Domain System Software Release Notes

Software Release 10.2
Order No. 005809-A05

First Printing: November 1989

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. Information in this publication is subjet to change without notice.

RESTRICTED RIGHTS LEGEND. Use, duplication or disclosure by the Government is subject to restrictions as set forth is subdivision (b) (3) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013. Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304

10 9 8 7 6 5 4 3 2 1

Preface

This document describes standard Domain operating system (Domain/OS) software for Software Release 10.2 (SR10.2). It includes an overview of new and changed functionality, software installation information, documentation references, and a list of known errors and limitations.

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 products are installed in a similar manner. If you are familiar with Apollo SR9-based systems and this is your first time using an SR10-based system, read *Making the Transition to SR10 Operating System Releases* (011435-A02) to understand the differences between SR9-based and SR10-based operating systems. We also have included an online document titled *System Software Changes at SR10 and SR10.1* to help you determine the functions added at those releases. You can find both documents online in the **/install/doc/apollo** directory. Note that the document *System Software Changes at SR10 and SR10.1* is not necessary for your installation or for determining your need for SR10.2. The document is only intended as a reference, so that you can read about functionality changes at previous releases.

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.v.10.2__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 Tables

Chapter 1: New Features at SR10.2

1.1 An Overview of this Release

Software Release 10.2 is a release of Domain system software that provides new functionality, performance enhancements, support for new products, and bug fixes for previous releases.

This chapter contains an overview of the new hardware and software, compatibility information, and new functionality. Major changes made to standard Domain system software since SR10.1 include the following:

- Support for new hardware
- Domain/X11 X Window support
- POSIX compatibility
- TCP/IP enhancements
- Network enhancements
- Improved graphics support
- New diagnostics

For the benefit of customers who may be new to the SR10-based software releases, we have included two online documents for your reference. *System Software Changes at SR10 and SR10.1* describes functionality that was added at SR10 and SR10.1. *Making the Transition to SR10 Operating System Releases* (011435-A02) includes information that will help you make the transition from SR9-based releases to SR10.2. You can find both documents online in the **/install/doc/apollo** directory. Note that the document *System Software Changes at SR10 and SR10.1* is not necessary for your installation, or for determining your need for SR10.2. The document is only intended as a reference, so that you can read about functionality changes at previous releases.

1.2 Layered Products Available with SR10.2

In the list that follows, we indicate the layered products available with SR10.2 and their latest version number. Layered products have their own release documents online in the **/install/doc/apollo** directory for your reference.

The table is *not* intended to help you determine which layered products are available for purchase. For specific information about purchasing layered products, contact your local Apollo sales representative.

Product	Version
Domain/Access	3.0
Domain/Ada	3.0
Domain/C	6.7.m and 6.7.mpx
Domain/C++	1.2.1
Domain/Core	9.5
FTN	10.7.m and 10.7.mpx
Pascal	8.7.m and 8.7.mpx
DTEK 4014	2.1
Domain/Dialogue	2.02
Open Dialogue	1.01
D3M	6.0
DSEE	3.3.2
DPAK	4.1
DPCC	3.1
DPCE	3.1
DPCI	4.0
DPSS/Mail	2.3
GPIO	10.2
NIDL	1.5.1
NFS(R)	2.1
LSLOCK	1.1
Domain 5080	3.3.1
OmniBack	1.1
PHIGS	1.1
PHIGS_RT	1.1
GSR	2.6
2DGMR	2.2.1
2DGMR_RT	2.2.1
3DGMR	2.7
3DGMR_RT	2.7
SCAT	2.0
SNA 3270	2.3
SNA 3770	1.4
SPE	2.1
Transcript	2.1
LU6.2	1.2
Domain X.25	2.3
GKS	1.0
GKS_RT	1.0
Knowledge Broker	1.0
Technet	1.1
TML	1.0

1.3 Compatibility Information

The following subsections describe changes in SR10.2 that can result in incompatiblities with pre-SR10.2 releases.

1.3.1 Registry Server Incompatibility

The SR10.2 registry server includes new functionality that is exploited by the SR10.2 clients and the portable Passwd Etc product (see Subsection 1.6.2 of these notes). This section describes only an incompatibility issue.

The SR10.2 registry server is incompatible with the SR10.1 registry server, (rgyd) due to a data loss bug that existed in the SR10.1 version.

If any of the registry servers are running the SR10.2 registry server, all other servers (master and replicas) must also be running the SR10.2 rgyd. Failure to keep the master site and a replica site running the same revision of the code will cause the master to defer updates to that replica. As soon as the master site and replica site are again running compatible versions of the server code, all deferred updates will be propagated to the replica.

You can copy the SR10.2 rgyd onto earlier systems without updating the operating system; the SR10.2 server can run on nodes running SR10 and SR10.1.

We recommend that all sites update their registry sites to the SR10.2 rgyd as soon as possible, even if the registry site nodes are not updated to run the SR10.2 OS.

NOTE: Updating a registry site to the SR10.2 server code will not modify the registry database. All registry information will be preserved.

1.3.2 Compiler Incompatibility

Compiler versions that run on SR10 (see version numbers below) generate incorrect COFF object code when run under SR10.2.

Note that programs compiled with these older compilers will still run on 10.2 nodes. The binaries do not change, so you will not need to recompile.

You can find out if you have an earlier version compiler by using the **-version** compiler option. For example, type:

```
$ ftn -version
```

The incompatible compilers have the following version numbers:

Domain FORTRAN	10.00
Domain Pascal	8.09
Domain/C	6.00

NOTE: The compilers that are compatible with SR10.2 are:

Domain FORTRAN	10.6 or later
Domain Pascal	8.6 or later
Domain/C	6.6 or later

If you do not have the later version compilers, please contact your Apollo sales representative.

1.3.3 C Compiler Install Incompatiblity

The m68k C compiler (Version 6.6) installs a version of **/lib/syslib.881** that is not compatible with the **/lib/syslib.881** for SR10.2.

If you install SR10.2 to your node and then install the C compiler, you may receive an `undefined global` error because of the incompatible library. Note that this is not a problem with compiler versions later than 6.6, or with the Series 10000 compilers (Version 6.5).

If this is a new SR10.2 install, or if you have involed the disk, the workaround is as follows:

1. Install SR10.2.
2. Install the C compiler (6.6) using a separate **config** file.
3. Copy <*//authorized area*>**/install/ri.apollo.os.v.10.2/lib/syslib.881** to

   //target/lib/syslib.881

4. Reboot the node.

If you are updating from SR10 or SR10.1 to SR10.2, deselect the C compiler from the **config** file. If the compiler is the same **config** file as the OS, deselect CC from that **config** file.

1. **config -s** <*//authorized area*> **-c** <*config_filename*>

```
CONFIG > deselect cc
CONFIG > update OS full
CONFIG > config OS full
CONFIG > exit
```

2. Install the software using the changed **config** file.
3. Reboot the node.

If you need to reinstall the C compiler, use a separate **config** file by creating a new one. Copy the file as indicated above each time you install the C compiler, then reboot the node.

1.3.4 /usr/include/apollo/pfm.h

/usr/include/apollo/pfm.h has changed and may result in type mismatches when the **include** file is recompiled. Note, however, that executables remain fully compatible.

1.3.5 POSIX

We made the following changes to the operating system to be compatible with POSIX. They may result in incompatibilities with previous releases:

More than two slashes at the beginning of a pathname will be compressed to a single slash. (One or two slashes at the beginning of a pathname will not be changed.)

The operating system validates process groups in a different way:

- The process group of a process that has been executed cannot be set.
- The process group of a session leader cannot be changed.
- The process group ID of a process may be set to its **pid**.
- The process group ID of a process may be set to the process group ID of another process in the same session.
- **Tty** process group IDs may only be set to an existing process group in the same session. (They may also be set to the process id of a process in the same session.)

1.3.6 File Changes

The following files have been changed to include bug fixes at SR10.2, and may result in an incompatibility:

/sys/ins/fontn.ins.pas
/sys/ins/fontn.ins.c
/usr/apollo/include/fontn.h

1.3.7 pad_$dm_cmd Change

The **pad_$dm_cmd**() call can now return with a new completion status, defined as **pad_$insuff_rights** in the **pad_$** insert files. The **pad_$insuff_rights** completion status is new at SR10.2; previous versions of the **pad_$dm_cmd**() call would never return with this status. Associated with the new **pad_$insuff_rights** completion status is the following standard error message:

```
insufficient rights to perform DM command (display manager/Pad manager
```

The **pad_$dm_cmd** can be executed by a process whose user ID is either **root** or **locksmith** or is equal to the user ID of the person logged into the Display Manager.

1.3.8 New /usr/bin/yacc

The **/usr/bin/yacc** C compiler code has been updated at SR10.2. If you use the bsd4.3 version, you must use matching versions of **/usr/bin/yacc** and **/usr/lib/yaccpar**. The new files and the old and new limits are described in Subsection 1.8.4.

1.3.9 Aegis Print Manager Incompatibility

When installing the SR10.2 Aegis print services (/**sys/hardcopy**), you must update the print manager node before updating the print server nodes. Failure to do this will prevent an SR10.2 print server from restarting. If you cannot update the print manager node, copy the /**sys/hardcopy** tree to the print manager node and restart the print manager. The SR10.2 /**sys/hardcopy** tree will run on an SR10.1 node.

1.4 Hardware Support

SR10.2 provides support for the new Series 2500 workstation, the new DN3500 and DN4500 Desktop Visualization Systems, SCSI devices, and physical networks. The following subsections briefly describe new hardware.

1.4.1 Support for the Series 2500 Workstation

SR10.2 provides support for the new Series 2500 workstation. The Series 2500 workstation has its own documentation as indicated in Chapter 3. An overview of the product follows:

Series 2500 workstations are new 20-Mhz, 68030-based, low-end workstations that are compatible with other Apollo systems. The Series 2500 is available in 4MB, 8MB, 12MB, or 16MB configurations, with storage options that include optional internal Winchester disks in 100MB or 200MB sizes, and optional external CTAPE. The system can use a 15-inch 1024 x 800 monitor or a 19-inch 1280 x 1024 monitor. Network options include Apollo Ring, ETHERNET, or IBM Token Ring.

1.4.2 Support for the Desktop Visualization Systems

SR10.2 provides support for the new DVS. The DVS graphics controller has its own documentation as indicated in Chapter 3. An overview follows:

DVS is a new high-resolution (1280 x 1024) color graphics controller. DVS is available in both 8-plane and 40-plane versions. The 8-plane DVS controller can be used with DN3500, DN3550, and DN4500 workstations. The 40-plane DVS controller can be used with DN3550 and DN4500 workstations. Each version of the controller is a 2-board set that consists of a Transform Processor board connected to an 8-plane or 40-plane Array board. The 2-board set uses two PC AT compatible bus board slots within the workstation.

1.4.3 Hardware Previously Supported in PSKs

SR10.2 is the first major release to include support for the Multifunction Peripheral Controller (WD Controller), Apollo's SCSI 8-MM Tape Drive (used with Omniback), the SCSI 1/2-inch reel-to-reel tape drive, the SCSI 1/4-inch cartridge tape drive, and the new 70Hz monitor. These devices were previously supported in Product Support Kits (PSKs).

1.4.4 Support for SCSI Devices

At SR10.2, Apollo supports Small Computer Systems Interface (SCSI) devices that allow you to write your own SCSI device drivers. The **scsi_$** interface is not supplied with the base software but is available when you purchase the GPIO layered product.

The **scsi_$** routines allow you to communicate with SCSI devices connected to an Apollo workstation. Apollo's implementation complies with the ANSI X3.131-1986 standard.

For more information, see the SCSI routines in the online man pages **(scsi_intro)** and the online GPIO Release Notes.

1.4.5 Support for New Physical Networks

SR10.2 supports additional physical networks, including ETHERNET (802.3), ATR, and IBM Token Ring (802.5).

1.5 New and Changed Operating System Features

This section describes new or changed Domain/OS functionality in SR10.2.

1.5.1 POSIX Compatibility

The SR10.2 version of Domain/OS is compatible with the IEEE Standard Portable Operating System Interface for Computer Environments, Version 1003.1 (IEEE Std. 1003.1-1988). Details of Domain/OS compatibility are online in **/install/doc/apollo/os.v.10.2__posix.** Note that we do not conform to the following requirements:

- Interrupted opens change the access times of the opened file.
- We do not modifiy access/modify/change times on files correctly for read and write I/O calls.
- We fault on `disk full` rather than return the number of bytes that we could not write.
- Various terminal interface problems are under investigation.

1.5.2 Domain/X11 Included in Base Operating System

At SR10.2, Domain/X11 X Window System has been added to the base operating system. You have the option of automatically starting the X server when you install SR10.2. The minimum subset requires approximately 2.5MB of disk space (server, font space, and RGB color database).

1.5.3 New Audit Subsystem

SR10.2 provides an audit subsystem that allows administrators to establish auditing on selected nodes within an Apollo network. The audit subsystem allows you to:

- Select individual nodes to be audited.
- Specify event types that you want to monitor.
- Collect the log files that are generated on each audited node.
- Output the contents of log files in text form for reporting purposes.

Although the audit subsystem is included as part of SR10.2, it must be installed explicitly. When installing SR10.2 on an audited node or on the auditor node, you must respond to specific prompts regarding access controls and audit tools. Please be sure to read *Managing the Audit Subsystem* (016339) for specific information about installing and managing the audit subsystem.

1.5.4 New Commands

Note the following changes in SR10.2:

- Enhancements to **make(1)** and **vi(1).** For changes to **make(1)** see the **make(1)** man page.
- A new command, **mk(1)**, from the AT&T toolkit (**mk(1)** is similar to the existing **make(1)** utility).
- A new command, **pax(1)**, for POSIX compatibility (**pax(1)** is a new form of tape-archival utility).
- The **dstdump** utility, which dumps the debug symbol information from a COFF object file.

For more information regarding these changes and enhancements, see the online help files.

1.5.5 Enhancement to BSD makewhatis

The **makewhatis** script (**/usr/lib/makewhatis**) is used to create the **/usr/man/whatis** database. This database is used by **whatis(1)** to look up the man page header line for BSD commands, special file names, system calls, and library functions.

The command syntax is: **makewhatis** *[directory name]*

where *directory name* is the top-level man page directory. If a directory is not specified, the default is **/usr/man**.

At SR10.2, **makewhatis** scans both the formatted and unformatted (nroff source) man pages in the **/usr/man** tree, and creates the appropriate entry in **/usr/man/whatis** for each page. Prior to SR10.2, **makewhatis** scanned only formatted or unformatted pages, but not both. This prevented users from having entries for both formatted and unformatted man pages in the **whatis** database.

makewhatis should be run after initially installing the system software, and whenever any additional man pages are installed.

1.5.6 New option to /etc/edns

We have added an option to the tool **/etc/edns.** This option permits the administrator to specify a prefix to be used for cataloging diskless nodes in the **ns_helper** database. This cataloging occurs automatically when the subcommand **init** or **update** is invoked in **edns**. The default prefix is **diskless_$**. An alternate prefix for diskless nodes is specified as follows:

```
$ edns -na[me] prefix
```

Using this option can cause problems if shell programs depend on scanning for the default prefix. Also, if names with the default prefix already exist in **ns_helper**, they will not be modified even if **edns** is run with a new prefix. If only names with the new prefix are desired, the old ones must be deleted from the **ns_helper** database. The new prefix must be specified each time **edns** is invoked, or the default prefix will be used.

1.5.7 Enhancement to the tctl Command

The **tctl** command has added support for **/dev/tty0***x* devices. Previously, **tctl** would only function properly with **/dev/sio***x* devices. For example, if **/dev/tty01** were used with **/etc/getty**, and you issued a **tctl -line 1** to the **/dev/tty01** device, the line would reset, causing a line drop and an eventual unexpected logout. The added support comes in the form of an additional switch in **tctl** for **/dev/tty0***x* devices. The **-TTY***x* option should be used when programs such as **/etc/getty** or **siologin** use **/dev/tty0***x* devices.

Examples:

In a **/etc/ttys** file the following could be used:

```
tty01    "/etc/getty std.9600"  vt100  on
sio2     "/etc/getty std.9600"  vt100  on
```

To modify or view **tctl** settings for **tty01**, use:

```
tctl -tty 1
```

To modifiy or view tctl settings for **sio2**, use:

```
tctl -line 2
```

See /sys/**siologin/siomonit_file** and /sys/**siologin/startup_sio.sh** for additional examples.

1.5.8 Error Library

The **error_$** routines are portable, and extract the error text from the global library to a separate file read by the error routines. Also, **/lib/error** is being eliminated. The routines are moving to **/lib/kslib**. By removing the error texts to an external file, more global A space becomes available.

1.5.9 Security Fix for sendmail

sendmail includes a new flag, **-Xsmtpdebug,** which fixes a previous security problem. The flag addresses the security hole exposed by the well-publicized Internet bug. This flag is used to enable **smtp** debug mode. It may be specified only when **sendmail** is being started from **root** (it uses **uid,** not **euid**). Attempts by unauthorized users to use this flag are logged in the system log.

1.5.10 Enhancement to Domain/OS Sockets

The number of user Domain/OS sockets available has been increased at SR10.2 from 23 to 64 (for m68k systems). We now provide sufficient user Domain/OS sockets for each user process. This is important, as NCS applications are becoming more prevalent.

1.5.11 New Sector Striping Mode for invol Command

We have a new sector striping mode for the **invol** command. Sector striping is a disk striping technique that allows you to treat multiple physical storage disks as a single logical volume. Sector striping, also called interleaving, scatters sequential blocks across different drives in a multidisk set. In addition to allowing large files to span across more than one drive, this addressing mode increases the disk I/O bandwidth, since multiblock transfer is likely to involve all disks working in parallel.

The **invol** command automatically selects the new sector striping mode when requested, yet calls out multiple drivers per controller (for example, four disks on two controllers). The mode is automatically selected if:

- Sector striping is requested.
- There are fewer controllers than disks.
- There are the same number of disks per controller (for example, four disks evenly scattered across two controllers).

In this mode, the low-order disk address bits select among the controllers while middle-order disk address bits select among disks on the same controller.

You can achieve the performance benefits of sector striping (normally requiring a single disk per controller to reach those improvements) and allow for the added capacity (larger volumes) of combining more disks than controllers.

1.5.12 Paging File Size

At SR10.2 the paging file size has increased to 640 blocks (for m68k systems). Run **invol**, then select option 8 to modify the paging file size.

1.5.13 Involing a SCSI Storage Module

It is not necessary to add the manufacturer's badspot list to the Apollo badspot list if your device is a SCSI storage module. The SCSI device manages the manufacturer's badspots (they do not appear on the badspot list from **invol**).

When using option 6 of **invol** on a SCSI disk (to initialize the physical badspot list), the following message is displayed:

```
The manufacturer supplied badspots are handled by the disk
controller, and therefore do not appear in this badspot list.
```

Also, when **invol** is used to list the badspot list on a SCSI disk (option 7), the following message appears:

```
The manufacturer supplied badspots are handled by the disk controller;
there is no need to manually add the badspots listed on the drive.
```

1.5.14 Changes to the Release Image

Please make note of the following changes to the release image:

- **'node_data/boot_shell** has been moved to **/sys/boot_shell**. The boot shell itself is not changing, just its location.
- The file **/lib/error** no longer exists. A new file **/usr/apollo/lib/stcode.db** contains the error text.
- An internal restructure of the graphics library **/lib/gprlib** has resulted in a new file called **/lib/awslib**. This new file contains window-managing procedures.

1.5.15 Change to netman

netman now sends diagnostic output to a system log in:

'node_data/systmp/netman.out

This file is truncated (length = 0) at each boot request, and will never grow beyond one or two blocks. This means that no action is required to clean it up or keep it at a reasonable size.

1.5.16 Addition to lcnode

We have added the flag **-r(igorous)** to the **/etc/lcnode** command. The flag causes **lcnode** to perform retries when acquiring Apollo Token Ring topologies. The list takes approximately three times longer, but the accuracy has been dramatically improved. The flag has no effect on other network types. The **-r** option also eliminates duplicate lines.

1.5.17 Changes to the Loader

At SR10.2, the loader installs a program's external symbols into the private KGT if the program has any installed library (**-inlib**) requirements. We made this change to allow the libraries to reference routines in the main program. Programs that load libraries with **loader_$load** can also use this behavior to export main program routines and data by marking the program with a dummy installed library.

We have also included a new call, **loader_$unload** which unloads previously loaded object modules. See the online help file for more detail.

1.5.18 Changes in Available Kernel Space

At SR10, portions of the virtual address space were rearranged. As a result, we inadvertantly reduced the amount of virtual address space available for certain kernel operations on 68010-based machines. (User address space is unaffected.)

The affected machine types are: DN300, DN320, DN550, DN560, DSP80, and DSP80A. No other machine types are affected by the problem.

Affected nodes have a reduced ability to handle remote file operations. The extent of the new limitation depends on the machine type. The worst case occurs on the DN550, which is unable to handle most requests to create directories from remote nodes. This problem generates the following error message:

```
no space available (OS/MST manager)
```

The solution to the above problem results in two new limitations imposed on the affected machine types, as described below. Note that these limits pertain to all SAU2, SAU3, and SAU5 machine types, including those that are 68020-based. It is expected that most customer configurations will not approach these new limits, but it is necessary to be aware of them.

The amount of kernel space available to support per-process virtual memory resources is reduced in certain machines by 50 percent. Also, there are no implications for SAU2, SAU3, and SAU5 machines which do not boot diskless and are not partners for diskless nodes. There are many more of these resources available than could ever be used by a single node. There are implications only for diskless nodes and their partner nodes. This applies to:

SAU2: DN300, DN320, DN330
SAU3: DSP80, DSP90
SAU5: DN550, DN560, DN570, DN580, DN590

Other machine types have not changed their virtual memory support.

Note that the size of the virtual address space of these machines is not being reduced. only the total number of per-process virtual memory resources that may be used across all processes, including diskless child node processes.

If all diskless children connected to a given partner node, and the partner node itself, create the maximum number of processes (that is, 56), there are enough per- process virtual memory resources to fully support six nodes (five diskless nodes, and their partner). Of course, not every node runs at the maximum 56 processes, each using the average number of these resources. The actual number of diskless nodes supported by a single partner node depends upon type and volume of activity.

In addition to the above, the limit on the number of netrequest (**remfile**) servers which may be run on SAU5 (DN5*xx*) machines is reduced from three to two. This limit is imposed on SAU5 machines only. It does not apply to SAU2, SAU3, or any other machines.

The system automatically imposes this new limit on SAU5 (the limit for all other SAUs is 3, and is imposed by the **netsvc** command). No error message results if either **netsvc -s 3** or **netsvc -s** (the latter defaults to three) is issued.

1.5.19 Malloc

You should include **<apollo/shlib.h>** in those modules of an installed library which use **malloc**, **free**, or **realloc**. Doing so assures that references to these routines resolve to those defined by the application, if any. See **malloc**(3) (BSD) and **malloc**(3C) (SysV).

1.6 Network Enhancements

The following subsections describe network enhancements included in this release.

1.6.1 Changes to the Network Computing Kernel

SR10.2 contains Version 1.5.1 of the Network Computing Kernel (NCK). The following subsections describe enhancements to NCK in Version 1.5.1.

1.6.1.1 New glbd Option

The Global Location Broker (GLB) daemon, **glbd**, now has a **–listen** option that allows you to restrict the address families on which a GLB listens. This option supports special configurations where access to a GLB is restricted to a subset of hosts in the network or internet.

1.6.1.2 New glb_obj.txt Configuration File

The GLB is an object identified by a Universal Unique Identifier (UUID). This UUID has a default value. The new **glb_obj.txt** configuration file allows you to override the default value by specifying a different GLB object UUID for a particular host. This file supports special configurations that require several disjoint GLB databases (each of which is possibly replicated). In most networks and internets, there is only one GLB database (possibly replicated), and hosts do not need to have a **glb_obj.txt** file.

1.6.2 Changes to the Registry at SR10.2

We have added a password override file in this release:

/etc/rgy/passwd_override

This administrative file allows you to override password, GECOS, home directory, and shell entries in the registry database. Using the override capabilities, you can locally control individual machines. For example, you can prevent individuals, groups, or organizations from logging onto a particular system. To implement an override, add the appropriate information to the **passwd_override** file for each machine on which the override should take place. See the **passwd_override** help files for more information.

1.6.2.1 Establishing Override Policy

As for all registry properties, you establish override policies using the **edrgy** command. **edrgy** lets you establish override policy for Domain machines (Apollo nodes) and non-Domain machines (non-Apollo nodes). Override policy establishes whether the following types of overrides are allowed:

- Password exclusion (preventing users from logging onto a specific machine by entering a nonvalid string in the **passwd_override** password field).
- Root password overrides. (Note that in Domain/OS the override policy you establish for root also applies to **%.locksmith**. If you allow **root** overrides, you are also allowing **%.locksmith** overrides. If you exclude **root** overrides, you are also excluding **%.locksmith** overrides.)
- Nonroot password overrides.
- Nonpassword data overrides (GECOS, home directory, and shell).

1.6.2.2 New or Changed Registry Commands

The following commands have been enhanced at SR10.2 to accommodate the new override functionality of Password Etc. Please read the help files for complete descriptions:

- **bin/login** - signs a user on to the system initially and changes from one user to another
- **bin/su** - temporarily changes user identity
- **bin/passwd** - changes user password file information
- **bin/chfn** - changes the GECOS information entry in the password file
- **bin/chsh** - changes a user's login shell
- **bin/chhd** is new at SR10.2 - changes a user's home directory.

1.6.3 Source Routing Service on IEEE 802.5 Token Ring Networks

At SR10.2, we support source routing extensions on Apollo nodes that use the IEEE 802.5 Network Controller-AT.

Source routing is an optional service that is transparent to Domain and TCP/IP protocols. It allows Apollo systems on IEEE 802.5 networks to communicate over commercially available source routing bridges; for example, the IBM Networks Bridge Program version 1.1.

To use source routing, all Apollo nodes on the IEEE 802.5 network must run SR10.2 and have source routing turned on. Nodes that run SR10.1 can communicate with nodes that run SR10.2 only when these nodes are on the same IEEE 802.5 ring.

If your site is a Domain Internet, that is, you run Domain internet protocols with Domain network numbers, there are special requirements that apply to internet configurations that include IEEE 802.5 networks using source routing.

- Any number of IEEE 802.5 rings connected by source routing bridges form one Domain network, with one Domain network number.
- We do not support diskless booting over source routed bridges. Connect diskless Apollo nodes and their partners on the same IEEE 802.5 ring.

By default, the source routing service is enabled. To disable it (because, for example, the network consists of only one ring), add the following lines to the **/etc/rc** file immediately after **/etc/rtsvc**. Then uncomment the line(s) associated with the network **RING802.5_AT** and device number.

```
if [ -f /etc/ritr ]; then
(echo "Turning off ring 802.5 source routing" > /dev/console)
#  /etc/ritr -d 0 -o 1>/dev/null 2>>`node_data/system_logs/startup_ritr.log
#  /etc/ritr -d 1 -o 1>/dev/null 2>>`node_data/system_logs/startup_ritr.log
   fi
```

1.6.4 TCP/IP Enhancements

The enhancements to TCP/IP at SR10.2 consist of the following features:

- Performance gain
- Address Resolution Protocol (ARP) aging
- TCP/IP support for IEEE 802.5 networks
- Security enhancements to Domain/OS sockets
- New **/etc/rc.local** startup file
- Enhancements to the **/etc/ifconfig** command
- Extensions to the **trpt** utility

- Changes to the name server
- The **gated** routing daemon available in **/domain_examples**
- Extensions to **ioctl** routines
- Miscellaneous enhancements
- File-typing enhancements for **ftp**

The following subsections briefly describe these features. For complete information about all of these features except the **ftp** file-typing enhancements, see the manual *Configuring and Managing TCP/IP* (008543), which we have revised. Note that **ftp** file-typing enhancements are described at the end of this section.

1.6.4.1 Performance Gain

This version of TCP/IP performs significantly faster than previous versions, with slower nodes showing the most gain. This performance gain was accomplished by eliminating unnecessary data copies and wait states.

1.6.4.2 Address Resolution Protocol (ARP) Aging

At SR10.2, TCP/IP removes entries from the internal address mapping table that have not been used in 20 minutes. This prevents your address mapping table from containing host entries that may no longer be valid because of changes to a host's physical address.

1.6.4.3 TCP/IP Support for IEEE 802.5 Networks

TCP/IP now supports connection to IEEE 802.5 (IBM Token-Ring) networks. To define an IEEE 802.5 network interface with the **/etc/ifconfig** command, use **itr0** or **itr1** as the interface name. For example:

```
/etc/ifconfig itr0 <ip-address>
```

Apollo nodes now support the following four physical network interfaces:

Network Interface	**Interface Names**
Apollo Token Ring	dr0 dr1
IEEE 802.3 ETHERNET	eth0 eth1
IEEE 802.5 IBM Token Ring	itr0 itr1
Serial Line (SLIP)	sl0

In addition, TCP/IP supports source routing for 802.5 networks as specified in RFC 1042. All nodes in the set of interconnected 802.5 source routing rings -- that is, nodes on the 802.5 network connected by Media Access Control (MAC)-level bridges must have the same Internet network ID so that the nodes appear to be on the same physical local area network.

1.6.4.4 Security Enhancements to Domain/OS Sockets

TCP/IP at SR10.2 implements two security enhancements for Domain/OS sockets: protected sockets and secure raw sockets. Only **root** can bind a socket to a socket address where the port number is less than 1024, because port numbers less than 1024 are considered privileged. If you use socket calls, any socket with a port number less than 1024 must run as **root**. Any raw socket, regardless of the port number, must run as **root**. Your applications that use protected or raw sockets must either be run only by **root** or have the Set-User-ID (SETUID) bit set and be owned by **root**.

Note that standard utilities using protected or raw sockets, such as **rlogin**, **rcp**, and the **/etc/ping** command, have the SETUID bit set, and are owned by **root** so that they can run.

1.6.4.5 New /etc/rc.local Startup File

This version of TCP/IP includes a new **/etc/rc.local** startup file. At SR10.2, we've made the following changes to this file:

- Changed the default options for the **/etc/routed** command to **-f** (flush all routes from the local routing table) and **-q** (run in quiet mode). We recommend you remove the **-q** option when configuring gateway nodes.
- Changed the default for the **/etc/ifconfig** command to enable trailers for the IEEE 802.3 (eth0 and eth1) network interface. This means that, by default, Apollo hosts perform trailer link-level encapsulation on ETHERNET packets only when sending to hosts that have requested trailers.

  If the host you're configuring requires that trailer link-level encapsulation be disabled, add the **-trailers** parameter to the **eth0** or **eth1** **/etc/ifconfig** command lines. When you disable trailers for a particular host, that host never performs trailer link-level encapsulation even if the receiving host has requested trailers.
- Changed the location of the **syslogd** command so that **syslogd** now starts after the **ifconfig** command. At SR10.2, **syslogd** requires that an interface be configured with **/etc/ifconfig** if you are using TCP. Therefore, if you use the **rc.local** file that we shipped with SR10.1, **syslogd** will fail.

Chapter 3 in *Configuring and Managing TCP/IP* lists the situations that require you to edit the **/etc/rc.local** file. For information about how to edit the items in the **rc.local** file, see Appendix A of that manual.

1.6.4.6 Enhancements to the /etc/ifconfig Command

The **/etc/ifconfig** command configures a node's physical network interface(s). Typically, you invoke the **/etc/ifconfig** command in the node's startup file (such as **/etc/rc.local**) to specify physical layer information such as Internet addresses or subnet masks.

At SR10.2, we've added the following parameters to the **/etc/ifconfig** command:

macaddr

Change or display the Media Access Control (MAC) address for IEEE 802.3 (ETHERNET) networks. Use this parameter if you are running communications software (such as Apollo/TECHnet) that requires a different MAC address than the one supplied by Domain/OS. Note that **macaddr** is the only **ifconfig** parameter that you can invoke when the **tcpd** server is not running.

source_routing

Enable IEEE 802.5 source routing as specified in RFC 1042. By default, source routing is enabled when you configure the 802.5 network interface. Specify this parameter only when bringing up the 802.5 interface.

-source_routing

Disable IEEE 802.5 source routing as specified in RFC 1042. Specify this parameter only when bringing up the 802.5 interface.

For more information about using these parameters, see the **ifconfig** manual page.

1.6.4.7 Extensions to the trpt Utility

The **trpt** (transliterate protocol trace) utility helps you troubleshoot TCP/IP problems by displaying TCP packet traces.

At SR10.2, the **trpt** utility implements the following additional command line options:

-m Print output values in decimal format.

-u Print output values in unsigned decimal format.

For a complete description of the **trpt** utility, see Appendix D in *Configuring and Managing TCP/IP*.

1.6.4.8 Changes to the Name Server

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

The name server program is public domain software distributed by the University of California at Berkeley. The BSD name server and the associated resolver routines are called the Berkeley Internet Name Domain (BIND) utility. Domain/OS SR10 and SR10.1 shipped this public software with some changes required to port the name server to Domain/OS. At SR10.2, we've modified the name server program to include several changes. We have:

- Ported the BSD BIND program, Version 4.8, which required changes to the **named.boot** file.

- Created a separate, private, dynamic library to contain the BIND resolver routines.
- Provided the **/etc/nmconfig** utility to allow users to specify which name-address resolution method to use (**/etc/hosts** or **/etc/named**).
- Implemented name-address resolution using type managers and objects.

  For detailed information about the changes to the name server program, see Chapter 4 in *Configuring and Managing TCP/IP*. Appendix C in that manual describes changes to the named database files.

1.6.4.9 The gated Routing Daemon Available in /domain_examples

The **gated** daemon is a routing daemon that runs on a gateway. The **gated** daemon uses the Routing Information Protocol (RIP) and HELLO protocol, in addition to the Exterior Gateway Protocol (EGP).

At SR10.2, we supply the sources to the **gated** public domain software in the **/domain_examples/tcp/gated** directory. You can read the code and accompanying documentation in that directory as an example of how we ported to Domain/OS. Note that we support **gated** as an example only, and are not committing to full support at this time.

For more information about using the **gated** sources, see Appendix F in *Configuring and Managing TCP/IP*.

1.6.4.10 Extensions to ioctl Routines

BSD4.3 UNIX[*] allows processes to access TCP/IP information by directly seeking into **/dev/kmem**. Because Domain/OS does not locate TCP/IP information in **/dev/kmem,** we've implemented three additional **ioctl** routines to allow **gated** access the TCP/IP internal tables. These new **ioctl** routines are the following:

- SISCGARPTAB returns a copy of the internal address mapping table. This table associates Internet addresses with local physical addresses and is updated using the Address Resolution Protocol (ARP).
- SISCGIFENT returns a copy of the interface control block for one network board on the node. (Each network interface board has its own interface control block.)
- SISCGRTTAB returns a copy of the internal routing table. This table lists the accessible destination addresses and specifies which gateways to use from the local network to reach each destination network.

For information about using these new **ioctl** routines, see Appendix F in *Configuring and Managing TCP/IP*.

[*] UNIX is a registered trademark of AT&T in the USA and in other countries.

1.6.4.11 Miscellaneous Enhancements

At SR10.2, the TCP/IP server supports

- Directed broadcasts as summarized in RFC 1009. (Chapter 5 in *Configuring and Managing TCP/IP* describes how to enable directed broadcasts with the **tcpd -b** command.)
- Internet Control Message Protocol (ICMP) time stamp as described in RFC 792.

The TCP/IP server no longer supports the debug bit value, 0010, which reported data in TCP packets. The information proved to be too verbose and meaningless for debugging purposes.

1.6.4.12 File-Typing Enhancements for ftp

Domain/OS files are stored as typed objects such as **unstruct, coff**, and **uasc**. Even though Domain/OS supports typed objects, our **ftp** utility did not support typed objects prior to this release. So if users transferred typed objects using **ftp**, the **ftp** utility would, by default, label the object type **unstruct** (at SR10 and SR10.1), or **uasc** (prior to SR10). The **ftp** utility did not handle typed objects because it is based on the BSD version of **ftp,** and the BSD UNIX operating system does not have the concept of typed objects.

At SR10.2, we added the typed object functionality to **ftp** so that files transferred by **ftp get** or **put** commands in any Domain/OS environment retain their original object type. The **filetype** command is useful for storing typed objects on non-Domain/OS systems that might not support typed objects. For example, you might want to store your **coff** object files on a non-Domain/OS disk and, when you retrieve them, set **filetype** to **coff**.

To retain an object's type when transferring objects via **ftp**, issue the **filetype** command before transferring the objects with either the **get** or **put** commands. The **filetype** command remains in effect until you issue another **filetype** command, so you can perform multiple transfer commands for objects of the same type. To display the current local and remote settings, type **filetype** without any arguments.

The **filetype** command has the following syntax:

filetype *filetype_name* **[local | remote | both]**

where:

filetype_name — Can be any one of a set of supported Domain/OS or user–defined object types such as coff, unstruct, record, default.

local — Specifies that this **filetype** applies to **get** operations to retrieve remote files and store them on the *local* system.

remote — Specifies that this filetype applies only to **put** operations to send local files to *remote* systems.

both — Specifies that this filetype applies to *both* **get** and **put** operations.

If you do not use the **filetype** command to set the file type, the files will be assigned the default type, **unstruct**.

When you set the filetype on a remote system, the system can support this feature only if it is an Apollo system running SR10.2 (or later) software. If you specify the **remote** option and the remote system does not support this feature, you get a message and the remote filetype remains unchanged.

If you do not specify an option when specifying the **filetype** command, the default value is **both**. If **both** is set and the remote system does not support this feature, the local **filetype** will be set to the specified filetype while the remote filetype will remain unchanged.

The following is a sample **ftp** session using the **filetype** command.

```
% ftp wilderness
.
. /* Display current filetype settings. (None set.) */
.
ftp> filetype

Local target files will be created with the Domain default filetype.
Remote target files will be created with the remote system's default
filetype.
.
. /* Set filetype to coff for both remote and local files. */
.
ftp> filetype coff both
Local target filetype set to coff.
  200 Filetype set to coff.
  Remote target filetype set to coff.
.
. /* Transfer local file to remote system, retaining filetypes. */
.
ftp> get /bin/ls /tmp/ls_file_type
.
.
226 Transfer complete.
local: /tmp/ls_file_type remote: /bin/ls
14354 bytes received in 0.15 seconds (92 Kbytes/s)
.
. /* Display filetypes of files using ls -Tl command. */
.
% ls -Tl /tmp/ls*
coff      -rwxr-xr-x   1 zoo   14288 Mar 10 13:11 /tmp/ls_file_type
```

1.6.5 ETHERNET Performance Enhancements

We have enhanced TCP/IP and the ETHERNET microcode and driver to increase performance in the ETHERNET network for SR10.2.

1.6.6 UUCP Modification

The SR10.2 **/usr/lib/uucp/uucico** command is equivalent to SR10.0 **/usr/lib/uucp/uucico.real**; there is no **/usr/lib/uucp/uucico.real** in SR10.2. Therefore, all uucp accounts should have **/usr/lib/uucp/uucico** as the log-in shell.

1.7 Enhancements to the DM and Graphics Environments

The following subsections describe new and changed functionality in the Display Manager and the graphics environment. They also describe the bundled Domain/X11 X Window system.

1.7.1 Pads/Windows Increased

At SR10.2, the maximum number of pads has been increased from 40 to 100, and the limit on the maximum number of windows has been increased from 50 to 110.

1.7.2 New DM Commands

We have two new DM commands, **dmio** and **wmgr**. Use the **dmio** command to take down the DM Command and Error windows when you are using an X window manager. **dmio** was specifically added for X share-mode support.

dmio -on	Put up DM windows permanently.
dmio -off	Don't want DM input/output unless DM command needs it - then put windows up until input is completed.

wmgr turns the DM window management on and off.

wmgr -on
wmgr -off

For more information about **dmio** and **wmgr**, see the online help files. **wmgr** is also documented in *Using the X Window System on Apollo Workstations* (015213).

1.7.3 New DM Behavior

Please note that when a DM window is moved even one pixel off screen, input is not allowed in that window.

You will get an error message when attempting the **wi** command or attempting to send a fault to a foreign window.

Key defs made with **pad_$def_pfk** for input pads will also be applied to the related transcript pad.

1.7.4 Change for 4-Plane Nodes

There has been a change to the DM. If you run the X server on 4-plane nodes, color slots 14 and 15 in the color map are not used.

1.7.5 Changes to Startup Files

There have been two changes to /sys/**node_data/startup*** files (DM startup scripts that read and execute when the DM is first booted). The new startup file entry is as follows:

```
# To disable DM window management, uncomment the following command.
# Uncomment if running X in X-owns-root mode
#
# wmgr -off
#
# The following sets a default DISPLAY environment for X.
#
env DISPLAY ’:0’
```

The files are installed on your node in /sys/**dm/startup*** with a date code so that we don’t alter your customized startup files. Use the data in the new files for DM and X window management information.

New files are also installed in /sys/**dm/startup_templates***. Your existing files are renamed, but you have the opportunity to merge any of your personal changes into the new files.

NOTE: Do not start any servers from **’node_data/startup**.*xxx* files. Servers started from these files cannot be expected to run between logins. Use the appropriate **/etc/rc** files to start servers.

1.7.6 New pad_$set_erase

A new system call, **pad_$set_erase** has been included in this release. **pad_$set_erase** changes a pad’s erase mode, which determines how the pad is redrawn. See the online help file for more information about this call.

1.7.7 Using /usr/apollo/bin/kbm

Any key assigned a keyboard modifier function via the **kbm** command will not be sent to the DM.

1.7.8 System Color Map File Changes

The system color map has been moved from /sys/**dm/color_map** to **‘node_data/etc/dm_display/color_map** and to **/etc/templates/dm_display/color_map**. At boot time, the color map in **/etc/templates/dm_display/color_map** is copied into **‘node_data/etc/dm_display/color_map**. You can then modify the color map for your node by editing **‘node_data/etc/dm_display/color_map**.

The purpose of this change is to allow diskless nodes to have different color maps from their paging partners. For further information on diskless nodes, refer to one of the following manuals:

Managing Aegis System Software (010852)
Managing BSD System Software (010853)
Managing SysV System Software (010851)

1.7.9 lcm (load_color_map) Enhancement

There are three new options for the **lcm** command:

-r

Loads the true map. If no pathname is given, **lcm** loads the true color map from the file in **'node_data/etc/dm_display/color_map.rgb**. This file contains 256 color slots (000 -255). True color is available on the DN590, DN3550B, DN4500B and the DN10000VS. The default color map for true color is a linear ramp (0 0 0, 1 1 1, 2 2 2, etc.).

-l

Lists the contents of the display's color map. The output has the same format as the color map in **'node_data/etc/dm_display/color_map**. The display's color map is not changed if you use this option.

-s *<i> <r> <g> <b>*

Sets the color slot <i> to the rgb value specified. *<i>* is a decimal integer, from 0 through 15 on 4-plane displays, and from 0 through 255 on displays with 8 or more planes. *<r>, <g>, and <b>* are integers ranging from 0 to 255 that represent the brightness of the red, green, and blue components of the color value at slot *<i>*. You may use more than one **-s** option if you want to set more than one color slot.

1.7.10 New Option for cdm Command

The **cdm** command has a new option, **-p 7**, for use with the 3550B and 4500B systems. **-p 7** specifies shared color mode, in which both true color and pseudo color are available at the same time. With this option, the DM and other pseudo color applications can run in pseudo color, and true color applications can run in true color at the same time.

In shared color mode, pseudo color has 7 bits; the high-order bit is dropped. True color has 21 bits (7 red, 7 green, 7 blue); the low-order bits are dropped. There are 128 colors available in pseudo color and 8.35 million colors available in true color.

1.7.11 16-Bit Characters

Previously, GPR text commands only worked for input streams of 8-bit characters (ASCII and European text). Therefore, the system could not support languages with more than 255 different characters, such as Japanese. 16-bit fonts permit up to 65,535 characters. SR10.2 introduces a new set of GPR text routines that support 16-bit character streams. All ideographic text (and any other text with character codes above 255) requires these routines. For more information about this new functionality, see Appendix A (16-Bit Fonts and Characters).

1.7.12 Full Latin-1 Support

GPR now supports Latin-1 input. See Appendix A for more information.

At SR10 and SR10.1, the release notes described 8-bit font restrictions with PostScript printers (characters that would not print). In this release, the Latin-1 character set is fully supported on PostScript printers.

1.7.13 Domain/X11 X Window System

Domain/X11 is Apollo's implementation of the MIT X Window System. Version 1.2 of Domain/X11 is bundled with SR10.2 and supports Version 11 of the X protocol. The principal current releases of MIT's Version 11 are Release 2 and Release 3, usually referred to as R2 and R3. Domain/X11 V1.2 is essentially an R3 release except that the server does not contain the R3 bug fixes and support for backing store. Domain/X11 V1.2 supports and assumes the R3 Toolkit intrinsics, the R3 fonts (clients that need to use the old R2 fonts must add the old X11 font directory to the font path), and R3 versions of the clients.

The principal changes from the V1.0 version, which runs on SR9.7, include new system files (described below), a new location for the server, a server log, a general change of paths from **/sys/x11** to **/usr/X11**, support of UDS (UNIX Domain Sockets), an updated termcap, more efficient use of disk space, new run-time libraries with globally-recognized symbols allowing for smaller clients, support for starting the server at boot time, and Makefiles and Imakefiles for all clients.

The changes from the V1.1 and V1.1.p R2 versions include the preferred use of **xinit** to start the X server; the removal of the **xclient** client, the Makefiles for clients, the **X.starterkit** directory, and the 3D GMR and Open Dialogue example programs; the use of R3 libraries and clients; support for the X Toolkit; and improved performance.

Domain/X11 has several parts:

- An Apollo implementation of the X server, Xapollo, that runs on Apollo systems in share mode with the existing Apollo window system, the Display Manager (DM). This means you can have familiar Apollo windows (editing/transcript pads, native graphics, Interleaf) together with X clients (**xterm, xclock**) on your display screen at the same time. You can choose either the DM or X as your primary window system and use either DM window management or an X window manager (such as **uwm**) to manipulate (grow, move, raise, and so on) both types of windows.
- A supported version of the **Xlib** C library. **Xlib** is the programmer's interface to the X Window System. Apollo's **Xlib** includes extensions for running Apollo native graphics programs (GPR, 2D GMR, and 3D GMR) in a local X window. The Domain/X11 **Xlib** has been slightly modified so that it will be installed at boot time as a set of globally known symbols that, when referenced, will load the library code that resolves the symbol. This reduces X client size. **Xlib** is type-stamped **any** for both **runtype** and **systype**.

- Supported run-time libraries, **/lib/xtlib** and **/lib/xawlib**, for binding with **-inlib**. (There are also full versions of these libraries in **/usr/X11/lib**: **libXt.a** and **libXaw.a,** respectively.)
- A set of FORTRAN and Pascal interludes that make it possible to write Pascal or FORTRAN clients that call **Xlib**, which is written in C. The interludes bridge the incompatible conventions of parameter-passing between C and languages such as Pascal and FORTRAN. Interludes exist only for **Xlib** subroutine calls; the many macros do not have interludes. The interludes' Pascal include files have been modified (corrected) since V1.0 and V1.1; therefore clients built using the V1.0 or V1.1 interludes require minor modifications.
- Two supported X clients: the terminal emulator **xterm** and the window manager **uwm**. These are nearly identical to the MIT versions of these programs. (Apollo's supported **xterm** is limited to the VT102 terminal emulation mode; although **xterm** includes a Tek 4014 emulation mode, that mode is not supported.)
- Support for R3 fonts, which are more numerous than R2 fonts. R3 font naming conventions allow for longer, more descriptive names.

The supported and unsupported clients, and Domain/X11's other enhancements are described in more detail on the Xapollo man page and in the manual *Using the X Window System on Apollo Workstations* (015213) that accompanies this release.

1.7.13.1 A Double Set of Links

As it comes off the original MIT distribution tape, X11 expects to reside in a series of standard directories under the UNIX **/usr** directory:

/usr/bin/X11
/usr/include/X11
/usr/lib/X11

These are the pathnames assumed in documentation and coded internally in many places. However, on Apollo systems, customers can have two UNIX environments installed at the same time: BSD4.3 and Sys5.3. If Domain/X11 software were installed according to the MIT pattern, customers with both environments would need two copies of Domain/X11, occupying twice the disk space.

When installed, Domain/X11 software resides in a series of **/usr/X11** directories:

/usr/X11/bin	(pointed to by **/usr/bin/X11**)
/usr/X11/include	(pointed to by **/usr/include/X11**)
/usr/X11/lib	(pointed to by **/usr/lib/X11**)
/usr/X11/src	(includes compressed **doc** directories)
/usr/X11/examples	(pointed to by **/domain_examples/X11**)

The links under **/usr** make the UNIX and MIT standard pathnames described in documentation work correctly. Domain/X11 is thus intended to be installed as a set of double links that, while perhaps somewhat confusing at first, give the system administrator and the end-user a lot of flexibility in deciding what pieces of X go where (see Chapter 2).

Typically, an administrative node contains a full local copy of the **/usr/X11** directory (or as much of it as the administrator chooses, but usually at least the bin and include directories). User nodes are normally installed with links back to the administrative node for the subdirectories **/usr/X11/bin, /usr/X11/include, /usr/X11/src** (if installed), and **/usr/X11/examples** (if installed). Note, however, that **/usr/X11/lib** has to be a local copy, except that the fonts subdirectory in it can be a link, as can the development libraries **libXt.a** and **libXaw.a.**

1.7.13.2 Software on the Release Tape

The directories and files that follow are named as they appear on a node once Domain/X11 is installed.

/etc/Xapollo
The server.

/etc/xdm
The MIT xdm client

/lib/x11lib, xtlib, xawlib
A run-time version of **Xlib** with globally-known symbols, and runtime versions of **Xtlib** and **xawlib** for **-inlib** binding. The **/lib/X11** directory contains **libX11.a**, a dummy version of **Xlib** so that Makefiles will continue to work properly; it also contains the real libraries **libXt.a** and **libXaw.a** so that clients can be linked with them and thus continue to run even if the libraries change. If clients are **-inlib** bound with **xtlib** and **xawlib**, however, they will have to be recompiled and relinked when these libraries are updated in the future. **xtlib** and **xawlib**, while source compatible from version to version, are not guaranteed to be binary compatible (typically, argument types may change). **Xlib**, on the other hand, is guaranteed to have binary compatibility from version to version.

/lib/x11paslib
Interludes for Pascal and FORTRAN.

/usr/X11/bin (pointed to by **/usr/bin/X11**)
Contains binaries for the Apollo-supported programs **xterm** and **uwm**, and for Apollo-supplied clients such as **xownroot** and **dmwin**. In addition, you will find the binaries for many MIT supported programs such as **xclock** and **xedit**, and the MIT unsupported examples, demos, and utilities.

/usr/X11/include (pointed to by **/usr/include/X11**)
Contains all header files.

/usr/X11/lib (pointed to by **/usr/lib/X11**)
Contains X libraries and various data files such as the RGB color database and the X fonts. Also, it contains system-wide startup scripts for use with **startx, xinit, and xdm** clients.

/usr/X11/src
This directory contains **tar** archives for the MIT documentation and O'Reilly examples and errata. See the**/usr/X11/src/README** file for details on unpacking these archives.

/usr/X11/src/doc
If the **doc.tar.Z** file in src is uncompressed and unarchived, this directory contains the original troff sources to the MIT R3 documentation.

/usr/X11/src/oreilly
If the **oreilly.tar** archive **src** file is unarchived, this directory contains source code to examples in the O'Reilly documention, and the errata listings to the O'Reilly documentation. (See Chapter 3 in these release notes for more information on the MIT and O'Reilly documents.)

/usr/X11/examples (pointed to by **/domain_examples/X11**)
Contains source and binaries for example programs.

/usr/X11/examples/x_and_gpr
Shows how to run Apollo GPR native graphics in a disowned X window. This directory contains an example with an iconic interface to a graphics window that displays a set of concentric circles.

/usr/X11/examples/x_and_gpr_input
This program demonstrates how to process events from both GPR and X. GPR events are used in the GPR window, while X events are used in the X windows.

/usr/X11/examples/dmwin
The source to the **dmwin** program tries to determine whether it is being run from an X window or a DM pad, and behaves differently in the two cases.

/usr/X11/examples/pascal
Shows an X client written in Pascal using the Pascal interludes to the **Xlib** C library.

Apollo does not include the sources or hardcopy for the MIT contributed software documentation (the **Xr11, andrew**, and **clue** subdirectories).

1.7.13.3 Links and Installation Hints

To run X, you must have a number of links set up on your node.

Link	Directory/File
//node_name/bsd4.3/usr/bin/X11	/usr/X11/bin
//node_name/sys5.3/usr/bin/X11	/usr/X11/bin
//node_name/bsd4.3/usr/include/X11	/usr/X11/include
//node_name/sys5.3/usr/include/X11	/usr/X11/include
//node_name/bsd4.3/usr/lib/X11	/usr/X11/lib
//node_name/sys5.3/usr/lib/X11	/usr/X11/lib
//node_name/bsd4.3/usr/man/X11	/usr/X11/man
//node_name/etc/X0.hosts	'node_data/etc/X0.hosts

If any of the above link names already exists as a file or directory, the link will not be created at installation. Either update the file or directory or remove the file or directory and create the appropriate link. (Links can be created manually using the UNIX **/bin/ln -s** command. If you don't have the privileges to create the links, ask your system administrator for help.)

Note that if any of the links does exist but resolves to the wrong text, you must first remove it using **/bin/rm** and then recreate it.

1.8 Other System Software Enhancements

The following subsections describe enhancements to features that are bundled in the system software.

1.8.1 Floating-Point Performance Enhancements

Since SR10.1, we have improved both the performance and the accuracy of many of the arithmetic intrinsic functions. For applications written in C, most routines are at least two times faster, and many are eight to ten times faster. For FORTRAN and Pascal applications, the **asin, acos, sinh, cosh, and atanh** functions are two to four times faster, and their accuracy is significantly improved. (These comparisons are for code compiled without the **__builtins** directive.)

Because of these improvements, the results of math functions running under SR10.2 may differ from the results produced by the same functions running under SR10.1. In general, the new results are more accurate.

When using the **-def __builtins**, calls that are substituted with in-line operations are optimal, and have been since SR10. However, calls that are not replaced with in line operations are still many times faster in SR10.2, even though the builtins switch is used.

The in-line operations emitted since SR10 are faster than SR9.7, and in fact performance of many library calls, without using **-def __builtins**, are faster at SR10.2 than they were at SR9.7.

1.8.1.1 FPP Service Routines

We have modified the **fpp_$** routines introduced at SR10.1 and SR10.0.p. Originally, if you tried to use these routines to read or write a mode that the user is prohibited from altering, the routines returned an error status. Now, if you try to read a fixed mode, the routines return a zero status and the value to which the mode is fixed. If you try to write to a fixed mode by assigning it the value to which it's fixed, the routines return a zero status.

1.8.1.2 Floating-point I/O

At SR10.0.p (for Series 10000 workstations), we introduced FORTRAN and Pascal I/O routines that convert floating-point operands to ASCII strings and vice versa. We now use these routines to perform conversions on m68k-based workstations as well.

In some cases the results of these routines may differ from the results of conversion routines used in previous releases. In general, the results at SR10.2 will be as accurate as or more accurate than the older results.

The following sections describe some important characteristics of the new routines.

We have added NaN and infinity capabilities. NaN or INF output will appear as follows:

Output width	*SNAN result*	*QNAN result*	*+Infinity result*	*-Infinity result*
1	N	N	I	I
2	NA	QN	+I	-I
3	NAN	QNA	+IN	-IN
4	SNAN	QNAN	+INF	-INF

For field widths greater than 4, output is left-justified.

- The routines correctly convert denormalized values to small decimal numbers, instead of flushing them to zero. On Series 10000 workstations, however, if the IEEE underflow mode is off (the default), denormalized values will be converted to zero.
- The current rounding and precision modes and any other user-alterable modes have no effect on the input or output of these routines. The routines never trap.
- You can't use a denormalized value as input to these routines. If you use a decimal value smaller than the smallest allowed normalized value for the current format, the routines interpret the value as zero.

1.8.2 C++ Support

The Domain/C header files have been modified in the following two ways to make them more compatible with Domain/C++ programs:

- Function prototypes, if they existed in the file before, are now enabled for C++ compilations.
- Files declaring tag names that conflict with function names have been changed. These files now declare different tag names for C++ so that there are no conflicts. For example, struct stat was changed to struct stat_t for C++, because there is a function named stat.

As a result of these changes, the Domain/C++ translator does more type checking on function calls and may therefore generate errors for code that previously compiled without errors.

1.8.3 Domain/DDE

This section describes the changes made to Domain/DDE for SR10.2, including:

- Support for FORTRAN Types (Domain/DDE now supports FORTRAN types byte and integer*1).
- C++ Language Manager (Domain/DDE now includes a C++ language manager).

By default, Domain/DDE invokes the C language manager for C and C++ programs. To use the C++ language manager instead, issue the property language c++ command.

With the exception of function calls, the C++ language manager supports the C++ expressions and declarations described in the textbook entitled The *C++ Programming Language* (012777) by Bjarne Stroustrup, published by Addison Wesley.

The C++ language manager provides all the capabilities of the C language manager. In addition, the C++ language manager allows you to select identifiers with embedded colons (:) and tildes (˜), such as identifiers that include the C++ scope resolution operator (::) or the C++ destructor operator (˜).

For example, the C++ language manager allows you to select the class member function named myclass::myfunc. The C language manager, by contrast, would select only myclass or myfunc, since colons may not appear in C identifiers.

1.8.3.1 Overload Resolution

A program may include more than one function with a single name. The C language allows function overloading, as does C++, or you may write identically named, private functions in separate modules.

The Domain/DDE C and C++ language managers now provide a mechanism for differentiating between these functions. You specify a particular function by providing its name and enough of its parameters to uniquely identify it. The syntax is as follows:

'amb(function_name, ***parameter_types)***

where *function_name* is the name of the function and *parameter_types* is a list of the data types of the function's parameters, separated by commas.

Throughout this discussion, consider the following C++ function declarations:

overload print;
void print(int);
void print(double);
void print(char*, int);

The Domain/DDE syntax for referring to these functions, in order, is:

'amb(print, int)
'amb(print,double)
'amb(print, char*, int)

If you can uniquely identify the function without listing all of its parameter types, you can omit the last parameter(s) and end the list with an ellipsis (...). The ellipsis represents zero or more parameters of unspecified type. For example, you can refer to the function

void print(char*, int);
with the Domain/DDE syntax
'amb(print, char*, ...)

You can use the **'amb** syntax in a Domain/DDE command wherever you can use a function name. If you use an ambiguous function name in a Domain/DDE command, Domain/DDE responds by listing the **'amb** syntax for all functions of that name. For example, if you issue the Domain/DDE command

break \print

Domain/DDE responds

```
dde> bre \print
     ?(dde) Unable to resolve identifier "\print" to single loca
     \\ovld.cxx\`amb(print,int)
     \\ovld.cxx\`amb(print,double)
     \\ovld.cxx\`amb(print,char*,int)
```

In general, each language manager implements overload resolution rules appropriate for the language it manages. Currently, the C and C++ language managers implement a simplified version of the Domain C++ Version 1.2 overload resolution rules. Overload resolution considers built-in type conversions but not user-defined conversions.

For example, the command

break 'amb(print,char)

resolves to the print(int) function; promotion of char to int is a built-in conversion.

1.8.3.2 Compiling Programs for Debugging

The Domain compilers take two different options for including debugging information in the compiled code: **-dba** and **-dbs**. The **/bin/cc** interface to the C compiler takes the **-g** option for including debugging information in the compiled code. The following discussion is intended to clarify the behavior of these three options:

- With **-dba**, the compiler performs no optimization on the code, not even the minimal amount of optimization specified by the **-opt 0** option.
- The **-dbs** option does not eliminate any optimizations: the compiler will perform the default level of optimization, or will obey the optimization instructions specified by the **-opt** option.
- The **-g** option to **/bin/cc** is identical to **-dbs**, not **-dba**; that is, **-g** does not eliminate any optimizations.

To mimic the behavior of **-dba** in **/bin/cc**, use the **-W0,-dba** option instead of the **-g** option.

You may want to eliminate all optimizations because they can cause problems in debugging. In particular, optimizing can merge statements, eliminate statements, and shorten variable lifetimes. If the compiler merges several statements to a single PC and the debugger stops program execution at that PC, the debugger will report `Resolved to more than one location` followed by a list of those statements. Merged statements can also produce seemingly illogical jumps in control flow. If the compiler eliminates a statement, the debugger can't set a breakpoint at it. If the compiler limits a variable's lifetime to a specific PC range, the debugger can't display a value for that variable when the program is stopped outside of that range; instead, it will display the message `No valid value here.`

1.8.4 New Version of yacc

SR10.2 includes a new version of **yacc** that accepts larger grammars. The updated files are:

/sys5.3/usr/bin/yacc
/bsd4.3/usr/bin/yacc
/bsd4.3/usr/lib/yaccpar

If your grammars exceed the limits of the old **yacc**, you should see an immediate improvement. If you run sys5.3, you will have no compatibility problems. However, if you run bsd4.3, be careful to use matching versions of **yacc** and **yaccpar**. The new **yacc** will not run with the old **yaccpar** (**yaccpar** is a C source code that serves as the skeleton for the parser that **yacc** generates). You can find out whether you have the correct **yaccpar** when you compile the **yacc** output known as **y.tab.c**. If you get a *new* bsd4.3 **yacc** and an *old* bsd4.3 **yaccpar**, the combination won't work, and **cc y.tab.c** will produce compiling errors.

When bsd4.3 **yacc** runs, it tries to open the file **/bsd4.3/usr/lib/yaccpar** and use the contents as the skeleton parser. Make sure that the pathname points to the correct version.

Sys5.3 **yacc** and bsd4.3 **yacc** now have the same code. Sys5.3 **yaccpar** and bsd4.3 **yaccpar** are also the same. At SR10.1, and in previous beta releases of SR10.2, the **yacc** programs for the two kinds of UNIX were different. The only remaining difference between the two **yacc** programs is the systype of the coff file.

Here are the old and new limit values: (old values are those supported at SR10.1).

	Old	**New**	
ACTSIZE	12000	30000	Size of the action table
MEMSIZE	12000	30000	State space
NSTATES	750	16000	Number of states
NTERMS	127	1000	Number of terminal symbols
NPROD	600	3000	Number of productions in the grammar
NNONTERM	300	1000	Number of nonterminal symbols
TEMPSIZE	1500	16000	Temporary storage area
CNAMSZ	5000	5000	Size of an array of char holding token
LSETSIZE	600	4000	Number of distinct lookahead sets
WSETSIZE	350	4000	Working set size

1.9 Diagnostic Enhancements

At SR10.2, we have made several enhancements to the diagnostics as indicated in the following subsections.

1.9.1 The Diagnostic EXecutive (DEX)

DEX enhancements are as follows:

- We incorporated a modification to the **load -remove** command. It now removes the diagnostic file from memory with the variant name previously loaded.
- We added support for new devices such as the IBM Token Ring, Magtape, and Exabyte Tape.
- We added a fix for the auto command when under control of configuration RAM only.
- DEX contains full support for the new Series 2500 workstation.

1.9.2 GRTEST (Graphics Test) Revision 6.12

At SR10.2, we separated GRTEST from SAX. All of the files needed to run GRTEST under SAX are now part of the GRTEST release. We also eliminated the driver shell script to run GRTEST. All sax_reports and interaction to SAX are handled directly by GRTEST itself.

We also fixed the **-brief** option. Previously, **-brief** would not limit the standard output as it should. Now, when invoked with **-brief**, GRTEST will only print out essential information.

1.9.3 System Acceptance EXerciser (SAX)

The SR10.2 revision of SAX has the following enhancements over SR10.1.

SAX Executive

We added **-partner**[*4-7*] for support of 8 networks.

Ctape

We increased **wear_pass_count** maximum value to 2200.

Networks

- Added support for 802.5 ring.
- Added support for up to 8 networks.
- Added support for partnering to non-native networks for the express intent of verifying routers.

Winchesters

We added support for up to 8 controllers
    controllers #0-3 have support for 4 drives, 0-3
    controllers #4-7 have support for 1 drive, 0

We changed the name of win tests to reflect controller, drive pair as follows:
    controller 0 drive 0 ... win_0_0
    controller 0 drive 1 ... win_0_1
    .
    .
    .
    controller 7 drive 0 ... win_7_0

Memory Test

Memory test now includes paging to remote nodes if **net.sh** is running, or if **-partner**[*n*] is specified in SAX argument list.

1.9.4 Other Diagnostic Changes

win7a.drvr Rev 3.3: Built February 15, 1989 3:35:59 pm (EST).

This version fixes a bug that occurred when running **win7a.dex** in an EMT window. Extra carriage returns were being issued during every command sent to the driver.

win7b.drvr Rev 2.9: Built January 20, 1989 1:05:05 pm (EST)

Known defects were occurring on the disk. The error was detected, but the incorrect sector was reported as sector 0 (zero). This has been corrected.

win7b.dex, win7b.drvr (SAU7)

A bug in test 170 for Winchester disk drives was fixed. The test would fail after running the controller diagnostic test (the default). A badspot was present on a diagnostic cylinder.

win8.drvr (SAU8)

Dual Winchester disk drive support was added.

disp7e.dex Rev 2.4: Built January 18, 1989 10:44:49 am (EST)

This diagnostic now contains support for the new 70-Hz color display monitor. We also made a change to catch CPU memory to display memory BLT errors that were not being detected.

disp7c.dex Rev 2.6: Built November 28, 1988 10:26:44 am (EST)

Test #170 was not testing the D/A converters (DACs) voltage output levels correctly as per the DAC specification.

itr.dex (sau7) Rev 2.0: Built December 9, 1988 4:37:44 pm (EST)

Test 660 (transmit to self) was enhanced to check for nonerror conditions and output warning messages instead of error messages. We also added multicontroller capability.

itr.dex (sau8) Rev 2.0: Built December 9, 1988 5:26:26 pm (EST)

Test 660 (transmit to self) was enhanced to check for nonerror conditions and output warning messages, instead of error messages. We also added multicontroller capability as well as fixes to potential timing problems due to incorrect controller board initialization.

ring7b.dex (sau7) Rev 3.4: Built June 1, 1989 5:45:41 pm (EDT) and

ring8b.dex (sau8) Rev 3.4: Built June 1, 1989 5:54:04 pm (EDT)

1. **bit_blaster** test (previously test 10) is now executed as two separate tests positioned with the digital and analog loopback tests (new test numbers are 207 and 307). We also moved tests 11 - 13 to 10 - 12 to fill the gap left by the repositioning of **bit_blaster**. New test numbers are used as a reference for the remainder of this description.

2. Test 11; **alt_adrs** now prints a warning that the test is not run when the ring board is jumpered as unit 1. This is because the diagnostic hook only allows this test to be run when the board is jumpered as unit 0. The test previously failed if run on unit 1.

3. Test 10; **at_xt** test is now enabled for all revisions of the Single-Board Apollo Token Ring (ATR) module. It was previously disabled for an expected release of the board in which a necessary diagnostic hook was not working. This version of the board was never released and the hook has been fixed on its replacement version.

4. We added a message to tests 400 and 401 to warn of possible failures if these tests are not run on a single node network.

Chapter 2: Installing SR10.2

We have updated the manual *Installing Software with Apollo's Release and Installation Tools* (008860). That manual describes the tools for installing SR10.2 and layered products. This chapter includes specific installation information that is not included in that manual. Please be sure to read the information in this chapter that describes the various install configuration sizes, including (Tables 2-1 and 2-2). Systems that have smaller disk drives may not be able to accomodate the larger install configurations.

NOTE: In order to install SR10.2, you must use the tools from the SR10.2 boot tape in the **/install/tools** directory. Install tools from previous releases will not work. For more information, see the section titled "Loading New Products into an Authorized Area" in Chapter 5 of *Installing Software with Apollo's Release and Installation Tools* (008860).

2.1 Note About the Hardware Acceptance Test

Before you install software, please note that when you run the hardware acceptance test on a DN3500 node, there is a possibility that you could receive error messages.

A 4MB DN3500 that is heavily configured may exhibit `rectangle not acquired` or `overload` error messages while running the acceptance test. This is not a fundamental system problem and will not appear if the acceptance test is rerun with a more constrained configuration.

2.2 Special Install Notes for Beta Sites

Although the **minst** installation tool is capable of updating an SR10.1 node to SR10.2, you cannot use **minst** to write over an existing version of SR10.2 (beta customers who may update). You must either delete or move the old product tree from the Authorized Area. The only requirement is that **minst** (and **distaa**) not see the old OS tree as a valid OS product and try to reuse it. Do this by issuing the command:

Aegis: **dlt -l** *<authorized area>***/install/ri.apollo.os.v.10.2**
Unix: **rm -r -v** *<authorized area>***/install/ri.apollo.os.v.10.2**

Note that you may have to be root to do this.

You may also choose to move the existing prerelease version:

Aegis: **mvf** *<authorized area>*/**install/ri.apollo.os.v.10.2** @
<authorized area>/**install/ri.apollo.os.v.10.2.old**

Unix: **mv** *<authorized area>*/**install/ri.apollo.os.v.10.2** \
<authorized area>/**install/ri.apollo.os.v.10.2.old**

If you use **distaa** to load SR10.2, and had an earlier prerelease version of SR10.2 installed, make the new override file active after the load completes. Use the command:

Aegis: **cpf** *<authorized area>*/**install/templates/apollo/os.v.10.2/ov.***<template>*
<authorized area>/**install/overrides/ri.apollo.os.v.10.2**

Unix: **cp** *<authorized area>*/**install/templates/apollo/os.v.10.2/ov.***<template>*
<authorized area>/**install/overrides/ri.apollo.os.v.10.2**

where ov.*template* is the name of the override file associated with the SR10.2 configuration you loaded (for example, **ov.aegis_small**.) If you loaded the entire operating system, the appropriate override file is **ov.large**.)

2.2.1 Contents of /install/doc/apollo Directory

If you have a prerelease version of SR10.2 (beta customers who may update), please note that the install will not overwrite the contents of the **/install/doc/apollo** directory. The result is that you will still have older versions of the documents in that directory. The workaround is to copy the latest version of the **/install/doc/apollo** directory from the <authorized area>, or delete the existing files before you update.

2.2.2 Possible Warning Messages

NOTE: If you are updating from a prerelease version of SR10.2, you may see some of the warnings listed on the following pages. The warnings occur because we have replaced certain files with links. When updating to a later baselevel (or if you have customized your files from an earlier SR10-based release), please ignore the warnings:

```
WARNING:/sys/dm/color_map would replace a link - item is ignored
WARNING:/sys/dm/fonts/f5x9.b.iv would replace a link - item is ignored
WARNING:/usr/X11/lib/libX11.a would replace a link - item is ignored
WARNING:/usr/X11/lib/libXaw.a would replace a link - item is ignored
WARNING:/usr/X11/lib/libXt.a would replace a link - item is ignored
WARNING:/usr/X11/lib/liboldX.a would replace a link - item is ignored
WARNING:/bsd4.3/bin/login would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/crt0.o would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/gcrt0.o would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/lib2648.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/lib300.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/lib300s.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/lib4013.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/lib4014.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/lib450.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libF77.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libI66.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libI77.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libU77.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libcurses.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libdbm.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libf77plot.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libl.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libld.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libln.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libmp.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplot.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplot2648.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplot7221.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplotaed.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplotbg.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplotdumb.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplotgigi.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libplotimagen.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libtermcap.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libtermlib.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/libvt0.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/liby.a would replace a link - item is ignored
WARNING:/bsd4.3/usr/lib/mcrt0.o would replace a link - item is ignored
WARNING:/sys5.3/bin/login would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/crt0.o would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/gcrt0.o would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/lib300.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/lib300s.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/lib4014.a would replace a link - item is ignored
```

```
WARNING: /sys5.3/usr/lib/lib450.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libF77.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libI77.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libPW.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libcurses.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libgen.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libl.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libld.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libmalloc.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libnsl.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libplot.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libtermcap.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libtermlib.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/libvt0.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/liby.a would replace a link - item is ignored
WARNING: /sys5.3/usr/lib/mcrt0.o would replace a link - item is ignored
```

2.3 Updating From SR10.1 to SR10.2

If you install SR10.2 on a node running SR10.1, and you do not run **invol**, you must set the paging size (**invol** option 8) after installing SR10.2 and before rebooting. The new paging file size at SR10.2 is 640 for m68k systems.

2.4 Canned Selection Files and Configurations

SR10.2 is shipped with 13 pairs of selection and override files, plus one configuration file that works with any pair of selection/override files. Use these files to install your software. *Installing 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 product components and their sizes. The last subsection describes what product components are loaded into the AA and can consequently be installed with the configuration file **/install/templates/apollo/os.v.10.2/cf.os**.

Selection files for the various Domain/OS sizes reside in //*<authorized_area>*/**install/templates/apollo/os.v.10.2**. The choices are as follows:

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

2.4.1 Selection Component Descriptions

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

sysboot   The program that loads Domain/OS into memory.

sau/n base   A directory containing the required stand-alone utilities.

sau/n diagnostics   Offline hardware diagnostics used for troubleshooting hardware problems.

sau_sys help   Help files for offline hardware diagnostics.

systest   A directory containing various online system tests and exercisers.

systest/ssr_util   A directory containing field service utilities.

install utilities   Domain/OS commands used by one or more of the installation tools.

domain_examples   A directory containing online programming examples.

sr9.7_compatibility   A directory containing files to be used in a network where some nodes are running SR10.*x* and some nodes are running SR9.7.

standard fonts   The standard system fonts (installed on every node) are in the directory **/sys/dm/fonts**.

optional fonts   All 8-bit fonts not considered standard. Nonstandard fonts include the families 'charter' and 'new century schoolbook' in all sizes (8, 10, 12, 14, 18, 24) and styles (roman, bold, italic and bolditalic), oblique and boldoblique styles of 'courier' and 'times' in all sizes, 'courier' in all styles and all but standard sizes (8, 14, 18, 24), all standard font families in size 8, all styles, an Old-English font, some graphics fonts (chess, symbol), and unusual point sizes (7, 9) for some of the standard fonts. These are in **/sys/dm/fonts**.

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

lib
The Domain/OS libraries.

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

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

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

com
A directory containing a large set of Aegis environment commands.

sys base
The top-level Domain/OS system directory.

sys/help
A directory containing help files for the Aegis environment.

sys/ins
Apollo-specific include files for Aegis software development.

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

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

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

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

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

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

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

bsd4.3 etc	BSD environment commands that reside in **/etc** and either have nonidentical counterparts or no counterparts in a SysV environment.
bsd4.3 usr	BSD environment commands that reside in **/usr** and either have nonidentical counterparts or no counterparts in a SysV environment.
sys5.3 base	All commands and files that are specific to the SysV environment except those included in the sys5.3 **etc** or sys5.3 **usr** components.
sys5.3 etc	SysV environment commands that reside in **/etc** and either have nonidentical counterparts or no counterparts in a BSD environment.
sys5.3 usr	SysV environment commands that reside in **/usr** and either have non identical counterparts or no counterparts in a BSD environment.

2.4.2 Selection Component Tables

The following tables list the software components that are loaded into your Authorized Area if you use the predefined selection files. They also specify the sizes of each component that is installed. They should help you determine the particular selection file that is most appropriate for your use and disk sizes. The first table covers the small and medium sized selections; the second table covers the large selections.

Note that these tables give the total size of the Authorized Area, and the size of the software that will be installed on your node if you choose one of the standard templates. You can reduce the size of the software that is installed on the node by using a customized configuration file instead of the one supplied in **install/templates/apollo/os.v.10.2/cf.os**. If you use a customized configuration, the messages displayed during the **config** operation indicate the amount of disk space used by your selections.

Each row in the 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.

Each column corresponds to a particular predefined selection file. For example, the AVM column defines the contents of the **aa.aegis_sys5.3_medium** selection file. The key above Table 2-1 lists the meanings of the one-character selection file identifiers.

NOTE: The disk where the authorized area is located must have a minimum of 10 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)

TABLE 2-1. AA Size for Small and Medium Selections

Component	Size (MB)	Selection File Code						
		AS	ASP	AM	BM	VM	ABM	AVM
sysboot	0.01	X	X	X	X	X	X	X
sau/n base	15.3	X	X	X	X	X	X	X
sau/n diagnostics	14.9							
systest	4.3							
systest/ssr_util	2.1							
install utilities	0.5	X	X	X	X	X	X	X
domain_examples	1.7							
sr9.7_compatibility	3.4							
optional fonts	2.7			X	X	X	X	X
lib	6.2	X	X	X	X	X	X	X
etc base	4.5	X	X	X	X	X	X	X
guaranteed commands	1.9	X	X	X	X	X	X	X
com	2.0	X	X	X			X	X
sys base	15.3	X	X	X	X	X	X	X
sys/help	1.7							
sys/ins	1.2		X	X			X	X
sys/source	0.3							
usr base	4.7	X	X	X	X	X	X	X
usr/apollo/include	0.9		X	X	X	X	X	X
usr/apollo/man	1.7							
usr/X11	10.2							
usr/games	2.7							
usr/new	4.8							
bsd4.3 base	1.3				X		X	
bsd4.3 etc	0.4				X		X	
bsd4.3 usr	12.4				X		X	
sys5.3 base	1.8					X		X
sys5.3 etc	0.4					X		X
sys5.3 usr	15.9					X		X
Total (Approximate)	135.2	50.4	52.5	55.5	66.4	70.4	69.3	73.3

TABLE 2-2. AA Size for Large Selections

Component	Size (MB)	Selection File Code					
		AL	BL	VL	ABL	AVL	ABVL
sysboot	0.01	X	X	X	X	X	X
sau/n base	15.3	X	X	X	X	X	X
sau/n diagnostics	14.9	X	X	X	X	X	X
systest	4.3	X	X	X	X	X	X
systest/ssr_util	2.1	X	X	X	X	X	X
install utilities	0.5	X	X	X	X	X	X
domain_examples	1.7	X	X	X	X	X	X
sr9.7_compatibility	3.4	X	X	X	X	X	X
optional fonts	2.7	X	X	X	X	X	X
lib	6.2	X	X	X	X	X	X
etc base	4.5	X	X	X	X	X	X
guaranteed commands	1.9	X	X	X	X	X	X
com	2.0	X			X	X	X
sys base	15.3	X	X	X	X	X	X
sys/help	1.7	X			X	X	X
sys/ins	1.2	X			X	X	X
sys/source	0.3	X			X	X	X
usr base	4.7	X	X	X	X	X	X
usr/X11	10.2	X	X	X	X	X	X
usr/apollo/include	0.9	X	X	X	X	X	X
usr/apollo/man	1.7		X	X	X	X	X
usr/games	2.7		X	X	X	X	X
usr/new	4.8		X	X	X	X	X
bsd4.3 base	1.3		X		X		X
bsd4.3 etc	0.4		X		X		X
bsd4.3 usr	12.4		X		X		X
sys5.3 base	1.8			X		X	X
sys5.3 etc	0.4			X		X	X
sys5.3 usr	15.9			X		X	X
Total (Approximate)	135.2	93.8	112.0	116.0	117.1	121.1	135.2

NOTE: An additional minimum of 10 megabytes of free space must be available during the installation from media.

2.4.3 Software Loaded into the Authorized Area

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

2.4.3.1 Small Aegis (aa.aegis_small)

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

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

You do not get the following:

- **/domain_examples**
- **/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 of **/com**, including:

  — Open System Toolkit utilities (**crty**, **crtyobj**)
  — Serial line communication commands (**em3270.***xxx*, **siorf**, **siotf**)
  — Spelling checker software (**fserr**)

- Any programming tools, including the high-level debugger (**dde**), **/com/db,** or any include files (**/sys/ins** or **/usr/include**)

The small Aegis selection (**aa.aegis_small**) requires approximately 50 megabytes. See column AS in Table 2-1.

2.4.3.2 Small Aegis for Programmers (aa.aegis_small_prog)

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

- The high-level debugger (**dde**) and **/com/db**.
- All of these include files:

  — **/sys/ins** (***.ins.*** files for Domain/OS calls)

- — **/usr/include/apollo** (***.h** files for Domain/OS calls)
- — **/usr/include** (***.h** files for BSD or SysV calls)

The small Aegis selection for programmers (**aa.aegis_small_prog**) requires approximately 53 megabytes. See column ASP in Table 2-1.

2.4.3.3 Medium Aegis (aa.aegis_medium)

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

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

The Aegis medium selection (**aa.aegis_medium**) requires approximately 56 megabytes. See column AM in Table 2-1.

2.4.3.4 Large Aegis (aa.aegis_large)

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

- Hardware diagnostics
- **/systest,** including **/systest/ssr_util**
- **/domain_examples**
- **/sr9.7_compatibility**
- **/sys/help**
- **/sys/source**

The Aegis large selection (**aa.aegis_large**) requires approximately 94 megabytes. See column AL in Table 2-2.

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

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

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

You do not get the following:

- **/domain_examples**
- **/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**)

The BSD medium selection (**aa.bsd4.3_medium**) requires approximately 66 megabytes (see column BM in Table 2-1). The SysV medium selection (**aa.sys5.3_medium**) requires approximately 70 megabytes (see column VM in Table 2-1).

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

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

The large BSD selection (**aa.bsd4.3_large**) requires approximately 112 megabytes (see column BL in Table 2-2). The large SysV selection (**aa.sys5.3_large**) requires approximately 116 megabytes (see column VL in Table 2-2).

2.4.3.7 Combination Medium Selection Files

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

The combined Aegis and BSD medium selection (**aa.aegis_bsd4.3_medium**) requires approximately 70 megabytes (see column ABM in Table 2-1). The combined Aegis and SysV medium selection (**aa.aegis_sys5.3_medium**) requires approximately 73 megabytes (see column AVM in Table 2-1).

2.4.3.8 Combination Large Selection Files

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

The combined large selection for all three environments (**aa.large**) requires approximately 135 megabytes. See column ABVL in Table 2-2.

2.5 Known Bugs and Limitations in minst

2.5.1 SAUs and Install Targets

Expert mode **minst** does not yet check that each of the install targets will receive the /**sau** (stand-alone utilities directory containing the Domain/OS kernel) necessary for proper node booting.

Expert mode **minst** does not check that each of the install targets exists.

Expert mode **minst** does not check that any of the install targets share the same physical volume as the Authorized Area when purging old Domain/OS products from the AA. Disk space will not be freed up as expected in such a case.

2.5.2 Update vs. New Install

minst does not know about Update installs. Every install is a new install. This may lead to a problem when moving a node from a previous version of Domain/OS to the new version without initializing the target disks. YOU MUST RETAIN THE SAME ACL MODEL. The install does not know how to close an open ACL model; it does know how to open up a closed model.

If your install has open ACLs, use open ACLs when you update. If your install has closed ACLs, use closed ACLs when you update.

2.5.3 New Template File for ACLs

We have included a "canned" template file that you can use with the **inprot** (install protections) tool to change *open* Domain/OS ACLs to *closed*. The file is located in:

*<//authorized area>***/install/templates/apollo/os.v.10.2/ip.closed_sysv**

The **inprot** tool is located in:

*<//authorized area>***/install/tools/inprot**

For more information about **inprot**, see *Installing Software with Apollo's Release and Installation Tools* (008860).

2.5.4 Bugs Fixed in the Current Version of minst

Numerous faults have been found and corrected since the last major release of **minst** in December of 1988. The list follows, in reverse chronological order:

- The help file *<AA>***/install/help/minst.hlp** has been corrected and updated.
- Use the pathname resolution service appropriate to the version of the OS on the node running **minst**.
- If **minst** cannot find a questions file or template questions file, allow one chance to pull it in from file 1 of media, then try again.
- Novice mode is now independent of questions file.
- **minst** now checks that the tools it needs (**rbak_sr10** or **rbak_sr9**) are present in the Authorized Area **tools** or **tools_sr9** directory, or in the directory from which **minst** is invoked, before continuing.
- Correctly sense which OS (pre-SR10 or SR10.*x*) is present on the node running **minst**.
- In Expert mode, when doing *separate* load and installs, **minst** now asks: `Do you wish to load more software to you Authorized Area (and install same to the target node(s)), or are you done?  : [ load_&_install  done ]`
- **minst** now correctly decides whether to load the tools from file 1 on media. It is no longer necessary to manually issue the **rbak** command to forcibly load the contents

of file 1 on media.

- **minst** no longer displays certain product names and their associated document names twice (this was a substring matching problem).
- **minst** tells you it's finding or fixing the release index and doing temporary file cleanup.
- **minst** checsk the free space of the disk volume that actually contains the Authorized Area. This bug is in the other RAI tools (**distaa, install**) and has not yet been fixed; the free space of the root volume of the node containing the AA is checked -- no problem when the AA is on the root volume.
- If **minst** finds that there is insufficient free space in the Authorized Area disk volume to contain *any* configuration of the OS product, you will be told this fact before being asked if you wish to prune old OS products from the AA. If there is still insufficient free space to hold even the smallest OS product configuration, minst will exit.
- When minst sends output to the serial I/O port (or to the vt100 terminal emulator), it pauses after 16 lines of output and issues the prompt `*** Press <RETURN> to read more ***`. Previously, long output messages would quickly scroll out of view.
- Sizing estimators return different values depending on whether a *PRISM* product (***.p***) is being sized or not.
- When in the **install++** phase, warns that the install choice(s) may be made after release notes are listed for reading.
- Allows **quit** as an option to the proceed/reselect prompt for multiple selected products.
- Allows purging of the Authorized Area whenever Domain/OS is presented on media, regardless of availability of free space in AA.
- Always asks if installing Domain/OS.
- When selecting what to install, presents the list of release notes of the loaded products and asks that they be read before making install selection(s).
- When installing Domain/OS, some questions are not asked:
  - — no install selection necessary in either Novice or Expert mode
  - — in Expert mode, no `separate/together` install question or `all/select` load question is necessary.
- In Expert mode, asks for the list of install targets before asking for the list of SAUs to install.
- Added **minst** start and end timestamping.
- Eliminated shutdown target message when no install happened.

- **minst** does not attempt to load/install when the AA purge step cannot free up enough space in the AA to hold the product.
- **minst** does not attempt to install a product that was not successfully loaded by **distaa**.
- When purging old Domain/OS product objects from the Authorized Area, and trimming the Authorized Area or install target filesystem to minimum bootable form, removes only objects that are on the same physical volume as the Authorized Area. Doesn't follow links to other nodes or filesystems. Always leaves the release index for the product in the AA for use by RAI tools.
- Reworded the confusing proceed/continue question to load/install.
- States clearly that Authorized Area and (in Novice mode) install target pathnames must exist.
- When selecting products for loading or installing from a list of choices, gives feedback when an incorrect choice is made and allow retry.
- Clearly states how to get the default answer to a question.
- Clearly states that the first character is usually sufficient to answer a question.
- **minst** deletes temporary files on entry rather than at exit.
- Added shutdown and reboot message when **minst** exits after installing Domain/OS (or launching remote installs of Domain/OS). Reminds user to shutdown and reboot target *after* install has completed, not just when **minst** exits.
- Reworded welcome message.
- Describes how to read release notes when the list is presented.
- **minst** no longer lies to **install++**, **config**, and **install** about the open or closed ACL model to use for Domain/OS.
- **minst** redisplays the selection templates list when a choice is not confirmed.
- Added help as an option to some **minst** questions.
- Displays full pathname to any document (release notes, transition guide) for a selected product for loading.
- Shows the pathname of the answers file when written.
- Asks about ACL model (open or closed) and /**sau** list entry *only* when Domain/OS will be installed.
- Made warning and error messages more informative.
- Fixed serial I/O mode scrolling of <*AA*>**/install/tools/config_info_file.**
- Fixed version comparison when selecting product(s) from a list. For example, **minst** wasn't able to distinguish **4.0.p** from **4.0**.

- Checks that the Authorized Area (and in Novice mode, the install target) path exists and is accessible. Echos the full pathname of the AA and target.
- Added novice mode **minst**. Former **minst** interface now available as Expert mode.
- Added to sign-on message some explanatory text to the effect that `The use of minst is optional.` To leave minst at any time, press <CTRL-Q>". Provided **quit** as early escape alternative after banner.
- Removed node recataloging from **minst**.
- Accepts **/sau***N* as well as **sau***N* (**N = 2-10**) as a response to the list of saus question.
- No longer allows user-specified options to **distaa** or **install++.**
- Doesn't quit the **minst** program when the wrong media volume is in the drive.
- Doesn't use display pads when it's inappropriate to do so (for example, on a serial I/O line).

2.6 New Query for Domain/Ada

The optional product, Domain/Ada, includes new files in the base operating system. If you intend to run Domain/Ada with SR10.2, answer appropriately when the install asks:

```
QUESTION: Do you want a local copy of the /usr/apollo/ada,
                   a link to another node or neither?
```

These files provide interfaces to the base operating system for programs written in Ada. In the past they were shipped with the Domain/Ada product, but they are now shipped with the products that they provide interfaces to. They continue to be installed in **apollolib**, an Ada program library under **/usr/apollo/ada**.

2.6.1 Installing Domain/Ada V3.0 after SR10.2

If you are going to install Domain/Ada V3.0 after SR10.2, you must prevent the Domain/Ada install from overwriting the Ada base operating system interfaces. After installing SR10.2, but before installing Domain/Ada v3.0, rename **apollolib** to **apollolib.sr10.2** so that it won't be overwritten. To rename **apollolib**, either first change the permission of its parent directory, for example **chmod 777 /usr/apollo/ada/isp_m68k**, or else become root.

After installing Domain/Ada V3.0, change **apollolib**'s protections by following the procedure that follows. Then **cd** to **apollolib.sr10.2** and move the saved files to **apollolib**, using **mv -f * ../apollolib**. Remove the (now empty) **apollolib.sr10.2** directory.

This procedure will not be necessary for versions of Domain/Ada released after V3.0.

2.6.2 Compiling the Installed Files

Unlike the files that were shipped with Domain/Ada, these have not been compiled in advance. It is your responsibility to compile them.

If the Domain/Ada version is not newer than V3.0, you must first change the protections of **apollolib**. In the **apollolib** directory, as root if necessary, use **chacl -B . .imports .lines .nets .objects**.

Once the protections are correct, in the **apollolib** directory compile everything with **a.make -v -f *.a**.

2.7 Sendmail Configuration Files

The installation process removes any **/usr/lib/sendmail.fc** (frozen configuration files) that exist; they are not compatible with the SR10.2 version of sendmail. If you wish to use frozen configuration files, refreeze them after the install has been completed (using the **/usr/lib/sendmail -bz** command).

2.8 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.

2.9 Media Types

We distribute SR10.2 on streaming cartridge tapes or magtapes. The media types are as follows:

For cartridge tape distributions:

CRTG_STD_SFW_1
CRTG_STD_SFW_2
CRTG_STD_SFW_3
CRTG_STD_SFW_BOOT_1

For magtape distributions:

MT_STD_SFW_1
MT_STD_SFW_2
MT_STD_SFW_3
MT_STD_SFW_4
MT_STD_SFW_5

The following boot floppies ship with the magtape distrubution:

FLP5_BASIC_1 - 4
FLP5_TOOLS_1 - 5
FLP5_LIBRARY_1 - 6
FLP5_PREP7
FLP5_BOOT7
FLP5_PREP8
FLP5_BOOT8
FLP5_REMOTE2 - 8
FLP5_UCODE4

Chapter 3: Documentation

This chapter lists the documents that are new or revised since SR10.1. It also includes any changes or corrections to documentation that we were not able to update for this release. For a complete list of technical publications that relate to Domain system products, refer to the *Apollo Documentation Quick Reference* (002685). Note that release documents for optional products can be found online in the **/install/doc/apollo** directory.

Also refer to the online help file that lists manuals available at SR10.2. Aegis users type **help manuals**. BSD and SYS5 users type **man manuals**.

A number of new documents were released at SR10 and SR10.1. Chapter 6 of *System Software Changes for SR10 and SR10.1* (also online in the **/install/doc/apollo** directory) lists the manuals that were new or revised at SR10 or SR10.1. New or revised manuals since SR10.1 are described in this chapter.

3.1 New or Revised Documents that Ship with Base Software

The following manuals include information about new functionality in SR10.2. These release notes and the following books ship with the installation media.

Installing Software with Apollo's Release and Installation Tools (008860-A02)

Making the Transition to SR10 Operating Systems Releases (011435-A02)

Configuring and Managing TCP/IP (008543-A02)

Managing the Audit Subsystem (016339-A01)

Using the X Window System on Apollo Workstations (015213-A02)

3.2 New or Revised Documents at SR10.2

This section lists new and revised documents that we are introducing at SR10.2. These manuals do not ship with the base software, but will be available at SR10.2.

Managing NCS Software (011895-A01)
(formerly *Managing the NCS Location Broker*)

Network Computing Architecture (010201-A01)
(formerly *Network Computing Architecture Protocol Specifications*)

Using Domain Diagnostics Volume 1 (009329-A01)

Using Domain Diagnostics Volume 2 (011775-A01)

BSD Command Reference (005800-A01)

SysV Command Reference (005798-A01)

Xlib Programming Manual, Vol. 1 (011241-A00)

Xlib Reference Manual, Vol. 2 (013418-A00)

X Window System User's Guide, Vol. 3 (015534-A00)

Domain Series 2500 Installation Instructions (015462-A00)

Domain Series 2500 Owner's Guide (015463-A00)

Servicing the Domain Series 2500 (015461-A00)

Domain Series 2500 Technical Reference (015945-A00)

Installing the Apollo Token Ring Network Controller-AT (010616-A01)

Installing the 802.3 Network Controller-AT (010614-A00)

Installing the 802.5 Network Controller-AT (014448-A01)

Servicing Domain Monitors (009349-A01)

Installation Instructions for the Desktop Visualization System Upgrades (016608-A00)

Servicing the Domain Personal Workstations and Servers (007859-A02)

Installing Graphics Controllers in the PC AT-Compatible Bus (010409-A02)

Unpacking and Installing the Multiple Disk Expansion Module for the Series 10000 (016654-A00)

Unpacking and Installing the Disk/Multiple Disk Expansion Module for the Personal Workstations (016411-A00)

Operating the Disk/Multiple Disk Expansion Module on the Personal Workstations or Series 10000 (012947-A02)

Product Servicing Summary for the Multiple Disk Expansion Module for the Series 10000 (016655-A00)

Product Servicing Summary for the Disk/Multiple Disk Expansion Module on the Personal Workstations (012945-A02)

Update to Domain Hardware Site Planning Specifications

Domain Floating-Point Guide (015853-A00)

Domain/OS Programming Environment Reference (011010-A01)

The *Domain/OS Programming Environment Reference* (011010-A01) reflects a substantial reorganization to the manual. Here are the highlights of this revision:

- A new overview of Domain/OS features and environments
- An enhanced description of the Domain/OS programming environment
- Greatly expanded documentation on COFF, installed libraries, and Domain calling conventions

In addition, this manual still contains the documentation for the Aegis binder (**bind**) and Aegis librarian (**lbr**).

The BSD and SysV tools and supplemental documents that had been part of the SR10 version of the manual (011010-A00) are documented in four new manuals for SR10.2:

Domain/OS SysV User's Guide 017269-A00
Domain/OS SysV Programmer's Guide 017270-A00
Domain/OS BSD UNIX User's Manual 017271-A00
Domain/OS BSD UNIX Programmer's Manual 017272-A00

3.3 Accessing Help Files

At SR10.2, you can access online documentation about programming calls for any environment via the **help** or **man** commands. Prior to SR10, only 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, 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 BSD or SYS5 shell, 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 BSD or SYS5 shells.

We also provide introductory section pages for each set of calls, accessed by specifying the prefix:

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

We changed the default definition of the HELP key in BSD and SYS5 environments so that the key will read **man** pages rather than help files.

3.4 Knowledge Broker

A new online documentation retrieval system called the Knowledge Broker is available with SR10.2. Knowledge Broker will replace Apollo's Domain/Delphi product and increase Delphi's capabilities. Its introductory documentation package contains SR10.0 versions of the following manuals:

Retrieving Information with the Knowledge Broker (011270-A01)

Administering the Knowledge Broker (014952-A00)

Publishing with the Knowledge Broker (014951-A00)

Getting Started with Domain/OS (002348-A00)

Display Manager Command Reference (011418-A01)

Aegis Command Reference (002547-A01)

Domain/OS Call Reference (Vols. 1-2) (007196-A00)

BSD Command Reference (005800-A00)

BSD Programmer's Reference (005801-A00)

SysV Command Reference (005798-A00)

SysV Programmer's Reference (005799-A00)

3.5 Domain/X11 Documentation

Domain/X11 documentation consists of user documentation, programming documentation, and MIT documentation. Manuals and help files are as follows:

User Documentation

Using the X Window System on Apollo Workstations (015213-A02)

Online help files, especially **X, Xapollo, Xserver, xterm, and uwm**.

Major MIT Documents

X Window System Release Notes & Installation Guide

Xlib C Language X Interface, March 1 1988

Inter-Client Communication Conventions Manual

A Simple X11 Client Program (Hello World)

X Window System Protocol (Scheifler)

X Toolkit Widgets--C Language X Interface

X Toolkit Intrinsics--C Language X Interface

How to Write a Widget (Asente & McCormack, Usenix 88)

From O'Reilly Associates:

X Protocol Reference Manual, Vol 0 (017140)

X Toolkit Programming Manual, Vol. 4 (017131)

X Toolkit Intrinsics Reference Manual, Vol. 5 (017132)

X Window System Quick Reference (017141)

Xlib Programming Manual, Vol. 1 (011241-A00)

Xlib Reference Manual, Vol. 2 (013418-A00)

X Window System User's Guide, Vol. 3 (015534-A00)

From Prentice-Hall

Introduction to the X Window System (017133)

3.6 Correction to Domain C Language Reference Manual

The current *Domain C Language Reference* (002093-A00) states incorrectly on page 4-99 that the #elif directive is supported by the Domain/OS preprocessor (cpp).

Instead the manual should state the following:

There are three preprocessors; one is for Aegis, one is for BSD4.3 and one is for SysV. The Aegis and BSD4.3 preprocessors do not support the #elif directive, but the SysV preprocessor does support the #elif directive.

3.7 Correction to Aegis Command Reference Manual

The Aegis **wd** command no longer takes a hyphen (-) as an argument to indicate that the command should take the pathname from standard input. Instead you should use an asterisk (*) without a filename. The description in the *Aegis Command Reference* (002547) still documents the hyphen.

3.8 Changes to TCP/IP Documentation

The following sections list changes to TCP/IP documentation at SR10.2. Note that *Configuring and Managing TCP/IP* (008543) has been revised to include new information. See Chapter 1 for new TCP/IP functionality added at this release.

3.8.1 Changes to Using TCP/IP Network Applications

Page 5–35. Add the following section:

Subsection 5.8.9 Retaining Domain/OS Objects with filetype

filetype filetype_name [local | remote | both]

The filetype command allows you to set the object type of Domain/OS or user–defined objects before transferring them with **get** or **put** operations.

Without any argument, the **filetype** command reports the current settings for the filetype used for transfers. Use this command with one of the arguments to change the file type after you've connected to a host and before you issue any **get** or **put** commands. Specify the filetype you want to set, which can be any one of the Domain/OS supported or user–defined object types such as **coff, unstruct, record**, or **default**. Then specify whether the file type applies to **get, put**, or **both** operations. Specify **local** to set the file type for **get** operations when retrieving remote files and storing them on the *local* system. Specify **remote** to set the file type for put operations when sending local files to *remote* systems. Specify **both** to set the file type for both **get** and **put** operations.

This command is useful for storing typed objects on non–Domain/OS systems that might not support typed objects. For example, you might want to store your COFF object files on a non–Domain/OS disk and, when you retrieve them, you set the filetype to **coff**.

The **filetype** command remains in effect until you issue another filetype command, so you can perform multiple transfer commands for objects of the same type.

If you do not use the **filetype** command to set the file type, the files will be assigned the default type, **unstruct**.

When you set the filetype on a remote system, the system can support this feature only if it is an Apollo system running SR10.2 (or later) software. If you specify the remote option and the remote system does *not* support this feature, you get a message and the remote filetype remains unchanged.

If you do not specify an option when specifying the **filetype** command, the default value is **both**. If **both** is set, and the remote system does not support this feature, the local filetype is set to the specified filetype and the remote filetype remains unchanged.

The following is a sample **ftp** session using the **filetype** command.

```
/* Set filetype to coff for both remote and local files. */

ftp> filetype coff both
Local target filetype set to coff.
        200 Filetype set to coff.
        Remote target filetype set to coff.
```

3.8.2 Changes to Man Pages and Help Files

The following *online* TCP/IP man pages and help files have been updated or added:

arp, dtcb, ftp, ftpd, gethostbyname, hostname, hostns, ifconfig, intro(4n), mailaddr, named, netstat, nmconfig, resolver(3), resolver(5), routed, tcpd, trpt

3.9 Changes to Managing System Software Books

We've uncovered the following errors in the *Managing System Software* books. The primary citation in each case refers to the Aegis version of the book. If the error also occurs in one or both of the UNIX versions, a reference in parentheses after the page or section number specifies the page or section in the UNIX books as (BSD;SysV). If the citation applies only to one of the UNIX versions, the semicolon (;) will appear before or after the reference, as appropriate.

(BSD;SysV)
(;SysV)
(BSD;)

On page 2-21 (2-19;2-21), in Task 3, the manual says to add **cps /sys/ns/ns_helper** to the startup files in **/sys/node_data**. Instead, uncomment the following lines in the **/etc/rc.user** file:

```
# if [ -f /sys/ns/ns_helper ]; then
# (echo " ns_helper\c" >/dev/console)
# /sys/ns/ns_helper &
# fi
```

The **SysV** version also speaks of the "**/etc/rc** user script," which should read "**/etc/rc.user**."

On page 2-39, correct a typographical error in the fifth paragraph of Section 2.7.2. Replace **rc.use** with **rc.user**. Also on page 2-39, the lines supplied in the **/etc/rc.user** file which start the print server are as follows:

```
# if [ -f /sys/hardcopy/prsvr ]; then
# (echo " prsvr\c" >/dev/console)
# /sys/hardcopy/prsvr <config file name> &
# fi
```

On page 2-41, correct a typographical error in the first text paragraph. Replace "via the **cp** command" with "via the **cps** command."

On page 2-42 (2-33;2-35), the lines to uncomment in the **/etc/rc** file are:

```
# if [ -f /sys/spm/spm ]; then
# (echo " server_process_manager\c" >/dev/console)
# /sys/spm/spm &
# fi
```

In the **SysV** version, the lines appear correctly, but delete the preceding lines, which read:

> To invoke it from the **/etc/rc** file for DM startup files,
> uncomment the following line in the file:

```
# cps /sys/spm/spm -n server_processor_manager
```

On page 3-7, in Section 3.3.2, correct a typographical error in the last paragraph; substitute "temporary" for "tempory".

On page 3-10 (3-14;3-14), in the description of "Starting Servers on a Remote Node", substitute "takes **-cp**, **-cpo**, and **-cps** switches" for "**cp**, **cpo** and **cps** local process commands."

Also on page 3-10 (3-14;3-14), the example should read:

crp -on //trout -me -cps /etc/ncs/glbd

On page 3-11 (3-15;3-15), in Table 3-1, under the column **SID of Process** in the row **Shell commands**, correct a typographical error; substitute **user.server.none** for **user_server.none**.

At the top of page 3-12, the example is incorrect. To start the print server at a shell prompt, type the following:

$ **/sys/hardcopy/prsvr** *config filename>*

On page 3-12 (3-16;3-16), in the third paragraph under Section 3.5.8 (3.5.7;3.5.7), further information is necessary. Most servers do not requirerRead and write rights to the **‘node_data** directory, because most servers now place their files in either **‘node_data/tmp**, **‘node_data/systmp**, or **‘node_data/system_logs**. Using the **crp** command from an SR9.7 node requires write access to **‘node_data,** since mailbox files are created there.

On page 3-27 (3-32;3-32), under Task 4, replace subtask 1 with the following:

Uncomment the following lines in **/etc/rc.user**:

```
# if [ -f /sys/net/netman ]; then
# (echo " netman\c" >/dev/console)
# /sys/net/netman &
# fi
```

The **netman** server will start automatically whenever the partner boots.

The lines in the SysV version are essentially correct, except that it should read "the partner's **/etc/rc.user** file" rather than "the partner's /sys **/etc/rc.user** file".

Correct a typographical error on page 3-31 in the first line of Section 3.1.2; substitute "monitoring" for "moniotoring".

Correct the log-in log file name in Section 3-12.1. The correct name is **'node_data/etc/login_log.conf**.

On page 3-32, correct a typographical error in the third paragraph; substitute "create" for "creat".

On page 3-33, add the information that, since the log file resides in the **'node_data/etc** directory, all comments about protecting the log file apply to that directory, too. Remove the parenthetical sentence about **k** rights.

On page 3-39 (3-42;3-43), to start **netman** from the **/etc/rc.user** file, uncomment the following lines:

```
# if [ -f /sys/net/netman ]; then
# (echo " netman\c" >/dev/console)
# /sys/net/netman &
# fi
```

The **netman** server starts automatically whenever the partner boots.

On page 3-40 (3-42;3-43), under "Starting the Tablet Server," replace the discussion of starting from the **/etc/rc** file with the following:

To start the Tablet Server, uncomment the following lines in the **/etc/rc.user** file:

```
# if [ -f /sys/dm/spb1 ]; then
# (echo " bitpad\c" >/dev/console)
# /sys/dm/sbp1 /dev/sio2 L &
# fi
```

On pages 3-42 and 3-43, all occurrences of **/dev/Hy0x** should be **/dev/tty0x**.

On pages 3-43, 3-46, 3-47, all occurrences of **'node_data/siologin_log** should be **'node_data/login_log**.

On page 3-46 (3-49;3-49) **'node_data/siomonit_log** should be **'node_data/system_logs/siomonit_log**, and **'node_data/siologin_log** should be **'node_data/login_log**.

Correct a word transposition on page 6-67 (9-120;9-134) in the **rexecd** manual page in the "Password incorrect" diagnostic message from the top of the page. The phrase "wrong was password" should be "wrong password was.*(rq.

The information on the **edrgy** tool in the two UNIX versions of the Managing System Software books is not complete. The last paragraph of Section 4.2.6 (page 4-8) in each manual discusses UNIX restrictions. There is an additional restriction which is not found there.

If UNIX restrictions are enforced, the org field of an account must be either **none** or **%**. You may not enforce UNIX restrictions and have users belonging to any other organization.

3.9.1 Corrections to SR10.1 Release Notes

- In Section 2.2.2 of the *SR10.1 Domain System Software Release Notes* we incorrectly stated:

  All SR10.1 created tapes can be restored on SR9.7 and SR9.7.1 systems.

  In order for a cartridge tape to be restored on SR9.7 or SR9.7.1 systems, you must have created the tape with the **pre_sr10** switch. If you did not use the pre_sr10 switch, restore the cartridge on an SR10.*x* node, and rewrite with **wbak_sr9.7.**

- In Section 8.2.19 of the SR10.1 Domain System Software Release Notes, we incorrectly identify a new macro as **'fault_no**. The correct name for this macro is **'fault_status.**

  When Domain/DDE catches a target program signal, it takes the following actions:

  sets 'signo to the UNIX signal number for the fault, or
  sets 'fault_status to the Aegis status code for the fault.

  invokes the alias 'after_fault.

  You can modify the definition of 'after_fault to take actions appropriate to the kind of fault. For example, the following definition instructs Domain/DDE to display the message

alias 'after_fault if 'signo==2 -then [[sh echo 'interrupt - continuing']; go -ignore

- Also in Section 8.2.19 of the SR10.1 Domain System Sofware Release Notes, we incorrectly state that the **-delete ignores** command instructs Domain/DDE to deliver the specified signal to the target program whenever it occurs, without stopping execution of the program and without notifying the user.

  In fact, the **-delete ignores** command instructs Domain/DDE to return to its default behavior for the specified type of signal, that is, to stop program execution and notify the user whenever the signal occurs.

3.9.2 Change to Managing Domain/OS and Domain Routing in an Internet

Please change the following definition in the manual *Managing Domain/OS and Domain Routing in an Internet*(005694), Appendix A, page A-2:

Over-run
The controller received a frame larger than the size
allowed by the IEEE 802.3 protocol.

Change to:

Overrun
This controller had no memory bus bandwidth available to hold an
arriving frame. Overrun errors usually indicate that the node
is heavily used.

Chapter 4: Bugs, Limitations, and APRs

This chapter describes bugs and Apollo Problem Reports (APRs) that exist or have been fixed in SR10.2.

For information regarding bugs and APRs at SR10 and SR10.1, see the online document *System Software Changes at SR10 and SR10.1* located in the file **/install/doc/apollo/os.v.10.1__changes.**

NOTE: Bugs and fixed bugs related to the media install tool (**minst**) are documented in Chapter 2 of these notes.

4.1 Bugs/Limitations Existing in SR10.2

The following bugs and limitations are reported in SR10.2 software:

4.1.1 Bugs in Domain/OS

On some configurations, if you boot diskless the `bytes loaded` message is not erased until the DM is initialized.

When you enter the boot PROM after a shut or a crash, the first command entered at the prompt is lost. Hit return more than once to correct the problem.

Any program which forks and has both the parent and children doing I/O concurrently, does not work over mbx channels. Thus, programs such as **bc, ksh, tip, cu, and script** do not work over **crp**.

We have experienced a problem with the **tar** command. You must execute either **mt -f /dev/r??? -rewind** or **rbak -rewind** before using the tar command.

The maximum number of nodes that you can boot diskless from a single mother node is 11.

4.1.2 Problems with Earlier Versions of SR10.2

If you have a prerelease version of SR10.2 (beta customers who may update), please note that the install will not overwrite the contents of the **/install/doc/apollo** directory. The result is that you will still have older versions of the documents in that directory. The workaround is to copy the latest version of the **/install/doc/apollo** directory from the <authorized area>, or delete the existing files before you update. Please see Chapter 2 of these notes for related issues regarding prerelease versions of SR10.2.

4.1.3 Install Bug

Install++ can get unnecessary warnings about links versus directory conflicts for some layered products.

4.1.4 Important Note About C and SR10.2

Do not install Domain/C version 6.6 or earlier after installing SR10.2; doing so can cause problems. If you must install Domain/C version 6.6 or earlier after installing SR10.2, be sure to reinstall the **/lib/syslib** library from the SR10.2 authorized area. (The pathname will be **//AA/install/ri.apollo.os.v.10.2/lib/syslib**, where AA is the pathname of your authorized area. It is okay to install Domain/C version 6.7 after installing SR10.2.

4.1.5 Series 2500 SIO Problem

You may see garbled characters when receiving multicharacter data on any of the 3 SIO lines on a Series 2500 system.

The UARTs used on the Series 2500 have a lower tolerance to baud rate errors. The problem is seen most often on modems where the baud rate error can be greater than 1%. For example, an incoming baud rate of 1220 will fail if the baud rate on the Series 2500 is set to 1200. The problem is baud rate independent.

To correct the problem, add two stop bits to the incoming serial data. The Series 2500 can tolerate baud rate errors greater than 5% with two stop bits. The two stop bits are *not* necessary for transmitting from the Series 2500.

4.1.6 Problem with Socket Support

The operating system maintains an internal pool of buffers used in the transfer of messages to and from sockets. This pool is limited in size. As a result, if programs using sockets (via **msg_$, ipc_$, dds/ncs)** fail to retrieve messages from their sockets, it is possible for the operating system pool to fill. In this case, the operating system may crash as it is unable to obtain a buffer. To avoid this possible crash, programs should retrieve messages from sockets as quickly as is practical. Obviously this problem becomes more critical as the number of programs using sockets increases.

This problem is not new at SR10.2. However, since the number of available user sockets has increased, the likelihood of a crash resulting from a shortage of buffer space in an environment of intensive socket usage may have increased.

4.1.7 Language and Tools Bugs

The **-s** option to **ld** strips the output file of unnecessary symbols. If you use this option when creating a **pic** (position-independent code) output file, **ld** fails to set the F_STRIPPED flag in the COFF file header.

If you redirect the output of **lbr**, it creates (and redirects) only a portion of the library file.

4.1.8 Ethernet Problem

/etc/ifconfig will not behave correctly on DN5*xx*-T machines with ETHERNET. **/etc/ifconfig eth0** *aa:aa:aa:aa:aa:aa* followed by **/etc/ifconfig eth0** will show 08:00:1e:NO:DE:ID, rather than the *aa:aa:aa:aa:aa:aa* address.

4.1.9 Bugs in the Display Manager

The following bugs are known to exist in the display manager:

If you use CPB to pop windows out of **invis_group**, thereafter TN finds only some of the windows popped.

With X owning root and **uwm** as the window manager, a call to **pad_$make_icon** makes a DM icon instead of a **uwm** icon.

When you log in a window as a user (with an empty shell field in the registry), you get a **/bin/sh** shell even though the **/etc/environ** file and the parent shell both specify Aegis.

Text in partially obscured alarm windows does not scroll properly.

Wrong key defaults can occur in a BSD environment on a SAU2 node.

The DM is unable to successfully allow a user to login if the calendar date is past November 1997. The workaround is to fix the node's calendar before booting the DM.

When closing windows, the cursor does not always go to a valid window.

With the new DM, a mouse up-transition will be given to the window made active by the down-transition. This may not be the same window that received the down-transition, for example if the command was **icon**.

When the DM is not the window manager, it still pops a window to do input.

Exiting **emacs** can leave the pad in raw mode. A workaround is to close the window.

Assignment of a tagged regular expression has problems if the newline character is part of the pattern.

/bsd4.3/bin/echo of a message to **/dev/display** omits the the first argument. A workaround is to put the entire string in single quotes.

4.1.10 Bugs in GPR

It is possible to add the color red to a monochrome color map on the DN3000 without getting an error returned.

GPR sometimes cannot create arcs with extremely large coordinates.

gpr_$inq_disp_characteristics changes the pad scale factor to (1,1). This change may cause the DM to place text incorrectly in frame mode. To get around this problem, either initialize GPR before calling **gpr_$inq_disp_characteristics** or restore the pad scale factor with the **pad_$set_scale** call.

You cannot access overlays in Domain/Dialogue.

At the present time, you cannot use the **gpr_$set_auto_refresh** call for overlays. Instead, you must write your own refresh procedure. This procedure must call **gpr_$clear** with the color parameter set to 0 before calling any draw routines.

4.1.11 Using gmf_$copy_plane, gmf_$copy_subplane, and gmf_$restore_plane

The following information explains the use of **gpr_$enable_direct_access** with **gmf_$** calls. The description below has been reported as a bug several times. However, this is not a bug (please read the explanation that follows):

gmf_$copy_plane (and **gmf_$copy_subplane**) has been reported as a problem with various nodes. If the created image is redisplayed via **gmf_$restore_plane**, the only result is that a portion of the window banner in the picture is printed, and it is overwritten with garbled font data. If borrow mode is used, nothing is displayed from **gmf_$restore_plane**.

gpr_$enable_direct_access should be called before any call to **gmf_$copy_plane, gmf_$copy_subplane or gmf_$restore_plane**. Since **gmf** is not layered on **gpr**, it is the responsibility of the application to make the call to **gpr**.

Polygon fill decomposition technique **gpr_$fast** traps support the filling of complex polygons, except in the case where two polygon edges are parallel and have zero pixels between them (that is, they're coincident). In this case, an integer divide by zero will occur, and the fill will fail. Use **gpr_$precise_traps**, or **gpr_$non_overlapping_tris**, to eliminate the problem.

4.1.12 Network Bugs

The following bugs exist in the networking environment:

If you turn Domain routing on and off repeatedly with the **rtsvc** command, the routing process, on rare occassions, does not die when it should. A system crash may result if this process is left running. After turning off routing, check to see if the Domain routing process is running when one or fewer networks are enabled for routing. If the routing process still exists, shut down and reboot your node.

The command **lcnet -n** *//node_inaccessible*, where *//node_inaccessible* is a node that cannot be reached, never times out. In such a case, kill the **lcnet** command. Use **bldt -n** *//node_inaccessible* to check if you can reach a node before running **lcnet**.

4.1.13 COM-ECMB Bug

For nodes on which only UNIX has been installed, that is the /com directory does not exist, a script for the COM-ECMB product will not work. This is due to the fact that the script **/sys/drivers/ecmb/build_ddf.sh** relies on **/com/sh**. The workaround for this is to invoke the script as

```
$ //<node_that_has_/com/sh>/com/sh build_ddf.sh
```

where *<node_that_has_/com/sh>* is the name of a node on which **/com/sh exists**.

Note that this invocation is performed only once at COM-ECMB product installation.

4.1.14 Incorrect Font File Problem

Previous software releases (SR10 and SR10.1) included the following definition files:

/sys/ins/fontn.ins.pas
/sys/ins/fontn.ins.c
/usr/apollo/include/fontn.h

A mismatch between the Pascal and C type declarations may cause problems when a font file written by a C application needs to be read by a Pascal application, or vice versa. The **space-type** field of the **fontn_$table_tver1** data structure differs so that C programs would read and write the low bit of the **space_type** field, but Pascal applications would read and write the high bit of the same field. The Pascal definition was incorrect, and the SR10.2 Pascal file now conforms more closely to the older C definitions.

Most font-handling applications, Pascal or C, should be able to recompile with the new definition files and immediately be able to write correct font headers. Unfortunately, font files may exist that were written with the old, incorrect Pascal font header declarations. You may need to rewrite your font-handling applications to recognize the **space_type** field in these font files.

The **/domain_examples/fix_font** directory contains the source form of a program that reads font files and (if necessary) corrects the space_type value. The **/sys/dm/fix_font** file is the executable image of that program. The command line has the form

```
fix_font <pathname> ...
```

where *<pathname>* is the pathname of a font file. You may specify more than one pathname on the command line, but you may not use wildcards if you run the program from an Aegis (**/com/sh**) shell.

Please be careful: There is no reliable way for **fix_font** to distinguish between font files and any other kind of files. If you feed anything but an Apollo font file to **fix_font**, it may change bits you didn't want changed.

Note that all fonts released by Apollo at SR10, SR10.1, and SR10.2 used the correct (low-order) values in the **space_type** field. Only fonts created by customer applications may contain erroneous **space_type** values, and then only if the application was written in Pascal. In the future, Apollo software will be modified to accept and (where reasonable) to fix font files that contain erroneous **space_type** values.

4.1.15 Limitation on Non-Flow Control Applications

Non-flow control applications are load dependent when running on SIO lines. Some of the following applications that run without flow control are:

SLIP, DPCI, UUCP, EMACS

These applications are load dependent and may affect data integrity under a heavy load. Reliability decreases with increased data rates.

4.1.16 Limitations for BSD Commands

Some commands can exhibit undefined behavior when processing 8-bit character data. Caution is suggested, as in some cases the 8-bit data will not be preserved.

The bsd4.3 archive command, **/bsd4.3/bin/ar,** does not accept the **-xo** option (-xo means to extract a file but preserve the original creation or modification date of the file).

4.1.17 Limitation for rwvol Command

At SR10 and later releases, the **rwvol** command must not be used to write disk blocks unless the INPROCESS environment variable is set to TRUE when using it online. The offline versions of **rwvol** continue to work correctly. Further, we suggest that **fixvol** be used in place of the online version of **rwvol**.

4.1.18 Mnemonic Debugger Limitation

MD commands such as LD and EX are slow when using striped disks. Once booted up, Domain/OS performance will be unaffected, however.

4.1.19 Bugs in Mail

We have noticed that using sendmail to send messages with a large number of names in the header often causes duplicate messages to be sent. We believe this behavior occurs because the header exceeds the default name limit imposed by the sendmail program (we use the Berkeley limit).

The installation process removes any **/usr/lib/sendmail.fc** (frozen configuration files) that exist as they are not compatible with the sr10.2 version of sendmail. If you wish to use frozen configuration files, refreeze them after the install has been completed (using the **/usr/lib/sendmail -bz** command).

4.1.20 No logo on DN560/DN660

Domain/OS doesn't display the logo during boot on DN560 and DN660 nodes.

4.1.21 GSR Limitation

Fill tiles are restricted to valid plane oriented main memory devices and to valid hidden display memory (HDM) devices. Pixel oriented main memory devices and display memory devices, such as windows or borrowed display, are not supported.

4.1.22 RGY Server Limitation

If any of the registry servers are running the SR10.2 rgy server, all other servers (masters and replicas) must also be at SR10.2. You can copy the SR10.2 registry server (**rgyd**) onto earlier systems without updating the operating system; the SR10.2 server can run on nodes running SR10 and SR10.1.

4.1.23 Bugs in Domain/DDE

Argument information is sometimes unavailable in FORTRAN.

Domain/DDE cannot find correct addresses for variables in registers in code ranges that have been removed in optimization.

The **-nwp** option prevents Domain/DDE from creating a separate window for the target program. If you use **-nwp** when debugging a program that requires user input, Domain/DDE fails with a SIGTTIN fault.

If Domain/DDE encounters data within a routine's instruction stream (a case statement's branch table, for example, or operands to syslib calls), the disassembly display goes blank.

Domain/DDE can't print the contents of virtual addresses from F8000000 to FFFFFFFF. A request such as:

```
dde> print ^integer32(16#fc070000)^
```

results in the following error message:

```
^: ?(dde) No read access to virtual address fc070000
```

4.1.24 Unsupported Open Dialogue Example

The example in the file **/bsd4.3/usr/dlg/examples/unsupported/geo** does not work on any SR10.2 machines. The example fails because it uses its own fonts; these fonts were created under X11 R2, and SR10.2 is running X11 R3 fonts. Three of these fonts can't be loaded by X: **scale_font, scale_4_font**, and **pattern_font**.

4.1.25 TCP/IP Bugs and Limitations

When using **ftp** or **telnet**, you cannot specify a host name of X or x because of a bug in **inet_addr**() that interprets x or X as localhost. Note that you can always add an alias for a host name to the **/etc/hosts** file.

Our implementation of **ftpd** during ascii transfers looks at each character that it receives from the network. If it receives a CR then a NL from the network, the NL goes into the output buffer, ignoring the CR. If it receives a CR then anything else, it ignores the other character and puts the CR into the output buffer. If you have a file that contains imbedded CR in the data stream, the character that follows the CR will be lost.

Because Domain/OS does not support the **chroot** command, you cannot configure an anonymous **ftp** account.

The **ftp** utility does not support the following **ftp** options: structure page, mode block or mode compressed.

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.

/etc/mkhosts normally takes a command line argument that is the pathname of the input file to use. The command generates output files with the pathname plus the extensions **creates the** *pathname*.**dir** and *pathname*.**pag** files, but continues to use **/etc/hosts.dir** and **/etc.hosts.pag**, or, if unavailable, **/etc/hosts**. To avoid this, do not try to rename the **/etc/hosts.pag** or **/etc/hosts.dir** files.

The **rwhod** and **ruptime** commands do not show idle time.

If a service that **inetd** listens for is a single–threaded service (i.e., **inetd** waits on that socket when it is in use, that is, **talkd**) as opposed to a multithreaded service (**inetd** forks a daemon that connects on that socket, (for example, **tftpd**) and that service is in use when **inetd** receives a SIGHUP signal (`kill -HUP inetd`), the single–threaded service does not get reconfigured.

When using the **ftp put** command on files with **sunique** set, ftp *reports* the wrong remote filename; however, it creates the remote filename with the correct name.

rlogind does not correctly handle out–of–band data used for stop/no-stop negotiations used in flow control.

The following files should have the following unix modes and owners:

```
-rwsr-xr-x  1 root   18039 May 23 17:18 /etc/ping
-rwsr-xr-x  1 root   13358 May 18 09:26 /sys5.3/usr/bin/rcp
-rwsr-xr-x  1 root   5487  May 18 09:26 /sys5.3/usr/bin/remsh
-rwsr-xr-x  1 root   12313 May 18 09:26 /sys5.3/usr/bin/rlogin
-rwsr-xr-x  1 root   13358 May 18 09:26 /bsd4.3/bin/rcp
-rwsr-xr-x  1 root   5487  May 18 09:26 /bsd4.3/usr/ucb/rsh
-rwsr-xr-x  1 root   12313 May 18 09:26 /bsd4.3/usr/ucb/rlogin
```

4.1.26 Bugs in Aegis Printing Services

When installing the SR10.2 Aegis print services (/sys/**hardcopy**), you must update the print manager node before updating the print server nodes. Failure to do this will prevent an SR10.2 print server from restarting. If you cannot update the print manager node, copy the /sys/**hardcopy** tree to the print manager node and restart the print manager. The SR10.2 /sys/**hardcopy** tree will run on an SR10.1 node.

If a text file containing the following text is sent to an imagen printer,

```
1
^L
2
```

the 2 overwrites the 1 on the first page. The 2 comes out okay at the top of the second page.

The footer string **foot -#-!** should give the page number and file name left justified. Instead, it outputs the page number followed by an @.

prflib quits with a communications failure if a print manager is not active on the network.

When using the tektronix driver **/sys/hardcopy/drivers/tektrnx** with a multibus Ikon board number 10085 and Domain/vers 1.0, you must load patch m0032_8905. Failure to do this will result in the print server quitting with an undefined global error. The problem will be fixed in the next release of Domain/vers.

4.1.27 Bug in UUCP

The **uucp** commands create **tty** lock files with an unreleased (and therefore, undocumented) format. This means that user programs cannot use these files for checking locks.

4.1.28 Bug in the Knowledge Broker

The Knowledge Broker installation shell scripts found in

/sys/kb/install/install_kb.sr10.sh
/sys/kb/install/install_kb.sr9.7.sh

contain a test which determines whether or not to replace the following global library files (found on the target node) with the versions of these files found in the Knowledge Broker authorized area.

/lib/dmllib
/lib/dialoglib
/lib/ddslib

This test returns incorrect results in SR10.2, causing the libraries in the authorized area (which are pre-SR10.2) always to overwrite the newer libraries on the target node. Please refer to the hardcopy document *Read This First* included with the Knowledge Broker media for a description of the workaround for this error.

4.1.29 Domain/CommonLISP Limitation

The current version of Domain/CommonLISP (version 3.0) will not run on SR10.2. When available, Domain/CommonLISP Version 4.0 will.

4.1.30 Limitation in /com/ulkob Command

It should be noted that **/com/ulkob** does not enable you to unlock an object if you are not running in process mode, or if the file is not remote.

4.1.31 etc/mkdsk Command is not Documented

Man pages refer to **mkdisk**, which is an older version of this command.

4.1.32 Limitation in /etc/mkdev Command

You should not specify the **'node_data** pathname with **/etc/mkdsk** in the case of a diskless node. For diskless nodes, refer to **//partner_node/sys/node_data.node_id**.

4.1.33 Restriction for /etc/mkcon Command

/etc/mkcon should not be used before the display manager takes over the screen.

4.1.34 Domain/X11 Known Bugs and Limitations

The following subsections describe Domain/X11 bugs, limitations, and special notes.

4.1.34.1 Domain/X11 Limitations

You cannot cut and paste between X and DM windows. X and DM windows do not share X window properties.

Parent-relative window backgrounds aren't drawn in children of DM-owned roots or children of disowned windows. The background is drawn in the parent, but X doesn't draw in windows it doesn't own.

If **twm** references any keys that have been (1) excluded from X (using the **-K keyboard.config** file), or (2) not assigned to a keycode that our keyboards send, the keyboard locks up and **twm** must be killed. Resetting the server also unlocks the keyboard. This happens, for instance, with the HOME key, whose keysym is defined in **keysymdef.h** as 0xFF50, but no Apollo keycode is currently assigned to 0xFF50, as can be seen in **/usr/lib/X11/keyboard/kb3.xprkbd**.

Do not reference any key in your **.twmrc** which is excluded (using the **-K keyboard.config file**) or not defined on the Apollo keyboard (for example, HOME).

Users with a keyboard 3 running in compatibility mode should use the keyboard 2 configuration file (for example, **/X11/lib/keyboard/xmodmap.kb2sample**), because X thinks a keyboard 3 in compatibility mode is actually a keyboard 2, and the necessary X events won't be sent.

We do not support the **xmodmap** pointer that reassigns mouse buttons. As a result, the mouse buttons cannot be remapped for left-handed users, for instance.

Multiple colormaps are not fully supported.

Since the use of a node's colormap is shared between the DM and the X server, use of it by X clients has limitations. The following chart shows colormap slot accessibility for clients of shared X with two different DM modes, as well as clients of borrow mode X.

	read/write	**read only (used by DM)**	**no access (used by X server)**
share X/color	N - 11	9	2
share X/monochrome	N - 4	2	2
borrow mode X	N - 2	0	2

In this chart, N represents the number of colormap slots on a node. Monochrome mode is a mode which can be set with the DM command MONO -ON so that the DM will only use 2 slots, and these slots will have the colors black and white. Colormap slots which are inaccessible to X clients always occupy the two highest numbered slots. These are used by the X server for its cursor.

As can be seen, use of shared X in color mode on a 4-plane node provides only 5 colormap slots with read and write access. Two alternatives providing more colormap flexiblity are to use shared X with the DM set to monochrome mode, or to use the X server in borrow mode. Using shared X in monochrome mode provides 12 colormap slots with read and write access. Using the borrow mode X server provides 14 colormap slots with read and write access.

The X server uses the last two slots of the color map for cursor colors. The DM no longer uses these last two slots, because it now uses just three pairs of colors on a 4-plane system. If you do issue the DM command MONO -ON, your X clients can allocate more colors.

The DM was designed to pop (raise) all DM pads before allowing the user to perform any operations. This means that all DM pads are unobscured before a user interacts with them. The X Window System was designed to allow the user to perform operations in obscured windows. If X owns the root and an X window manager is being used, it is possible to damage DM pads and X windows or both. The damage occurs because the DM pads are not being popped (raised) by the X window manager before some operations are performed. These operations are input, edit, and the following DM commands: PN, EE, TT, TB, UNDO, TDM, XP, XD. The damage does not cause data corruption; it can be repaired by refreshing the screen or window. Some X window mangers are designed to pop the DM pads and others are not. **uwm** and **mwm** are notified with a Configure Notify Event and will eventually pop the DM pad. When **mwm** or **uwm** pops the DM pad, the damage is repaired. **twm** is not notified and it will not pop the DM pad. If **twm** is used, damage to obscured DM pads is displayed until the screen or window is refreshed.

Since we do not support the MIT X Clients in **/usr/X11/bin**, we only expect them to work in the BSD environment. Using them in any other environment except BSD will lead to unexpected results.

Using **xterm** and **uwm** in environments other than BSD will work as expected.

The function call to XApolloOwnRectangle can only reacquire ownership of a rectangle for X; it is not possible for X to acquire ownership of a window unless it was created by or previously owned by X.

4.1.34.2 Domain/X11 Notes

This subsection provides various notes that don't fit into the category of bugs or limitations.

xterms started from Bourne shells (**/bin/sh**) ignore SIGINT, SIGQUIT and SIGTERM. This means that an **xterm** started in the foreground will not exit when a SIGINT (for example, ^C, ^Q) is entered in the shell from which the **xterm** was started. The easiest way to kill an **xterm** that has been started from a Bourne shell is to exit the shell by sending an EOF, for example a ^D.

If the X server is invoked directly or from a script after logging in, X won't stop when the DM tries to log out. The two supported ways of starting **Xapollo** are via **/etc/rc** and via **xinit,** which is used by the script **/usr/x11/bin/startx**. If you have difficulty with logging out, first kill the X server before issuing the DM command to log out, or issue the command:

```
kill -HUP xinit
```

The SR10.1 include files **/sys/ins/xgpr.*** have now been subsumed by **/sys/ins/gpr.*.** They are no longer available, and your #includes for them should be deleted. Use **/sys/ins/gpr.*** instead.

Clients that want to use R2 font paths should include **/usr/lib/X11/fonts/oldx11/** in the default font path, (for example, via **xset**'s **fp+** switch):

```
xset fp+ /usr/lib/X11/fonts/oldx11/
```

If you get the message `XIO: Operation would block` from an X client program you're developing, it can mean that you've made the mistake of defining the variable **errno** somewhere in one of your C modules. Wherever you declare **errno,** you should use the following declaration:

```
extern int errno;
```

If you declare **errno** without declaring it **exterm**, your program may refer to a different location than the various shared libraries. One consequence of this is that Xlib's internal code can pick up incorrect data.

4.1.34.3 Domain/X11 Bugs

This subsection lists the various bugs that have been reported:

When running a reparenting X window manager, any attempt to move the cursor from inside a DM pad into the screen background using an arrow key can cause the DM to lose the keyboard focus while the cursor still appears inside the DM pad. When the keyboard focus gets lost, attempts to type characters result in one of the following DM messages:

```
(EN) Cursor is not in any window
   No character under cursor
```

To recover, use the mouse to move the cursor a small distance.

When using a reparenting window manager and creating a new DM pad, sometimes cursor droppings appear in the upper left corner of the newly created pad.

When the **mwm** Motif window manager is started from a DM pad, error messages can cause a deadlock involving the DM and the X server. This problem can occur in any client which performs an X Grab operation, then outputs information to a DM pad via **stdout** or **stderr** during the grab. Remember that an implicit passive grab is activated whenever you hold down a mouse button: avoid logging MotionNotify events to a DM pad. Workaround: start **mwm** and other such clients from an **xterm**.

Reparenting window managers never see button events over DM rectangles.

X clients sometimes cannot establish a connection to the server when they're stamped sys5.3 and run from a sys5.3 environment.

There is a problem displaying rubberbanding with **uwm**. Workaround: issue an **xownroot -off** and an **xownroot -on**.

The DM loses track of the cursor and reports the error message `(EN) cursor is not in any window` when the user tries to type after moving the cursor via the arrow keys in such a way that the cursor would leave a DM pad. For example, if you use arrow keys to move from a DM transcript pad to its corresponding input pad, and you hit the down-arrow key once too often (going too far and ending up in X territory), the problem occurs. The DM block cursor looks like it's in the DM window, but the window won't accept typing until you move the mouse slightly, or use the arrow keys to move in reverse, back into the window.

If you press <ESC> and <RETURN> in an **xterm** window, the **xterm** will not accept any more keyboard input. To work around this, press <CTRL> and the middle mouse button (<M2>) to produce **xterm**'s Modes Menu. Select "Soft Reset."

4.2 Bugs Fixed in SR10.2

The following bugs were previously reported in release notes, but have since been fixed.

4.2.1 Display Manager

The following DM bugs are fixed:

- You could not boot the DM over an SIO line.
- The DM converted a tab character into eight times the width of a space character; as a result, the spacing of text with tabs was incorrect.
- If you opened a file just as another was closing it, you could get a lock on the **.bak** file rather than the file itself, even though your header said you had the file itself. Then someone else could come along and open the actual file. In this case, you could

get a `name not found` error message when you tried to close the file.

- You could not undo a paste operation when the paste buffer was longer than 64 lines for an input pad, or 512 lines for an edit pad. The undo buffer overflowed, you got a `nothing to undo` error message, and you could not undo previous commands.
- The DM did not properly handle command input lines that ended with an equal sign ("="). You could use such lines if you enclosed the argument in single quotes.
- The SHIFT/READ and M3 commands did not work when the pathname was at the beginning of the first line of a file.
- **Auto_refresh** problems on DN590 nodes occured while an application was in true color mode. This has been corrected in both the DM and GPRLIB.
- The **auto_refreshed** bitmap was incorrect after the window was grown.
- The **ctm.ins.c** include file was missing the declaration for the procedure **ctm_$mark_read_only.**
- When a direct graphics program exited, sometimes subsequent output to a pad (via a command like LD) would be incorrectly formatted. This was caused by a problem in the Pad Manager.
- The TI command no longer goes to read-only pads.
- When you login to the DM, the correct entry is now put in the **utmp** file.
- If you used the std.color font on a 19" monochrome monitor, you could cause the cursor to jump to a position outside the window. A fix in the DM's cursor settling code has fixed this problem.
- A call to **pad_$dm_cmd** would fail if the **person_id** of the process making the pad call was not the same as the **person_id** of the user logged into the DM.
- When a cut or paste of greater than 512 lines in an edit pad occured, the DM's undo buffer overflowed and the error message `Paste buffer is too large for undo buffer` was displayed.
- Support for the SYS5 ioctl VMIN and VTIME has been added to the pad manager. The support has been added only for RAW mode input (**-echo**).

4.2.2 BSD and SysV Commands and Utilities

The **passwd** command bypassed any policy restrictions that the system administrator had set on password contents (for example, minimum length, contents), whether for the organization or for the registry as a whole.

The SCCS utilities didn't accept commands with filenames that exceeded 32 characters.

4.2.3 BSD Commands and Utilities

- The **size** command could not deal correctly with large COFF objects; it failed with a `has too many fields` error message from **awk**.
- The BSD **lpr** command failed to start the printer daemon **lpd** if **lpd** was on a remote workstation. **lpr** would spool the print job, and the job would be printed when **lpd** was started by a local **lpr** process or the administrative command **lpc**.

4.2.4 Domain/DDE

The following existed in the Domain Distributed Debugging Environment:

- **STEP –SIGNAL** did not deliver the given signal to the target.
- FORTRAN alternate entry points were not supported.
- Argument information was sometimes unavailable in FORTRAN.
- If you were debugging an **obj** format module and set your environment to a module, no source file would be displayed and line number references would not be recognized.
- Variables declared in unnamed C inner scopes could not be referenced because the compiler did not provide information about the address range associated with the scope.
- Domain/DDE could not find correct address for variables in registers in code ranges that had been optimized away.
- Domain/DDE could not step through a C statement that included auto-initializations.
- A simple **go** command in the middle of an action list was not handled correctly.
- You could not use property **fork –parent –child** to follow the new process as well as the original process, if the new process was created by using **pgm_$invoke**. This was an operating system limitation.
- You could not use the **step –over** command to step through code that was loaded read-only; for example, when you attached to a running program, or when you stepped into a shared library.
- The **step -signal** command did not deliver the given signal to the target.
- Domain/DDE was sometimes unable to produce an assembly language display because the buffer that we used to store the disassembled code was too small.

4.2.5 Language and Tools

- When creating an output file, **obj2coff** now correctly ensures that the output file has execute rights for owner, group, and other. Previously, the protection for an **obj2coff** output file was assigned by the containing directory.
- The sys5.3 **cpp** sometimes incorrectly inserted a newline.

- Improperly positioned carriage returns in source code caused irrelevant syntax errors from the C preprocessor.
- The **-s** option to **ld**, which strips line number entries and symbol table information from an object file, failed to delete expunged symbols from **pic** files (object files containing position-independent code).
- **ld** sometimes found insufficient space for auxiliary symbol table entries.
- **ld** sometimes faulted when retrieving a block data subroutine from a library.
- **bind** sometimes retrieved the same block data subroutine module several times.

4.2.6 GPRLIB

The following bugs have been fixed in **gprlib**.

- Two problems related to passing very large rectangles through **gpr** were fixed.
- There was a bug with GPR UNIX signal handling which adversely impacted Ada programs using **pause**() to implement tasks.
- If you did a **gpr_$open_bitmap_file** (gpr_$readonly, ... bits, ...); followed by a **gpr_$set_bitmap** (bits, st); the **gpr_$set_bitmap** call returned a bad status `Bitmap is read-only`.
- If you did a tile fill to a 24-plane main memory bitmap, all but the lower 8 bits were zero.
- While doing a **printf** with GMR running, buffer swap was corrupted. The pad seemed to do a swap as well.
- Pixels were missing from lines drawn from 45 to 135 degrees. The missing pixels were near the clipping boundaries.
- **gpr_$arc_3p** worked incorrectly with clipping enabled if coordinate values were larger than 10935. The Display Manager would hang until `display acquire time out` occured. This problem was reported when using **gpr_$arc_2p**. With very large arcs, each point along the arc had to be calculated. To further slow it down, each point was being clipped to the clip window, which made the arc call slow. An alternative arc clipping method was developed to increase the performance of the routine.
- Autorefresh on DN590 nodes for true color/double buffering modes has been fixed.

4.2.7 Domain/OS

If you adjusted your system time with any of the tools that are used for this purpose, the system could use or report an incorrect value.

Domain/OS bugs reported at beta have been fixed as follows:

- Ctrl C in **xterm** from a Korn shell was sometimes ignored by the target process.

- Depending on the type manager, redirection from the C shell sometimes did not work.
- If command-line substitutions were done inside the startup script for a Korn shell, the shell would sometimes hang.
- When exiting or shutting down from the DM, a node could sometimes crash with a 40004 `reference to illegal address.`
- If the network was unstable, the **mbx_helper** process could fail with a read concurrency violation.
- Moving the mouse while **salvol** was running could cause **salvol** to fail.
- The rename system used as (**rename** *x,x*) would trash the file in cases where the file was not local, or in an NFS partition. It would behave correctly only if the file was on a local disk.
- Creating more than 45 /**com** shells could crash a DN3000.
- While in service mode, when using the single user shell, **systype** would not be set properly.
- In certain cases, the **uds** client server link was slower than the TCP link.
- **wbak -f 1 -dev ct -l -rewind -full** would not report an error, but nothing was written to tape.
- If you issued an **ioctl** to change the handling of tabs, it had no effect in sys5.3.

4.2.8 TCP/IP

The following TCP/IP problems have been fixed.

- Configuration problems that occurred during Beta on nodes having multiple Apollo Token Ring controllers or an Apollo Token Ring controller and an Internet Interface Controller (IIC) have been fixed.
- Record–structured file transfers were broken at SR10.1. You can now transfer record files using the **ftp struct** command.
- Prior to this release, the **rlogin** command would hang when logging into a *fast* machine (DN10000) from a *slower* machine (such as a DN3500).
- Performing rapid open and close operations on multiple sockets caused the **tcp** server to fail.
- The documentation for TCP/IP 3.1 did not define the meaning of state 0 when you invoked the **netstat** –A command. If a 0 appears in the PCB column, it means that a socket is in the process of being shut down.
- Prior to this release, if you used **telnet** to log in to a DN10000, you would be prompted for the password before your user name.

- There was a lack of security on reserved ports at SR10. Now programs must run as root to open a raw socket.
- If you invoked the **remsh** command from a workstation running only the SysV environment, **remsh** would incorrectly try to invoke the BSD command, **/usr/ucb/rlogin**. It now correctly invokes **/usr/bin/rlogin**.
- The **ftp quote** command did not work.

4.2.9 Domain X11

The following bugs have been fixed in Domain/X11.

- Text was occasionally displayed on top of the Modes menu in **xterm** when a large file was being catted.
- The first pixel at the start of wide polylines was missing; when the line width was greater than 1 and the **cap_style=projecting**, the cap didn't draw at the beginning.
- The **source.tar** files didn't have fixed header files.
- The hold key caused the DM to treat an X window as if it were a pad (that is, the DM drew a box and put the letter "R" in its header line). (Applied to DM-root mode, with **wmgr -on**.)
- Using an X window manager to position DM edit or transcript pads so they were partially off-screen, or partially obscured by other windows, even when active, had the result that scrolling text in these windows did not always work correctly.
- XWarpPointer didn't permanently update the pointer position in the window. Moving the pointer after XWarpPointer had been executed caused the pointer to return to its position before the warp. If any mouse button was pressed before the cursor was moved, the warp pointer position was updated and the pointer did not return to its initial position.
- After a popup, for example an **xterm** popup, was popped down, the cursor sometimes moved to the bottom left corner of the screen.
- Something kept moving the cursor if you ran **mwm** and moved one of the windows (either X or DM); when you released the mouse button, the cursor would appear to be in the correct place, but when you jiggled the mouse, you'd see the cursor in the bottom left corner of the window.
- Extraneous material appeared between the **xterm** scrollbar and command window, or the borders of graphics windows would shift, when running with a reparenting window manger.

4.2.10 Diagnostics

The following diagnostic bugs have been fixed.

- SAX failed with an error code 140003 on a diskless node that was brought up in the NORMAL mode.
- SAX hung if a child node crashed. If a mother node and a diskless child node were running SAX, the child node crashed.
- SAX reported the SAU7 SCSI ctape as being online and available for testing even though no tape was in the drive. The **ctape** test eventually failed when RBAK reported the unit was not ready.

4.3 Changes to the mkapr Tool

Since Apollo has merged with Hewlett Packard, the Internet address shown in the **mkapr** manual pages has also changed. If you expect to send APRs to Apollo via E-mail, please be sure that the MAIL_PATH field in your **.aprinit** file has the following Internet address:

apr_cs_admin@apollo.hp.com

While the old address (**apr_cs_admin@apollo.com**) may work for you, there is no guarantee that it will be correct. Note the following changes to **mkapr** (make Apollo Problem Report):

- The SR9.7 OBJ version of **mkapr** is now in **/install/sr9.7_compatibility/sr9.7_executables/com/mkapr**
- The SR10 COFF **cmpexe** object (both m68k and a88k) is now in **/usr/apollo/bin/mkapr**
- Both SR9.7 and SR10 versions are stripped of the debug symbol info; resulting executables are 40% smaller
- The annoying `can't change phone number in command line mode` bug is fixed.
- The MAIL_OPTIONS field is added to **.aprinit** to specify the sendmail(8) options to be used with **/usr/lib/sendmail**; default options are **-om -oi**
- **mkapr** will warn if aliases database needs to be built before sending mail
- **mkapr** will use DPSS/mail (**/com/mail**) only if sendmail is *not* available and DPSS/mail *is* available; MAIL_OPTIONS is ignored in this case.
- mkapr will always allow system services to report errors directly to the transcript pad; verbose mode (**-v** switch) will give info about **mkapr** actions such as locally saving copies of files and extended error and status messages.
- **mkapr** now has a **-c** switch to force **mkapr** to run in command line mode rather than with the Domain/Dialogue display.

- **mkapr** now allows you to specify the print command to be used when the PRINT action is requested. A new field has been added to the **.aprinit** file; it is called PRINT_COMMAND, and is given a default value of **prf -left 0.5 -wrap**. You may change the command string value to be whatever is appropriate for the your execution environment.
- **mkapr** no longer makes direct calls to the **/lib/prflib** print library. It is now your responsibility to ascertain, maintain, and use a correct print environment.
- In a bsd4.3 environment, the shell environment variable PRINTER can be used to indicate a printer to use in place of the default assumed by the **/usr/ucb/lpr** command. In a sys5.3 environment, the shell environment variable LPDEST is used by the **/usr/bin/lp** command. In an SR10.*x* Aegis environment, the file **~/user_data/startup.prf** may be used to provide needed information for the **/usr/apollo/bin/prf** command. In a pre-SR10 Aegis environment, the file **((^HOME))/user_data/prf.db** serves the same purpose.
- **mkapr** now allows the user to specify the directory path to be used when the SAVE action is requested. A new field has been added to the **.aprinit** file; it is called SAVE_PATH and has the default value of **.** (current working directory). The user may change this value to whatever is appropriate for the user's execution environment.
- When the SAVE action is requested, **mkapr** creates the output files used by the PRINT and SEND actions (the former is human-readable, the latter is intended to be sent to Apollo via E-mail); these files are initially created in the **/tmp** temporary directory; they are then moved to the directory specified by SAVE_PATH. The filenames are:

apr.<APR_ID>.p	the print file
apr.<APR_ID>.s	the send file

- The APR ID is now prominently displayed in **mkapr 2.04**
- The Dialog-based graphic display has been rearranged to accommodate the above changes.
- **mkapr** now deletes all temporary files it created during its execution when the EXIT action is requested. **mkapr** will not delete temporary files belonging to other **mkapr** runs; those files are deleted when the respective **mkapr** runs exit, or they will be removed when the node is next rebooted. Temporary file names are now of the form:

  ```
  apr.<APR_ID>.*.<PROCESS_ID>
  ```

  where * stands for any of **m, c, s, e, v, p**. The process ID is unique to the executing process (it is assigned by the OS). In the UNIX command shells, the variable **$$** will contain this value. It does not appear in the Aegis command shell environment, but can be seen by issuing the command **pst -un**.
- **mkapr** now has a popup edit help box for the editable text fields. This help box summarizes the editing information given in the next section.

4.3.1 Problem with mkapr

Since there currently is no documentation about editing text in a Domain/Dialogue interface, we explain the procedure in the following paragraphs.

If you select a field in the Domain/Dialogue interface for editing (by pointing and clicking with M1 and M3 or F1 and F3), a small filled triangle appears in the field as near to the cursor arrow as possible. This marks the insertion point for text.

If you type in the field, light square brackets appear at the far left and far right of the field (they can be hard to see). While those brackets are present, **mkapr** has not yet seen the text you have typed. Any **Print, Send, or View** action will use the field values known to **mkapr** (most likely these are empty string values).

If you press <RETURN>, all text from the left margin of the text field to the text insertion point (triangle) is passed to **mkapr**'s internal service routines for processing. The square brackets disappear to indicate that the editing of the field has completed.

There is no limit to the number of pending field edits. By starting a field edit, then pressing <TAB> to proceed to the next field (you think you have finished the edit), and starting to type in the new field, you now have two field edits underway. The edits must be completed with <RETURN>.

<TAB> is intended for quickly moving from one field to another without changing the field value. <RETURN> will move from one field to another, but will have drastic effects on the field values.

4.4 Fixed APRs

The following list indicates priority 1 and 2 Apollo Problem Reports (APRs) that we have fixed in SR10.2.

000DBBFF	aegis_kernel	Loss of LU's until gateway rebooted
000DBD78	sio	Newline delay problems at 9.5 and 9.6
000DBE74	aegis_kernel	Hanging node with Control Q
000DBEBF	sio	Siologin process not refreshed upon disconnect
000DC055	systest	boot fails power off in normal mode
000DC06C	dm	auto refresh problem with grown windows
000DC087	gpr	gpr_$set_cursor_pattern problem
000DC0E7	nfs	nfs group access fails if not primary group
000DC116	sio	SIO transmission pause SR9.7
000DC1C1	aegis_kernel	/com/crddf dff_file doesn't display the field
000DC1D1	dm	TI command moves cursor to read only pad
000DC1F1	tape/floppy	Sys crashes attempting a ctrl Q during wbak
000DC21F	dm	DM cursor jumping
000DC29A	sio	SIORF fails with the replace -r option
000DC36D	dm	Jumping cursor using non-default fonts
000DC3B3	unix_clib	trouble with pipes parentchild and no output
000DC400	aegis_backup	Rbak creates bad pad for scrolling
000DC429	spe	problems with tcp/bsd4.2 ver 3.1
000DC478	print_server	Timeout problem with Apple Laserwriters
000DC4D2	nck	Bug in VMS NCK Release notes
000DC52F	com/crp	EMT does not work correctly in CRP window
000DC55A	install	/usr/include does contain /usr/apollo/include
000DC568	dm	/sys5.3 and bsd4.3 bin/mesg requires root
000DC577	mbx	MBX_$OPEN hangs
000DC582	gpr	click on files failes to produce text fonts/f
000DC5B6	gpr	cursor freeze with Matrox board
000DC5EC	rgy	Option regarding validity of passwd at login
000DC5FD	aegis_kernel	Wrong system date crashes node with 3030007
000DC606	gpr	gpr problem on meerkat
000DC68D	gpr	missing pixels from clipped lines
000DC68F	aegis_backup	WBAK needs to give error on bad tape marks
000DC691	ctm	missing cmd in ctm.ins.c
000DC692	aegis_backup	RWMT -i ctape controller timeout in SR9.7.1
000DC6DB	mbx	mbx sample programs hang
000DC6E1	a_net_eth	Patch for A0002 crash
000DC71E	install	Cannot backup reg information on stand alone
000DC74B	install	Install++ of pure Aegis as links fail
000DC7AF	unix	Tar on magtape not working at SR10.1
000DC7E4	float_pt	Bad EXP value with -cpu fpa1
000DC813	gpio	User request iomap interferes with ctape
000DC860	print_server	LPR/LPD does not work as documented
000DC885	aegis_kernel	NFS not allowing execution of apollo executables
000DC8CD	dm	File locking problem from remote node
000DC8CE	dde	TB -args goes wild
000DC91B	a_net_eth	DPCI-ENET does not work with SR10.1 beta

000DC923	dm	Jumping cursor
000DC948	print_server	Prsvr returns odd address error at startup
000DC958	aegis_kernel	Auto refresh problem at SR10 and SR10.1
000DC973	sio	stty dec crtkill doesn't work with siomonit
000DC989	aegis_backup	/USR/APOLLO/BIN/MT does not function right
000DC994	unix	chsh gets seg. fault if not root
000DC995	a_net_atr	Unable to use ring 1 at sr10.1
000DC9B9	aegis_kernel	Multiple files created with the same name
000DC9D2	aegis_kernel	Cannot boot DN300 diskless at SR10.1
000DCA01	sio	-NLD option of TCTL does not change from 0
000DCA0B	emt	/COM/EMT interm ff doesn't work on 9.7 compat
000DCA0D	aegis_shell	Shell fails to interpret ./ in pathname
000DCA22	dbx	DBX internal err printing array elem:IOT trap
000DCA31	aegis_kernel	Coprocessor protocol violation on 68882 node
000DCA37	gpr	Problem with text path up
000DCA3D	unix	System V VI
000DCA3F	aegis_kernel	'Object not found' when executing files
000DCA44	gpr	Crash '/lib/grlib:.text' offset 9f8
000DCA54	gpr	>40 fonts
000DCA57	unix_clib	Problems with mmap().
000DCA7D	dm	Display Manager edit of NFS_FILE type uids
000DCA96	gpr	Bit BLT to main memory bug on AT
000DCA9C	print_server	SR10.1 DSP-90 V-80 printer undefined global
000DCAA2	print_server	Printing with -wrap option
000DCAA8	tcp/ip	RLOGIN does not work
000DCAC4	aegis_kernel	Unimplemented instruction on 68882
000DCAFB	install	MINST fails if no name is given for AA
000DCB01	sio	Problem with IOCTL on siolines
000DCB06	gpio	Cartridge tape controller timeouts at SR10.1
000DCB13	print_server	Printronix Printer Problem in SR10
000DCB15	a_net_eth	A-NET-ETH
000DCB16	gpr	Fill pattern bug on AT
000DCB29	aegis_kernel	Phase II Boot Shell
000DCB34	dm	PAD_$DM_CMD takes on userid of person logged
000DCB3C	com/netsvc	NETSVC -P reports incorrect paging size
000DCB4A	com/mtvol	Mounting floppy with mtvol
000DCB56	unix	Termcap delays are ignored
000DCB5C	gpr	Linestyle bug if scale set to -1
000DCB62	a_net_eth	ETHERNET_MICROCODE
000DCB93	aegis_kernel	NAME_$READ_DIR_LC in Fortran causes problems
000DCB94	install	edacl_sp does not get executed
000DCB95	dm	dm gets corrupted after a sequence of events
000DCBA0	tcp/ip	Unable to talk or write to 'user.group'
000DCBAE	mkapr	MKAPR enhanced by adding a 'save' button
000DCBB8	float_pt	Invalid casting of C Compiler with FPA

000DCBD5	nfs	NFS truncates files
000DCBF2	aegis_kernel	Invol problem with SR10.0.p
000DCC0C	aegis_kernel	cp change the file type when copying
000DCC0D	gpr	Borrow mode font problem
000DCC2F	unix	BSD4.3 ls still broken
000DCC3A	rgy	Problems with EDRGY
000DCC3D	install	Invalid proj error on RAI install at SR9.7
000DCC47	domain/dialog	Limits to Dialogue file sizes
000DCC53	tcp/ip	Rlogin to a SUN ^C hangs pad
000DCC5F	rgy	Two accounts
000DCC82	sau/invol	INVOL suboptions 2 & 3 don't work on striped
000DCCA9	streams	inquire and bitmap files fails
000DCCB3	unix	Search & Replace segmentation Faults in Vi
000DCCDD	com/mvf	Compiling C program in an NFS environment
000DCD13	unix	SYS5.3 date and time report inconsistent
000DCD17	rm/idm	IDM error from enabling gpr_$no_event
000DCD2D	gpio	VME csrs not available for 24-bit devices
000DCD3C	print_server	SR10.1 problem printing with Genicom printer
000DCD3D	print_server	SR10.1 SPE and TEK 4692 do not work
000DCD66	tape/floppy	00280022 crash on DN590T w/ESCI c-tape
000DCD6C	x_window_sys	server crash with Xdrawseg more than 128
000DCD90	cc_tools	CC 6.6 SysV.3 string length
000DCD9A	domain/dialogu	String_field fails with long character string
000DCD9C	streams	error after numerous backspace commands
000DCDA4	unix	BSD4.2 /bin/sh does not process unstruct file
000DCDAB	dm	DM login problem
000DCDC5	gpr	invalid poligon fill on frame mode of GPR
000DCDD7	sio	SIO line and the quite signal problem
000DCDFA	rgy	getpwnam fails
000DCE03	aegis_backup	If only 'r' on directory WBAK fails on link
000DCE0A	aegis_kernel	Problem with obty for type obj under SR10
000DCE0B	aegis_kernel	/etc/server -p starts processes with person
000DCE3C	aegis_kernel	SR9.7.4/9.7.5 does not boot diskless fails
000DCE59	print_server	SR9.7 pathname longer 100 characters crash
000DCE5F	tcp/ip	Pasting in Telnet session to VAX/VMS
000DCE7D	emt	EMT Xmit fails at SR10.1 to VAX ok at 9.7
000DCE92	dde	dde produces incorrect results when exam vars
000DCEB0	unix	^Y pseudo tty security problem
000DCEB3	rgy	rgyd consumes large amounts of cput time gro
000DCEB8	aegis_backup	rbak/wbak link name case
000DCEC2	dbx	dbx problems with large files
000DCECA	unix_langtool	Position Independant code with /bin/cc
000DCEE1	com/cpt	9.7 cp[ft] -sacl com not preserve Unix sup(8)
000DCEE4	aegis_kernel	processes are not killed at logout
000DCEF2	unix_syscalls	The BSD4.3 getrusage system call fails

000DCEFF	gpr	multiline clipping problem on 3xxx4xxx
000DCF0E	aegis_swtools	/usr/ucb/last - too many spaces
000DCF15	print_server	lpr - not reliabley recognize certain files
000DCF22	nfs	NFS networked mounts
000DCF24	dm	mesgy fails is users isa non super user
000DCF5E	unix_clib	Sys5.3 'ioctl' doesn't work correctly w/keyboard
000DCF73	aegis_kernel	entire ftn array allocated at load time: sr10
000DCF90	gpr	problems with gpr_$arc_3p
000DCFA1	gpr	Node crash with reference to illegal address
000DCFA3	aegis_kernel	monitor goes mono using getty on login atmpt
000DCFAD	aegis_backup	WBAK reports recovery error DN3500 at PSK2
000DCFB1	unix_langtool	Symbol table overflow in /bin/ld
000DCFB5	unix	Sys5 Make error - no more available streams
000DCFBA	x_window_sys	Xterm -l or -lf option not enabling logging
000DCFD7	unix_syscalls	dn4500 sr9.7.5; last digit trunc in transcrip
000DCFDD	aegis_backup	/sys5/bin/tar on a SCSI 6250 magtape
000DCFE9	aegis_kernel	Security Problem
000DCFF1	rgy	RGYD aborts during normal maintenance
000DCFF7	gpr	msr aborts jobs with filled polygons on 570
000DD000	unix	unix ld -l command hands
000DD039	aegis_kernel	Accounting startup
000DD042	com/bind	/com/bind mangles ACLs
000DD07A	unix	AR in MAKE - cant make symbol directory
000DD07F	install	Unix sys5 print utility problems
000DD080	print_server	Unix bsd4.3 print utility problems
000DD084	dbx	DBX fails to come up on small test program
000DD08B	aegis_kernel	open system tooklit doc
000DD08E	unix	/etc/accton gives bad eventcount key
000DD090	ncs	NCS brokers talk with TCP/IP by default
000DD09C	aegis_backup	Tape processing at workstations
000DD0B4	aegis_backup	Wbak problems
000DD0BF	print_server	Banner size hardwired to 8.5 by 11 in sr10.1
000DD0D2	unix	slow performance with user accounting enabled
000DD0D9		print_serverSR10 prsvr and Versatec plotter does not work
000DD0EC	a_net_atr	Irma board does not work with SR10
000DD0F8	aegis_backup	Using 'tar' cannot write 2 archives to ctape
000DD0FB	aegis_kernel	urently require a kernel recompilation of 96
000DD11E	aegis_kernel	/etc/server dangerous when setuid
000DD125	com/crp	CRP w/emt from 10.1 to 10.1 causes seg probs
000DD128	unix_clib	ioctl and fcntl in bsd4.3 are not completely
000DD12C	unix	/bsd4.3/bin/start_ksh outputs 2 motd
000DD13E	x_window_sys	colour of images
000DD13F	x_window_sys	cursor chagnes colour/disappears
000DD140	x_window_sys	problem is pixmap become corrupted
000DD141	x_window_sys	region functions do not seem to work

000DD142	gpr	remote borrow gpr screen timeout problem
000DD143	x_window_sys	cursor still drops blobs
000DD146	x_window_sys	font corrupt
000DD14D	x_window_sys	Problem with XSubtractRegions
000DD158	rgy	RGYD cannot initialize slave database across
000DD16E	tape/floppy	node crashwith floppy and SCSI I/O together
000DD194	rgy	EDRGY change of account makes rgy too lose
000DD195	x_window_sys	Multiple defines error in SR9.7 sys5 & X11
000DD1F2	aegis_kernel	sr10.1
000DD1FD	x_window_sys	Missing Pascal and Fortran interude routines
000DD1FE	x_window_sys	Malfunctioning interlude routines in v1.0
000DD20B	aegis_kernel	substitute to other uids
000DD215	aegis_kernel	sr10 existf doesn't recognize sr9.7 objects
000DD21F	aegis_backup	sr10.1 rbak backups
000DD24E	gpr	bug if text path is not to the right
000DD26F	aegis_kernel	system accounting
000DD28E	aegis_backup	wbak to stdout
000DD299	rgy	passwd modifies registry fulname
000DD29A	rgy	'mesg y' can not change mode
000DD2B0	rgy	Using getgrent() floods network with requests
000DD2B8	tcp/ip	Aegis 10.1 install missing login
000DD2D1	spe	TCTL does not toggle -INSYNC -RTS et al
000DD2DF	gpr	gpr problem with distributed building
000DD2EE	unix_syscalls	SysV fcntl() with N_DELAY option
000DD2F1	sau/salvol	a bug in the salvol on sr10.1
000DD306	tape/floppy	rwmt cannot handle all -bl lengths
000DD311	dm	key def for shifted ^ keys same as unshifted
000DD320	dde	DDE source line pointer aligment
000DD32B	tape/floppy	wbak -f cur SCSI failure
000DD32C	edfont	edfont - trouble moving origina in large font
000DD32E	tape/floppy	rwmt cannot read ascii files on scsi cntlr
000DD345	unix	bsd 4.2 Domain/IX Bourne-shell continue-comma
000DD34C	tcp/ip	vt100 then telnet hoses tty
000DD352	tcp/ip	Same rlogin problem reported in DC972
000DD355	aegis_backup	Wbak -dev ct -f 1 . does not backup anything
000DD366	sio	Siologin on dsp90 doesnt logout properly
000DD377	aegis_kernel	registry services lost periodically
000DD392	print_server	sr10.1 Printronix printer does not work on ds
000DD39A	aegis_backup	rwmt '-rf f' does not work
000DD39B	tcp/ip	/etc/tcpd (tcp/ip daemon) dying in same spot
000DD3A6	dbx	dbx problems under SR9.7
000DD3BF	aegis_backup	Error using WBAK -DEV c -F 1 STEVE -L ...
000DD3C6	dde	lose control stepping in phigs execs to end
000DD3C8	unknown_graphi	Dialbox problem with X windows gprlib
000DD3E2	unix	sys5.3 newgrp command ignores group password

000DD3E8	install	/usr/apollo/bin/login not installed properly
000DD403	install	Inprot can't create sid entry with no rights
000DD40E	aegis_kernel	Socket limit on DSP10020 too low
000DD40F	unix_mail	mail fails with 'set crt=20'
000DD420	tape/floppy	Tar to dual scsi magtapes fails
000DD424	unix	cntrl chars func keys cause xterm to stop
000DD44B	unix	bsd4.3 script command doesn't fork a shell
000DD46B	unix_langtool	Get or put outside of file error in 9.7
000DD476	unix_mail	Biff still fails under sr10.2 beta
000DD47C	x_window_sys	sys crash when large number of widgets create
000DD47D	x_window_sys	long text string broken in X
000DD47F	x_window_sys	cursor turned off when X app exits
000DD482	unix_clib	bug in sr10.1.p system V msg queue code
000DD486	dde	DDE gives error(s) on c progs with >1 name
000DD488	rgy	edrgy: changing required entries
000DD48E	nfs	umount -a does not unmount nfs file systems
000DD497	unix_langtool	'aux table overflow' error in /sys5/bin/ld
000DD49F	x_window_sys	Xterm -l does not produce log file
000DD4AD	aegis_backup	wbak -stdout
000DD4B2	print_server	BSD spooling system inoperble
000DD4D7	gpr	piced fill patterns bleed
000DD4F7	rgy	edrgy prob doesn't change props if owner
000DD4FE	print_server	patch_m68k_8908_0046.cps -n option for prmgr
000DD4FF	unix	sys5 ls -l breaks without rgyd
000DD512	rgy	sr10 import_passwd registries rgy Unix
000DD513	dbx	dbx gives wrong value for extern var in C
000DD514	tcp/ip	tcpd terminates abnormally when routes age
000DD51B	x_window_sys	X shared mode
000DD526	print_server	CGM Plotter v 1.0 problems with multiple plot
000DD52D	dm	DM loosing context of cursor
000DD52F	dbx	dbx incor vals & exe errs when pr arr mems
000DD533	rgy	cvtrgy sr10.2 beta 'unable to find uid ...'
000DD536	rgy	edrgy - cannot change rgy info as root
000DD548	rgy	10.1 rgyd breaks UNIX group syscalls on 10.2
000DD55B	nfs	NFS problems between apollo and HP
000DD57B	rgy	rgy_merge 'unauthorized to perform operation'
000DD588	com/cmf	cmf can't handle derived names
000DD58A	dde	debugging from 4500->prism dies with SIGTTIN
000DD58B	sio	cbreak mode broken in SR10.2
000DD5B5	gpr	Direct mode breaks frame mode on Meerkat III
000DD5BD	x_window_sys	Access violation in IDM$$GET_RECT_PTR line491
000DD5EC	aegis_kernel	SR10.1.p doesn't allocate all of phys mem
000DD60D	aegis_kernel	SPE using irq3 as global device crashes340001
000DD61D	gpr	gpr_$set_auto_refresh returns bad status
000DD66F	domain/dialogu	Dialogue problem crashing when writing

000DD67F	x_window_sys	XPutImage gets offset with obscuring windows
000DD68B	unix	/usr/lib/graf/dev.ps missing from SR10.2 psd
000DD69E	siologin	Arg #1 isn't being passed to the startup_sio
000DD6B6	float_pt	sin cos tan etc are not using 881 chip
000DD6B8	aegis_kernel	dm command ex releases global devices twice
000DD6BF	aegis_kernel	SR10.x Virtual Memory Allocation Problem
000DD6F3	gpr	insert files gpr.ins.c and types.h conflict
000DD6F4	rgy	sr9.7 registry database file limit
000DD6F6	unix	vi editor doesn't work using rlogin or getty
000DD752	unix	sys 5 man fails for 'X' man page
000DD7A1	print_server	Online help for PRSVR/CONFIG is wrong
000DD7A2	print_server	prf on-line help has errors
000DD7A8	unix	output re-direction is not consistent
000DD7BA	unix_mail	sendmail bug causes mail dsta to be lost
000DD7C2	sau/invol	invol option 2 fails on 760mb disk
000DD7E2	gpio	crash 120031 when no user intr rout defined
000DD863	aegis_kernel	degradation of KGT mapping with lots of symbols
000DD87E	dm	XCreatePixmap fails if depth is not 1 or 8
0D880568	aegis_kernel	x25file_svr process dies if ring failure
0D881947	systest/rwmt	RWMT behaving differently
0D881950	dm	wlfutr - fixed
0D881951	dm	wl9.5 - same as previous one - fixed
0D882141	vfmt	vfmt string limit of 512 chars?
0D882267	netmain	netmain outputting garbage
0D882283	com/acl	acl cmd not returning appropriate status
0D882586	systest/rwmt	rwmt crashing processes
0D882623	aegis_kernel	SIO line hanging using emt to a vax
0D882729	vfmt	vfmt broken for dec field width > 20
0D882777	aegis_kernel	node not responding to rapid cts change
0D882836	aegis_backup	fundamental problem with wbak...
0D882857	naming_server	Problems with name_$ status codes
0D883064	emt	new line delay option to emt broken
0D883130	unix_langtool	Bug in AR when adding zero length file
0D883167	unix_langtool	bug in ar reference to illegal address
0D883514	sau/calendar	cal_$encode sick
0D883548	aegis_shell	Suspended in process in Unix
0D883746	sio	deadlock using xon/xoff
0D883770	aegis_kernel	directories being lost by MFACTS on DN3000
0D883835	aegis_shell	When you crp you ~user_data/startup should
0D883889	print_server	Provide status messages to users
0D884066	aegis_shell	Case insensitivity in the OS
0D884087	unix	ls -R segmentation fault
0D884112	com/cpt	cpt should verify that it not deleting
0D884115	com/cpt	The cpf -r command deletes target file even
0D884179	streams	streams problem between 9.0 and 9.2.3

0D884490	systest/rwmt	rwmt -w removes internal & trailing blanks
0D884565	streams	NAME_$READ_DIR not returning documented status
0D884699	domain/dialogu	likes to have gpr_$set_cursor in dialog
0D884725	sio	prob with misbehaving sio lines on dspxxs
0D884784	gpr	/COM/CPSCR inverts bitmap on DN300
0D884849	systest/rwmt	RWMT not working with large files
0D884954	install	crucr doesnt work when given from vt100 window
0D884980	systest/rwmt	rwmt fail if multiple line feeds at EOF
0D885017	aegis_kernel	crp and login to look for user_data/sh/startup
0D885032	aegis_kernel	name_$resolve_stop_stort copying characters
0D885084	dm	cant undo paste >512 lines
0D885111	domain/dialogu	need more than 2 decimal in real field
0D885142	dm	Loss of input events when window switched
0D885247	gpio	pbu_$wait never gets control back on quit
0D885278	streams	stream_$inquire lies
0D885386	aegis_kernel	something about getting 2 copies of spm
0D885604	domain/dialogu	need more than 2 decimal in real field
0D885706	dm	ld prints all the files on one line
0D886066	gpr	gpr bug affecting within-gpr mode on dn3000
0D886275	dm	lost mouse button up events
0D886604	help_files	set size with grave accents doesn't always
000DD6D4	dde	DDE crashes when printing a variable
000DD82C	com/dpat	10.2 dpat does not work correctly
000DD8E8	gpr	control -return missing downstroke key event
000DD6A8	gpr	gpr_$terminate basic heap storage manager
000DD7EB	gpr	gpr_$terminate basic heap storage manager
000DD891	gpr	mouse button position info bug
000DD8C0	gpr	auto refresh slower on coyote at sr10.2 beta
000DD2A5	dm	PAD_$LOAD_FONT
000DCED2	gpio	/user/include/apollo/pbu.h errors
000DD1E8	gpio	GPIO interrupt problem
000DD5F9	gpio	AQDEV error in SR10
000DD762	gpio	pbu_$device_interrupting crashes as 120011
000DD581	aegis_kernel	Program runs forever at sr10.1/10.2
000DD519	aegis_kernel	renaming sets of files with UPPERCASE pattern
000DCCE5	aegis_kernel	security hole in Apollo OS
000DCCE7	aegis_kernel	mts_$create_default_desc fails on 9.7 file
000DCDE8	aegis_kernel	Network moves
000DCF21	aegis_kernel	1protect - control local protection
000DD22B	aegis_kernel	profile doesn't give corr val for % time used
000DD6D3	aegis_kernel	SR10.1 cp command corrupts compound exe files
000DC8E2	aegis_kernel	ACL cannot handle path names 256 characters
000DC99A	aegis_kernel	Unix groups don't work setgroups(2) broken

4.5 Open APRs

The following list indicates the priority 1 and 2 Apollo Problem Reports that remain open.

000DD282	tcp/ip	SO_LINGER and SO_REUSEADDR options for socket
000DD2B4	uucp	Tip sets port to 8 bits causing problems
000DD857	tcp/ip	tcpstat -h -n not enough space
000DD7F9	vstreams	fstat and cloneable devices
000DD671	tcp/ip	corrupted files with ftp
000DCCD3	tcp/ip	System calls 'setsockopt and getsockopt'
000DD737	tcp/ip	ftp trashes byte following 0D used to NULL
000DD808	tcp/ip	rwho report on FM display login
000DD856	tcp/ip	VT100 Telnet combined cause VI problems
000DD868	tcp/ip	sr10.1 ftp's 'mget' fails on DG mv4000
000DD3E9	uucp	'tip' always sets even parity on lines
000DD3D9	tcp/ip	Telnet session hangs when file size > 512k
000DD2A6	rgy	floating point error from dde on fpa systems
000DD479	rgy	rgy_merge does not work
000DD4E3	rgy	virgin registry login
000DCE5B	rgy	problems with reg
000DD633	gpr	gmf copy plain fails if x dim > 4096
000DCA19	gpr	clipping prob with large arcs and lines
000DD1C5	gpr	GPR_$INQ_DISP_CHAR call causes strange side e
000DD42D	gpr	drawing arcs that use line widths > 1
000DD42E	gpr	displaying polylines greater than 1
000DCF64	gpr	pgon Fill doesn't work with clipping
000DD033	install	System installation tools
000DD6A4	float_pt	floating point operand error
000DD83C	float_pt	floating point overflow in bsd4.3
000DD693	float_pt	undefined globals
000DCDB4	float_pt	OS trap handler returns execution
000DD78C	float_pt	support for the FPA board
000DCF80	unix_langtool	f77 fails to load BLOCK DATA from library
2B6AAD9C	netmain	Problems extracting data from large log files
000DD449	com_ecmb	tcpd dies in Ethernet driver on dsp90
000DD727	unix	math.h include file incomplete in bsd4.3
000DD6AD	aegis_kernel	unable to obtain sfcb hash table mutex lock
000DC93F	aegis_kernel	NET-ETH timeout problem for domain level svc
000DD81A	aegis_kernel	errlog_$traceback output useless
000DD79B	aegis_kernel	'unstruct' type mgs fails bkwd get @ map seg
000DD088	sio	VT100 emulator buffering problem
000DD45B	emt	emt to vax 4800 only at sr10.1
000DD62E	aegis_kernel	SR10.1 EMT 'input buffer overflow' not 9.7
000DD6C9	sau/invol	error reading config file
000DD83D	sio	Input buffer overrun problem

000DD887	tape/floppy	Can't write after magtape EOT marker detected
000DD8DE	streams	unix dir i/o broken with 3rd party type mgrs
5B547ED0	sio	tctl reports 0 for new line delay
000DD6A3	siologin	If DCD drops Siomonit still runs.
000DD469	install	distaa calculates needed space from boot volume
000DC277	aegis_backup	WBAK problem with large amounts of data
000DCCEE	sio	DSPs on SIO1 2 &3 intermittantly hang
000DCD8C	aegis_kernel	Downcase does not lower case of command flags
000DD150	aegis_kernel	Config does not allow for the absence
000DD1BF	aegis_kernel	64 process limitation
000DD1DC	tape/floppy	cpio -icvd gives Error: out of phase
000DD3B9	com/mvf	mvf command loses a file
000DD3E0	unix_clib	Can't kill background process
000DC924	install	installation needs some improvement
000DCB24	install	Replacing links with install++
000DCB39	install	Install update 10.0 to 10.1 causes problems
000DCB3E	install	Minst problems
000DCBF5	install	Boot from ctape install leaves some acls open
000DCCFD	tape/floppy	Problem acquiring floppy drive in sr10.1
000DCE08	aegis_kernel	Diskless booting of DN590-T
000DCE71	install	Replacing links with install++
000DD005	aegis_kernel	aegis 9.7.0.6 crashes when floppy is used
000DD0F9	install	Protection Templates need to be provided
000DD10D	install	ACL changed to from BSD4.3 to sys5.3
000DD17E	aegis_kernel	tb can't addresses: infinite loop once
000DD232	aegis_backup	WBAK thru SIO dumb term doesn't multi volume
000DD264	install	install problem
000DD29B	unix_clib	C statfs function returns incorrect values
000DD2BF	unix_syscalls	unix sys call 'fseek'
000DD2F2	install	sr10.1 version of the cartridge SW
000DD32D	unix_syscalls	bsd ioctl(0TIOCFLUSH&n) fails - TIOCFLUSH
000DD32F	aegis_kernel	tb functionality removed with SR10
000DD367	a_net_eth	'object not found' from rbak to remote node
000DD386	unix_syscalls	sys5's fopen(<file> 'a+') doesn't work
000DD3B1	streams	ec2_$wait doesn't catch pipe info with gpr
000DD3CC	install	installation faults
000DD3F4	tape/floppy	dn570 multibus magtape
000DD445	aegis_backup	wbak error message when no EOF
000DD478	unix	syslogd does not write marks
000DD501	aegis_kernel	diskless boot
000DD5AC	install	distaa doesn't load DOC files
000DD5C6	tape/floppy	handling multi-fime MT in unix
000DD62D	aegis_kernel	pbu_$control swap bytes does not work
000DD735	sau/invol	760 marten hw or invol failure
000DD736	aegis_kernel	netstat -config says no SCSI ctape exists

000DD739	com/mtvol	can not mtvol w1 in /etc/rc.user
000DD782	sio	File transfer with emt and sio lines
000DD79F	emt	EMT james during XMT1 of file
000DD7A0	unix_clib	sys5 & bsd43. rename() deletes file
000DD7B9	sio	tctl and documentation
000DD7BD	install	SR10.1 msg tape media will not install
000DD80E	aegis_kernel	node hangs during reboot
000DD84C	sau/invol	760 marten hw or invol failure
000DD574	unix_syscalls	sys5 telnet doesn't set pty speed properly
000DD63F	com/crp	sometimes cannot 'crp-on' to 10.2 node
000DD743	install	sr10.2 installmissed /usr/spool/lp/model
000DD7E8	sio	sr10.2 etc/getty has trouble with fonts
000DD84A	aegis_kernel	future needs for heterogeneous networks
000DD0F7	unix_syscalls	Signal handling within revcfrom()
000DCF7C	aegis_kernel	Needed enhancement to config utility and boot
000DCBD2	aegis_kernel	patch 143 using /com/invol crash A0002
000DD2FC	aegis_kernel	deadly embrace when a type mgr tries to get a
000DC83E	aegis_backup	CDC magtape hangs after doing ^Q in WBAK cmd
000DCE6E	aegis_kernel	DN3000 at SR9.7 crashes during shut
0D886772	aegis_kernel	cpf ./ seems to confuse cpf
000DD507	print_server	aegis printing manual
000DD597	print_server	versatec plotter V80 bw model 1200
000DC919	doc	Document need to quotes around shell keywords
000DD456	doc	getxx of domain_examples is m68k type
000DD8BE	dde	dde trouble with 2-dim array values
000DCF2B	domain/dialogu	dp_$terminate_dpd doesn't release memory
000DD681	domain/dialogu	problem with rubberbanding on AT
000DD3B4	domain/dialogu	menu popup remnant in graphics area
000DCA79	domain/dialogu	Bug in Domain/Dialog 2.0
000DD485	domain/dialogu	Heap size jumps 600k when techs rearranged
000DD573	unknown_graphi	sys5 graphics ad is broken/not user friendly
000DD31F	gpr	crash problem with window grow
000DD4F3	unknown	lbr2ar gives segmantation fault
000DD5DB	unknown	phigslib in /etc/sys.conf prevents node boot
000DD637	rgy	cvtrgy cannot convert corrupted 9.7 registry
000DD8B2	dm	Arrow up key sometimes bombs out with newer
000DD167	aegis_swtools	Sigp wildcard error
000DD494	unknown	bad colume label after 760MB striped chuvol
000DD4D2	unknown	Have need for at least 64 pseudo ttys
000DD549	dm	editing user_data/startup_dx.xx file
000DD657	vfmt	VFMT crash(/lib/streams:.test' offset 270E0)
000DD8BD	mbx	ksh hangs in CRP windows
000DD5DD	dm	Input event behave diff. from 9.7 to sr10
000DD27D	naming_server	ns_helper is not functioning correctly.
000DC6D2	domain/dialogu	Extra refreshes occurring in graphics_area

000DD10A	doc	MKAPR does not mail correctly
000DCD93	unix_clib	Drand48() producing segmentation fault
000DCF1E	dm	nfs_file_owner
000DD030	dm	pad_$close_frame or popen cause triangle 'A'
0D880344	dm	wlfutr - tagging in reg exp multi lines
0D885590	doc	pad_$inq_disp_type has dummy output option
0D884306	dm	The following substitute command: s/@n{E}/
0D885225	dm	If you are scrolled up transcript pad
0D886158	dm	s/{?*}/@1;/ actually places 2 semi-colons
000DC50F	dm	xi DM command on dn3000 blank created
000DC30D	dm	use of gpr_$inq_window_id can cause node hang
000DCB2D	dm	Mouse movement is shakey after desired position
000DCF61	dm	problem with graphics window
000DD03C	dm	problems with DM starting new shell
000DCE77	dm	Icon button iconifying the other session
000DD08C	dm	XI command does not produce printable files
000DC3BF	dm	Problem with search and substitute command
000DD6FD	x_window_sys	Problem with sys5 only clients

Appendix A: New GPR Calls

This Appendix describes the GPR subroutine calls that are new or changed for SR10.2.

A.1 New GPR Online Examples

There are several new online GPR examples in C that are part of SR10.2. They are included in the directory /domain_examples/gpr_examples. These examples will be incorporated into the main directories for GPR C language examples (/domain_examples/cc_examples) when the C compilers are next released.

A.2 GPR Calls and Data Types

This section describes the concepts, data types and subroutine calls that are new for GPR.

A.2.1 Concepts

A.2.1.1 New Event Types

Five new keyboard input events have been added. These included support for the Latin-1 character set, full support of the Low-Profile Model II keyboard, shift and control key modified mouse button events, and keyboard focus change events. In addition, gpr_$set_quit_event has been added as a general replacement for smd_$set_quit_char.

The five new GPR input event types are:

- gpr_$coded_keys – represents an ASCII or Latin-1 character.
- gpr_$function_keys – provides full function key support.
- gpr_$physical_keys – Up/down physical events for all keyboard keys
- gpr_$kbd_enter_window – Keyboard focus enter window
- gpr_$kbd_left_window – Keyboard focus leave window

gpr_$coded_keys supports the 7-bit ASCII and 8-bit Latin-1 character sets. (Latin-1 includes ASCII in positions 0-127, and has the additional characters necessary for Western European languages in positions 128-255.) The keyset has 256 characters. The first 128 keys are ASCII characters; these keys are equivalent to the first 128 keys of the gpr_$keystroke keyset. The second 128 keys are the European letters and symbols from the Latin-1 character set.

gpr_$coded_keys replaces gpr_$keystroke. The gpr_$coded_keys and gpr_$keystroke events are mutually exclusive, i.e., an application can not simultaneously enable both events in the same attribute block.

Unlike gpr_$keystroke, gpr_$coded_keys does not support function keys. To get function key support, use gpr_$function_keys.

gpr_$function_keys represents 256 key transitions of the 64 keyboard function keys (all gray keys, escape, delete, backspace, return, tab, control, caps lock, repeat, left and right shift, left and right alt). Each function key on the keyboard has four possible transitions: down, shift-down, control-down, and up. Keys 0-63 of the keyset represent the down transitions; keys 64-127 represent the shift-down transition;, 128-193 represent the control-down transitions; and keys 194-255 represent the up transitions. See /sys/ins/kbd.ins.{pas,ftn,c} or /usr/apollo/include/kbd.h for the keyset values. This keyset includes the keys 128 to 255 within the gpr_$keystroke keyset. However, the keyset values of gpr_$function_keys are *not* the same as the equivalent keyset values of gpr_$keystroke keyset.

Not all function key events can be generated from the older Low-Profile Model I keyboard (DN300, DN400 and DN500 series) Events from these keyboards are limited to the equivalent function key events of the gpr_$keystroke keyset.

The gpr_$function_keys and gpr_$keystroke events are mutually exclusive, i.e., an application cannot simultaneously enable both events in the same attribute block. The gpr_$coded_keys and gpr_$function_keys keysets contain all the characters and function keys available in gpr_$keystroke, plus many new key codes.

For gpr_$coded_keys, gpr_$function_keys, and gpr_$keystroke events, the system translates the *physical* key codes into the appropriate *logical* key codes. The translation takes into account the keyboard subtype or language (North American, German, French, etc.). See /sys/ins/kbd.ins.{pas,ftn,c} or /usr/apollo/include/kbd.h for keyset values.

gpr_$coded_keys and gpr_$function_keys are sufficient for most applications. However, some applications, such as terminal emulators, need access to the actual physical key transitions. The physical key codes are numbers that are hard-wired into the system. gpr_$physical_keys is the event type that gives an application complete access to these physical keyboard transitions. The gpr_$physical_keys keyset contains 128 key down transitions and 128 key up transitions. (The Apollo keyboards actually have slightly less than 128 keys.) These events represent the actual physical transitions of the keyboard; they are unaffected by any modifiers keys or the keyboard subtype (language). Applications that use gpr_$physical_keys must perform their own translation.

gpr_$physical_keys can be enabled concurrently with gpr_$keystroke, gpr_$coded_keys or gpr_$function_keys. gpr_$physical_keys takes precedence over the others, i.e., if a keystroke could generate both a gpr_$physical_keys and one of the other event types, *only* the gpr_$physical_keys event is delivered to the application.

The Low-Profile Model II keyboard actually transmits down/up events for each key. The Low-Profile Model I keyboard does not generate down/up events for each key. For this keyboard pseudo up/down events are generated by the system. For example, if the keyboard transmits "A". The following events are generated: left-shift-down, a-key-down, a-key-up, left-shift-key-up. Thus, on the Model I keyboards, receipt of gpr_$physical_keys from the modifier keys (shift, control, etc.) do not necessarily correspond to actual state changes of these keys.

When an application runs under the Display Manager, keyboard events are tied to the mouse cursor. When an application, receives a gpr_$left_window event, it cannot receive keyboard input. If the application is running under an X window manager, however, it is possible for the *keyboard focus* and the *mouse focus* to be in different windows. (The focus determines which window will receive input.) To find out when the keyboard focus has entered and left the window, an application can enable gpr_$kbd_entered_window and gpr_$kbd_left_window. These events are similar to gpr_$entered_window and gpr_$left_window, which occur when the mouse focus enters and leaves the window. A gpr_$kbd_entered_window event occurs when the keyboard focus enters the window; gpr_$kbd_left_window occurs when the keyboard focus leaves the window.

If the application has not enabled gpr_$kbd_entered_window or gpr_$kbd_left_window, both the locator and the keyboard must be in the window in order to receive key events. It the keyboard focus is in the window, but the locator focus is not, GPR will reject the keystroke and sound a beep.

There are new mouse events in the gpr_$buttons keyset. These events can be used with the Low Profile Model II keyboard. These new events include shift-key and control-key modified mouse button events. The new keyset values are:

Event	Constant	Alternate Constant
Press the left mouse button.	KBD3_$M1D	a
Press the left mouse button and left shift key.	KBD3_$M1S	!
Press the left mouse button and control key.	KBD3_$M1C	C – 0x01 FORTRAN – char(1) Pascal – chr(1)
Release the left mouse button.	KBD3_$M1U	A
Press the middle mouse button.	KBD3_$M2D	b
Press the middle mouse button and left shift key.	KBD3_$M2S	"
Press the middle mouse button and control key.	KBD3_$M2C	C – 0x02 FORTRAN – char(2) Pascal – chr(2)
Release the middle mouse button.	KBD3_$M2U	B
Press the right mouse button.	KBD3_$M3D	c
Press the right mouse button and left shift key.	KBD3_$M3S	#
Press the right mouse button and control key.	KBD3_$M3C	C – 0x03 FORTRAN – char(3) Pascal – chr(3)
Release the middle mouse button.	KBD3_$M3U	C
Press the fourth mouse button (on a mouse with four buttons).	KBD3_$M4D	d
Press the fourth mouse button and left shift key.	KBD3_$M4S	$
Press the fourth mouse button and control key.	KBD3_$M4C	C – 0x04 FORTRAN – char(4)

Event	Constant	Alternate Constant
		Pascal – chr(4)
Release the fourth mouse button.	KBD3_$M4U	D

A.2.1.2 Pixel, Projection, and Video Formats

Traditionally, color raster display devices have consisted of an array of pixels. Each pixel in such an array has, in the past, been represented by a 1-bit, 4-bit, 8-bit, or 24-bit number. This arrangement is still very common in low-cost displays. However, in complex, high-performance displays such as the DN10000VS, each pixel is represented by either 40 bits or 80 bits. 40-bit pixels are available on the DN3550B, DN4500B, and DN10000VS. Drawing into pixels with this number of bits would be a challenge for application programs were it not for the fact that the bits in each pixel are formatted into functional subgroups.

The *pixel format* is the scheme by which the 40 or 80 bits in each pixel are divided into functional subgroups. The pixel format may be chosen by an application on a window-by-window basis. For example, one window might have a single 8-bit pseudo color image buffer and 72 unused planes (on an 80-plane display). Another window might have two 24-plane true color image buffers, a 24-plane Z-buffer, and an 8-plane alpha buffer.

The DN3550B, DN4500B, and DN10000VS provide a large number of formats from which an application can choose. data type description. These pixel formats afford application programs the choice ,window-by-window, of functional subgroups, or *projections*, of bits. Available subgroups are:

- Image data
- Overlay data
- Z-buffer data
- Alpha data (not yet available)
- A second buffer for any of the above subgroups

The image consists of the red, green, and blue planes of the pixel. (Or a single plane for monochrome.) It contains the image that appears on the screen. If double buffering is available on your device, you can request a second buffer of image planes.

At the present time, overlay planes are available on the DN3550B, and DN4500B, and DN10000VS. Future devices may also have them. The overlay planes contain an image that can be overlaid on the screen without destroying the image beneath it. Because the data in the image planes is not destroyed, you do not have to save and restore the image in the image planes. When you clear the overlay from the screen with gpr_$clear, the obscured image will reappear. When none of the bits of an overlay pixel are turned on, the pixel is transparent. The overlay planes have their own color map, which has four entries. It is in pseudo color. Entry 0 is transparent. This entry cannot be changed. Overlay planes are useful for such things as scrolling text on top of a picture, pop-up menus, dialogue boxes, and animation, where the animation would take place in the overlay planes and the background would reside in the image planes.

The alpha buffer is unused at the present time.

The Z-buffer determines which object is in front of another. GPR does not use this buffer. For information on the Z buffer refer to *Programming with 3D Graphics Metafile Resource*.

The application program chooses a pixel format that incorporates all the various projections required by calling gpr_$initialize. However, an application must also be able to control which projection, i.e., which functional subgroup, of the pixels it will manipulate in any given graphics operation. Each projection of the overall pixel format appears to applications as a separate and distinct GPR bitmap. The gpr_$initialize call both sets up a pixels format and allocates a bitmap for a particular projection within that pixel format. The application specifies which projection to use for the initial bitmap via a *projection format* parameter to gpr_$initialize.

Application programs can get access to other projections of a pixel format; The gpr_$allocate_projection call creates bitmaps for other projections. For example, suppose an application needed double-buffered 8-plane pseudo color with four overlay planes. The application would call gpr_$initialize to create a bitmap for the first image buffer and would call gpr_$allocate_projection twice, once to create a bitmap for the second image and once to create a bitmap for the overlay planes.

When double buffering is available, an application can write to either buffer but can only display one buffer. Usually, an application will write to one buffer while displaying the other. The *video format* determines which buffer is being displayed. The video format for the initial bitmap is set in the gpr_$initialize call. To change the video format, call gpr_$select_display_buffer.

A.2.1.3 Display Resources

Display resources are used in GPR initialization and in the new pixel format calls to specify the environment in which the application is running. Display resources are analogous to the display modes used with the old gpr_$init call, except that display resources do not specify pseudo color or true color. The types of display resources are:

- Pad – the application runs in a Display Manager pad. Analogous to direct mode.
- Screen – the application uses the entire screen. Analogous to borrow mode.
- Memory – the application uses a main memory bitmap. Analogous to no-display mode.
- X window – the application runs in an X window.

- Frame – the application runs within the frame of a Display Manager window.

A.2.1.4 16-Bit Fonts and Characters

Previously, GPR text commands only worked for input streams of 8-bit characters, e.g., ASCII or Latin-1 text. Therefore, the system could not support languages with more than 255 different characters, such as Kanji. 16-bit fonts permit up to 65,535 characters. SR10.2 introduces a new set of GPR text routines that support 16-bit character streams. These are gpr_$text16, gpr_$inq_text16_extent, gpr_$inq_text16_offset, gpr_$inq_character16_width, and gpr_$set_character16_width. They correspond exactly to the existing routines for 8-bit text. The only difference is that the 16-bit text routines require 16-bit integers instead of 8-bit characters as input. It is possible to use these routines with input streams of 8-bit characters as well by first converting the characters to 16-bit integers. For example, the 8-bit code 0x61 is represented in 16 bits as 0x0061.

SR10.2 introduces 16-bit fonts for Kanji (in three sizes) and for Hangul (in two sizes). These 16-bit fonts include the standard ASCII characters with codes below 127, so it is possible use either the new 16-bit routines or the existing GPR routines for 8-bit text to access ASCII characters. The 8-bit routines will not let you input character codes greater than 255, so the new 16-bit routines must be used to input all ideographic text codes.

It is the application's responsibility to specify valid character codes for the loaded font. Illegal codes are printed as blanks. Hence character codes above 255 print as blanks if a Latin-1 font is loaded, while character codes between 128 and 255 print as blanks if one of the new Kanji or Hangul fonts is loaded. Also, the new Kanji and Hangul fonts support the unshifted JIS coding scheme, so all character codes with a 1 in either bit 7 or 15 in its binary representation are illegal for these fonts.

Use gpr_$load_font_file to load 16-bit fonts as well as 7- and 8-bit fonts.

GPR does not load 16-bit fonts on the DN300 series of workstations.

A simple viewing program that lets you view the new Kanji and Hangul fonts, viewJIS, is provided in /systest/ssr_util.

A.2.1.5 Nondestructive Cursors

GPR now supports two types of cursors:

- Software cursors
- Hardware cursors (or nondestructive cursors)

Software cursors are available on all workstations. The DN10000VS, DN3500A, DN3550A/B, and DN4500A/B are currently the only workstations that support hardware cursors. To specify the cursor mode on devices supporting both types of cursors, call gpr_$set_cursor_mode.

In software cursor mode, the cursor can interfere with drawing. GPR helps applications avoid cursor interference with drawing by taking down the software cursor when the application calls gpr_$init. It is up to the application to put it back with gpr_$set_cursor_active (true, status). The software cursor can interfere with drawing and direct access in unpredictable ways. It is the GPR application's responsibility to get it out of the way with a call to gpr_$set_cursor_active (false, status). Software mode is the default for all devices.

The hardware cursor does not interfere with drawing. You can draw in the cursor area and the graphics will still be there when the cursor moves. The cursor does not destroy the graphics it covers, hence the name *nondestructive cursor*. There is no need to take down the cursor before drawing and and activate it after. Thus, hardware cursors have an important performance advantage. The cursor is always "active" – GPR does not take down the cursor on gpr_$init, and ignores requests from the user to take it down and put it up.

If you want to use the nondestructive cursor, but your application must be backward-compatible to run on devices that do not support it, use gpr_$set_cursor_active calls to take down and bring up the cursor and just set the cursor to the appropriate mode for the device; your application will still run faster in hardware mode. If you are writing a new application to run on a device that supports nondestructive cursors, leave out the gpr_$set_cursor_active calls for even better performance.

If your application uses hardware cursors and you want to make the cursor invisible, use gpr_$set_cursor_pattern to set the cursor to a pattern containing all zeros. Although it is still possible to make a software cursor invisible by calling gpr_$set_cursor_active (false), we recommend that you use gpr_$set_cursor_pattern for this purpose instead, as it works for both software and hardware cursors.

A.2.1.6 Pixel Mode Direct Memory Access

Previously, we made direct display access available via the gpr_$remap_color_memory, gpr_$inq_bitmap_pointer and gpr_$enable_direct_access calls. Between them, these calls allowed virtual memory access to one plane of display memory at a time. gpr_$remap_pixels enhances the function of gpr_$remap_color_memory in that it allows you to specify memory mapped pixel-by-pixel access instead of plane-by-plane access in hardware supporting that configuration. Pixel mode access to display and HDM memory is currently available on the DN3500A, DN3550A/B, DN4500A/B, and DN10000VS.

The memory mappings for display bitmaps and bitmap files are identical. The default memory mapping for any bitmap is by planes. In plane mode, the bits in the bitmap are accessed in plane order. In pixel mode access, the bits of a bitmap are accessed in pixel order.

Pixel mode access offers the following advantages:

- It gives you the ability to treat a color image on the screen as if it were an array in main memory. This ability is especially useful for imaging applications. Previously, these applications had to store color graphics in an array in main memory and then do a graphics operation to copy it on to the screen. With pixel mode access, as you compute the pixel image, you can store your image right on the screen. This results in superior performance.

- It allows you to create your own color graphics without using the GPR drawing calls. This results in greater control and flexibility for applications.

gpr_$open_bitmap_file and gpr_$remap_pixels both use the bitmap group header to specify the desired memory mapping.

Use the pixel_size field of the bitmap group header to specify the size of the pixel you want. You will probably want one of the following pixel sizes:

- 1 – for monochrome, plane access
- 8 – for 8-plane pseudocolor and for the red, green, or blue component of a true color map
- 24 – for true color

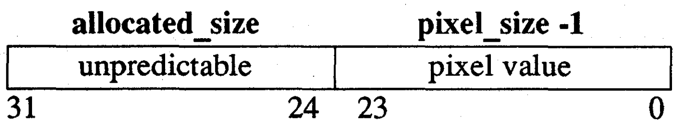
The actual pixel size that you get, however, is returned in the allocated_size field in the bitmap group header. It is based upon the pixel_size you specify. At present, the allocated sizes returned for the display bitmap are:

Specified Pixel Size	**Returned Allocated Size**
1	1
2-8	8
9-32	32

For compatibility with future devices and future software releases, however, your program should examine the actual allocated_size returned by GPR. An allocated size of one indicates that your program should access the pixels in plane mode. A value of eight indicates that you should access the pixels as an array of 8- bit bytes, and a value of 32 indicates that you should access the pixels as an array of 32-bit longwords.

The pixels are mapped with plane 0 as the low-order bit, e.g., for an eight-plane pixel, the binary value 10000000 would set plane 7 to 1 and all the remaining planes to 0.

When the allocated_size is greater than the pixel_size, the low-order bits of the pixel are significant, and the bits above the pixel_size have unpredictable values. The figure below illustrates a pixel_size of 24. The returned allocated_size is 32. The value for the pixel is in bits 0 - 23.



Do not assume that the bits above the pixel_size contain zeros, because in many hardware implementations they are random values.

All writes to memory are based upon allocated_size. For display bitmaps, you must be careful when pixel_size does not equal allocated_size because any data in the high-order bits will affect the image being displayed. This is not an issue for file bitmaps because the data above the pixel_size is not displayed.

There are three types of bitmap access, which correspond to the allocated sizes returned in the bitmap group header:

1. Plane mode
2. 8-bit (byte) pixel access
3. 32-bit (longword) pixel access

Plane mode access treats an X x Y x D bitmap (X bits wide, Y bits high, D bits per pixel) as D consecutive arrays of X x Y bits. The figure below illustrates the order of bits for plane access for an eight-bit pixel.

Plane Access

Byte 0

Pixel Number	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Byte 1

Pixel Number	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

In plane mode, the leftmost bit in the data becomes the leftmost bit on the screen.

Eight-bit pixel access (often called "chunky planar") treats an X x Y x D bitmap as D/8 consecutive arrays of X x Y *bytes*. Use chunky planar mode for 4-plane and 8-plane pseudocolor, and to separate the red, green, and blue planes in true color (useful for imaging). For true color, you specify three 8-bit "chunks". If you specified a pixel_size of 4, planes 4 - 7 (bits 7 - 4) are random values. The figure below illustrates the order of bits for pixel mode access for eight bits per pixel.

8-Bit Pixel Access

Byte 0

Plane Number	7	6	5	4	3	2	1	0
Bit Number	7	6	5	4	3	2	1	0

Byte 1

Plane Number	7	6	5	4	3	2	1	0
Bit Number	7	6	5	4	3	2	1	0

32-bit pixel access (often called "chunky") treats an X x Y x D bitmap as an array of X x Y longwords. It may be used for true color bitmaps. True-color bitmaps often contain 24-bit pixels, of which eight bits each are allocated to red, green, and blue. Therefore, 32-bit mode is useful for storing pixel data in consecutive bits. The allocated_size that you get for a true color bitmap however, is a 32-bit pixel. Red is in bits 23 - 16; green is in bits 15 - 8; blue is in bits 7 - 0; bits 31 -24 are unpredictable. The figure below illustrates the order of bits for a 32-bit pixel in which 24 bits are actually being used.

32-Bit Pixel Access

Longword 0

Plane Number	← unpredictable →								23	→	0
Bit Number	31	30	29	28	27	26	25	24	23	→	0

A.2.2 Data Types

gpr_$display_config_t

The following new configurations have been added:

- *gpr_$color10_1280X1024* – A DN10000VS 40- or 80-plane color display
- *gpr_$color7_1280x1024* – A DN3500/DN4500, 8- or 40- plane color display
- *gpr_$mono9_2kx1k* – A 1280x1000 or 1000x800 monochrome display

gpr_$controller_type_t

Two new controller types have been added:

- *gpr_$ctrl_color_10* – DN10000VS
- *gpr_$ctl_color_7* – DN3500/DN4500

gpr_$event_t

The following new event types have been added:

- gpr_$coded_keys – an ASCII or Latin-1 character
- gpr_$function_keys – a function key
- gpr_$physical_keys – an untranslated keyboard character
- gpr_$kbd_entered_window – the keyboard entered the window. Direct mode is required.
- gpr_$kbd_left_window – the keyboard left the window. Direct mode is required.

gpr_$resource_type_t

Used by gpr_$initialize and gpr_$inq_pixel_formats to identify the type of display resource in which the application will run. This data type takes the place of the display modes used for gpr_$init. This is a predefined enumerated type in Pascal and C. FORTRAN does not support enumerated types, but you can simulate this type by declaring an INTEGER*2 variable. Possible values are:

gpr_$memory_bitmap

A main memory bitmap. This has no effect on the screen.

gpr_$pad_id

A display-manager window

gpr_$pad_frame_id

A display-manager frame

gpr_$screen
The entire screen

gpr_$x_window_id
An X window

gpr_$rm_id
Reserved for future use

gpr_$init_options_t
Initialization options. This is a predefined enumerated type in Pascal and C. FORTRAN does not support enumerated types, but you can simulate this type by declaring an INTEGER*4 variable. Possible values are:

gpr_$no_clear
Do not clear the screen. This may be used in conjunction with a display resource type of gpr_$screen.

gpr_$init_options_set_t
This is a predefined set of gpr_$init_options_t type in Pascal. C and FORTRAN do not support set types, but you can simulate this type by declaring a long int variable in C or an INTEGER*4 in FORTRAN. This set has 32 members. This data type specifies the set of options for the gpr_$initialization routine. At the present time, unless you want the gpr_$no_clear option, you should set this to a null value.

gpr_$pixel_format_t
This is a predefined record type in Pascal and a predefined structure type in C. FORTRAN does not support record/structure types, but you can simulate this type by declaring a 16-element array of INTEGER*4. This type describes the pixel format of a bitmap. The table below shows the fields of the gpr_$pixel_format data type.

Name of Field	Data Type	Element # in FTN Array
length	integer32	1
pixel_mode	integer32	2
image_depth	integer32	3
buffer_count	integer32	4
red_depth	integer32	5
green_depth	integer32	6
blue_depth	integer32	7
ovlay_mode	integer32	8
ovlay_depth	integer32	9
ovlay_buffer_count	integer32	10
z_mode	integer32	11
z_depth	integer32	12
z_buffer_count	integer32	13
alpha_mode	integer32	14
alpha_depth	integer32	15
alpha_buffer_count	integer32	16

Although you do not need to set all of the fields, you must set the fields sequentially, e.g., you cannot set alpha_buffer_count without setting all of the fields that precede it.

Description of each field:

length

The number of members in this record (excluding this field) that you set. The purpose of this field is to allow the number of fields to grow in the future without breaking existing code.

pixel_mode

The type of image. Presently available types are:

- gpr_$pixel_pseudocolor
- gpr_$pixel_truecolor

image_depth

Number of planes in the image (red, green, and blue planes or pseudo color planes)

buffer_count

Number of image buffers

red_depth

Number of red planes for true color.

green_depth

Number of green planes for true color.

blue_depth

Number of blue planes for true color.

ovlay_mode

Type of overlay. Available types are:

- gpr_$ovlay_none – no overlays
- gpr_$ovlay_per_buffer – one overlay projection per buffer

ovlay_depth

Number of overlay planes

ovlay_buffer_count

Number of overlay buffers. Specify 1 for single buffer, 2 for double buffer.

z_mode

Type of z buffer. Available types are:

- gpr_$z_none – no z planes
- gpr_$z_per_buffer – one z projection per buffer

z_depth

Number of z planes

z_buffer_count

Number of z buffers. Specify 1 for single buffer, 2 for double buffer.

alpha_mode

Type of alpha buffer. Available types are:

- gpr_$alpha_none – no alpha planes
- gpr_$alpha_per_buffer – one alpha projection per buffer

alpha_depth

Number of alpha planes

alpha_buffer_count

Number of alpha buffers. Specify 1 for single buffer, 2 for double buffer.

The available pixel formats for the DN3550B and DN4500B are listed below.

gpr_$pixel_format_array_t

An array of up to gpr_$max_formats elements. Each element of the array has the data type gpr_$pixel_format_t.

gpr_$pixel_format_array_ptr_t

A pointer to an array of data type of gpr_$pixel_format_array_t.

gpr_$proj_format_t

This is a predefined record type in Pascal and a predefined structure type in C. FORTRAN does not support record/structure types, but you can simulate this type by declaring a 4-element array of INTEGER*4. This type specifies a projection for a bitmap. A projection is a set of planes that can be grouped together to form a display bitmap. The table below shows the fields of the gpr_$proj_format data type.

Name of Field	Data Type	Element # in FTN Array
length	integer32	1
proj_mode	integer32	2
proj_buffer	integer32	3
reserved	integer32	4

Description of each field:

length

The number of fields in this record (excluding this field). The purpose of this field is to allow the number of fields to grow in the future without breaking existing code.

proj_mode

The bitmap projection. The following projection modes are available:

- gpr_$proj_mode_argb – the alpha, red, green, and blue planes
- gpr_$proj_mode_ovlay – the overlay plane(s)
- gpr_$proj_mode_zzzz – the z planes

proj_buffer

The number of the buffer being used for the projection. This can be 1 or 2 for a device with double buffering. For a device that does not offer double buffering, this can only be 1.

gpr_$video_format_t

This is a predefined record type in Pascal and a predefined structure type in C. FORTRAN does not support record/structure types, but you can simulate this type by declaring a 2-element array of INTEGER*4. This type determines which buffer of a bitmap is displayed. The table below shows the fields of the gpr_$video_format data type.

Name of Field	Data Type	Element # in FTN /
length	integer32	1
video_buffer	integer32	2

Description of each field:

length

The number of fields in this record (excluding this field). The purpose of this field is to allow the number of fields to grow in the future without breaking existing code.

video_buffer

The buffer being displayed. This can be 1 or 2 for a device with double buffering. For a device that does not offer double buffering, this can only be 1.

gpr_$16bit_character_array_t

This is a predefined array of unsigned 16-bit integers. It is used by gpr_$text16, gpr_$inq_text16_extent, and gpr_$inq_text16_offset to specify an array of 16-bit characters.

gpr_$cursor_mode_t

Used by gpr_$set_cursor_mode and gpr_$inq_cursor_mode to indicate the cursor mode. This is a predefined enumerated type in Pascal and C. FORTRAN does not support enumerated types, but you can simulate this type by declaring an INTEGER*2 variable. Possible values are:

gpr_$software_cursor

Software cursor mode

gpr_$hardware_cursor

Hardware cursor mode

gpr_$cursor_mode_set_t

Used to declare a new element (avail_cursor_modes) in gpr_$disp_char_t. This is predefined set of gpr_$cursor_mode_t in Pascal. C and FORTRAN do not support set types, but you can simulate this type by declaring a short int variable in C and an INTEGER*2 variable in FORTRAN. It specifies a set of cursor modes.

gpr_$disp_char_t

The following new fields have been added to support cursor modes:

avail_cursor_modes

Indicates the available cursor modes for the devices. This field is of type gpr_$cursor_mode_set_t in Pascal, a short int variable in C, and an INTEGER*2 variable in FORTRAN.

default_cursor_mode

Indicates the device's default cursor mode. This field is of type gpr_$cursor_mode_t in Pascal and C and an INTEGER*2 in FORTRAN.

NAME

gpr_$set_quit_event – defines the event which causes a quit to be sent to the process.

FORMAT

gpr_$set_quit_event (*event_type*, *code*, *status*)

DESCRIPTION

This routine sets the quit character event for a GPR application. It should be used in place of smd_$set_quit_char. This routine takes a GPR event and a keyset value as arguments.

Input Parameters

event_type

The event type, in gpr_$event_t format. Possible values are:

- gpr_$keystroke
- gpr_$coded_keys
- gpr_$function_keys
- gpr_$physical_keys

code

The keystroke or key code

Output Parameters

status

Completion status, in status_$t format.

NOTES

The default quit character is CTRL/Q.

The quit character event does not have to be enabled by gpr_$enable_events. If enabled, it will still generate a quit fault and NOT be delivered to the application.

NAME

gpr_$set_icon_opt – sets option for acquire-display behavior when window is an icon.

FORMAT

gpr_$set_icon_opt (*icon_opt*, *status*)

DESCRIPTION

Normally, when a program running in an iconized window tries to acquire the display, GPR blocks execution of the program. GPR does not return from gpr_$acquire_display, and execution of the application is suspended until the icon returns to the window state. Since gpr_$event_wait and gpr_$cond_event_wait both do an implicit acquire/release, GPR also blocks execution of applications waiting for events when they are in an iconized window. This routine gives applications the option of continuing execution when the window is iconized by the Display Manager.

Input Parameters

icon_opt

The option to block/not block execution of the program when the window is an icon, in gpr_$icon_opt_t format. Possible values are:

- gpr_$ok_if_icon
- gpr_$block_if_icon

Output Parameters

status

Completion status, in status_$t format.

NOTES

The purpose of this routine is to allow a program with multiple windows to continue executing when one window is an icon.

NAME

gpr_$initialize – initializes GPR.

FORMAT

gpr_$initialize (*resource_type*, *resource_id*, *options*, *size*,
pix_format, *proj_format*, *video_format*,
init_bitmap, *status*)

DESCRIPTION

This routine initializes the graphics primitives package, allocates and sets an initial bitmap, and sets the pixel format, projection format, and video format.

Input Parameters

resource_type

The display resource type, in gpr_$resource_type_t format. Possible values for this paramter are:

gpr_$memory_bitmap
A main memory bitmap. This has no effect on the screen.

gpr_$pad_id
A display-manager window

gpr_$pad_frame_id
A display-manager frame

gpr_$screen
The entire screen

gpr_$x_window_id
An X window

gpr_$rm_id
Reserved for future use

resource_id

The resource identifier. This is a 4-byte integer. This value can have the following meanings:

- The display unit, if the display resource type is gpr_$screen. This is a 4-byte integer. Currently, the only valid display ID for gpr_$screen is 1.
- The stream identifier for the pad or X window, if the display resource is gpr_$pad_id or gpr_$x_window_id.

- Any value, such as zero, if the display resource is gpr_$memory_bitmap.

options

The intialization options, in gpr_$init_options_set_t format. This is a predefined set of gpr_$init_options_t type in Pascal. C and FORTRAN do not support set types, but you can simulate this type by declaring a long int variable in C or an INTEGER*4 in FORTRAN. This set has 32 members. At the present time, the only initialization option that has been implemented is **gpr_$no_clear**, which specifies not to clear the screen. This may be used in conjunction with a display resource type of gpr_$screen. Unless you want the gpr_$noclear option, you should set this to a null value.

Input/Output Parameters

size

The size of the initial bitmap (or the size of the frame, if the resource type is gpr_$pad_frame_id) in gpr_$offset_t format.

If the display resource is gpr_$screen, you must set both dimensions of size to an integer between 1 and 8192 inclusive. If you provide bitmap dimensions smaller than the display memory of the node you are using, the size of the bitmap will match the dimensions you provide. If, however, you provide dimensions larger than the size of the display memory, the system will reduce the size of the initial bitmap to match the size of the display memory on your node. The origin of the bitmap is the top left corner of the screen.

If the display resource is gpr_$pad_id or gpr_$x_window_id, you must set both dimensions of size to integers between 1 and 8192 inclusive. If you provide dimensions larger than the current display window, the system sets the size of the bitmap equal to the current size of the display window. However, if you grow the display window, the bitmap will also grow, but cannot grow past the dimensions you specified. The origin is the top left corner of the display window.

If the display resource is gpr_$pad_frame_id, you must set both dimensions of size to integers between 1 and 32767 inclusive. For this display resource, size specifies the size of both the frame and the initial bitmap.

If the display resource is gpr_$memory_bitmap, you must set both dimensions of size to integers between 1 and 8192 inclusive. The size that the system allocates will equal the size that you specify.

pix_format

The pixel format, in gpr_$pixel_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The pixel format describes the configuration of the planes of a pixel.

proj_format

The projection format for the initial bitmap, in gpr_$proj_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The projection format specifies which set of planes will compose the initial bitmap.

video_format

The video format, in gpr_$video_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The video format specifies which buffer will be displayed.

Output Parameters

init_bitmap

The descriptor of the initial bitmap, in gpr_$bitmap_desc_t format. This is a 4-byte integer that uniquely identifies the bitmap. If the resource type is gpr_$pad_id, gpr_$pad_frame_id, gpr_$screen_id, or gpr_$x_window_id, the bitmap will be a display bitmap. This bitmap consists of the group of planes specified in the projection format. Since devices with fewer than 24 planes and main memory bitmaps have only one projection.

status

Completion status, in status_$t format.

NOTES

This call takes the place of gpr_$init in an application program. Programs may continue to use gpr_$init, but only if they do not require access to the capabilities made available through gpr_$initialize.

The display resource is the equivalent of the display mode used in gpr_$init, except that it does not specify true or pseudo color.

It is not necessary to set all of the members of the pixel format record. At a minimum, you must set the pixel_mode, the pixel_depth, and length = 2, and GPR will fill in the rest of fields for you.

The default projection format is the first buffer image planes. To get the default projection, set proj_format.length = 0.

The default video format is the first buffer. To get the default video format, set video_format.length = 0.

To find out what pixel formats your device supports, call gpr_$inq_pixel_formats.

To find out the characteristics of your device to use in your initialization, call gpr_$inq_display_characteristics.

If your program uses multiple windows, you must call gpr_$initialize for each window that uses GPR calls.

This routine initializes GPR, so that you can use the GPR routines. The only GPR calls you can make before gpr_$initialize are gpr_$inq_pixel_formats and gpr_$inq_display_characteristics.

For main memory bitmaps, there is no need to set the projection and video formats. Just specify a length of 0 for each.

For main memory bitmaps, the only fields of the pixel format that need to be set are the pixel mode, image depth, and red, green and blue depths for true color.

NAME

gpr_$inq_pixel_formats – returns the pixel formats available on the device.

FORMAT

gpr_$inq_pixel_formats (*resource_type*, *resource_id*,
max_formats, *format_size*,
num_formats, *formats*, *status*)

DESCRIPTION

This routine returns the pixel formats available on the device.

Input Parameters

resource_type

The display resource type, in gpr_$resource_type_t format. Possible values for this parameter are:

gpr_$memory_bitmap

A main memory bitmap. This has no effect on the screen.

gpr_$pad_id

A display-manager window

gpr_$pad_frame_id

A display-manager frame

gpr_$screen

The entire screen

gpr_$x_window_id

An X window

gpr_$rm_id

Reserved for future use

resource_id

The resource identifier. This value can have the following meanings:

- The display unit, if the display resource type is gpr_$screen. This is a 2-byte integer. Currently, the only valid display ID for gpr_$screen is 1.
- The stream identifier for the pad or X window, if the display resource is gpr_$pad_id or gpr_$x_window_id. Use stream_$id_t format. This is a 2-byte integer.
- Any value, such as zero, if the display resource is gpr_$memory_bitmap.

max_formats

The maximum number of formats requested. This is a 4-byte integer. The maximum number of formats available is defined in gpr_$max_formats.

format_size

The size in bytes of the format record. For example, if the buffer is ten 16-bit words in length, the program gives 20 as the value of this parameter. No checking is (or can be) done to verify that this length is correct; so unpredictable results are obtained if the program gives a size that is larger than the actual size of the buffer. This parameter allows the calling program to request that less than the full set of characteristics be returned. It also allows the program to continue to function correctly if the list is extended in the future. To get the size of the buffer, use the 'sizeof' function in C and Pascal. In FORTRAN, use the constant gpr_$pixel_format_size, which has been defined in gpr.ins.ftn, to get this value. This parameter is a 2-byte integer.

Output Parameters

num_formats

The number of formats returned in the pixel format array.

formats

An array of supported pixel formats for this device, in gpr_$pixel_format_array_t format. This is an array of gpr_$pixel_format_t.

status

Completion status, in status_$t format.

NOTES

Use this call to find a valid pixel format for gpr_$initialize and gpr_$inq_pixel_characteristics.

The display resource is the equivalent of the display mode used in gpr_$init, except that it does not specify true or pseudo color.

NAME

gpr_$inq_display_characteristics – returns information about the device or the external bitmap.

FORMAT

gpr_$inq_display_characteristics (*resource_type*, *resource_id*, *disp_len*, *disp*, *disp_len_returned*, *status*)

DESCRIPTION

This routine returns a variety of information about the nature of the actual display device or the external bitmap if a memory resource is specified.

Input Parameters

resource_type

The display resource type, in gpr_$resource_type_t format. Possible values for this paramter are:

gpr_$memory_bitmap

A main memory bitmap. This has no effect on the screen.

gpr_$pad_id

A display-manager window

gpr_$pad_frame_id

A display-manager frame

gpr_$screen

The entire screen

gpr_$x_window_id

An X window

gpr_$rm_id

Reserved for future use

resource_id

The resource identifier. This value can have the following meanings:

- The display unit, if the display resource type is gpr_$screen. This is a 2-byte integer. Currently, the only valid display ID for gpr_$screen is 1.
- The stream identifier for the pad or X window, if the display resource is gpr_$pad_id or gpr_$x_window_id. Use stream_$id_t format. This is a 2-byte integer.
- Any value, such as zero, if the display resource is gpr_$memory_bitmap.

disp_len

The size of the buffer (the disp parameter described below) in bytes provided by the calling program, which will contain the returned display or device information in bytes. For example, if the buffer is ten 16-bit words in length, the program gives 20 as the value of this parameter. No checking is (or can be) done to verify that this length is correct, so unpredictable results are obtained if the program gives a size that is larger than the actual size of the buffer. This parameter allows the calling program to request that less than the full set of characteristics be returned. It also allows the program to continue to function correctly if the list is extended in the future. To get the size of the buffer, use the been defined in gpr.ins.ftn, to get this value. This parameter is a 2-byte integer.

Output Parameters

disp

Returned display device characteristics in gpr_$disp_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN.

disp_len_returned

Actual number of bytes of data returned in the disp parameter. This is a 2-byte integer. It will always be less than or equal to the disp_len input parameter value.

status

Completion status, in status_$t format.

NOTES

This call takes the place of gpr_$inq_disp_characteristics.

The display resource is the equivalent of the display mode used in gpr_$inq_disp_characteristics, except that it does not specify true or pseudo color.

Use gpr_$inq_display_characteristics to determine your node's characteristics as it runs in a specified display resource. The characteristics include important information such as the size of the display screen and the number of planes. The call returns the characteristics in the disp parameter.

You can call gpr_$inq_display_characteristics at any time in the program. In fact, it is good programming practice to call gpr_$inq_display_characteristics prior to calling gpr_$initialize. By doing so, gpr_$inq_display_characteristics will return values (such as bitmap size and hi_plane_id) that you can use when you call gpr_$initialize. In the future, we may extend the list of data items returned into disp as we release new display devices with new characteristics. However, programs written to use the existing set of characteristics will still continue to operate correctly. Note that enumerated and set fields within disp will probably be extended in future releases.

Note that calling gpr_$inq_display_characteristics after gpr_$initialize has no effect on the current bitmap or its attributes. gpr_$inq_display_characteristics is a purely descriptive call.

Note that gpr_$inq_display_characteristics returns information reflecting the status of the display when the call was made. Therefore, if you call gpr_$inq_display_characteristics prior to acquiring the display, the returned information may not accurately reflect the future state of the window (since the window could move, change size, or be obscured.

NAME

gpr_$inq_bitmap_pixel_format – returns the pixel format for the specified bitmap.

FORMAT

gpr_$inq_bitmap_pixel_format (*bitmap*, *pixform*, *status*)

DESCRIPTION

This routine returns the pixel format for the specified bitmap.

Input Parameters

bitmap

The descriptor of the bitmap, in gpr_$bitmap_desc_t format. This is a 4-byte integer that uniquely identifies the bitmap.

Output Parameters

pixform

A pointer to the pixel format, which is in gpr_$pixel_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The pixel format describes the configuration of the planes of a pixel.

status

Completion status, in status_$t format.

NOTES

This call returns the entire pixel format, regardless of which projection a display bitmap uses.

To set the pixel format, call gpr_$initialize.

NAME

gpr_$inq_bitmap_proj_format – returns the projection format for the specified bitmap.

FORMAT

gpr_$inq_bitmap_proj_format (*bitmap*, *projform*, *status*)

DESCRIPTION

This routine returns the projection format for the specified bitmap.

Input Parameters

bitmap

The descriptor of the bitmap, in gpr_$bitmap_desc_t format. This is a 4-byte integer that uniquely identifies the bitmap.

Output Parameters

projform

A pointer to the projection format, which is in gpr_$proj_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The projection format specifies which set of planes of the bitmap is active.

status

Completion status, in status_$t format.

NOTES

Use this call to find the current projection format for the display bitmap.

The concept of projections is only applicable to display bitmaps.

To set the projection, call gpr_$initialize.

To change the projection, call gpr_$allocate_projection.

NAME

gpr_$inq_bitmap_video_format – returns the video format for the bitmap.

FORMAT

gpr_$inq_bitmap_video_format (*bitmap*, *vidform*, *status*)

DESCRIPTION

This routine returns the video format for the specified display bitmap.

Input Parameters

bitmap

The descriptor of the bitmap, in gpr_$bitmap_desc_t format.

Output Parameters

vidform

A pointer to the video format, which is in gpr_$video_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The video format specifies which buffer will be displayed.

status

Completion status, in status_$t format.

NOTES

The video format is set in gpr_$initialize. The default format is video_buffer = 1.

Since the video format is associated with the window rather than the bitmap, any bitmap descriptor associated with a window will return the correct video format for that window.

This routines only works for display bitmaps.

NAME

gpr_$allocate_projection – allocates a new projection for an existing bitmap.

FORMAT

gpr_$allocate_projection (*old_projection,options, proj_format, new_projection, status*)

DESCRIPTION

This routine allocates a bitmap for the specified projection of an existing display bitmap. This projection consists of a subset of the planes available in the pixel format.

Input Parameters

old_projection

The bitmap descriptor of an existing projection bitmap for this window, in gpr_$bitmap_desc_t format. Use the bitmap descriptor returned by gpr_$initialize.

options

The intialization options, in gpr_$init_options_set_t format. This is a predefined set of gpr_$init_options_t type in Pascal. C and FORTRAN do not support set types, but you can simulate this type by declaring a long int variable in C or an INTEGER*4 in FORTRAN. This set has 32 members. At the present time, the only initialization option that has been implemented is **gpr_$no_clear**, which specifies not to clear the screen. This may be used in conjunction with a display resource type of gpr_$screen. Unless you want the gpr_$noclear option, you should set this to a null value.

proj_format

A pointer to the projection format for the output bitmap, which is in gpr_$proj_format_t format. This data type is a record in Pascal, a structure in C, or an array in FORTRAN. The projection format specifies which set of planes in the bitmap will be allocated.

Output Parameters

new_projection

The descriptor for the new bitmap, in gpr_$bitmap_desc_t format. This is a 4-byte integer that uniquely identifies the bitmap.

status

Completion status, in status_$t format.

NOTES

Use gpr_$set_bitmap to establish an allocated bitmap as the current bitmap.

This routine is only valid for display bitmaps.

Applications will probably only want to do direct access in the Z buffer.

NAME

gpr_$text16 – writes text consisting of 16-bit characters.

FORMAT

gpr_$text16 (*t_array*, *t_arrayl*, *status*)

DESCRIPTION

This routine writes text consisting of 16-bit characters to the current bitmap, beginning at the current position. The text is written in the current font.

Input Parameters

t_array

The array of 16-bit characters to write, in gpr_$16bit_character_array_t format. This is an array of unsigned 2-byte integers.

t_arrayl

The number of 16-bit characters. This is a 2-byte integer.

status

Completion status, in status_$t format.

NOTES

This routine is not supported in frame mode. gpr_$text16 always clips to the edge of the bitmap, regardless of whether clipping is enabled.

The gpr_$text16 routine writes the 16-bit characters in the current font that correspond to the values of the characters in the specified 16-bit character array. If the font does not have a character that corresponds to a character in the array, gpr_$text16 leaves a space.

The font sets the size of this space. To change the size of the space left for unsupported characters, call gpr_$set_space_size. Note that this is not the size of the space character ' ' (0x20) if it is present in the font.

Text is written at the current position. The origin of the first character of the 16-bit character array is placed at the current position. Generally, the origin of the character is at the bottom left, excluding descenders of the character.

Upon completion of the gpr_$text16 routine, the current position is updated to the coordinate position where the next character would be written. This is true even if the text array is partly or completely clipped. However, the current position always remains within the boundaries of the bitmap.

Note that gpr_$text16 can only print 16-bit character arrays. If you want other kinds of data (e.g., numbers) printed, you must convert the data to a 16-bit character array before calling gpr_$text16.

NAME

gpr_$inq_character16_width – returns the width of a 16-bit character.

FORMAT

gpr_$inq_character16_width (*font_id*, *character*, *width*, *status*)

DESCRIPTION

This routine returns the width of the specified 16-bit character in the specified font.

Input Parameters

font_id

Identifier of the font. This is a 2-byte integer.

character

The specified 16-bit character. This is an unsigned 2-byte integer.

Output Parameters

width

The width (in pixels) of the specified 16-bit character. The width is the distance from the origin of the specified character to the following current position. This is a 2-byte integer.

status

Completion status, in status_$t format.

NOTES

To set a 16-bit character's width, use gpr_$set_character16_width.

The initial character widths are defined in the font file.

This routine returns the 16-bit character width in the local copy of the font. Initially, this is a copy of the font file; but the local copy may have been changed. Change in the local copy does not affect the font file or the use of the font by other processes.

If the specified character is not in the font, GPR returns a status of gpr_$character_not_in_font.

NAME

gpr_$set_character16_width – specifies the width of a 16-bit character.

FORMAT

gpr_$set_character16_width (*font_id*, *character*, *width*, *status*)

DESCRIPTION

This routine sets the width of the specified 16-bit character in the specified modifiable font.

Input Parameters

font_id

Identifier of the font. This is a 2-byte integer.

character

The specified 16-bit character. This is an unsigned 2-byte integer.

width

The width (in pixels) of the specified 16-bit character. The width is the distance from the origin of the specified character to the following current position. This is a 2-byte integer.

Output Parameters

status

Completion status, in status_$t format.

NOTES

To retrieve a 16-bit character's width, use gpr_$inq_character16_width.

The initial character widths are defined in the font file.

Negative widths are not supported.

To use routines that change fonts, you must first call gpr_$replicate_font to create a modifiable copy of a font. Font-modifying routines include gpr_$set_character_width, gpr_$set_horizontal_spacing, and gpr_$set_space_size, as well as gpr_$set_character16_width. These calls change only the local copy of the font. If you unload a font and reload it, the font is reset to the values in the font file.

If the specified character is not in the font, GPR returns a status of gpr_$character_not_in_font.

NAME

gpr_$inq_text16_extent – returns the extent of an array of 16-bit characters.

FORMAT

gpr_$inq_text16_extent (*t_array*, *t_arrayl*, *size*, *status*)

DESCRIPTION

This routine returns the x- and y-offsets that the specified array of 16-bit characters would span if written with gpr_$text16.

Input Parameters

t_array

The array of 16-bit characters to inquire about, in gpr_$16bit_character_array_t format. This is an array of unsigned 2-byte integers.

t_arrayl

The number of 16-bit characters in the array. This is a 2-byte integer.

Output Parameters

size

The width and height the written character array will occupy, in gpr_$offset_t format. This data type is 4 bytes long.

status

Completion status, in status_$t format.

NOTES

When the text path is gpr_$right or gpr_$left, the width is the x-offset. When the text path is gpr_$up or gpr_$down, the width is the y-offset.

To change the direction of text, use gpr_$set_text_path.

NAME

gpr_$inq_text16_offset – returns the x- and y-offsets of an array of 16-bit characters.

FORMAT

gpr_$inq_text16_offset (*t_array*, *t_arrayl*, *start*, *xy_end*, *status*)

DESCRIPTION

This routine returns the following offsets:

- The x- and y-offsets from the top left pixel of an array of 16-bit characters to the pixel that is the new current position after the array is written with gpr_$text16.
- The x- or y-offset from the top left pixel of an array of 16-bit characters to the origin of the array's first character.

Input Parameters

t_array

The array of 16-bit characters to inquire about, in gpr_$16bit_character_array_t format. This is an array of unsigned 2-byte integers.

t_arrayl

The number of 16-bit characters in the array. This is a 2-byte integer.

Output Parameters

start

The x- and y-offsets from the top left pixel of the character array to the origin of its first character, in gpr_$offset_t format. This data type is 4 bytes long.

xy_end

The x- or y-offset from the top left pixel of the character array to the pixel that will be the new current position after the array is written with gpr_$text16. This is the x-offset when the text path is specified as gpr_$right or gpr_$left. This is the y-offset when the text path is specified as gpr_$up or gpr_$down. This is a 2-byte integer.

status

Completion status, in status_$t format.

NOTES

A program can use the information derived from the "start" output parameter to set the current position to the character origin, rather than the top left corner of the array, before writing the array with gpr_$text16.

See gpr_$set_text_path for use of gpr_$right, gpr_$left, gpr_$up and gpr_$down.

NAME

gpr_$set_cursor_mode – Sets the cursor mode to software or hardware mode.

FORMAT

gpr_$set_cursor_mode (*mode*, *status*)

DESCRIPTION

This routine sets the cursor mode for the current bitmap and becomes the default for all bitmaps subsequently created.

Input Parameters

mode

The new cursor mode, in gpr_$cursor_mode_t format. Possible values are:

gpr_$software_cursor

Use gpr_$software_cursor to specify software cursor mode.

gpr_$hardware_cursor

Use gpr_$hardware_cursor specify hardware cursor mode.

Output Parameters

status

Completion status, in status_$t format.

NOTES

Although the hardware cursor is always active, an application can make the cursor invisible by calling gpr_$set_cursor_pattern with the bits in the cursor bitmap all set to zero.

gpr_$set_cursor_mode (gpr_$hardware_cursor) does an implicit gpr_$set_cursor_active(true). Thus, if you switch from hardware to software cursor mode, the cursor will still be visible.

EXAMPLE

This program requires the header file "my_include_file.c", which can be found in the same directory as this example.

```
/* Name of Program -- nondestructive_cursor */

/* NOTE: Before compiling this program, you must obtain 'my_include_file' and */
/*      store it in filename 'my_include_file.c'                        */

/* This program shows how you can display a nondestructive cursor wherever the
  mouse tracks.  It uses a combination of cursor and event calls to
  demonstrate this feature.  Enter <CTRL-Q> to exit from the program.
*/
#include <apollo/base.h>
#include <apollo/gpr.h>
#include "my_include_file.c" /*Contains the init, check, and pause routines.*/

gpr_$bitmap_desc_t  cursor_bitmap_descriptor;

/****************************************************************************/

main(void)
{

gpr_$event_t          event_type;
gpr_$position_t        mouse_position;
unsigned char          event_data;
gpr_$keyset_t         key_set;
gpr_$cursor_mode_t      cursor_mode = gpr_$hardware_cursor;
static gpr_$position_t   cursor_origin = {8,1};

  init(gpr_$direct);

  gpr_$set_cursor_mode (cursor_mode,&status);
     check("setting cursor mode");

  gpr_$acquire_display (&status);

  gpr_$set_cursor_origin(cursor_origin, &status);

/*Activate the locator.*/
  event_type = gpr_$locator;
  gpr_$enable_input (event_type, key_set, &status);
  while(1)
   {
   gpr_$event_wait(&event_type, &event_data, &mouse_position, &status);
   if (event_type == gpr_$locator)
     {
     gpr_$set_cursor_position(mouse_position, &status);

    /* Note that you can draw in the cursor region and the graphics
        will still be there when you move the cursor.
     */
     gpr_$circle(mouse_position, 1, &status);
         check("drawing circle");
     }
   }

/*Terminate the graphics package.*/
  gpr_$release_display (&status);
  gpr_$terminate(false, &status);
}
```

NAME

gpr_$inq_cursor_mode – Retrieves the current cursor mode.

FORMAT

gpr_$inq_cursor_mode (*mode*, *status*)

DESCRIPTION

This routine retrieves the cursor mode for the current bitmap.

Output Parameters

mode

The cursor mode, in gpr_$cursor_mode_t format. Possible values are:

gpr_$software_cursor

Specifies software cursor mode.

gpr_$hardware_cursor

Specifies hardware cursor mode.

status

Completion status, in status_$t format.

SEE ALSO

gpr_$set_cursor_mode

NAME

gpr_$remap_pixels – Remaps the display bitmap to pixel mode.

FORMAT

gpr_$remap_pixels (*section_number*, *group_header*, *status*))

DESCRIPTION

This routine changes the default memory mapping of a display or HDM bitmap from plane to pixel mode. Pixel mode access is currently available on the DN3500A, DN3550A/B, DN4500A/B, and DN10000VS.

Input Parameters

section_number

The number of the section to map. A *section* is a single component of the image. For example, if you specify a pixel_size of one for an eight-plane bitmap, the bitmap will have eight sections, numbered 0 - 7. If you specify a pixel_size of eight for a 24-plane bitmap, it will have three sections, numbered 0 - 2. Most of the time, you will want to set the section_number to zero. If you specify a nonexistent section, GPR will not return an error and will use a default of zero. This is a 2-byte integer.

Input/Output Parameters

group_header

This is a description of the bitmap, in gpr_$bmf_group_header_t format. The fields in the group header are:

n_sects

2-byte integer giving the number of sections in the group. GPR fills in this value for you. Image planes are the planes used for specifying the color.

pixel_size

2-byte integer representing the number of bits in the image buffer of each pixel in the group. If you set this to zero, you will get the default, which is a pixel_size of one (plane mode).

allocated_size

2-byte integer. The number of bits per pixel allocated. GPR fills in this value for you.

bytes_per_line

Number of bytes per row of bitmap (counting any padding at ends of scanlines). GPR fills in this value for you.

bytes_per_sect

Number of bytes per section. GPR fills in this value for you. If n_sects is zero, this value may also be zero.

storage_offset

UNIV_PTR to the beginning of the mapped section of the bitmap. GPR fills in this value for you.

Output Parameters

status

Completion status, in status_$t format.

NOTES

gpr_$remap_pixels allows direct access only to display and HDM bitmaps.

In order to use gpr_$remap_pixels, either the display or the HDM bitmap must be current. Use gpr_$set_bitmap to set the current bitmap.

You may wish to take the cursor down before reading or writing directly to the bitmap.

Use either the storage offset from the group header or the pointer returned from gpr_$inq_bitmap_pointer to get a pointer for writing directly to the bitmap. If allocated_size is eight, this pointer points to an array of eight-bit bytes. If allocated_size is 32, this pointer points to an array of 32-bit longwords.

If your allocated_size equals one (plane access), you must use gpr_$inq_bm_offset to get the bit offset for the bitmap.

gpr_$wrong_display_hardware can result if you attempt to map the display with a pixel size or section number it does not support.

For this release, if you specify a pixel_size of 32, the highest eight bits will be unpredictable.

When you enable direct access, you have access to the entire screen, even when you are in direct display mode. Thus, you can write outside the window. Use gpr_$inq_bitmap_dimensions and gpr_$inq_vis_list to determine the area on the screen where you application should draw. It is the application's responsibility to keep its drawing in the window. On the DN10000VS, you can use gpr_$set_clip_window and gpr_$set_clipping_active to keep your drawing within the window, but the clip rectangle mechanism does not work for direct access on other nodes.

Even though you can write to the entire screen, the origin of the coordinate system is still at the upper left corner of the window.

You can move easily between direct access and GPR drawing calls. However, any GPR drawing call (e.g., gpr_$line) automatically makes the memory mapping of the bitmap invalid. To use direct access again, a program must call gpr_$remap_pixels after the GPR drawing calls and before using the pointer from storage_offset or gpr_$inq_bitmap_pointer. Failure to call gpr_$remap_pixels before writing to the bitmap is a common cause of strange graphical results in programs using direct access.

gpr_$remap_pixels enables direct access for the current bitmap only. If you want to use direct access with another bitmap, you must call gpr_$remap_pixels again after the new bitmap becomes active.