The **getusershell**(3) routine reads the file /etc/**shells** to determine the valid shells that the node supports and returns a pointer to the list of shells. If /etc/**shells** does not exist, **getusershell**(3) returns a pointer to a list containing /**bin**/**sh** and /**bin**/**csh**.

If a user has a non-default log-in shell, they must include this shell in the file /etc/**shells** to be able to **ftp** into that node. Note that the default shells are /**bin**/**sh** and /**bin**/**csh**, not /**bsd4.3**/**bin**/**sh** and /**bsd4.3**/**bin**/**csh**. If a person specifies /**bsd4.3**/**bin**/**sh** or /**bsd4.3**/**bin**/csh (or any other shell) as their log-in shell, they must edit /etc/**shell**. (When users create accounts with /etc/edrgy they can specifiy a log-in shell.)

8.2.7 Programming with System Calls For Interprocess Communication

The *Programming with System Calls For Interprocess Communication* manual has not been revised at SR10 and will be updated at a later time. However, the online examples that correspond to the examples in the book have been updated and corrected. You should therefore refer to the online examples in the /**domain_examples** directory, and not use the sample code listed in the book.

8.2.8 Alarm Server Help Page

The help page for the **alarm_server** refers to the program as /etc/sys/**alarm**/**alarm_server**; the file is actually /sys/**alarm**/**alarm_server**.

8.2.9 Debugger Help Page

There is an error in the debugger help files /sys/**debug**/**help**/**Startup_files.hlp**. It should read, in part:

```
When dde is invoked:
first
   -do commands given on dde command line
then
   user interface dependent startup file installed with debugger, for example,
      /sys/debug/ui/debug_ui_std.startup
then
   user's working directory startup file
      ./.dderc
or if ./.dderc not present
   user's home directory startup file
      ~/.dderc
 ...
```

8.2.10 SysV Syntax Documentation

The SysV **intro**(1) documentation does not include a description of command syntax rules; however, the **getopt**(3c) and **getopt**(1) documentation refer to these rules by number in the **intro**(1) list.

8.2.11 Common Commands

The documentation for the following commands is incorrect in the *Command Reference* or *Managing Your Software* (for administrative commands) manuals.

8.2.11.1 The cpboot Command

The documentation for the /etc/**cpboot** command does not describe the **-dev ct** option that enables you to copy the /sys/**ctboot**, the boot file used when booting form a cartridge tape.

If you wish to build a bootable cartridge tape, you should specify **-dev ct** in place of the target directory. This will copy **ctboot**, the cartridge tape version of **sysboot**, from the source directory (usually /sys) onto the beginning of the cartridge tape. (Note: subsequent **wbaks** to the tape should use the **-sysboot** option to avoid overwriting **ctboot** on the tape.)

8.2.11.2 The edmtdesc Command

The documentation for **edmtdesc** states that you should use **f** for fixed record format; **edmtdesc** only accepts the uppercase **F**.

8.2.11.3 The lprotect Command

The documentation for the **/etc/lprotect** administrative command specifies incorrect options. There are no **-e** and **-d** options. There is a single **-rmtroot** option with three keywords:

all	Allows a remote root full root privileges
none	Does not allow a remote root any privileges
readonly	Allows a remote root read privileges

8.2.11.4 The mkapr Command

The following changes were made to the **mkapr** command after the hardcopy and online documentation were completed. The online **man** and **help** files are correct.

- The **mkapr** utility no longer uses DPSS/Mail as the mailer; instead it uses **/usr/lib/sendmail** with options **-bm -odi -oeq -oi -om**.
- The **.aprintit** file now contains a new editable **mail_path** field which specifies the destination address. The initial value, and new default destination address, is **apollo!apr_cs_admin**.
- The cancel button (and the cancel command in the serial line interface) has been replaced by a new button (command), "Another". This allows you to generate more than one Apollo Product Report during a run of the mkapr program. **Another** reinitializes the Product Report field values.
- The format of the message generated by **mkapr**'s **Send** button/command has changed. It is now encoded in a form that allows more rapid processing at Apollo Computer, Inc.
- The Customer Reference Number field is no longer initialized to a value generated by **mkapr**; it is initialized to zero. You may enter any 8-digit hexadecimal value.
- The Priority field has been renamed **Severity**. The values are the same. This field ranks the impact of the reported problem on your work.

8.2.11.5 The mkdevno Command

The online BSD **man** page for **/etc/mkdevno** is blank. You can copy the SysV page from **/sys5.3/usr/catman/a_man/man1/mkdevno.1m**.

8.2.12 Aegis Commands

The **index** help page has out-of-date information in the section on Network Registry Manipulation. The **edppo** and **edacct** commands are obsolete. They have been replaced by the **edrgy** utility.

The documentation in the help files and the *Aegis Command Reference* for the following Aegis commands is incorrect or incomplete.

8.2.12.1 The fserr Command

The documentation does not mention that all words in the dictionary must be in lowercase only. **fserr** ignores any uppercase entries in the dictionary.

8.2.12.2 The prf Command

The documentation for the **prf** command in the help files and the *Aegis Command Reference* does not mention the **-dia[logue]** option that replaces the separate pre-SR10 **prfd** command. There is a help file and a manual file for **prfd**; these apply to **prf -dialogue**. This documentation also refers to the **user_data/startup.prf** file and **˜/user_data/startup_prf** file; in both cases, the correct file is **˜/user_data/startup.prf**.

8.2.12.3 The pst Command

The **pst** command documentation for the **-pa[ging]** option states that paging information is provided on private page faults and global page faults. This is no longer true. Paging information is now provided on program execution faults and Data I/O faults (in addition to Disk paging I/O and Net Paging I/O).

8.2.13 The sa Command

The entries in *Managing BSD Software* and *Managing SysV Software* for the BSD /etc/sa(8) incorrectly state that "Domain/OS systems report half the process time as user time and half as system time". The accounting utility correctly reports separately the time spent in user and supervisor mode. Note, however, that the division of UNIX processing between user and supervisor mode varies among manufacturers and many of our system calls have a significant execution component in the global libraries.

8.2.14 BSD Commands

The documentation in the *BSD Command Reference* for the following commands is incorrect or incomplete.

8.2.14.1 The ls Command

Hardcopy pages for ls(1)in the *BSD Command Reference* are incomplete. They should note that **ls -l** will show a plus sign (+) at the end of the permissions string if the permissions have any extensions beyond standard UNIX permissions. These extensions include allowing p rights for any user besides the owner, k rights, organization rights (that is, the organization entry is not set "Ignore"), and extended ACLs (that is, permissions entries in addition to those for the owner, group, organization, and world). (The BSD documentation mentions the "+" indicator, but does not provide complete information.)

8.2.14.2 The man Command

The documentation on the **local** section specifier does not indicate that, if there is no local entry for a command, **man** returns an error message.

8.2.14.3 The tip Command

The **tip**(1) documentation should mention that when you are dialing out using /etc/**remote**, you should use one of the /**dev/sio** devices, rather than one of the /**dev/tty** devices because the SIO devices do not wait for the carrier.

8.2.14.4 The tset Command

The **tset**(1) documentation recommends using the following:

```
eval 'tset -s options...'
```

You must **set noglob** before issuing the **tset -s** command and **unset noglob** after it. This requirement is not documented.

8.2.15 SysV Commands

The documentation in the **man** pages and the *SysV Command Reference* for the following commands is incorrect or incomplete.

8.2.15.1 The chmod Command

The documentation for **chmod** incorrectly specifies that

chmod 111 *filename*

will add read rights for the file. At SR10 this is no longer true; **chmod 111** now only gives execute rights.

8.2.15.2 The ls Command

The hardcopy page for **ls**(1) in the *SysV Command Reference* is incomplete. They should note that **ls -l** will show a plus sign (+) at the end of the permissions string if the permissions have any extensions beyond standard UNIX permissions. These extensions include allowing p rights for any user besides the owner, k rights, organization rights (that is, the organization entry is not set "Ignore"), and extended ACLs (that is permissions entries in addition to those for the owner, group, organization, and world).

8.2.15.3 The glossary, locate, starter, and usage Commands

We document the **glossary**, **locate**, **starter**, and **usage** commands; however, they are not supported at SR10.

8.2.16 /usr/lib/vfont

There is a BSD **vfont(5) man** page. We do not support **/usr/lib/vfont**. This information should also be added to the BSD **domain(5) man** page.

8.2.17 The bstring(3) Call

The BUGS section of the documentation for the BSD **bstring**(3) call is incorrect. It states the following:

> The **bcopy** routine take parameters backward from **strcpy**. For example,
>
> **strcpy (foo, bar)**
>
> copies foo to bar, while
>
> **bcopy (foo, bar, 3)** copies bar to foo.

It should state the following:

> **strcpy (foo, bar)**
>
> copies bar to foo, while
>
> **bcopy (foo, bar, 3)** copies foo to bar.

8.2.18 Open System Toolkit

Using the Open System Toolkit to Extend the Stream Facility refers to the **itest** utility. This is now the **ios_test** utility. There is no **/com/itest**.

8.2.19 Debugger Manual

The following functionality has been added to the debugger since the *Domain Distributed Debugging Environment Reference* went to print:

- The **dde** command now takes the option **-nwp**, which indicates that no target program window should be created.
- In FORTRAN, the **cmplx** and **aimag** intrinsic functions are now supported.
- In Pascal, the **ord** standard function is now supported.
- Whenever a target program signal is caught by the debugger, a special alias is invoked called **'after_fault**. You can use this alias to execute debugger commands that are dependent on the kind of fault encountered. The values **'signo**, giving the UNIX signal number, and **'fault_no**, giving the Aegis fault status code, are set following each fault. Example:

```
alias 'after_fault if 'signo==2 -then [[sh echo 'interrupt - continuing']; go -ignore]
```

- In C, Pascal, or FORTRAN expressions, the special functions **‘long** and **‘short** can be used to typecast values to a long or short integer respectively.
- Three **dde** commands have been added; the following pages document them.

  — **ignore**
  — **delete ignores**
  — **list ignores**

ignore Ignore kind of signal.

FORMAT

ignore *signal*

ARGUMENTS

signal

Ignore "signal," a signal name or number.

DESCRIPTION

Ignore all signals of the given kind. When a signal of this type occurs, deliver the signal to the target program, but do not stop execution or notify the user.

By default, the debugger catches all signals before they are delivered to the target program and notifies the user. You can use "go -ignore" or "step -ignore" to avoid delivering a signal that has already been reported.

EXAMPLES

Ignore the given UNIX signal name.

```
dde> ignore SIGQUIT
```

Ignore the given UNIX signal number.

```
dde> ignore 8
```

Ignore the given Aegis status code.

```
dde> ignore 16#00120026
```

RELATED COMMANDS

delete ignores
go
list ignores
step

delete ignores

Delete one or all ignored signals.

FORMAT

delete ignores {*signal* | **-all**}

ARGUMENTS

signal

Do not ignore "signal," a signal name or number.

-all

Do not ignore any signals.

DESCRIPTION

Do not ignore signals of the given kind. When a signal of this type occurs, deliver the signal to the target program, but to not stop execution or notify the user.

EXAMPLES

Do not ignore the given UNIX signal name.

```
dde> delete ignore SIGQUIT
```

Do not ignore the given UNIX signal number.

```
dde> delete ignore 8
```

Do not ignore signals with the given Aegis status code.

```
dde> delete ignore 16#00120026
```

Do not ignore any signals.

```
dde> delete ignores -all
```

RELATED COMMANDS

ignore
list ignores

list ignores List all ignored signals.

FORMAT

list ignores

ARGUMENTS

None.

DESCRIPTION

List all signals that are currently ignored.

The debugger lists the signals in a format suitable for input, so you can save them in a command file or paste them into the input pad.

EXAMPLES

List all ignored signals.

```
dde> list ignores
ignore 8 ; #(SIGFPE) floating point exception fault (UNIX/signal)
ignore SIGQUIT ; #(SIGQUIT) quit fault (UNIX/signal)
```

Save ignored signals to a command file for input at a later debugger session.

```
dde> list ignores >ignore
```

RELATED COMMANDS

delete ignores
ignore

8.3 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.

8.3.1 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. The **pl_mask** parameter, 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. The **length** parameter 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. The **length_ret** parameter is the number of bytes returned by the call. The **time_stamp** parameter 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;
```

8.3.2 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 2DGMR 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.

8.3.2.1 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.v.2.01** that are available with this release. For 2DGMR information see the *Programming with Domain 2D Graphics Metafile Resources* manual (Order No. 005097-A00).

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.

8.3.2.2 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>]
```

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....
```

8.3.2.3 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 port1
```

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

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

8.3.2.4 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.

8.3.2.5 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**.

The **dial_driver_test** tool allows you to check dial box functionality without initializing the dial server. The **dial_server_test** tool 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 the following:

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

8.3.2.6 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.

8.3.2.7 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 A: 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 **login** request 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 document 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.

debug_sr9.2 — Development of software for pre-SR9.5 systems is not supported at SR10.

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.

fst 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**. The **fst** command will not be supported at the next major release. (**fst**(1))

lbr_sr9.2 Development of software for pre-SR9.5 systems is not supported at SR10.

lkob The **lkob** command is supplied in the **/com** directory at SR10 for changeover compatibility reasons only.

login If you do not specify the **-c** option, the **/com/login** command now invokes the default log-in shell, and not necessarily the Aegis shell.

lopstr The **lopstr** command is supplied in the **/com** directory at SR10 for changeover compatibility reasons only.

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 -dialogue**.

rldev The **rldev** command is supplied in the **/com** directory at SR10 for changeover compatibility reasons only. See the GPIO release document for more details.

salrgy Replaced by the **rgy_admin** utility.

scrch Replaced by the **/usr/apollo/scrattr** command.

sh8 The standard Aegis shell, **/com/sh**, now provides 8-bit support.

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.

Note that several commands that manipulate files now require a **-p[airwise]** option. This option is described in detail in Chapter 1.

acl The **acl** command has been changed to incorporate the SR10 protection mechanisms.

chn This command now requires you to use a **-p[airwise]** option if you specify more than two file tokens.

cpf This command now requires you to use a **-p[airwise]** option if you specify more than two file tokens.

cpt This command now requires you to use a **-p[airwise]** option if you specify more than two file tokens.

crl This command now requires you to use a **-p[airwise]** option if you specify more than two file tokens.

crp The **crp** facility now supports pad and tty traits; if the remote node has the spmio type and manager, **crp** enables remoted pad and tty operations. Otherwise, a warning that pad support is not available is issued. This command is now a link to **/usr/apollo/bin/crp**.

crtyobj Several options have been added and deleted; this command is now a link to **/usr/apollo/bin/crtyobj**.

date Several options have been added to provide support for European languages and time/date formats.

dlf This command now requires you to use a **-p[airwise]** option if you specify more than two file tokens.

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** option has been added.

mvf This command now requires you to use a **-p[airwise]** option if you specify more than two file tokens.

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/nd** will no longer work; they will have to use the appropriate system calls instead.

pst The **pst** command now displays the name of the program being run by a process on the local node if the process has not been assigned a name.

rbak The **rbak** command no longer reads responses from **errin**; instead, it reads from **/dev/tty**. It can also now read input from standard streams in addition to tape. This command is now a link to **/usr/apollo/bin/rbak**.

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**.)

vctl This command has been enhanced and now works on a standard stream as well as a VT100 teminal emulator.

wbak The **wbak** command takes a new option, **-presr10**. If you use this option when you create a tape on an SR10 system, an SR9.X system running the standard pre-SR10 **rbak** can then read the tape. There are separate **/com/wbak** and **/usr/apollo/bin/wbak** commands with different interfaces. Also, **wbak** no longer reads responses from **errin**; instead, it reads from **/dev/tty**, and it can now write output to standard streams in addition to tape.

wd The **wd** command is now built in to the Aegis shell; **/com/wd** is a shell script. As a result, programs that **pgm_$invoke /com/wd** will no longer work; they will have to use the appropriate system calls instead.

A.1.2.3 New Commands

The following commands are new as of SR10:

lst Lists the number of blocks of storage used in a directory 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 wildcards, 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.

bldt Displays the time at which the operating system was built (**bldt**(1)).

chacl Changes an access control list; intended for UNIX users; new at SR10 (**chacl**(1)).

coffdump Dumps selected parts of a COFF object file (**coffdump**(1)).

cpacl Copies an access control list; intended for UNIX users; new at SR10 (**cpacl**(1)).

cpscr Copies the current display to a file (**cpscr**(1)).

crddf Creates, displays or modifies a Device Descriptor File (**crddf**(1)).

crp Creates a remote process. Note: If you use **crp** in a UNIX shell and include a command line in the **crp** invocation, you must quote any wildcards that you include in the command line.

crp now supports the pad and tty traits; if the remote node has the spmio type and manager, **crp** enables remoted pad and tty operations. Otherwise, a warning that pad support is not available is issued (**crp**(1)).

crpad Creates a transcript pad and window (**crpad**(1)).

crty Creates a new type (**crty**(1)).

crtyobj Creates a type object module for binding with type managers; several options have been added and deleted. (**crtyobj**(1)).

dbacl Domain/Dialogue-based tool for changing protections; new at SR10 (**dbacl**(1)).

dde High-level debugger; new at SR10 (**dde**(1)).

dial_server_request
Initializes a dial box (**dial_server_request**(1)).

dlty Deletes a type (**dlty**(1)).

dspst Displays process statistics in a graphic format (**dspst**(1)).

edfont Edits a character font; revised at SR10 to edit 8-bit fonts (**edfont**(1)).

edmtdesc Edits a magnetic tape descriptor file; this program is identical to /etc/**edmtdesc** (**edmtdesc**(1)).

emt Emulates a dumb terminal. This command now allows you to use either /**com**/**tctl** or /**bin**/**stty** to set SIO line characteristics (**emt**(1)).

esa Displays the address of an external symbol (**esa**(1)).

french_to_iso Converts an old-format French font to an 8-bit ISO standard font format; new at SR10 (**french_to_iso**(1)).

german_to_iso Converts an old-format German font to an 8-bit ISO standard font format; new at SR10 (**german_to_iso**(1)).

hpc Produces a histogram of the program counter during program execution (**hpc**(1)).

intm Installs a type manager (**intm**(1)).

inty Installs a type (**inty**(1)).

kbm Displays the keyboard characteristics and sets the key assignments for the compose, long_shift, and short_shift functions; new at SR10 (**kbm**(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 (**lbr2ar**(1)).

lcm Loads a color map (**lcm**(1)).

llib Lists installed libraries (**llib**(1)).

llkob Lists locked objects (**llkob**(1)).

lsacl Lists an 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 (**mkapr**(1)).

mksinit Make a SysV STREAMS initialization file from a **master.d** file; new at SR10 (**mksinit**(1)).

mt The BSD **mt** program (**mt**(1)).

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**(1)).

obj2coff Converts obj format binaries (produced by pre-SR10 compilers) to COFF format; new at SR10 (**obj2coff**(1)).

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**. It can now read input from standard streams in addition to tape. This command no longer reads responses from **errin**, instead it reads from **/dev/tty** (**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)).

scrto Sets the screen timeout for display blanking (**scrto**(1)).

stcode Translates a status code value to a text message (**stcode**(1)).

strinfo Print information about currently running (SysV) Streams; new at SR10 (**strinfo**(1)).

swapul Reads lines from standard input, rearranges text so that underlines follow, rather than precede, characters in the output stream; formerly in the Domain/IX **/etc** directory (**swapul**(1)).

swedish_to_iso Converts an old-format Swedish font to an 8-bit ISO standard font format; new at SR10 (**swedish_to_iso**(1)).

swiss_to_iso Converts an old-format Swiss font to an 8-bit ISO standard font format; new at SR10 (**swiss_to_iso**(1)).

systype Displays a file's UNIX version stamp; was in /etc in previous releases (**systype**(1)).

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 (**tb**(1)).

tpm Sets the mouse or touchpad mode (**tpm**(1)).

trconf Configure a (SysV) Stream with trace modules; new at SR10 (**trconf**(1)).

trmon Monitor a Stream with trace modules after configuration; new at SR10 (**trmon**(1)).

ts Displays the module name and time stamp stored in an Apollo object module; previously in /systest/ssr_**util** (**ts**(1)).

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 (**uk_to_iso**(1)).

vsize Sets or displays VT100 window settings (**vsize**(1)).

vt100 VT100 terminal emulator (**vt100**(1)).

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 option, **-presr10**, which allows you to make a tape on SR10 that is readable on an SR9.X system by the standard pre-SR10 **rbak**. It can also write output to standard streams in addition to tape (**wbak**(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 are links to these files. In cases where the /**com** command expands derived wildcards, 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.

Where a command has a UNIX manual page, we list the BSD manual section number (normally (8)).

chuvol Changes the UID of a volume (**chuvol**(8)).

copyright A file containing copyright statements that are displayed by /etc/**rc**; new at SR10.

cpboot Copies the system boot file (**cpboot**(8)).

crpty Create a pseudo-tty (**crpty**(8)).

ctnode Catalogs a node; there are separate /com/**ctnode** and /usr/apollo/bin/**ctnode** commands with different interfaces (**ctnode**(8)).

ctob Catalogs an object in the file system (**ctob**(8)).

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.

drm_admin Data replication manager administration tool (/etc/ncs/**drm_admin**); formerly in /sys/ncs.

edmtdesc Edits a magnetic tape descriptor file; this program is identical to /usr/apollo/bin/**edmtdesc**. (**edmtdesc**(8)).

edns Edits the **ns_helper** database (**edns**(8)).

edrgy Edits the registry database; new at SR10 (**edrgy**(8)).

environ A file that specifies the default values of the SYSTYPE and ENVIRONMENT variables for the node; new at SR10.

environment Displays the current environment and installed environments; new at SR10 (**environment**(8)).

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 (**find_orphans**(8)).

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 (**group**(5)).

import_passwd Imports the contents of standard UNIX /etc/passwd and /etc/**group** files into the registry; new at SR10 (**import_passwd**(8)).

invoke_script Invokes one of two scripts, an Aegis shell script and a Bourne shell script, that are specified as input arguments, depending upon the environment(s) installed on the system; new at SR10 (**invoke_script**(8)).

invol Initializes a disk volume; the **invol** utility has been rewritten at SR10 (**invol**(8)).

lb_admin Location broker administration tool; formerly in /sys/ncs (**/etc/ncs/lb_admin**(8)).

lcnet Lists connected networks (**lcnet**(8)).

lcnode Lists connected nodes (**lcnode**(8)).

llbd Local location broker daemon; formerly in /sys/ncs (**/etc/ncs/llbd**(8)).

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; new at SR10.

lprotect Protects the local node from remote root access; new at SR10 (**lprotect**(8)).

mkdevno Creates a device number; new at SR10 (**mkdevno**(8)).

mother Determines the boot device or mother node of a diskless node; used by /etc/rc; new at SR10.

netmain Network statistics tool (**netmain**(8)).

netmain_chklog Cleans up bad **netmain** log files (**netmain_chklog**(8)).

netmain_note Places a message in the network error log (**netmain_note**(8)).

netmain_srvr Network statistics server; formerly in /sys/net (**netmain_srvr**(8)).

netsvc Sets or displays network services (**netsvc**(8)).

nodestat Displays node statistics; **/com/netstat** is now a link to this file (**nodestat**(8)).

obty Sets or displays the type of an object; there are separate **/com/obty** and **/usr/apollo/bin/obty** commands with different interfaces (**obty**(8)).

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 (**org**(8)).

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 (**probenet**(8)).

rgy_admin Registry administration tool; new at SR10 (**rgy_admin**(8)).

rgyd Registry daemon; new at SR10 (**rgyd**(8)).

rgy_merge Tool for merging registries; new at SR10 (**rgy_merge**(8)).

rtchk Tests traffic between adjacent routers (**rtchk**(8)).

rtstat Displays internet router statistics (**rtstat**(8)).

rtsvc Sets or displays internet routing service (**rtsvc**(8)).

salacl Salvages and merges an access control lists; this utility takes a new **-l** option that lists the UIDs of the ACL files that are merged (**salacl**(8)).

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 (**salvol**(8)).

server Runs the program or command specified by its arguments, by default with process ID **user.server.none**; new at SR10 (**server**(8)).

show_lc A compatibility tool that shows which obsolete system calls are being made by an object module; new at SR10 (**show_lc**(8)).

shutspm Shuts down SPM on a node (**shutspm**(8)).

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 (**syncids**(8)).

sys.conf Global library configuration file, a link to **'node_data/etc/sys.conf**; new at SR10.

sys_sh A sys5.3 Bourne shell that is installed on all nodes and is run by **init** to interpret /etc/rc; new at SR10.

templates A directory containing template configuration files, such as a template **/etc/rc**; used to boot diskless nodes; new at SR10.

uctnode Uncatalogs a node; there are separate versions of the **uctnode** command in the **/com** and **/usr/apollo/bin** directories with different interfaces (**uctnode**(8)).

uctob Uncatalogs an object; there are separate versions of the **uctob** command in the **/com** and **/usr/apollo/bin** directories with different interfaces (**uctob**(8)).

ulkob Unlocks an object; there are separate versions of the **ulkob** command in the **/com** and **/usr/apollo/bin** directories with different interfaces (**ulkob**(8)).

uuid_gen Generates an ASCII UUID (Universal Unique Identifier) (/etc/ncs/**uuid_gen**(8)).

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. 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.

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.

arp Address resolution display and control program (**arp**(1m)).

dtcb Dump TCP control block program; a debugging tool (**dctb**(1)).

ftpd DARPA Internet File Transfer Protocol server (**ftpd**(1m)).

gateways This file contains information about routing gateways (**gateways**(4)).

gettable Gets NIC format host tables from a host (**gettable**(1m)).

hostns Maps /etc/**hosts** to /etc/**named** (**hostns**(1m)).

hosts The host name database (**hosts**(4)).

hosts.equiv Specifies equivalent hosts for login purposes (**hosts.equiv**(4)).

hostname Sets or prints the name of the current host system (**hostname**(1)).

htable Converts a NIC standard format host table (**htable**(1m)).

ifconfig Configures network interface parameters (**ifconfig**(1m)).

inetd The internet “super-server”; starts other internet daemons (**inetd**(1m)).

inetd.conf The configuration file for **inetd** (**inetd.conf**(4)).

mbd Memory buffer dump; a troubleshooting utility (**mbd**(1)).

mkhosts Generates a hashed host table (**mkhosts**(1m)).

named The Internet domain name server (**named**(1m)).

named.boot The boot file for /etc/**named**.

named.ca Configuration file for /etc/**named**.

named.hosts Configuration file for /etc/**named**.

named.local Configuration file for /etc/**named**.

named.rev Configuration file for /etc/**named**.

networks Network name database (**networks**(4)).

nshost Maps /etc/**named** to /etc/**hosts** (**nshost**(1m)).

ping Sends ICMP ECHO_REQUEST packets to network hosts (**ping**(1m)).

protocols Protocol name database (**protocols**(4)).

resolv.conf Contains information used by **named** (**resolv.conf**(4)).

rexecd Remote execution server (**rexecd**(1m)).

rlogind Remote login server (**rlogind**(1m)).

route Manually manipulates the network routing table (**route**(1m)).

routed Network routing daemon (**routed**(1m)).

rshd Remote shell server (**rshd**(1m)).

rwhod System state server (**rwhod**(1m)).

services Service name database (**services**(4)).

syslogd System message logging server (**syslogd**(1m)).

tcpd Replaces **tcp_server** used by pre-SR10 Aegis TCP/IP (**tcpd**(1m)).

telnetd DARPA Telnet protocol server (**telnetd**(1m)).

tftpd An adaptation of the Massachusetts Institute of Technology 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 Internet file transfer program (**ftp**(1c)).

logger Makes entries in the system log (**logger**(1)).

netstat This command has had two options added and two other options changed (compared to SR9.7 BSD4.2 **netstat**). (**netstat**(1)).

rlogin Remote login command (**rlogin**(1c)).

remsh The BSD **rsh** utility (**remsh**(1c)).

ruptime Shows status information of hosts on the network (**ruptime**(1c)).

rwho Reports who is logged in on hosts on the network (**rwho**(1c)).

telnet User interface to the DARPA Telnet protocol (**telnet**(1c)).

tftp An adaptation of the Massachusetts Institute of Technology implementation of the **tftp** protocol (**tftp**(1c)).

whois Internet user name directory service (**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 This command has been changed so that it now is exactly like the UNIX version of **ftp**.

tcpstat This command has been changed so that it now is exactly like the UNIX **netstat** command.

telnet This command has been changed so that it now is exactly like the UNIX version of **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 the AT&T System V Release 3 of the UNIX operating system. You should also see the **domain**(1) manual page or page in the *SysV Command Reference* for information on SR10 extensions to SysV and for System V commands that are not supported as SR10.

A.1.6.1 Obsolete or Unsupported Commands

As of SR10, SysV text formatting facilities are part of The DOCUMENTER'S 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(3X), 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 and ACLs (**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 UID) is actually updated when the node reboots (**date**(1)).

ld The **ld** loader no longer invokes **/com/bind**, but is a separate utility. The **ld** command 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. The **ls -l** option displays a plus sign (+) to the left of the permissions string if the object's permissions contain any extensions to standard UNIX semantics (**ls**(1)).

mkdir The SysV **mkdir** command 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)).

swapul The **swapul** command has been moved from the /**etc** directory to /**usr/apollo/bin**; its functionality has not been changed (**swapul**(1)).

tar This command can now handle typed files, supports ACLs, supports blocking factors of 1-20, and has a default blocking factor of 20. Performance has also been improved (**tar**(1)).

time The **time** command now reports separately the time spent in user and supervisor mode (**time**(1)).

who This command has several added options and extensions (**who**(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 Handles special functions of DASI 300 terminals (**300**(1)).

450 Handles special functions of DASI 450 terminals (**450**(1)).

accton System accounting tool (/**etc/sa**(1m)).

captoinfo Converts a **termcap** description into a **terminfo** description (**captoinfo**(1m)).

dump Dumps selected parts of a COFF object file (**dump**(1)).

f77 FORTRAN 77 Compiler (f77(1)).

gencc Creates a front end to the cc command (**gencc**(1m)).

getopt Parses command options (**getopt**(1)).

getty The SysV environment uses the 4.3BSD version of this command (/etc/**getty**(1m)).

greek Terminal character set filter (**greek**(1)).

help Provides on SCCS messages and commands (**help**(1)).

infocmp Compares or prints **terminfo** descriptions (**infocmp**(1m)).

init The SysV environment uses the 4.3BSD version of this command (/etc/**init**(1m)).

ksh The Korn shell (**ksh**(1)).

list Provides a C source listing from a COFF file (**list**(1)).

mcs Manipulates a COFF file comment section (**mcs**(1)).

mkdev The SysV **mkdev** command takes options to create Apollo-specific devices (/etc/**mkdev**(1m)).

mknod Creates a special file (/etc/**mknod**(1m)).

mt The SysV environment now supports the BSD version of the **mt** utility (**mt**(7)).

prof Displays profile data (**prof**(1)).

prtconf (**prtconf**(1m)).

sa System accounting reporting and maintenance tool (/etc/sa(1m)).

sendmail The SysV environment now supports the BSD version of **sendmail** (**sendmail**(1m)).

strace Prints SysV STREAMS trace messages (**strace**(1m)).

strclean SysV STREAMS error logger cleanup program (/etc/**strclean**(1m)).

streamd A daemon that monitors interrupt vectors for STREAMS devices (**streamd**(1m)).

strerr SysV STREAMS error logger daemon (/etc/**strerr**(1m)).

suid_exec (/etc/suid_exec)

timex Times a command and reports process data and system activity (**timex**(1)).

uucheck	Checks the UUCP directories and permissions file (**/usr/lib/uucheck**(1m)).
uucico	File transport program for the UUCP system (**uucico**(1m)).
uucleanup	cleans up the UUCP spool directories (**/usr/lib/uucleanup**(1m)).
uusched	Scheduler for the UUCP file transport system (**uusched**(1m)).
Uutry	Tries to contact a remote system via UUCP with debugging on (**/usr/lib/Uutry**(1m)).
uuxqt	Executes remote command requests (**/usr/lib/uuxqt**(1m)).

A.1.7 BSD Commands and Files

The following subsections list commands that run in the BSD environment that are obsolete, changed, or added as of SR10. You should also see the **domain**(1) online manual page or page in the *BSD Command Reference* for information on SR10 extensions to 4.3BSD and for BSD commands that are not supported as SR10.

A.1.7.1 Obsolete or Unsupported Commands

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

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)).
eyacc	**eyacc** is not supported by 4.3BSD (**eyacc**(1)).
fix_cache	Obsoleted by changes to the protection mechanism (**fix_cache**(1m)).
flush_cache	Obsoleted by changes to the protection mechanism (**flush_cache**(1m)).
hostid	This command is not supported at SR10.
mkptnr	**netman** now performs all functions that were done by this command (**mkptnr**(1m)).
ratfor	(**ratfor**(1)).
sup	Obsoleted by changes to the protection mechanism (**sup**(1m)).
update_slave	Obsoleted 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 (**ld**(1)).

ls The **ls -l** option displays a plus sign (+) to the left of the permissions string if the object's permissions contain any extensions to standard UNIX semantics (**ls**(1)).

makewhatis Several changes have been made to the 4.3BSD version of the **/usr/lib/makewhatis** program to support using formatted manual pages as input to the script. The script now can take a single option. If the option is **-s**, the script runs as before, on manual pages in source format. If you supply any other option, or no option, the script will use the formatted manual pages under **cat1-cat8**, **catn** and **catl**. Additionally, **makewhatis** will now take the nameline from text between NAME and SYNOPSIS or NAME and DESCRIPTION, takes the section number from the suffix of the file name, and includes only the names listed on the nameline, not any different ones in the title.

You should not use **makewhatis -s** to add information to the Apollo-supplied **whatis** file because we do not supply manual pages in source format, and the existing file may become damaged.

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 (used to specify inheritance mechanisms) are now inherited from the parent directory (**mkdir**(1)).

netstat This command has had two options added and two other options changed (compared to SR9.7 4.2BSD **netstat** (**netstat**(1)).

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)).

swapul The **swapul** command has been moved from the **/etc** directory to **/usr/apollo/bin**; its functionality has not been changed (**swapul**(1)).

tar This command can now handle typed files, supports ACLs, supports blocking factors of 1-20, and has a default blocking factor of 20. Performance has also been improved (**tar**(1)).

time The **time** command now reports separately the time spent in user and supervisor mode (**time**(1)).

whatis The **whatis** command has been moved from the **/usr/lib** directory to **/usr/man**; its functionality has not been changed (**whatis**(1)).

who This command has several added options and extensions (**who**(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.

ac Provides log-in accounting information (**ac**(8)).

accton Starts and stops system accounting (/etc/sa(1m)).

atq Prints the queue of jobs that were created by the **at** tool that are waiting to run (**atq**(1)).

atrm Remove jobs that were spooled by **at** (**atrm**(1)).

biff Informs the system whether you want to be notified when mail arrives (**biff**(1)).

coffdump An extended version of the SysV **dump** command; SR10 does not support the 4.3BSD **dump** command (**coffdump**(1)).

compress Compresses and expands data (**compress**(1)).

comsat The server process for **biff** (**comsat**(8c)).

error Analyzes and disperses compiler error messages to the source file and line where the errors occurred (**error**(1))

f77 The FORTRAN 77 compiler (**f77**(1)).

file The BSD environment uses the System V version of the file command (**file**(1)).

finger Provides information about users (**finger**(1)).

getty Opens and initializes tty lines, invokes login (/etc/**getty**(8)).

gprof Displays an execution profile of C or FORTRAN 77 programs (**gprof**(1)).

init The process control initialization program that is invoked as the last step of the boot procedure (/etc/**init**(8)).

ksh The System V Korn shell (**ksh**(1)).

logger Makes entries in the syslog system log (**logger**(1)).

mkdev The **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)).

ping Send Internet ICMP ECHO_REQUEST packets to network hosts (**ping**(8)).

pmerge A Pascal file merger (**pmerge**(1)).

prof Displays profile data (**prof**(1)).

rdist A remote file distribution program for maintaining identical copies of files on multiple hosts (**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 The remote host description file (/etc/**remote**(8)).

sa System accounting reporting and maintenance tool (/etc/sa(1m)). (sa(8)).

sccs The front-end utility for the Source Code Control System (sccs(1)).

shutdown Shuts down the system at a given time (/etc/**shutdown**(8)).

struct Translates A FORTRAN program specified by a file into a Ratfor program (**struct**(1)).

syslogd The System message logging daemon (/etc/**syslogd**(8)).

tcopy Copies a magnetic tape (**tcopy**(1)).

tftp An adaptation of the Massachusetts Institute of Technology implementation of the **tftp** protocol (**tftp**(1c)).

trpt Prints a readable description of TCP trace records (**trpt**(8c)).

vacation A program that returns a message to the sender of a message telling that you are on vacation (**vacation**(1)).

vgrind The **vgrind** program formats the program sources (specified as arguments) using **troff(1)** (**vgrind**(1)).

whois Internet user name directory service (**whois**(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. You should also see the online **domain** manual pages or **domain** pages in the *SysV Call Reference* for information on SR10 extensions to SysV and for System V calls that are not supported as SR10.

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.

default_acl This call is no longer required. At SR10 it is a no-op (**default_acl**(2)).

set_sbrk_size This call is no longer required. At SR10 it is a no-op (**set_sbrk_size**(2)).

soft_link This call is now identical to **symlink**(2). Both now support soft links by using textual links (**soft_link**(2)).

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 and were not in SR9.7 and earlier Domain/IX SysV, or are significantly changed since SR9.7.

FORTRAN Library Calls:

abort Terminates a FORTRAN program (**abort**(3F)).

abs, iabs, dabs, cabs, zabs
A FORTRAN absolute value (**abs**(3F)).

acos, dacos FORTRAN arccosine intrinsic function (**acos**(3F)).

aimag, dimag FORTRAN imaginary part of complex argument (**aimag**(3F)).

aint, dint FORTRAN integer part intrinsic function (**aint**(3F)).

asin, dasin FORTRAN arcsine intrinsic function (**asin**(3F)).

atan, datan FORTRAN arctangent intrinsic function (**atan**(3F)).

atan2, datan2 FORTRAN arctangent intrinsic function (**atan2**(3F)).

bool: and, or, xor, not, lshift, rshift
FORTRAN bitwise Boolean functions (**bool**(3F)).

conjg, dconjg FORTRAN complex conjugate intrinsic function (**conjg**(3F)).

cos, dcos, ccos
FORTRAN cosine intrinsic function (**cos**(3F)).

cosh, dcosh FORTRAN hyperbolic cosine intrinsic function (**cosh**(3F)).

dim, ddim, idim
Positive difference intrinsic functions (**dim**(3F)).

dprod Double-precision product intrinsic function (**dprod**(3F)).

exp, dexp, cexp
FORTRAN exponential intrinsic function (**exp**(3F)).

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)).

iargc Returns the number of command-line arguments (**iargc**(3F)).

index Returns the location of a FORTRAN substring (**index**(3F)).

len Returns the length of a FORTRAN string (**len**(3F)).

log, alog, dlog, clog
FORTRAN natural logarithm intrinsic function (**log**(3F)).

log10, alog10, dlog10
FORTRAN common logarithm intrinsic function (**log10**(3F)).

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)).

mod, amod, dmod
FORTRAN remaindering intrinsic functions (**mod**(3F)).

rand, irand, srand
Random number generator (**rand**(3F)).

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

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

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

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

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

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

strcmp: lge, lgt, lle, llt
String comparison intrinsic functions (**strcmp**(3F)).

system Issues a shell command from FORTRAN (**system**(3F)).

tan, dtan FORTRAN tangent intrinsic function (**tan**(3F)).

tanh, dtanh FORTRAN hyperbolic tangent intrinsic function (**tanh**(3F)).

Other Calls:

acct Enables or disables process accounting (**acct**(2)).

acct Per-process accounting file format (**acct**(4)).

default_acl Changes the default file protection environment; this is an extension to the SysV call set (**default_acl**(2)).

dirent File system independent directory entry (**dirent**(4)).

dup2 Duplicates an open file descriptor (**dup2**(3C)).

eqnchar Special character definitions for eqn; an extension to SysV (**eqnchar**(5)).

filehdr File header for common object files (**filehdr**(4)).

fstab File system table (**fstab**(4)).

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 SysV Stream (**getmsg**(2)).

getoid, geteoid
Gets the real and effective organization IDs (**getoid**(2)).

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

getut: getutent, getutid, getutline, pututline, setutent, endutent, utmpname
Accesses a **utmp** file entry (**getut**(3C)).

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)).

ldgetarname Retrieves the file name of a member of an archive file (**ldgetarname**(3X)).

ldgetname Retrieves the symbol name for a common object file symbol table entry (**ldgetname**(3X)).

ldgetstring Retrieves a string from the common object file string table (**ldgetstring**(3X)).

ldlread, ldlinit, 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)).

ldsgetname Retrieves the section name for a common object file entry entry (**ldsgetname**(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)).

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)).

madvise, mincore
Berkeley's proposed memory management system calls to advise the system of a process' expected paging behavior (**madvise**(2)).

malloc, free, realloc, calloc, mallopt, mallinfo
Fast main memory allocator (**malloc**(3X)).

malloc, free, realloc, calloc
Memory allocator with debugging support (**malloc.dbg**(3C)).

man Macros for formatting manual pages. (**man**(5)).

manx Macros for formatting entries in this manual (**manx**(5)).

master Master configuration database for STREAMS modules and drivers (**master**(4)).

mm The MM macro package for formatting documents (**mm**(5)).

mmap, msync, munmap
Berkeley's proposed memory management system calls to map a file system object into virtual memory. The implementation is based on proposals published by Berkeley. These calls are provided in Domain/OS as extensions to the SysV environment (**mmap**(2)).

monitor Prepares execution profile (**monitor**(3C)).

mprotect Berkeley's proposed memory management system calls to assign protection to region mapped in virtual memory (**mprotect**(2)).

ms Text formatting macros (**ms**(5)).

mset, mclear Berkeley's proposed memory management system calls to set and clear semaphores for processes mapped in virtual memory (**mset**(2)).

mv A **troff** macro package for typesetting viewgraphs and slides (**mv**(5)).

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)).

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

putmsg Sends a message on a SysV Stream (**putmsg**(2)).

putpwent Writes a password file entry; an extension to SysV (**putpwent**(3C)).

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)).

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

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

setoid Sets the organization ID (**setoid**(3)).

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

soft_link, soft_unlink
Creates or deletes soft links; an extension to SysV (**soft_link**(2)).

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

stdio Standard buffered input/output package (**stdio**(3S)).

stime Sets the time (**stime**(2)).

syms Common object file symbol table format (**syms**(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)).

times The **times** call now reports separately the time spent in user and supervisor mode. (**times**(2)).

ttyslot Finds the slot in the **utmp** file of the current user (**ttyslot**(3C)).

uadmin Administrative control (**uadmin**(2)).

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)).

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. You should also see the online **domain** man pages or **domain** pages in the *BSD Call Reference* for information on SR10 extensions to 4.3BSD and for BSD calls that are not supported as SR10.

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 no-op (**default_acl**(2)).

set_sbrk_size This call is no longer required. At SR10 it is a no-op (**set_sbrk_size**(2)).

soft_link This call identical to **symlin**(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 Turns accounting on or off (**acct**(2)).

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 Bessel functions 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)).

console Console interface (**console**(4)).

crpio The interface between a window and process created by **crp** (**crpio**(4)).

ctio Cartridge tape I/O interface (**ctio**(4)).

default_acl Change default file protection environment (**default_acl**(2)).

display The window device for the display manager (**display**(4)).

dsk Winchester, floppy, and storage module interface (**dsk**(4)).

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)).

filehdr The file header for common object files (**filehdr**(5)).

flmin, flmax, ffrac, dflmin, dflmax, dffrac, inmax
Return extreme values (**flmin**(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)).

gethostbyname, gethostbyaddr, gethostent, sethostent, endhostent
Gets a network host entry (**gethostbyname**(3N)).

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 setorgfile
Gets the organization file entry (**getorgent**(3)).

getpid Gets the process ID (**getpid**(3F)).

getttyent, getttynam, 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 the legal user shells (**getusershell**(3)).

global_devices A directory for custom device descriptor files (**global_devices**(4)).

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, lnblnk, 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)).

ldfcn Common object file access routines (**ldfcn**(5)).

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

ldgetarname Retrieves the file name of a member of an archive file (**ldgetarname**(3X)).

ldgetname Retrieves symbol name for common object file symbol table entry (**ldgetname**(3X)).

ldgetstring Retrieves a string from the common object file string table (**ldgetstring**(3X)).

ldlread, ldlinit, 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)).

ldsgetname Retrieves the section name for a common object file entry entry (**ldsgetname**(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)).

lgamma The log gamma function (**lgamma**(3M)).

linenum Line number entries in a common object file (**linenum**(5)).

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)).

madvise mincore
One of Berkeley's proposed memory management system calls. The implementation is based on proposals published by Berkeley. (**madvise**(2)).

malloc, free, falloc
Memory allocator (**malloc**(3F)).

malloc, free, realloc, calloc
Memory allocator with debugging support (**malloc.dbg**(3)).

matherr Error-handling function (**matherr**(3M)).

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 (used to specify inheritance mechanisms) are now inherited from the parent directory (**mkdir**(2)).

mmap, msync, munmap
Berkeley's proposed memory management system calls to map file system objects into virtual memory. The implementation is based on proposals published by Berkeley (**mmap**(2)).

monitor, monstartup, moncontrol
Prepares execution profile (**monitor**(3)).

mp Multiple precision integer arithmetic (**mp**(3X)).

mprotect Berkeley's proposed memory management system calls to assign protection to a region mapped into virtual memory (**mprotect**(2)).

mset, mclear Berkeley's proposed memory management system calls to set and clear semaphores for processes mapped in virtual memory (**mset**(2)).

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

netrc Autologin profile for ftp users (**netrc**(5)).

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

pad The interface to processes opened on a transcript pad (**pad**(4)).

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)).
reloc	Relocation information for a common object file (**reloc**(5)).
rename	Renames a file (**rename**(3F)).
res_mkquery, res_send, res_init, dn_comp, dn_expand	Resolver routines (**resolver**(3)).
resolver: resolv.conf	Resolver configuration file (**resolver**(5)).
scnhdr	Section header for a common object file (**scnhdr**(5)).
setoid, seteoid, setroid	Sets the organization ID (**setoid**(3)).
setreoid	Sets the 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)).
sleep	Suspends execution for an interval (**sleep**(3F)).
cbrt, sqrt	Cube root, square root (**sqrt**(3M)).
socketpair	Create a pair of connected sockets (**socketpair**(2)).
soft_link, soft_unlink	Creates or deletes soft links (**soft_link**(2)).
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)).

syms The common object file symbol table format (**syms**(5)).

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

system Executes a UNIX command (**system**(3F)).

time, ctime ltime gmtime
Returns system time (**time**(3C)).

times The **times** call now reports separately the time spent in user and supervisor mode (**times**(3C)).

topen, tclose, tread, twrite, trewin, tskipf, tstate
f77 tape I/O routines (**topen**(3F)).

traper Traps arithmetic errors (**traper**(3F)).

trapov Traps and repairs floating-point overflow (**trapov**(3F)).

trpfpe, fpecnt Traps and repairs floating-point faults (**trpfpe**(3F)).

ttynam, isatty Finds the name of a terminal port (**ttynam**(3F)).

ttys Terminal initialization data (**ttys**(5)).

ualarm Schedules a signal after a specified time (**ualarm**(3)).

unlink Removes a directory entry (**unlink**(3F)).

usleep Suspends execution for a time interval (**usleep**(3)).

vgrindefs **vgrind**'s language definition database (**vgrindefs**(5)).

vhangup Virtually "hangs up" the current control terminal (**vhangup**(2)).

wait Waits for a process to terminate (**wait**(3F)).

A.2.3 Domain/OS System Calls

At SR10, several calls and interfaces have been added, changed, or 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 interruptible.

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 Aegis Environment*.

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_$clear_inlib	Clears the installed library list
loader_$cond_load	Ensures that a module is loaded
loader_$inlib	Creates a private installed library
loader_$inquire_image	Gets information about a loaded image
loader_$inquire_known_lib	Gets information about a libraries known to a loaded image
loader_$inquire_section	Gets information about an image section
loader_$inquire_stat	Gets information about the load history of the calling process
loader_$kg_define	Adds a symbol or section to the known global table (KGT)
loader_$kg_lookup	Looks up a symbol in the KGT and returns its attributes
loader_$load	Loads an object module
loader_$lookup_stack_size	Gets the stack size requirement of an image
loader_$lookup_start_addr	Gets the start address of an image

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 operations, 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 on a remote node by a process that is run in a **crp** window and the program 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 new option, **pad_$curs_off**, has been added to the **pad_cre_opt_t** set type. This option allows the pad to be created with the **cursor_lock** option, that prevents the DM **tn** command from placing the cursor in the pad, set to true. By default, the **cursor_lock** option is set to false.

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.

There is a 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**, and **/usr/include/apollo/fontn.h**.

pm_$errout is a new externally available 2-byte integer declared in the C and Pascal **base.*** include files. It describes the current number of the errout stream, which is program-dependent (obj vs. COFF), and should be used in any library code which is not actually bound with a program; specifically, this includes dynamically loaded libraries which are used with **-inlib**.

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

The following changes have been made to the /systest directory:

- 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/bin/tb**.
- We have added **vmstat**, which reports virtual memory statistics to the /systest/ssr_util directory.
- The **lst** command has been moved from the /systest directory to the /com directory.
- The **ctboot** utility has been moved from the /sys directory to the /systest directory.

Appendix B: Making the Transition to SR10 Operating System Releases

This appendix contains changes, corrections and additions to *Making the Transition to SR10 Operating System Releases*.

B.1 Changes to Making the Transition to SR10 Operating System Releases

The following subsections contain changes and corrections to specific sections of *Making the Transition to SR10 Operating System Releases*. The heading for each section is the same as the heading in the printed manual, and the page numbers used are the numbers in the printed manual.

B.1.1 Login Monitoring

The correct pathname for the login monitoring file, mentioned in the last paragraph of page 1-8, is **/etc/login_log.conf**, which is a link to **`node_data/etc/login_log.conf**. The file is not **`node_data/login_log.config**.

B.1.2 Project Lists

The description of project list matching in the second paragraph on page 1-19 (starting with "The access rights....") is not entirely accurate. Replace that paragraph with the following paragraph:

The access rights allowed by the project list consist of the logical OR of the access rights allowed to each group in your project list. The project list is only created if either the **PROJLIST** environment variable is true or the **SYSTYPE** environment variable value is **BSD***xxx* (e.g., BSD4.3) at log-in.

B.1.3 The Print Environment

The information in this section, on page 2-9 is not completely accurate or up-to-date. You should replace this section with the following text:

The SR10 Aegis print environment consists of a new print architecture which adds a print server manager to the **prf** and **prsvr** commands.

The architecture is based on the Apollo Network Computing System, and is structured as a series of filters and drivers. Features include job query and control and printer query and control. A sample printer driver and sample filter combine with a new manual to allow programmers to expand the system.

The Aegis print environment supports all previously supported devices plus the the Tektronix 4639d color printer. The new print environment can queue jobs from SR10 nodes to SR9.7 printer nodes and print jobs queued from SR9.7 nodes.

We have expanded UNIX printing by allowing the use of **/dev/lp**, which frees it of the requirement to use **prf**.

For more information on the Aegis print environment see *Printing in the Aegis Environment*. For more information on the BSD or SysV print environments see the SysV and BSD *Managing System Software* books and *Programmers Reference* manuals.

B.1.4 The /etc/environ File

On page 2-12, add a note that the format of the **.environ** file is the same as that of **/etc/environ**.

B.1.5 Protected Subsystems

On page 2-19, add the following to the paragraph on protected subsystems: However, since programs run by scripts are run out-of-process by default, a shell script with the following format will not work:

```
subs –up
do something which requires subsystem access
subs –down
```

You can make this work in an Aegis shell by setting the **INPROCESS** environment variable true. Note that the **INPROCESS** compatibility mechanism will be removed in the next major release.

B.1.6 Protection Inheritance

Make the following changes to the "Protection Inheritance" section on pages 2-21 and 2-22:

- Change the references to the **acl** command in the second paragraph of this section to **edacl**.
- Delete the **–inh_pgo** option. This option is obsolete. You can specify that any of the person, group, or organization fields be inherited from the process by specifying **i** rights for that field.

B.1.7 The Local-Access-Only Attribute

Delete the second paragraph in this section on page 2-22, and replace it with the following:

You can use the **chacl**, **lsacl**, **edacl**, and **acl** command to set and display this attribute.

B.1.8 The cvtrgy Tool

The hardcopy version of *Making the Transition to SR10 Operating System Releases* does not state that **cvtrgy** is located in the **/install/tools** directory. You should update the manual pages 3-6 through 3-13 on **cvtrgy** to indicate the command's location and to state that you must specify **/install/tools/cvtrgy** to run this program.

You should insert the following paragraph after the second paragraph on page 3-11 (that is, after the paragraph that ends "You can of course, change the default password for the SR10 "root.staff.none" to be the same as the "root.staff.sr_9.none" account"):

Note that before you delete the _sr9 SIDs (to reduce clutter in the registry) you should change the ACLs on any files owned by these SIDs so that the files are owned by active accounts. Otherwise, you may have to become root in order to be able to access the files or modify the protections.

B.1.9 The crpasswd Tool

Both the second paragraph on Page 3-14 (in the "Converting from SR10 to SR9.7" subsection) and the fourth paragraph on page 3-14 (in the "Converting /etc/passwd and /etc/group Files" subsection) state that the version of **crpasswd** that is provided for use by SR9.7 nodes in a mixed network is installed in the **/sr9.7_compatibility/sr9.7_executables** tree. At SR10.1, this file is located in the **/install/tools/crpasswd** file.

B.1.10 Starting and Administering the glbd

The **crp** command following the second paragraph in this section has a typographical error and can be simplified. The **llbd** in the command should be **glbd**. The command line should read:

crp -on ***//remote_node*** **-cps -n glbd 'fB/etc/ncs/glbd -create -first'.**

B.1.11 Starting the rgyd Process

You should delete the text starting with last paragraph on page 3-18, "You can start another instance of rgyd ...." and all remaining text to the end of the "Starting the rgyd Process" section. Replace it with the following paragraph:

If you create the empty file /etc/**daemons/rgyd**, the node's /etc/**rc** file, as shipped, will ensure that the daemon is restarted every time the node is rebooted.

B.1.12 Operating Mixed Networks

Change the last sentence in the first paragraph of this section on page 3-21 to read as follows:

Generally speaking, the major problem in mixed networks is that pre-SR9.7 nodes will not be able to see files and/or read directories on SR10 nodes, usually because of insufficient rights or incompatibilities in directory formats.

B.1.13 Print Services

The description of print services on page 3-27 is not completely accurate or up-to-date. Replace this section with the following paragraphs:

The new Aegis print architecture works transparently in a mixed environment that contains SR10 and SR9.7 print servers. Two switches, **–pre10** and **–check**, have been added to the **prf** command to provide some additional control in mixed environments.

By default the SR10 **prf** command first submits a print job using the SR9.7 format (it creates a file in /sys/**print/queue** and copies the data file to /sys/**print/spooler**). The SR10 **prf** command then queries the print managers on the network to find the specified printer name. If the name is found to be an active SR10 printer, the job is added the the print manager's queue. If the specified printer does not exist, **prf** returns with no error message, assuming the job has been queued to an SR9 print server.

The **–pre10** switch eliminates the query to the print manager. Use this option only when you are sure there are no SR10 print managers on the network.

The **–check** switch is used if you want to be sure you are queuing a job to an SR10 printer. Adding the **–check** switch to your command line or **startup.prf** file will cause **prf** to report an error and not queue the job if the specified printer name is not an active SR10 printer. This switch will become the default switch in a future release.

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.

The new features of the SR10 architecture are, of course, not supported on an SR9.x node.

B.1.14 Protection Incompatibilities in Mixed Networks

You should change the table on page 3-23 by moving the "g" from the last line of the "SR9.x sees" column to the first line of the column and changing the notes. The table should now look like:

SR10 right(s)	SR9.x sees	Notes
p	pg	g right is seen if p is on
r	r	no change
w	cale	never a subset of cale
x	s	execute = search
!k	d	if not k, d
k	!d	if k, not d
	n	this right will always be on

B.1.15 Command Search Rules

The last line on page 3-28 refers to the "$(systype)" link and the first line on 3-28 refer to the $(systype)/usr directory. In both cases, the word systype is in the wrong case; it must be an uppercase SYSTYPE.

B.1.16 Bind Utility

Change the third bulleted item in "The Bind Utility" on page 4-3 by replacing all instances of **–mergebss** to **–allocbss**. The paragraph should now read as follows:

- At SR10, you must use the **–allocbss** bind option whenever you compile a C program with either /**bin**/cc or with /**com**/cc and the **–bss** option. The **–allocbss** option allocates space for globals, and is required, even if the C program consists of only one source file. When you bind with **–allocbss**, you cannot run the output file through the binder again.

If you compile by using /**bin**/cc **–W0 –nbss** or /**com**/cc (without the –bss option) you should use **–mergebss** to merge the resulting multiple global data sections. Conversely, if you compile with /**bin**/cc or /**com**/cc **–bss**, you *must* use **–allocbss** in the final bind operation to create a single data section and to put all the uninitialized global variables (which are not assigned a section during compilation) into the section. Unlike **–mergebss**, the **–allocbss** option will not merge any existing uninitialized global data sections into the .bss section.

B.1.17 Libraries

Add the following paragraph to the "Libraries" section on page 4-4:

We are shipping an SR9.7 version of /**lib**/**streams** to correct an error you may encounter when running in a mixed network. If you run a program on an SR9.7 node that uses the **ios_$close** call to close a temporary file that is resident on an SR10 node, the call will return the error status f0001, **object not found**. The operation has correctly occurred on the SR10 node, but the return of that status code may cause the program to fail. Install the /**lib**/**streams** from /**SR9.7_compatibility**/**SR9.7_executables**/**lib**/**streams** on the SR9.7 node to correct this behavior. Please note that this version of the streams library supports only Version 3.1 of TCP/IP. If you are running Version 3.0 of TCP/IP, you should not install this version of /**lib**/**streams** on your SR9.7 node.

B.1.18 Transition Aids

The discussion of the replacement of the backslash convention on page 4-10 should include the following information:

You can no longer use a file's pathname as the "root" of a pathname with backslashes. For example, in an Aegis shell script, the following construction is not valid, while the backslash equivalent was valid at SR9.7

```
^0/../program2
```

where program2 is another file in the same directory as the script.

B.1.19 The inlib Command

Delete the section on the **inlib** command on page 4-17. At both SR10.0 and SR10.1 the performance penalty of this command is insignificant.

B.2 Protection and ACL Overview

The *Managing System Software* manuals provide a detailed view of protection and ACLs at SR10, and several sections in *Making the Transition to SR10 Operating System Releases* describe changes that were made to ACLs and protections. However, the documentation does not provide a unified description of the differences between the SR10 protection model and both the standard UNIX model and the pre-SR10 models, and their implications for users converting to SR10. The following sections discuss the SR10 protection model in the context of both UNIX and pre-SR10 Domain software, and describes some of the transition issues. See the *Managing System Software* manuals for a detailed description of protections and ACLs at SR10, and the tools that you use to manage them. Also see **Installing Software with Apollo's Release and Installation Tools, and particularly Chapter 6 of that book, for information on how to manage protection when you install SR10-based software.**

B.2.1 Overall Network Protection Model

The Apollo system protection model allows each installation to determine the best way to organize the protection of files and directories for that installation. The model supports the full range of protection choices, from totally open to tightly protected, and anywhere in between.

Some installations give node owners full control over their nodes, including the system software. In an open network, all the files on a node's disk "belong" to the owner of the node, who is free to protect, delete or move the files at his discretion. (There are some exceptions, of course. Certain system files must not be deleted or moved, or else the system will not operate correctly or will be open to intrusions. In the UNIX environments in particular, certain directories must be controlled since many procedures that run as root use programs from these directories and the susceptibility to Trojan horses is great.) However, within this framework, in an open network the node owner may tailor his node to meet his needs.

On other networks, nodes are viewed as being a group resource and are administered by a central group and tightly controlled. In a protected network, all system files on a node's disk "belong" to the system administrator and are protected against modification by users of the node. Protections are maintained so that only the system administrator can alter files not specifically private to a user. At SR10, software on such nodes is owned and generally controlled by root (locksmith), rather than sys_admin as in previous releases. Protected installations, especially those that run in the Aegis environment, may want to migrate their protections from control by sys_admin accounts to control by root accounts.

B.2.2 Object Protection Model

B.2.2.1 The UNIX Model

UNIX protects objects with three rights: read, write and execute (rwx). For a directory, the execute right is the right to search through the directory to resolve pathnames. UNIX recognizes three subject classes for protection: user, group and "other" (Aegis terminology refers to "user" as owner and "other" as world). Rights are granted to the owner of the object, a group associated with the object (and therefore people with accounts in that group) and the rest of the world. These rights are generally represented as **rwxrwxrwx**, with the leftmost **rwx** referring to the user rights, the middle **rwx** referring to the group rights, and the rightmost **rwx** referring to the "other" rights.

Protection for newly created objects is derived as follows: the user identity is obtained from the user-name identity of the creating process. The group identity is obtained from the group-name identity of the creating process (for SysV) or the group associated with the containing directory (for BSD). The object's initial protection is derived from the set of rights specified by the creating the program and and a mask, known as the umask, that is associated with the creating process.

The umask is a nine-bit mask in the standard "rwxrwxrwx" form that specifies rights that are to be automatically subtracted from all newly created files and directories. For example, if a file is created with rights of "rw-rw-rw-" and the umask is "----w--w-", the file will get rights of "rw-r--r--". A typical use of the umask might be to deny certain rights to "other" so that files that are created by applications running on behalf of the user will not, by default, be writable by all others. The value of the umask is set by the umask command.

B.2.2.2 SR10 Extensions to the UNIX Model

The SR10 protection implementation extends the UNIX model in three directions:

- We have extended the rights from **rwx** to **pwrxk**, where **p** specifies complete control over the object (i.e., the right to modify its protections), and **k** specifies that an object may not be deleted, even in an otherwise writable directory.
- We have extended the number of recognized organizational divisions from just two, person and group, to three, person, group and organization. This more closely matches the real world divisions that occur in large organizations and large networks. As a result, we have expanded the number of subject categories that can be specified for protection purposes from just three, user, group, and other, to include an organization entry. However, we have provided a mechanism for marking an object so that the organization identifier is ignored during access checking, so that the "pure" UNIX behavior can be maintained. Note that, the registry can use the organization identifier as a basis for control over user accounts. When you make the transition to SR10, and if you want to have a "pure" UNIX environment, you should make sure that the registry also does not use organizations.
- We have further extended the number of subject categories for protection to allow you to specify additional sιbjects in an access control list. Each subject so named is

referred to by a triplet "Subject IDentifier" (SID), consisting of a person, group and organization. SID's are specifiable exactly or with wildcards for any of the three fields (e.g. joe.os.r_d or %.%.mktg). The access control list also contains the rights granted or denied to the SIDs named in the list.

These extensions:

- Allow groups of people to exercise joint control over sets of files, since several people may have **p** rights to the files.
- Allow selected objects in a directory to be protected, through the **k** right, without restricting the protection of other objects coresident in the directory.
- Allow different access rights for specific listed subjects than for the rest of the world, including increased or lessened privileges.

Protection inheritance has also been enhanced. In addition to supporting the System V and BSD semantics for protecting newly created objects, every Domain/OS directory has two initial default ACLs, one for files and one for directories, in addition to the ACL that protects the directory itself. At the user's discretion, these initial default ACLs, rather than the UNIX inheritance rules, can be used to control the initial protection values for files and directories created within the directory.

The inheritance that can be specified in an initial default ACL is very flexible. The user can independently specify whether the user, the user rights, the group, the group rights, the organization, the organization rights, or the "other" rights get inherited from the initial ACL or from the creating process. Extended ACLs are also supported for initial default ACLs.

The UNIX umask is also supported in the Aegis environment and a **umask** command is provided for the Aegis shell. This is a UNIX compatible command, so that only **rwx** rights for person, group and other, but not organization, can be specified. (Note that files created by some other process on behalf of the first process' e.g. by the DM, do not get the benefit of the first process' umask.)

B.2.3 Interactions of UNIX Protection and ACLS

The standard UNIX system calls and commands used for setting or interrogating protection capabilities are not extensible and cannot support the additional information in an ACL. However, when one of these system calls or commands is used, its behavior may, nevertheless, affect or be affected by the extended protection information. This section describes the interactions between UNIX protection and ACLs.

In order to control this interaction, each file's ACL has an extended entry rights mask, whose bits correspond to **prwxk**. Whenever rights are to be granted as a result of an extended ACL entry, the rights in the ACL entry are first ANDed with the extended entry rights mask before they are used for access determination. Thus, if the extended entry rights mask specified "-r---", only read rights could be granted via an extended ACL. The use of a mask to limit permissions is required so that UNIX system calls and commands perform as expected, regardless of the presence of an extended ACL.

Initially, the extended entry rights mask is the OR of all the rights in the extended entries in the ACL, thereby allowing an extended ACL to grant all rights it contains. The subsequent value of the mask is determined by the **chmod()** system call. At SR10.0 and SR10.1 whenever a **chmod()** system call is issued, the following occurs:

- The "other" rights specified in that call becomes the new value of the extended entry rights mask.
- Any p and k rights are turned off in the group and world entries and in the extended entry rights mask.
- Any k rights are turned off in the user entry.
- The organization entry is marked as ignored.

Thus when **chmod()** sets a value for "other" rights, it is in effect saying that the MAXIMUM rights to be granted to anyone other than user or group are the "other" rights. If a given SID matches an extended ACL entry, the only possible effect of the extended ACL entry is to REDUCE that SID's rights to less than those that would be granted via "other" rights.

At the same time, the **chmod()** call removes the effects of the other Domain/OS extensions to the UNIX protection mechanism, that is, the special granting of p and k rights and the organization entry.

The UNIX **chmod** command, which uses the **chmod()** system call, supports two distinct forms. The first form sets the access rights to an absolute value; the second adds or removes specific parts of the existing access rights. The effect of these two forms on the extended entry rights mask is as follows.

The first form, illustrated by **chmod 664**, sets the "other" rights. In this case "4" (or bits "100"), is used as the middle three bits (rwx) of the extended entry rights mask. Also, as described above, **chmod** always sets the p and k bits, which are not supported by UNIX, to 0. So in this example, even if the mask was initially "prwxk", it is now -r---. Note that the extended ACL entries are not discarded, even if you specify **chmod** ***xx*0**. The list of entries is maintained but is rendered ineffective.

The second form, illustrated by **chmod g–r** or **chmod o+r**, currently also always changes the extended entry rights mask. For example, **chmod g–r** does not specify 'other' rights, so the rwx rights in the extended entry rights mask are not changed. However, the extended entry rights mask p and k rights are removed. If the mask was initially "prwxk" it is now "-rwx-". If the command does specify "other" rights, the extended entry rights mask will also be adjusted the way that "other" rights are. For example, **chmod o+r** will re-enable read rights in extended ACL entries that might have been masked off, but any p and k rights will be turned off. If the rights mask was "p--xk" the mask will now be "-r-x-".

Similar considerations apply to reporting the protections in effect for a file. The **stat()** system call is the standard way for a UNIX program to determine the access rights that apply to a file. Because this system call is well established, it cannot be changed to report the extended ACL rights. However, it is important that the system never underrepresent the rights that are in effect for a given file. Therefore, the value of "other" rights reported by the **stat**() system call includes the OR'ing of the following:

- the actual world rights
- the organization rights, if they are not marked as ignored
- the extended entry rights mask.

For example, if:

- the actual world rights are "-r---"
- the organization rights are "---x-"
- the extended entry rights mask was "-rw--"

the value reported by the **stat**() system call would be (ignoring p and k which are not returned):

```
    'r--' OR '--x' OR 'rw-'
  = 'rwx'
```

NOTE: If any Domain/OS extensions to the UNIX are in effect for a particular object, a **ls -l** command will show a + (plus sign) identifier at the end of the rights string, for example "-rwxr-x--x+".

Although the standard UNIX commands do reasonable things to extended ACLs, you can only get the full benefit of the extended protection features in a UNIX environment, if you use the extended UNIX commands provided by Apollo (**chacl**, **lsacl**, and **cpacl**). The standard UNIX commands cannot be coerced into retaining all aspects of the new protection semantics. Also, note that UNIX applications that use the **chmod**() system call will affect the extended protection on any files or directories to which they apply **chmod**(). The effect may be to limit the rights granted or it may disable the extended ACL or organization rights altogether. It will, however, never grant MORE rights to an entry listed in an ACL than is specified in "other" rights.

B.2.4 ACL Search Order

This section describes the matching rules that are used for access checking. In the description *%* is the wildcard specifier in an ACL entry and *[xxxx]* indicates a all possible values for that field, including the wild card.

To understand how the system determines access rights, you must understand the search order within an ACL. ACL entries are always searched from the most specificity to least. For example, entries with a person name are always searched before entries with a wildcard (%) in the person field. Entries of the form *person.group.%* are searched before entries of the form *person.%.%* since the allowed group is exactly specified.

1. The "user" entry is checked first. If there is a match and the rights are not to be "ignored", that entry is used.

2. Extended ACL entries of the form *person.[xxxx].[xxxx]* are searched for an exact match in the person field and any other non-wildcarded fields. If a match is found, that entry is used after application of the extended entry rights mask.
3. The "group" entry is checked next. If there is a match, the rights are not to be "ignored", that entry is used.
4. Extended ACL entries of the form *%.group.[xxxx]* are searched for an exact match in the group field and any non-wildcarded org field. If a match is found, that entry is used after application of the extended entry rights mask.

   NOTE: In steps 3 and 4, if project lists are enabled, then each entry from the user's project list is substituted for his current project to form an SID. Each SID is then checked against the "group" entry (step 3) or against all extended ACL entries of the form *%.project.[xxxx]* (step 4). That is, the checks in steps 3 and 4 are repeated for each possible SID, including the effective SID. The rights from each match, if any, are concatenated (OR'ed) and used to determine access. (The project list is only created if either the **PROJLIST** environment variable is true or the **SYSTYPE** environment variable value is **BSD***xxx* (e.g., BSD4.3) at log-in.)
5. The "org" entry is checked next. If there is a match and the rights are not to be "ignored", that entry is used.
6. Extended ACL entries of the form *%.%.org* are searched for an exact match in the organization field. If a match is found, that entry is used, after application of the extended entry rights mask.
7. The "other" entry is used if no previous match was found.

B.2.5 Changes in Protections Between SR9.7 and SR10

The following subsections discuss changes have been made in the way ACLs worked in SR9.7 and the way protection works in SR10. They also discuss some of the implications of these changes.

B.2.5.1 Identifiers

The node ID has been eliminated from the SID and from ACLs. As a result, you cannot control access by node ID, and SR10 nodes ignore any node ID values specified in pre-SR10 ACLS. However, we have added several mechanisms to limit access to local objects. These are discussed in the "Additional Protection Capabilities" subsection.

B.2.5.2 Required Entries and Ownership

At SR10, four protections entries are always present: user, group, organization, and other. These correspond to the standard UNIX protections, with the organization extension. As part of these entries, each object now has a user (owner) and is associated with a specific group and organization. Any member of the group or organization will be granted, at a minimum, the rights specified in the organization or group field.

When you copy a file from a pre-SR10 node to an SR10 system using **cpf –sacl** to preserve the protections the required entries are given the user, group, or organization ID or “none” and the protection rights are marked “Ignore”. Any rights specified for *%.%.%* are converted into the “other” required entry. The remaining SR9.7 ACLS are simply converted into extended ACLS. For example, a file ACL might look as follows on an SR9.7 node:

```
$ acl /misc/mail_names
Acl for /misc/mail_names:
 bill.%.%.%              pgndwrx
 %.backup.%.%              ----wr-
 %.%.r_d.%               ----wrx
 %.%.%.%                 -----rx
```

If you use the **cpf –sacl** (on the SR10 node) to copy the file, you will then see the following protections for the new file:

```
$ cpf //h/misc/mail_names test2 –sacl
$ acl test2
Acl for test2:
Required entries
 none.%.%               [ignored]
 %.none.%               [ignored]
 %.%.none               [ignored]
 %.%.%                  -r-x-
 Extended entry rights mask:    prwx-
Extended entries
 bill.%.%               prwx-
 %.backup.%               -rw--
 %.%.r_d                -rwx-
```

You can use a **–conv[ert]** option to the SR10.1 (but not SR10.0 or pre-SR10) **cpf, cpt,** and **acl** commands to convert the SR9.7 acl to SR10 required protections. This option sets the user entry in the target from the first *person.%.% entry* in the source with P rights. The first *%.group.% entry* in the source becomes the group entry in the target, and the first *%.%.org entry* in the source becomes the organization entry in the target. For example:

```
$ cpf //h/misc/mail_names test3 –sacl –conv
$ acl test3
Acl for test3:
Required entries
 bill.%.%               prwx-
 %.backup.%               -rw--
 %.%.r_d                -rwx-
 %.%.%                  -r-x-
 Extended entry rights mask:    -----
```

Note that **rbak** does not have a **–convert** option. However, if you use **rbak** to restore a tape that was generated by a pre-SR10 **wbak** onto an SR10 node, the **–convert** style conversion will be done automatically.

We have also provided a mechanism that enables SR10 protections to mimic pre-SR10 Aegis protections. You can mark any combination of the person, group, and organization required entries to be ignored. If you mark all three of the required entries this way, the protection extended ACLS behave in the pre-SR10 Aegis manner. You cannot mark the "other" entry (or extended ACL)s to be ignored.

You should also note the changes that we made for full UNIX compatibility, such as the extended entry rights mask, that are discussed in the previous section.

B.2.5.3 Inheritance Mechanism

Before SR10, DOMAIN/IX allowed you to use UNIX right inheritance by having special initial directory and file ACLs. In this case both the "identities" and the permissions for new files and directories were inherited from the process (with BSD group id inherited from the containing directory). At SR10 this mechanism has been removed; the mode of protection inheritance is now coded directly in the initial file and initial directory ACL required entries. You can specify any possible combination of identity inheritance and rights inheritance for the required entries. (However, you cannot mark the extended ACL entries to inherit rights from the process.) For example, you could specify that all files created in directory obtained their user, group, and organization identities and protections from the process, with an additional extended ACL entry that gives you pwrx rights.

In this case, **acl –if** would show something like:

```
$ acl test_dr –if
Initial (default) acl for files created under test_dr
Required entries                     For the current process:
 [from process]       [specified by process] bill.%.%  [specified by process]
 [from process]       [specified by process] %.none.%  [specified by process]
 [from process]       [specified by process] %.r_d.%   [specified by process]
 %.%.%               [specified by process]
 Extended entry rights mask:  prwx-
Extended entries
 bill.%.%            prwx-
```

B.2.5.4 Rights

At SR10 we have changed and simplified the protection rights as follows:

- We have eliminated **g** and **n** rights.
- We have combined the **c**, **a**, **l**, and e rights into a single directory w right.
- We have replaced the **d** with the **k** right. The **k** right is approximately the logical negation of the **d** right. An object with a **k** right cannot be deleted unless you are **root** or have **p** rights and use the dlf –f or dlt –f command. You can delete any object that does not have **k** rights if you have w rights for the directory.

- We have changed the s right identifier to x.

When accessing files or listing ACLs, SR9.7 automatically convert rights from SR10 format. SR10 nodes can convert protection rights from pre-SR10 format to SR10 format. The transition manual's "Protection Incompatibilities in Mixed Networks" section (corrected as noted in the previous section) describes how an SR10 node will see protections on SR9.7 nodes and vice versa.

NOTE: When you use the **cpf –sacl** command to copy from an SR9.7 node to an SR10 node, the exact results will differ depending upon whether you use the SR9.7 or SR10 version of the command. If you enter the cpf command on the SR9.7 node, **k** rights are set on the SR10 entries if the corresponding SR9.7 ACL entry did not have **d** rights set. If you enter the cpf command on the SR10 node, **k** rights are never set, independent of the status of the SR9.7 **d** right.

B.2.6 Tools for Manipulating Protections on Objects

In the Aegis environment, the tools used to specify, modify and show protections on objects are:

edacl	used to edit the protections of a specified object
acl	used to display the protections on one or more objects and to copy protections from one object to one or more objects
ld –r	used to display the current process's rights to listed objects

In the UNIX environments, the tools used to specify, modify and show protections on objects are:

chmod	standard tool for specifying protections on objects
chown	standard tool for modifying the owner of objects
chown	standard tool for modifying the group of objects
lsacl	extended tool for listing protections of objects
chacl	extended tool for changing protections of an object
cpacl	extended tool for copying protections from one object to one or more objects
ls –l	used to display the access rights to listed objects

B.2.7 Additional New Protection Capabilities

B.2.7.1 Local Access Only (LAO)

The "local access only" attribute may be used to deny all remote requests to an object. If this attribute is specified in an object's ACL, only local processes will be granted access to the object. This feature is supported only between SR10 nodes; SR9.7 will not respect the LAO attribute of an SR10 object. The **edacl** or **chacl** commands may be used to set or reset this attribute.

B.2.7.2 The lprotect Command

When a process runs as root, the system automatically grants that process full rights to any object, regardless of the access rights that have been assigned to the object. In a network of distributed nodes, the owner of one node may be suspicious of processes on other nodes that claim to be root. The **lprotect** command is used to specify whether remote processes running as root should be granted this special treatment, and if so, the extent of the special treatment they should be granted. Specifically, **lprotect** can specify that requests from remote root processes be treated as follows:

1. Do not grant any special rights to a remote requester that is root; treat it as if it were "other" or **user.none.none**.
2. Grant read rights to a remote requester that is root, in addition to any rights applicable to "other" or **user.none.none**.
3. Grant all rights to a remote root requester.

The following additional restrictions are enforced when **lprotect** is used to limit root rights:

- The server process manager (SPM) will not allow processes to be created as root unless an active login is done, whereby the user specifies the proper password for the root account (i.e., **crp –me** as root will not be honored).
- Invocations of remote setuid-to-root programs are disallowed.
- Remote requests to setuid-to-root on a local object are disallowed.

You should note that operations on a node configured with links to remote setuid-to-root programs cannot work in this mode. That is, if you use **lprotect** on your node, and you have a link from your node to some setuid-to-root program on another node, you cannot use that program, even though you have an entry for it in a local directory.

The **lprotect** command only applies to the node on which it it is issued. It may only be issued by root or by users that have been designated as node owners (see below).

B.2.7.3 Node Owners

It is often desirable for one or more users to have special privileges on a given node, so that certain operations may only be performed by root or by users who have been designated as **node owners**. The system interprets the ACL on the file **`node_data/node_owners** as specifying which users are to be considered as "owners" of the node; all users with **p** rights to this file are treated as node owners. To prevent subversion of this mechanism, the file **`node_data/node_owners** must itself be owned by root.

Currently, the operations controlled by **`node_data/node_owners** are:

lprotect	The ability to deny protection overrides to requests by remote roots
sigp	The ability to signal any process running on this node, rather than just processes with the same SID