SANYO
ICON

ICON/UXV™ Release 4.00

for M68k-based ICON Computers

Release Notes

SANYO/ICON

764 East Timpanogos Parkway
Orem, Utah 84057-6212
Telephone: 801 225-6888
Fax: 801 226-0651
Telex: 323938 ICONSYS

RELEASE NOTES

ICON/UXV™ Operating System

for M68k-based ICON Computers

Release 4.00

SANYO/ICON

764 East Timpanogos Parkway
Orem, Utah 84057-6212
Telephone: Corporate: 1-800-US-SANYO (877-2696)
Customer Service: 1-800-SANYO-US (726-9687)
Fax: 801 226-0651
Telex: 323938 ICONSYS

ICON/UXV 4.00 Release Notes
For M68k-based ICON Computers
Edition A
August 1990

Order No. **169-043-001**

Additional copies of this manual may be ordered by addressing your correspondence to the Sanyo/Icon corporate address above. Include the name of the manual and the order number.

Trademarks

Table of Contents

Chapter 1: Introduction

Introduction

These *Release Notes* contain important information about ICON/UXV Release 4.00 for M68k-based computers. ICON/UXV is Sanyo/Icon's version of UNIX® System V Release 3.2.

First, these *Release Notes* briefly describe the new features of this release. Then they give step-by-step instructions on how to install the release. The installation instructions explain how to determine whether or not the hardware on your ICON computer needs to be upgraded. They also explain how to back up all files on the system before doing the actual installation. Software notes, additional information about upgrading and installation, and compatibility notes are also provided.

For a complete description of the procedures used in the administration of an ICON computer running ICON/UXV Release 4.00, see the *System Administrator's Guide.*

Conventions Used in these Release Notes

In this document certain typesetting conventions are followed when command names, command line formats, files, and directory names are described. There are also conventions for displays of terminal input and output.

- Words that appear in **bold** font should be typed as they appear.
- Words that appear in *italic* are substitutable elements of text and you should replace them with appropriate values. These values may be file names or they may be data values, as applicable.
- CRT or terminal output and examples of source code are presented in `constant-width` font.
- Characters or words in square brackets, [ ], are optional. (Do not type the brackets.) Uppercase characters inside arrows (< >) are keyboard keys to be pressed, not entered.

When a command name is followed by a number, for example, **ed**(1), the number refers to the section of the system reference manual in which the manual page for that command is found. Manual pages ("man pages") from section (1), (1M), (4), (5), (7), and (8) appear in the *System Administrator's Reference Manual,* unless otherwise noted. Section (1) commands appropriate for programmers and manual pages from sections (2), (3), (4), and (5) appear in the *Programmer's Reference Manual.*

Examples in these release notes show the default system prompt for ICON/UXV, the dollar sign ($). They also show the default prompt when you log in as the superuser, the pound sign (#).

Installing ICON/UXV 4.00

Because installing ICON/UXV Release 4.00 involves reformatting all the SCSI hard drives on your ICON computer except the HD180 (an HD180 or an HSMD drive will need to be repartitioned only if it serves as the boot device), the installation will erase all data on those drives: system files, application files, and user files. For this reason, you need to back up all file systems on SCSI drives as the first procedure of the installation. Instructions for backing up the file systems are provided. Before performing any of the installation procedures in Chapter 3, it is recommended that you read the section "Important Installation Information," which is also in Chapter 3.

Supporting Documentation

The books in the first list below come with your ICON computer. Those in the second list come when you order the ICON/UXV Software Development Documentation package. (PN 175-015-001).

Books for users and system administrators:

- ICON/UXV System Administrator's Guide (PN 172-071-001)
- UNIX System V Release 3.2 User's Guide (PN 176-080-001)
- UNIX System V Release 3.2 System Administrator's Reference Manual (PN 176-080-003). (Contains the user's and system administrator's reference manuals under one cover.)
- UNIX System V Release 3.2 Framed Access Command Environment (FACE) User's Guide (PN 176-080-002)

Books for programmers:

- UNIX System V Release 3.2 Programmer's Guide, Volume I (PN 176-080-004)
- UNIX System V Release 3.2 Programmer's Guide, Volume 2 (PN 176-080-005)
- UNIX System V Release 3.2 Programmer's Reference Manual (PN 176-080-006)
- UNIX System V Release 3.2 Form and Menu Language Interpreter (FMLI) Programmer's Guide (PN 176-080-007)

Chapter 2: Release 4.00 Features

Release 4.00 Features

Release 4.00 maintains source and object code compatibility across the ICON computer family. Application packages designed for 3.3*x* releases of ICON/UXV will continue to work with Release 4.00.

ICON/UXV Release 4.00 provides the following new features:

- Enhanced Basic Networking Utilities
- Command Syntax Standard
- Signal Mechanism Enhancements
- Improved Terminal Support Facilities
  - Terminal Information Utilities Enhancements
  - AT&T Windowing Utilities Package
- **help** Facility Extensions
- New System Header Files
- Support for Alternate Date and Time Formats (see **cftime**(4) and **environ** (5) in the *System Administrator's Reference Manual*)
- ASSIST Menu/Forms Interface
- New **awk** (**nawk**)
- Improved Recovery of Files from **cpio** Archives
- System Administration Menus
- Line Printer Spooling Utilities
- Framed Access Command Environment (FACE)
- Form and Menu Language Interpreter (FMLI)
- Enhanced **curses**
- Enhanced Security Features
  - **lastlogin** Time
  - **loginlog** File
  - Shadow Password File
  - Enhanced Shell
  - Flexible SCSI Addressing
  - New Disk Cache Code
  - Additional Berkeley Commands and Utilities and Command Extensions
  - Berkeley Fast File System
  - TCP/IP Networking Support
  - Berkeley Networking Tools

Command Syntax Standard

getopt(1) allows shell procedures to parse command lines to check for legal options and to process option arguments. A new command, **getopts**(1), is an enhanced version of the **getopt**(1) command. **getopts** is consistent with and supports rules 3-10 of the UNIX system command syntax standard. (The standard is described on the **intro**(1) manual page.)

You may use **getopt** in shell scripts. However, you should use **getopts** instead of **getopt**; in future releases, **getopt** may no longer be supported.

To assist in the conversion of shell scripts that are affected by a change from **getopt** to **getopts**, a conversion command, **/usr/lib/getoptcvt**, is provided. (See the **getopts** manual page for details.)

Signal Mechanism Enhancements

A new set of system calls (see the **sigset**(2) manual page in the *Programmer's Reference Manual*) provides a mechanism to catch and hold signals without losing them during later processing, and to guarantee that a process reaches the signal handler before it is interrupted by another signal. Some additional signal-handling features, provided by other popular operating systems, are also available.

Improved Facilities for Supporting Terminals

Support for terminals is improved with new features of the Terminal Information Utilities and the new AT&T Windowing Utilities.

Terminal Information Utilities

The "Terminal Information Utilities" package (often called **curses/terminfo**) has the following new features:

- expanded support for terminal filters, soft labels, and new AT&T terminals
- new commands: **captoinfo**(1M) converts **termcap** entries to **terminfo** entries; **infocmp**(1M) compares two **terminfo** entries or prints entries in several formats. (Section (1M) is in the *System Administrator's Reference Manual*.)
- new options to the **tput**(1) command to initialize, reset, and learn the "long name" of a terminal
- an improved version of the **terminfo** compiler, **tic**
- new documentation on the manual pages and in the "**curses** and **terminfo**" chapter of the *Programmer's Guide* (Chapter 10)

All programs that ran with System V Release 2 **curses** will run with System V Release 3.0. You may link applications with object files based on the Release 2 **curses/terminfo** with the Release 3.0 **libcurses.a** library. You may link applications with object files based on the Release 3.0 **curses/terminfo** with the Release 2 **libcurses.a** library, as long as the application does not use the new features in the Release 3.0 **curses/terminfo**.

help Facility Extensions

Descriptions and examples of many additional commands, terms, and symbols have been added. See **help**(1), **glossary**(1), **starter**(1), and **usage**(1).

New System Header Files

New header files were added to **/usr/include**: **unistd.h** (definitions for symbolic constants introduced and used throughout the /usr/group Standards document, see **unistd**(4) in the *Programmer's Reference Manual*) and **limits.h** (definitions for commonly used values that vary from implementation to implementation, see **limits**(4) in the *Programmer's Reference Manual*). Several new definitions were added to the header file **/usr/include/sys/stat.h** to make it easier for programmers to write portable code.

Support for Alternate Date and Time Formats

The **cpio**, **date**, **ls**, **mount**, **pr**, and **sort** commands were changed to provide the date and time in the language and format determined by the value of the **LANGUAGE** environment variable. While the United States conventions remain the default, other languages can be supported by creating and installing a file for the language desired in the **/lib/cftime** directory. The content of that file includes: month and weekday names (full and abbreviated), default local time, date, pre-noon, and post-noon formats, and the default output of the **date** command if the **CFTIME** environment variable is not set. In addition, time zones and alternate time zones (such as daylight time) now can be defined in terms of hours and minutes using the **TZ** environment variable.

For more information, see the **date**(1), **timezone**(4), and **environ**(5) manual pages in the *System Administrator's Reference Manual*.

ASSIST Utilities

The ASSIST Utilities package provides on-line assistance for UNIX System V users. It is designed to help new users get started using the UNIX system and to help experienced users explore UNIX system facilities that lie outside their areas of expertise. Rather than hiding the UNIX system, ASSIST exposes and explains it. **assist** does this through four components:

- Menus: show categories of activities a user does on a computer and list UNIX system commands for each of these activities
- Command Forms: construct a UNIX system command line at the bottom of the screen as users answer questions about what they want to do; then, optionally, execute the command line.
- Key-word Search: suggest appropriate UNIX system commands based on key words typed by the user
- Walkthrus: demonstrate, interactively, important concepts underlying the UNIX system and commands that have their own user interfaces, including how to use ASSIST

In addition to these four components, the **astgen** commands lets you add to or modify menus and command forms.

For more information, see the **assist**(1) and **astgen**(1) manual pages in the *System Administrator's Reference Manual*.

New awk

awk is a programming language for information retrieval and data manipulation that is often used by people with no programming background. Customers using **awk** in an international environment must use the new version (**awk**), because the old version (**oawk**) does not support 8-bit code sets. Some of the other enhancements to this version of **awk** include:

- the ability to define functions
- new keywords: **delete, do, func, function, return**
- new built-in functions: **atan2, cos, sin, rand, srand, gsub, sub, match, close,** and **system**
- new pre-defined variables: **FNR, ARGC, ARGV, RSTART,** and **RLENGTH**
- the input field-separator variable, **FS,** and the third argument to **split** are treated as regular expressions
- the precedence of operations now matches C language precedence

To take advantage of these new features, you must use the **.B awk** command. Some of these enhancements may not be compatible with some existing **awk** programs. You can get the old version by typing **oawk.** For more information, see the "**awk** Tutorial" packaged with the *Release Notes,* and the "Installation/Upgrade," "Software Notes," and "Future Directions" sections of the *Release Notes.*

Improved Recovery of Files from cpio Archives

If errors are encountered while restoring a file from floppy disk using **sysadm** or **cpio,** you can now skip over the bad blocks and continue the restore with the next file. To invoke this procedure, select the **restore** command from the "**sysadm**filemgmt" menu or use the **-k** option described on the **cpio**(1) manual page in the *System Administrator's Reference Manual.*

System Administration Menus

To help you perform system administration tasks, ICON/UXV Release 4.00 provides a system of menus from which you can choose an appropriate subcommand. These menus are accessed by the command **sysadm,** pronounced "sys-admin" The system consists of ten menus for administrative tasks, such as "file management" and "user management," and a Main Menu that names these ten task-specific menus. Under the "user management" menu, for example, you can choose among menu items to add a group to the system, add a user, delete a group, delete a user, list groups, list users, modify defaults used by adduser, modify group attributes, or modify a user's login. Release 4.00 will prompt you for information and administer the system according to your responses. Of course, you can always still use the system administration commands separately from the *sysadm* menus, which afford you more flexibility. The system administration menus provide a user-friendly way of administering the system.

Line Printer Spooling Utilities

The Line Printer (LP) Spooling Utilities package has been expanded to offer three new features: access to forms, easier administration of filters, and a menu interface for administration. Forms are pieces of pre-printed paper (such as invoices or payroll checks) on which files can be printed. When requesting jobs from the new print service, you can have your files printed on pre-printed forms (such as invoices or payroll checks) that you have registered with the print service. Filters are used to convert the content of files to formats required by various printers. This new version of the print service simplifies the LP administrator's job of handling filters. Additionally, a new set of menus makes it easier for the LP administrator to perform tasks such as installing the LP print service. For more information, see Chapter 7 of the *System Administrator's Guide.*

Framed Access Command Environment (FACE)

The FACE user interface is designed to present the ICON/UXV system environment in a user-friendly manner. The user "sees" the ICON/UXV system through a world of frames containing menus of items that can be selected, forms of information to be filled out, text of any type, including help. Function keys are used to execute often-used commands. For more information, see the *Framed Access Command Environment User's Guide.* The Form and Menu Language Interpreter is a high-level language interpreter that allows developers to write user-friendly applications. For more information on programming

with FMLI, see the *Form and Menu Language Interpreter Programmer's Guide.*

Enhanced curses

The ICON/UXV system screen management library known as **curses**(3X) has been improved to support color text on terminals capable of displaying it. A default table of eight colors can be modified or expanded. The programmer simply defines a table of foreground-background color pairs. This table is used as any other screen attribute is.

Enhanced Security Features

The following features have been added to ICON/UXV Release 4.00 to improve system security. For details about these features, see the "Security Notes" section in these *Release Notes.*

login.secure Time

If the file **login.secure** exists, a password is required to log in and no more than 3 failed attempts to log in are permitted. After the 3 failed attempts, **login.secure** sleeps for 20 seconds and then exits.

lastlogin Time

To enhance security, the time a user last logged in will be displayed each time a user logs in to that account. It is recommended that you check this time message to make sure that it corresponds to the last time you logged in.

loginlog File Capability

The system can now record unsuccessful login attempts in a file. If the file **/usr/adm/loginlog** exists, any five consecutive unsuccessful login attempts will be logged there, or, if **login.secure** exists, any three consecutive unsuccessful login attempts will be logged. After five (or three) unsuccessful attempts, **login** will sleep for 20 seconds before dropping the line. If, however, a person has fewer than five (or three) unsuccessful attempts, none of them is logged.

Sticky Bit

Because public directories such as /tmp and /usr/tmp are writable by everyone, anyone may remove files from them. This situation poses a serious problem to the integrity of files contained in those directories, as well as to the overall security of the system. The sticky bit on a directory is now used to restrict the removal of files within that directory. Without the sticky bit, the standard ICON/UXV semantics for object removal are followed. By default, the sticky bit will be set for the public directories **/tmp** and **/usr/tmp**.

Shadow Password File

The shadow password file was developed to address a security concern in ICON/UXV. Previously, encrypted user passwords were stored in the password file (**/etc/passwd**), which was readable by all users. Encrypted passwords and their attributes (such as aging information) have been moved to an access-restricted file called the shadow password file (**/etc/shadow**) to enhance security. The shadow password file is readable only by its owner (**root**, by default). Three commands have been added, **passmgmt**(1M), **pwconv**(1M), and **pwunconv**(1M). In addition, the **passwd**(1) command has been changed. These commands should be used to update the password files.

Flexible SCSI Addressing

SCSI devices can be addressed to any of seven available IDs after two requirements are met: First, the boot drive has to be addressed at ID 4. Second, there has to be a tape drive– cassette or quarter-inch –at ID 1 or 3, respectively. After these requirements of the ROM have been met, any SCSI device can be addressed to any of the other IDs.

This greater addressing flexibility allows there to be more than 1 of any type of tape drive, for example, more than 1 half-inch drive. It also allows you to have more than the 3 disk drives allowable in previous releases. You can now have up to 6 disk drives on a system.

The flexible SCSI addressing scheme precipitated this release's new device-naming conventions. The conventions were modified to include the actual address of a device since devices are no longer associated with specific addresses.

As an aid to users, *devicename* aliases have been provided in this release. It is important to note, however, that the aliases do not map completely becausesome device names have not been included. Nevertheless, when a name was left out, it was because the new name is reminiscent of the old: "qic0" and "qic24." The new device-naming conventions are described in Appendix A of the *System Administrator's Guide*. A handy quick reference to the new conventions is included.

New Disk Cache Code

The new disk cache code smoothes out system performance to a relatively consistent level, dispensing with periodic dips in performance due to flushing dirty cache buffers. The new code is more active, causing more frequent flushes of the buffers, with each occurrence taking less total time than previously. An important aspect of the new code is that it becomes more aggressive in flushing as the cache becomes more dirty, thus preventing it from becoming very dirty and requiring excessive time for flushing.

In addition, the new disk cache code has cured a type of sudden system failure that usually appeared to be a file system corruption, but was, in fact, related to the cache code.

Additional Berkeley Commands and Utilities and Command Extensions

Some commands that had originated from Berkeley Unix (4.3BSD) have been fully implemented as System V commands. Instead of being run in compatible mode, the commands are now fully integrated, binary members of Release 4.00's command set.

For example, **w** and **more** are now fully implemented System V commands. In addition, **find** and **test** now support symbolic links. The **finger** command is now available in this release, as are **uuncode** and **uudcode**. And for incremental backup and restore, the **fdump** program is now available.

Berkeley Fast File System

The Berkeley fast file system provides high throughput rates and some useful file-handling tools. By clustering data that is sequentially accessed, and by providing two block sizes, among other techniques, the Berkeley file system allows access rates of up to 10 times faster than the traditional UNIX file system.

In addition, the Berkeley fast file system supports a file-locking mechanism, arbitrary-length file names, and other useful file management tools.

TCP/IP Networking Support

ICON computer systems use the industry-standard TCP/IP protocol on two industry-standard interfaces: Ethernet and Serial-Link. Ethernet and Serial-Link can be supported simultaneously, if desired.

Berkeley Networking Tools

The ICON/UXV-NET utilities package has been bundled into this release. The tools in this package is a subset of the networking tools developed at the Advance Research Projects Agency (ARPA) and the University of Calfornia at Berkely (UCB). The Berkeley utilities are from version 4.3 (4.3BSD) of their software.

These tools give you the ability to transfer files to remote systems; log in to remote hosts; execute commands remotely; and send and receive mail from remote hosts.

Chapter 3: Installation and Upgrade Procedures

Introduction

This chapter will help you install ICON/UXV Release 4.00 on your ICON computer as an upgrade from an earlier ICON/UXV release. The chapter starts with an Installation Checklist, which provides an overview of the major steps involved in the installation process. It also can show you at a glance how many more steps you have to go at any given point in the installation process.

Adhere to the instructions in this chapter carefully to guarantee a smooth upgrade. The Installation Checklist is provided to help you perform the steps in the right order.

The first section of this chapter, Upgrade Information, provides instructions for determining if any hardware on your ICON computer needs to be upgraded. Certain hardware and firmware components must be at a specific revision level before ICON/UXV Release 4.00 can be installed. Because the remainder of this section deals with actually upgrading hardware, if you find you don't need to upgrade, you can go directly to the second section, Back Up Data Files.

Unlike some ICON/UXV upgrades, this one involves reformatting all SCSI hard drives except the HD180. Section two of this chapter contains instructions for backing up your data files using the **fdump**(1M) program in anticipation of the reformat; the files will be erased during the reformat. The reformatting program on the distribution tape will automatically repartition your drives, leaving sufficient space for the root file system of this release. It will also provide as much as a 10% increase in usable disk space on the drives.

Since reformatting can take a lot of time and reformatting the HD180 nets no gain in disk space, there is no reason to reformat it. It does, however, need to be repartitioned if it serves as the boot drive, as does any HSMD drive serving as the boot drive.

If the boot drive on your system is an HD180 or (rarely) an HSMD drive, you will be repartitioning it before you start the actual installation. Section two also contains instructions for repartitioning a drive.

It is very important that you save configuration information (passwords, login directories, port assignations) from your existing ICON/UXV system so you can integrate it into the 4.00 system. Section three, Back Up System Files, contains instructions for backing up configuration information from your current system.

The fourth section, Install This Release, presents step-by-step installation instructions.

The fifth section, Integrate System Files, suggests methods of integrating configuration information into the new release.

The last section, Restore Data Files, shows you how to use the new **sysadm**(1M) (pronounced "sysadmin") menu's **restore** command.

About the Distribution Tape

This release is delivered on a single magnetic tape, either a cassette or quarter-inch cartridge. The tape is labeled ICON/UXV Release 4.00. It contains ten files:

0. install (tar archive format)
1. disk formatter (standalone format)
2. kernel files (tar archive format)
3. utility programs (tar archive format)
4. place holder (contains no files)
5. place holder (contains no files)
6. HSMD disk formatter (standalone format)
7. mini-root file system (file system image)
8. root ("/") file system (tar archive format)
9. "/usr" file system (tar archive format)

Other Restore Procedures

These Release Notes do not provide instructions for two other important procedures related to installation of the ICON/UXV system:

- partially restoring a system to bring it to a usable state or to remove a forgotten root password
- fully restoring a system either when there is a new disk, or when you need to increase the size of the disk partitions

Instructions for doing these types of restore operations are available in the ICON/UXV System Administrator's Guide.

Installation Checklist

Check off items as you complete them. Complete item 1 before proceeding to item 2, and so on. Instructions for carrying out each item will be found on the page cited.

1 Upgrade Hardware and/or Firmware ☐

Page 3-4

2 Back Up Data Files ☐

Page 3-7

3 Back Up System Files ☐

Page 3-10

4 Install This Release ☐

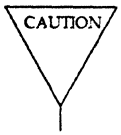
Page 3-11

5 Integrate System Information ☐

Page 3-34

6 Restore Data Files ☐

Page 3-39



If you install this new system (item 4) before you have upgraded hardware and backed up your existing system (items 1, 2, and 3), you may find yourself unable to execute items 5 and 6.

Upgrade Information

Some hardware components need to be at certain revision levels before you can install this release. Follow the instructions under "Upgrade Procedure" in this chapter. In addition, we strongly recommend you perform the two configuration upgrades described below to enhance system performance. Call Sanyo/Icon Customer Service for instructions on these: 1-800-SANYO-US (726-9687).

Hardware and Firmware Upgrades

1. HSMD Controller board

   The HSMD Controller board's firmware needs to be upgraded to version 2.2. We recommend you install this firmware whether or not you intend to upgrade software. It provides six bug fixes.

2. Cassette tape drives

   These must have TEAC part number MT-2ST/45S2-27 with firmware revision M. All S2-27 drives must be retrofitted with the M revision firmware if an older version of firmware is installed. Not all drives even have the firmware ROM installed, so this upgrade may require a drive exchange.

3. MT32 tape drives

   If an MT32 drive has a number 9500 SCSI controller board, it needs to be upgraded to number 9700. In addition, this change requires that new firmware be added to the MT32's main controller board.

   If this upgrade is not done, the MT32 tape drive will continue to work but only in unbuffered mode, which results in poor tape performance.

Configuration Upgrades

1. Unlike previous releases, ICON/UXV 4.00 supports two half-inch-type devices. If you are using both an MT32 half-inch drive and an 8mm tape drive (MT2000), they must have separate SCSI addresses. So if they have both been addressed at SCSI ID 2, the address of one of them needs to be changed.
2. We strongly recommend that DCP expansion memory local bus addressing be offset by 16 MB. The offset, if done, will aid Customer Service in troubleshooting system crashes.

Upgrade Procedure

You will use the Hardware Diagnostics Release 2.10 distribution tape to display configuration information to your console terminal's screen. Sanyo/Icon engineers will use that information to put together an upgrade package for your system. Follow the procedure below.

Starting Conditions	Machine off or operating system shut down.
Medium	*Hardware Diagnostics Release 2.10* distribution tape.

Step 1: After the operating system has been properly shut down, turn the keyswitch on the front of the ICON computer to RESET and then press the spacebar immediately after the following message is displayed:

```
Copyright 1988 Icon International, Inc.  ROM Version 2.3

Beginning autoboot
(press space bar to override) ...
```

If you didn't press the spacebar soon enough, just try Step 1 again.

Step 2: Load the Hardware Diagnostics Release 2.10 distribution tape and specify its drive as **ct0** if it is a cassette distribution tape, or **qic24** if it is a cartridge.

```
Device Name
  is0    Integrated SCSI Hard Disk
  hs0    HSMD Hard Disk
  ct0    Cassette Tape
  qic24  Quarter Inch Cartridge Tape
  mt0    Half Inch Magnetic Tape
Enter boot device name
> qic24
```

Step 3: You will see a display similar to the following one.

```
SYSTEM CONFIGURATION REPORT

Main CPU Board (CPU3) .....................................CPU3
     without Floating-Point Coprocessor
SFM4 Static Frame Memory board 00 .........................SFM4-00 CPU3 <-CPU3
Disk Cache Processor ......................................DCP3
SFM4 Static Frame Memory board 00 on CPU3 bus .............SFM4-00 CPU3 <-DCP3
Hard Disk Drive 1 .........................................SCSI Hard Disk 2
     CDC       94151-9          6226       0x120000
Cassette Tape Drive .......................................Cassette Tape
     TEAC      MT-2ST/45S2-27   RV G       0x000000
Cartridge Tape Drive ......................................QIC Tape
     ADSI      ADSI-T100        C          0x300000
Peripheral Communications Processor 0 .....................PCP-0
     No loop-back connectors on any ports

Initializing lower memory (4 MB) ...
press any key to continue ...
```

Step 4: Since the computer is in the standalone state (independent of the ICON/UXV operating system), the information on the screen cannot be sent to a printer but must be transcribed by hand from screen to paper. (If you have a printer attached to your terminal, and the terminal supports a "copy screen to printer" function, you may use that feature.)

Step 5: FAX the information to Sanyo/Icon Customer Service. The FAX number is (801) 226-0651. Call Customer Service's toll-free number to notify them that the FAX is on its way: 1-800-SANYO-US (726-9687). A Customer Service representative will contact you to recommend a suitable upgrade package for your ICON computer.

Back Up Data Files

You will be reformatting all the SCSI hard drives on your ICON computer except the HD180. If the HD180 serves as the boot drive, you will be repartitioning it. (Reformatting the HD180 nets no extra storage space.) In addition, any HSMD drive that serves as the boot drive will have to be repartitioned. Use the flow chart below to decide whether to reformat, repartition, or neither.

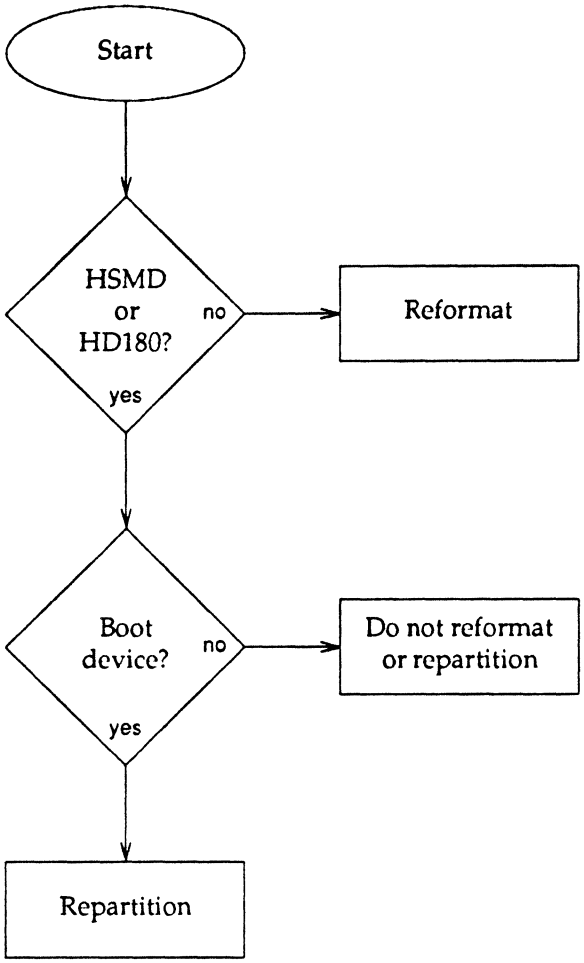


Figure 3-1: Disk Reformatting Decision Flowchart

Since reformatting or repartitioning will erase all file systems on those drives, use the fdump (file system dump) program to do a complete backup of the drives before attempting any of the installation procedures in this chapter. Follow the instructions below.

How to Use fdump

Whereas **tar**(1) and **cpio**(1) back up *only* as much data as can fit onto a single tape, the **fdump**(1M) program allows you to back up an entire ICON/UXV file system onto multiple tapes. The program tells you how many tapes you will need and prompts you to insert a new tape when needed.

Step 1: Use the **df**(1) command to display a list of the devices the file systems are stored on. The display will look something like this:

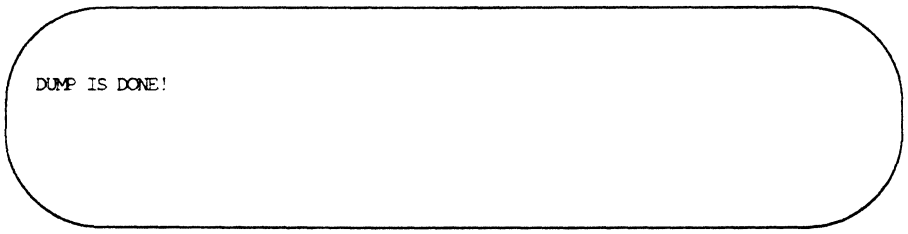
```
# df -b
Filesystem        kbytes    used   avail capacity  Mounted on
/dev/dsk/is0a     19927   12399    5536    69%    /
/dev/dsk/is0b     31947    4634   24119    16%    /swap
/dev/dsk/is0g    553032  310967  186762    62%    /usr
/dev/dsk/is1g    545720  305854  185294    62%    /usr2
#
```

Step 2: The **fdump** command you will use contains 3 keys, the tape size, the tape device, and last, the disk device the file system is stored on. For example, to back up /usr (above), you would type

```
# fdump 0cf 60 /dev/rqic24 /dev/is0g
```

The **0** key specifies that you want a complete, not partial, backup. The **c** key corresponds to the tape-capacity argument (**60**), and the **f** key corresponds to the tape-device argument (**/dev/rqic24**).

Shortly after the program starts, **fdump** will estimate the number of tapes it will take to complete the backup. After it has finished, **fdump** will display the following message:



Step 3: Repeat Step 2 until you have backed up the file systems on all the SCSI hard drives except the HD180 unless the HD180 is the boot drive, in which case you will back up its file systems, too.

NOTE

It is not necessary to back up the **/swap** file system.

If either the HD180 or an HSMD drive serves as the boot drive, you will have to repartition it, so back up its file systems using fdump. (Instructions for repartitioning an HD180 or HSMD drive are given in the third section of this chapter.)

Instructions for restoring the file systems you have backed up are given in the last section of this chapter. To do so, you will use the new interactive sysadm menu provided in Release 4.00.

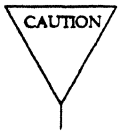
If you need help backing up your file systems, call Sanyo/Icon Customer Service at the toll-free number: 1-800-SANYO-US (726-9687).

Back Up System Files

Use the **tar** command to make a special backup tape called "Backup of ICON/UXV 3.*xx* Configuration Files." After you have installed 4.00, you will load these files into a temporary directory called **/usr/tmp**. You will then either merge configuration information from these files into 4.00 files, or use them as reference to configuration information that you will hand type into 4.00 files.

On this special backup tape, copy the following files:

- Any files that define your existing system, for example **/etc/inittab**, which contains login port definitions.
- Any files that you have added since the existing system was installed .
- Any ICON/UXV application and utilities files.



This is very important: When you copy the directories and files onto this tape, do not type a slash (/) at the beginning of a directory or file name. Otherwise, when you extract it from the tape, it has the potential of erasing the 4.00 file by the same name.

To copy, for example, **/etc/passwd** and **/etc/inittab**, onto a quarter-inch cartridge tape, you would type

 # cd /

 # tar cvf /dev/rqic24 etc/passwd etc/inittab

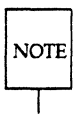
Later, after you have finished installing 4.00, you will extract these files into a working directory, **/usr/tmp**, by typing

 # cd /usr/tmp

 # tar xvf /dev/rqic0

This will create the files **/usr/tmp/etc/passwd** and **/usr/tmp/etc/inittab**. You can then use the editor to copy passwords and login port definitions from these files into their corresponding 4.00 files. The table on the next page lists the files to put on this special backup tape.

The special backup tape must have, but is not limited to, the files in the following table.



Save only the local changes you have made for files in the directories marked with a dagger (†). Saving the other files in these directories would take up a lot of space needlessly.

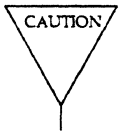
1.	/.profile	11.	/etc/passwd
2.	/dev/MAKEDEV.local	12.	/etc/rc
†3.	/usr/adm/	13.	/etc/rc.local
4.	/usr/mail	14.	/etc/termcap
†5.	/usr/spool	15.	/etc/inittab
6.	/usr/spool/lp	16	/etc/uxrc
7.	/usr/lib/uucp	17.	/etc/rc.pick
8.	/usr/local	18.	/etc/smiledisks
†9.	/usr/lib/terminfo	19.	/etc/smiledisks_*xy*
10.	/etc/group	20.	/etc/smileprinters

Install This Release

The basic facts about this release installation procedure are as follows:

Purpose	To upgrade an ICON/UXV operating system to Release 4.00.
Current Release	ICON/UXV 3.00, 3.10, 3.20, 3.30, 3.36, or 3.37.
Media	*ICON/UXV Release 4.00* distribution tape, either cassette or quarter-inch cartridge.
Time	About one hour, plus time to back up data and either reformat or repartition hard disk drives (if necessary).

Remember, reformatting a large-capacity SCSI hard drive can take several hours. Give users plenty of advance notice about the installation. Also, make sure there are no outstanding print jobs in any of the printers or the **/usr/spool** directory before you start the installation.



Make sure you have completed the first 3 items on the Installation Checklist before you start the installation process. Making a backup of the existing system is the only way to safeguard users' files.

About The Instructions

Screens show both system output and your input (what you should type in literally as it appears on the page). Your input shows up in boldface type slightly larger than that of system output.

We have tried to keep written instructions (Step 1, Step 2) to a minimum so you will spend more time installing than reading. However, input sometimes varies according to a system's configuration, so be sure to read all the instructions carefully.

We have assumed in these instructions that your distribution tape is a quarter-inch cartridge. If you have a cassette instead, make the following adjustments when typing device names: type **ct0** instead of **qic24** and **sc0c1r** instead of **sc0q3r**. (See Appendix A of the *System Administrator's Guide* for an explanation of 4.00's new device-naming conventions.)

We have also assumed that the SCSI devices on your system are addressed to the default ID #:

SCSI Device	ID #
Floppy drive controller	0
Cassette tape drive	1
Half-inch tape drive	2
Quarter-inch cartridge tape drive	3
First hard disk drive	4
Second hard disk drive	5
Third hard disk drive	6

Figure 3-2: SCSI Device Locations

If the SCSI addresses on your system do not follow this default addressing scheme, adjust your input accordingly.

The last portion of the installation involves configuring the new 4.00 operating system. At this time, you will use the "Backup of ICON/UXV 3.*xx* Configuration Files" tape you made (item 3 on the Installation Checklist). In addition to using this backup tape, you will also use an interactive program called "setup" to create a login account for yourself (the system administrator), and to create passwords for administrative logins (**powerdown, setup, sysadm**) and system logins (**adm, daemon, root,** others). As a final step, the setup program will offer to help you create entries in the */etc/inittab* file. You will answer "no" to this offer; you will carefully merge entries from the old */etc/inittab* into the new one after you finish the installation. (See the "Integrating System Information" section.) "Installation procedure."

Installation Procedure

Step 1: Boot from the distribution tape. Turn the keyswitch to RESET and then press the spacebar IMMEDIATELY after the following message comes up on the screen:

```
Copyright 1988 Icon International, Inc.  ROM Version 2.3
    Beginning autoboot
(press space bar to override) ...
```

If you didn't press the spacebar soon enough, just try Step 1 again.

```
Device Name
  is0    Integrated SCSI Hard Disk
  hs0    HSMD Hard Disk
  ct0    Cassette Tape
  qic24  Quarter Inch Cartridge Tape
  mt0    Half Inch Magnetic Tape
Enter boot device name
> qic24
```

If your boot drive is an HSMD drive, call Customer Support for help repartitioning it: 1-800-SANYO-US (726-9687).

If your SCSI boot drive is an HD180, go to Step 2. Otherwise, go to Step 3.

Step 2: To skip the formatting and repartition only, use the **-f** option for the disk formatter program (at position 1 on the distribution tape):

```
Boot loader -- Version 4.00
Load:  qic24:1 -f

Disk formatter -- Version 4.00
(Skipping formatting...)

Specify disk unit to format:
  is0       = First integrated scsi drive
  is1       = Second integrated scsi drive
  is2       = Third integrated scsi drive
  is(0)     = integrated scsi drive at scsi id 0
  is(1)     = integrated scsi drive at scsi id 1
  is(2)     = integrated scsi drive at scsi id 2
  is(3)     = integrated scsi drive at scsi id 3
  is(4)     = integrated scsi drive at scsi id 4
  is(5)     = integrated scsi drive at scsi id 5
  is(6)     = integrated scsi drive at scsi id 6
  hs00-hs73 = HSMD drives
is0
```

```
Mfg info = "CDC      94161-9    6226"

Skipping format...
Valid disk label found.
Caution...changing partition start blocks and lengths will
destroy file system integrity.

Disk unique ID = ""
```

When you are asked to set partition sizes, type an asterisk (*) to have the formatter automatically reset the **boot**, **swap**, and **/usr** partition sizes to the default for this release. You can then change the sizes of **/usr** and add custom partitions, as illustrated below.

```
PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    15639    15898
b swap partition           15899    10369    26268
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           26269   130384   156653
h                              0        0        0

Enter "*" to reset partitions or RETURN to continue: *
Enter unique disk name: <CR>

PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    20480    20738
b swap partition           20739    10240    30978
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           30979   125675   156653
h                              0        0        0
```

Each block is 1024 bytes in length. Say you wanted partition **g** (section 6) to be a 40MB **/usr**, and partition **h** (section 7) to occupy the rest of the space as partition **/home**. You don't need to figure out the ending block numbers; the program does it for you. You would add 40×1024, or 40960, to the starting block of **/usr**. Then print out the result:

```
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: h
Enter description: /usr
Enter starting block (decimal): 30979
Enter number of blocks (decimal): 40960
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue:  p

PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    20480    20738
b swap partition           20739    10240    30978
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           30979    40960    71939
h                              0        0        0
```

When you make the */home* partition, add 1 to the ending block of */usr*, then enter an asterisk (*) as the "number of blocks" to indicate the rest of the available space:

```
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: h
Enter description: /home
Enter starting block (decimal): 71940
Enter number of blocks (decimal): *
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: p
```

```
PARTITION TABLE
Description                 Start   Length      End
  Usable disk                 259   156395   156653
a Root partition              259    20480    20738
b swap partition            20739    10240    30978
c Entire disk                   0   156654   156653
d                               0        0        0
e                               0        0        0
f                               0        0        0
g /usr partition            30979    40960    71939
h /home                     71940    84713   156653

Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: 0

Enter "ok" if unique name and partitions are ok,
else RETURN to reset values and try again: ok

Label written

Disk initialized.

Boot loader -- Version 4.00
Load: qic24:2

Bootstrap program installer -- Version 4.00

From: qic24:3
..Bootstrap program loaded.

To:
   is0       = First integrated scsi drive
   is1       = Second integrated scsi drive
   is2       = Third integrated scsi drive
   is(0)     = integrated scsi drive at scsi id 0
   is(1)     = integrated scsi drive at scsi id 1
   is(2)     = integrated scsi drive at scsi id 2
   is(3)     = integrated scsi drive at scsi id 3
   is(4)     = integrated scsi drive at scsi id 4
   is(5)     = integrated scsi drive at scsi id 5
   is(6)     = integrated scsi drive at scsi id 6
   hs00-hs73 = HSMD drives
is0

Bootstrap program installed.

Boot loader -- Version 4.00
Load:
```

If your system has other SCSI drives that are not HD180s, go to Step 3. If there are no other SCSI drives, skip Steps 3 and 4.

Step 3: Format SCSI drives as illustrated in the screens that follow.

NOTE

When you are asked to specify the number of write/read patterns, specify a minimum of 5.

```
Boot loader — Version 4.00
Load: qic24:1

Disk formatter — Version 4.00

Specify disk unit to format:
   is0       = First integrated scsi drive
   is1       = Second integrated scsi drive
   is2       = Third integrated scsi drive
   is(0)     = integrated scsi drive at scsi id 0
   is(1)     = integrated scsi drive at scsi id 1
   is(2)     = integrated scsi drive at scsi id 2
   is(3)     = integrated scsi drive at scsi id 3
   is(4)     = integrated scsi drive at scsi id 4
   is(5)     = integrated scsi drive at scsi id 5
   is(6)     = integrated scsi drive at scsi id 6
  hs00-hs73 = HSMD drives
is0

Mfg info = "CDC       94161-9    6226"

This program destroys all data on the disk.
Enter "ok" then RETURN to continue; anything else will halt: ok
Initializing disk

Physical heads/Cyls: 9/967
Formatting...please wait...

sec size = 1024, nsectors = 156654
Logical geometry:
  cyls = 967, heads = 6, sectors = 27
Enter number of write/read patterns to run )0-14): 5
```

After the formatter completes the write/read patterns, you will be asked the disk's unique name. Simply press <return> and the formatter will generate a default name. Then the formatter will display a "partition table," which describes how the drive has been partitioned.

```
Enter unique disk name:  <RETURN>

PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    20480    20738
b swap partition           20739    10240    30978
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           30979   125675   156653
h                              0        0        0

Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue:
```

Each block is 1024 bytes in length. Say you wanted partition **g** (section 6) to be a 40MB **/usr** partition, and partition **h** (section 7) to be a **/home** partition that occupies the rest of the available disk space. You don't need to figure out block numbers; the program will do it for you. You would just add 40960 (40×1024) to the starting block of **/usr**. Then you would print out the result:

```
PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    20480    20738
b swap partition           20739    10240    30978
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           30979   125675   156653
h                              0        0        0

Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: h
Enter description: /usr
Enter starting block (decimal): 30979
Enter number of blocks (decimal): 40960
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue:  p
```

```
PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    20480    20738
b swap partition           20739    10240    30978
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           30979    40960    71939
h                              0        0        0
```

When you make the /home partition, type an asterisk (*) for the "number of blocks" to indicate "the rest of the available space:"

```
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: h
Enter description: /home
Enter starting block (decimal): 71940
Enter number of blocks (decimal): *
Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: p

PARTITION TABLE
Description                Start   Length      End
  Usable disk                259   156395   156653
a Root partition             259    20480    20738
b swap partition           20739    10240    30978
c Entire disk                  0   156654   156653
d                              0        0        0
e                              0        0        0
f                              0        0        0
g /usr partition           30979    40960    71939
h /home                    71940    84713   156653

Enter "a" through "h" to modify a partition,
   or "p" to print current configuration,
   or "0" to continue: 0
```

```
Enter "ok" if unique name and partitions are ok,
else RETURN to reset values and try again: ok

Label written

Disk initialized.

Boot loader -- Version 4.00
Load: qic24:2

Bootstrap program installer -- Version 4.00

From: qic24:3
..Bootstrap program loaded.

To:
   is0       = First integrated scsi drive
   is1       = Second integrated scsi drive
   is2       = Third integrated scsi drive
   is(0)     = integrated scsi drive at scsi id 0
   is(1)     = integrated scsi drive at scsi id 1
   is(2)     = integrated scsi drive at scsi id 2
   is(3)     = integrated scsi drive at scsi id 3
   is(4)     = integrated scsi drive at scsi id 4
   is(5)     = integrated scsi drive at scsi id 5
   is(6)     = integrated scsi drive at scsi id 6
   hs00-hs73 = HSMD drives
is0

Bootstrap program installed.

Boot loader -- Version 4.00
Load:
```

Step 4: Repeat Step 3 for each SCSI drive to be formatted, then proceed to Step 5.

Step 5: Next copy the mini-root file system from the distribution tape onto partition **b** (**/swap**) of the boot drive (ususally is0). Next, you will boot the system from partition **b**. (Be sure to type the parentheses in the **(is0,b)** command.)

```
Boot loader -- Version 4.00
Load: qic24:4

Copy program -- Version 4.00
From: qic24:7

To: is0:b
...
Copying

  512K bytes read...  512K total bytes written.
  512K bytes read... 1024K total bytes written.
  512K bytes read... 1536K total bytes written.
                        .
                        .
                        .
  512K bytes read... 7168K total bytes written.

Copy done.

Boot loader -- Version 4.00
Load: (is0,b)
```

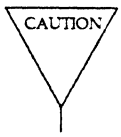
Step 6: After the system has booted, follow the steps in the next 2 screens, taking care to do the following things:

- Be sure to type the dot and slash in the **./sinstall** command.
- If you are not sure how to specify your console type, call Sanyo/Icon Customer Service: 1-800-SANYO-US (726-9687).
- If your distribution tape is either a cassette or a cartridge that you are using in a drive that is not addressed to the default SCSI ID for cartridge drives, type no to the question, "Is the distribution tape loaded in the QIC cartridge tape drive (device name sc0q3)?" When the system asks you for the drive's SCSI address, type **1 if it is a cassette drive addressed at the default SCSI ID for a cassette drive. If it is a drive-cassette or cartridge-not addressed to the default SCSI ID, type in the drive's actual address.**

```
Loading (is0,b)/vmunix...
Loading (is0,b)/dcunix...

ICON/UX Kernel Release 4.00: 12/05/89 16:40:36 MST (P/N 160-230-001 A0)

(C) Copyright 1986-1989 Icon International, Inc.
    All Rights Reserved.

    Portions of this software include material separately
    licensed and copyrighted by AT&T, the Regents of the
    University of California and Motorola, Inc.

Disk CPU: 2Mb Disk cache memory
Disk CPU: 6 Smile Ports exist
WARNING: clock gained 65 days -- CHECK AND RESET THE DATE!
Main CPU: 4Mb Virtual Memory Cache

INIT: SINGLE USER MODE
# /sinstall

Please select a terminal type or emulation:

        1.  ICON DT1200
        2.  DEC VT100
        3.  Wyse WY-50
        4.  Televideo TVI925
        5.  ADDS Viewpoint
        6.  Other

> 2
```

```
Note:  you may enter "quit" to most of the questions in the following dialogue
to terminate the installation process.

Is this a new machine or an existing installation? (n/e) n

Checking tape drive configuration...

Is the distribution tape loaded in the QIC cartridge
tape drive (device name sc0q3)?
y

Checking disk configuration...

Will you use the default SCSI disk as the boot device? y
```

```
File system information:

        Partition     Length   Mount Point    Overlapping Partitions
        ---------     -------  -----------    ----------------------
        sc0d4s0        20480   /
        sc0d4s1        10240   /swap
        sc0d4s6       125675   /usr

        Total 3 Partitions

Enter a partition/mount-point pair to update file system information,
"help" or "?" for help, "quit" to quit, or press RETURN with
no other input to continue.

> <RETURN>
```



Be sure to type **y** when asked if you wish to continue. If you press **<RETURN>** or anything else, the installation will be terminated! If it is terminated, restart by typing **./sinstall** at the prompt (#).

```
Checking file system information......

                        * * * WARNING * * *

        All file systems configured in the previous dialogue will
        now be completely erased.  If you have any data on your
        system that you wish to preserve, do not select the "new"
        option in the installation dialogue.

Do you wish to continue?  y

Initializing file systems.  This may take some time.

Make sure that the ICON/UXV Release 4.00 distribution tape is in
the tape drive.

Press RETURN to continue. <RETURN>
mount -f BSD43 /dev/dsk/sc0d4s0 /new
mount -f BSD43 /dev/dsk/sc0d4s6 /new/usr

It will take approximately twenty (20) minutes to extract files
from the distribution tape.

Please stand by....

Generating file system databases (/etc/fstab and /etc/checklist)
....
```

NOTE

The file system check messages at the bottom of the next screen are only an example; file system check messages vary from system to system.

```
It will take approximately 10 minutes to create device file
entries.

Please stand by....

Checking file systems.
/dev/rdsk/sc0d4s0: 493 files, 7499 used, 11804 free  (96 frags, 2927 blocks, 0.5
% fragmentation)
/dev/rdsk/sc0d4s6: 3384 files, 20495 used, 26464 free (1020 frags, 6361 blocks,
2.2% fragmentation)

Installation complete.
```

Step 7: Now reboot but do not override the autoboot this time.

If the date and time are not correct, type yes anyway; later on, the interactive setup program will prompt you for the correct date and time information.

```
Do you wish to reboot the system at this time? y

The system will now reboot.


syncing disks... done

Copyright 1988 Icon International, Inc.   ROM Version 2.3

Beginning autoboot
(press space bar to override) ...

Boot loader -- Version 4.00
Loading (is0,a)/vmunix...
Loading (is0,a)/dcunix...
```

```
ICON/UX Kernel Release 4.00: 02/07/90 16:40:36 MST (P/N 160-230-001 A0)

(C) Copyright 1986-1989 Icon International, Inc.
    All Rights Reserved.

    Portions of this software include material separately
    licensed and copyrighted by AT&T, the Regents of the
    University of California and Motorola, Inc.

Disk CPU: 2Mb Disk cache memory
Disk CPU: 6 Smile Ports exist
Disk CPU: initializing swap partition....
Main CPU: 4Mb Virtual Memory Cache

INIT: SINGLE USER MODE
# init 2
#
INIT: New run level: 2

Is the date Thu Feb 22 17:06:23 MST 1990 correct? y

The system is coming up.  Please wait.

This machine has not yet been used as a customer machine.  The messages that
follow are from checking the built-in file systems for damage that might have
occurred during shipment.  As long as you do not see either of the messages

                                BOOT UNIX

or                        FILE SYSTEM WAS MODIFIED

all is well.  If either message does come out, call your service representative.
However, the machine is still usable unless you are told otherwise.
```

```
Checking file systems:

** /dev/rdsk/sc0d4s6
** Last Mounted on /new/usr
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl Groups
** Phase 6 - Check Active File System Flag
3384 files, 20495 used, 26464 free (1020 frags, 6361 blocks, 2.2% fragmentation)

Node name set to generic.
mount -f BSD43 /dev/dsk/sc0d4s6 /usr
Loaded PCP #0 from /pcpimage
Process accounting started.
Expreserve notification performed.
Print services started.
Line printer scheduler started.

ICON SYSTEM CONFIGURATION:

System Processor:
        Processor Boards:   16 MHz CPU3 and DCP3
        ROM Revisions:      CPU Version 2.3, DCP Version 2.3.
        Main Memory Size:   4 Megabytes
        Disk Cache Memory:  2 Megabytes

System Peripherals:
        PCP0 Peripheral Communications Processor
        SMILE Host Board (6 ports enabled)
        SCSI 0 Unit 1:  ICON CS50 Cassette Tape Drive
        SCSI 0 Unit 3:  ICON CR60 Cartridge Tape Drive
        SCSI 0 Unit 4:  ICON HD180 Hard Disk Drive
```

Step 8: Log in as **setup.** The interactive setup program will prompt you for configuration and date and time information.

```
Welcome!  This machine has to be set up by you.  When you see
the "login" message type

                                setup

followed by the RETURN key.  This will start a procedure that leads you
through those things that should be done the "first time" the machine
is used.

Until you do so, the above filesystem shipment damage check will be run
each time you boot.

The system is ready.


ICON/UXV (generic:console)

login: setup
ICON/UXV Release 4.00, M68020 Version
Copyright (c) 1986-1990 Icon International, Inc.
All Rights Reserved
generic
Warning:  .lastlogin did not exist, creating it

                            Setup Procedure

Setup establishes this machine as yours and can make sure that no one else
uses it without your permission.  We assume that you have read about
"initial setup" in the GETTING STARTED chapter of the Owner/Operator Manual.

The first step is to set the timezone, date, and time of the system clock.

Current time and time zone is:  17:25 MST
Change the time zone: [y, n, ?, q] n
Current date and time:  Thu.  02/22/090  17:25
Change the date and time? [y, n, ?, q] n
```

Step 9: Now set up a login for yourself, the system administrator. Later, you will establish lgins for the other users by merging their logins from the old **/etc/passwd** file (on the special backup tape you made in item 3 of the Pre-installation Checklist) into into the new (4.00) **/etc/passwd** file.

```
        The next step is to set up logins.
The first one you make should be for yourself.

Anytime you want to quit, type "q."
If you are not sure how to answer any prompt, type "?" for help,
or see the Owner/Operator Manual.

If a default appears in the question, press <RETURN> for the default.

Enter user's full name [?, q]: Icon S. Administrator
Enter user's login ID  (or login name) [?, q]: isa
Enter user ID number (default 100) [?, q]: <RETURN>
Enter group ID number or group name (default 30) [?, q]: >RETURN>
Enter user's login (home) directory name. (default '/usr/isa') [?, q]: <RETURN>
User shell for this login (default:/bin/sh) [?, q]: <RETURN>

This is the information for the new login:
        User's name:    Icon S. Administrator
        login ID:       isa
        user ID:        100
        group ID:       30      (unix)
        home directory: /usr/isa
        usr shell:      /bin/sh

Do you want to install, edit, or skip this entry [i, e, s, q]]? i
Login installed.
Do you want to give the user a password? [y, n] y
New password: password will not echo as you enter it
Re-enter new password: password will not echo as you enter it
Do you want to add another login? [y, n, q] n
```

Step 10: Now the setup program gives you the opportunity to establish passwords for administrative logins (**powerdown**, **setup**), as illustrated below. You have the choice of doing so, or proceeding to the next step by typing "**q**."

```
NOTE:  Your password is very important.  It is the way that the computer
verifies that someone who attempts to login as you is indeed you.  If you
give it away to someone, they can do anything you can do and the machine does
not know the difference.

The next step is to establish passwords for the administrative logins
and commands.

Do you want to give passwords to administrative logins? [y, n, ?, q] y

The login 'powerdown' does not have a password.
Do you want to give it one? [y, n, ?, q] y
New password: the password is not echoed as you enter it
Re-enter new password: the password is not echoed as you enter it

The login 'setup' does not have a password.
Do you want to give it one? [y, n, ?, q] y
```

Step 11: Now the setup program gives you the opportunity to establish passwords for system logins (**root, daemon, others**), as illustrated below. Again, you may do so or type "**q**."

```
The next step is to establish passwords for the system logins that
do not already have them.  Once set, these passwords can only be
changed by the login or "root".

Do you want to give passwords to system logins? [y, n, ?, q] y
Do you want to give the 'root' login a password? [y, n, ?, q] y
New password: the password is not echoed as your enter it
Re-enter new password: the password is not echoed as your enter it

Do you want to give the 'daemon' login a password? [y, n, ?, q] y
```

Step 12: If your ICON computer system is part of a network, type in its node name. (If your system is not part of a network, you may still want to give it a name other than "generic.")

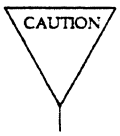
```
The next step is to set the node name of this machine.
This is the name by which other machines know this one.

This machine is currently called "generic".
Do you want to change it? (y, n, ?, q] y
What name do you want to give it? [q] hostname
Machine node name changed to hostname.
```

Step 13: Next you will set the characteristics of your system console according to its particular terminal type (TTY Type). If you need help specifying your terminal type, call Sanyo/Icon Customer Service: 1-800-SANYO-US (726-96787).

```
The next step is to check the terminal configuration for
the system console.  Do not change the line set (speed).

Console:  current characteristics:
Hangup Delay     none
Line Setting     console
TTY Type         dumb
Description      System Console

Enter a hangup delay, in seconds, or 'none' (default: none) [?, q]: <RETURN>

Available line settings:

        console   300UUCP   1200UUCP  2400UUCP  9600      19200     38400
        300       1200      2400      4800      9600C     19200C    38400C
        300C      1200C     2400C     4800C     9600UUCP  19200UUCP

Select a line setting (default: console) [?, q]: <RETURN>
Enter a new terminal type (default: dumb) [?, q]: vt100
Current description: System Console
Enter a new description (default: current description) [?, q]: <RETURN>
```

```
Console: new characteristics:
Hangup Delay    none
Line Setting    console
TTY Type        vt100
Description     System Console

    Do you want to install these new characteristics? [y, n, q] y
console now has new characteristics.

    Changes will become effective on next login.
```

Step 14: The setup program offers to create entries in the **/etc/inittab** file. This function involves some risk. (See the CAUTION below.) Instead of using this function, simply merge information from the old **/etc/inittab** into the new (4.00) one. You will find guidelines for executing this merge in the next section, "Integrating System Information."



If your system has either a Distributed Communication System (DCS1) board or a SMILE Host board, be sure to type "n" in response to the question, "Do you want to configure new terminals using global defaults?" Typing "y" could bring the installation to an abrupt halt. The setup program would then create getty entries in */etc/inittab* for all the terminal configurations the system could possibly support, perhaps consuming all the remaining space in the */swap* partition and all available memory.

To escape from the setup program's terminal configuration process, type **q** when asked to select a state.

```
The final step is to ensure that all terminal lines are in
the system configuration file.

    Terminal Configuration

    This section makes entries in the ICON/UXV terminal configuration file
(/etc/inittab) for all terminal devices created by MAKEDEV which do not
already have an entry in the file.  You may either configure each port
individually or select global default settings for all terminal ports.
If most ports will be allocated as ICON/PICK logon ports, you should
select the "global default" method of terminal configuration.

Do you want to configure new terminals using global defaults? n

Checking installed equipment against configuration files...
```

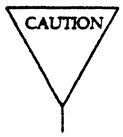
```
The following terminal ports are not presently configured:

ttya0  ttya2  ttya4  ttya6  ttya8  ttyaa  ttyac  ttyae
ttya1  ttya3  ttya5  ttya7  ttya9  ttyab  ttyad  ttyaf

Enter default configuration data for port ttya0:

Available states:

        off     unix    pick

Select a state (default: unix) [?, q]: q
This completes your initial setup of ICON/UXV on this machine.

Do you wish to install the ICON/PICK operating environment?
[y, n, ?, q] n

ICON system installation is now complete.  You may now log in.

ICON/UXV (hostname:console)

login:
```

You have successfully installed ICON/UXV Release 4.00. Proceed to the next section to integrate system configuration information into this release.

Integrate System Information

Now that you have finished installing 4.00, you are ready to integrate old configuration information from the special backup tape you made in item 3 of the Installation Checklist. First, you will extract the old files into a working directory, **/usr/tmp.**

If you are proficient at using **diff**(1) or **comm**(1), you may want to use these tools to integrate system information into the new system. You will still want to read the tips about each system file provided below, even if you don't follow the suggestions for how you might go about integrating the old information.



If you extract the old system's files into the root directory of the new system, all your work will be lost. You will have to reinstall Release 4.00 from scratch.

Extracting Files into /usr/tmp

To extract the files from the tape into **/usr/tmp** , type

 # cd /usr/tmp

 # tar xvf /dev/sc0q3r

Integrating System Information

There are various methods of integrating system information. Which one you use depends on the type of information you are integrating or the type of file it is in.

Method 1. Some system files can be copied directly into their corresponding 4.00 directory.

Method 2. Others require that sections of them be either merged with an editor or typed by hand into new 4.00 files, or conversely, that sections of 4.00 files be merged or typed into the old file, which then is brought over into the corresponding 4.00 directory.

Method 3. And some files should just be stored in /usr/tmp for future reference.

The figure below, Methods of Integrating System Information, shows which of the three methods above you should use to integrate information from each of the files you stored on your special backup tape. The number *1* means the file can be handled with Method 1 above, that is, it can be copied directly into its corresponding 4.00 directory. The number 2 next to a file means it can be handled with Method 2 above, and likewise with the number 3. Tips on integrating these files follow the figure.

Method	File	Method	File
2	/.profile	2	/etc/passwd
1	/dev/MAKEDEV.local	2	/etc/rc
1	/usr/adm/	2	/etc/rc.local
1	/usr/mail/	2	/etc/termcap
1, 2	/usr/spool/	2	/etc/inittab
3	/usr/spool/lp	2	/etc/uxrc
3	/usr/lib/uucp/	2	/etc/rc.pick
1	/usr/local/	1	/etc/smiledisks
1	/usr/lib/terminfo/	1	/etc/smiledisks_*xy*
2	/etc/group	1	/etc/smileprinters

Figure 3-3: Methods of Integrating System Information

/.profile

If you have made any of your own changes to the old default **.profile**, you may want to add them to the new file. Before adding them, examine each entry to see if it applies to Release 4.00. For example, Release 4.00's new device-naming conventions may have to be applied to an entry before it is added to its corresponding 4.00 file.

You may also use the old **/.profile**. If you choose to do so, examine the PATH environment variable in the new system. Make sure Release 4.00 hasn't added anything to the new default PATH.

/dev/MAKEDEV.local

In the rare case that you have added something to this file, just copy the old one on top of the new one.

/usr/adm

In the rare case that you have added any files to this directory, just copy them into the new **/usr/adm** directory.

/usr/mail

If you have any files in this directory, copy them up to the new **/usr/mail** directory.

/usr/spool/

If you have added any information or files to this directory, copy them up to the new directory. (There should not have been any outstanding jobs in any printer spooling area or other directories in **/usr/spool**.)

/usr/spool/lp/

If you have added any custom interface files to **/usr/spool/lp/interface**, you will want to save them for integration into the new spooling system when you set up the new line printer spooling system. Having the old **lp** interface files will help you do the setup.

/usr/lib/uucp/

The new **uucp** configuration files are in a different format but have names similar to the old ones. You will want to have the old ones stored in **/usr/tmp** for easy reference to phone numbers, system names, et cetera.

/usr/local/

If you have added anything to this directory, copy the entire file into the new directory.

/usr/lib/terminfo/

If by any chance you have added any files to this directory, copy them in their entirety into the new directory.

If you have terminfo sources, you should recompile those using the **tic**(1M) command. Otherwise, just copying changes into the new directory may work, although it is not guaranteed. Same as /etc/passwd (below).

/etc/passwd and /etc/group

Use an editor to merge passwords, login definitions, et cetera from the files in these directories into the corresponding 4.00 files. For example, to copy passwords:

Step 1: Open both the new and old **/etc/passwd** files.

Step 2: For any login entry in the new file, delete the corresponding one in the old one, if any.

Step 3: Merge the remaining parts of the old file into the new one.

In a similar fashion, merge the group IDs from the **/usr/tmp/etc/group** file.

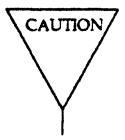
/etc/rc and /etc/rc.local

Any custom or local changes you have made to **/etc/rc** or **/etc/rc.local** in the old system should be put into the 4.00 **/etc/init.d/local**file . Note that it is not necessary to add any changes to start ICON/PWS Level II, ICON/PICK, or other optional products for your ICON/UXV system. Contact your Customer Service Representative if you are not sure if a specific change needs to be integrated.

/etc/termcap

Any changes should be added to the new **/etc/termcap** file. If the changes are too extensive and you want to use the old **/etc/termcap**, you may do so. Keep in mind that there may be some entries in the new **/etc/termcap** file that aren't in the old one. If that doesn't bother you, go ahead and use the old one. (Most programs use **terminfo** instead of **termcap**.)

/etc/inittab



You definitely cannot use the old **/etc/inittab**. The system will not boot correctly; it will just go berserk.

If you don't want to use the default 4.00 **/etc/inittab** file, you can open the old file and copy over everything after the initial **getty** entry for the **console**. For example, from the **/etc/inittab** file below, you would copy everything after and including line **01**. You would *not* copy line **co**.

```
mi::sysinit:/etc/devnm / | /etc/setmnt > /dev/console 2>&1
off:2:initdefault:
bl::bootwait:/etc/bcheckrc </dev/console >/dev/console 2>&1 #bootlog
bc::bootwait:/etc/brc 1>/dev/console 2>&1 #bootrun command
sl::wait:(rm -f /dev/syscon;ln /dev/systty /dev/syscon;) 1>/dev/console 2>&1
rc::wait:/etc/rc 1>/dev/console 2>&1 #run com
pf::powerfail:/etc/powerfail 1>/dev/console 2>&1 #power fail routines
co:2:respawn:/etc/getty console console vt100    # Main console (bottom)
01:2:respawn:/etc/getty tty01 9600 vt100         # Main line 1 (top)
02:2:pick:/etc/getty ttya0 9600 viewpoint-90     # PCP 0 line 0 (bottom)
03:2:pick:/etc/getty ttya1 9600 viewpoint-90     # PCP 0 line 1
04:2:pick:/etc/getty ttya2 9600 viewpoint-90     # PCP 0 line 2
05:2:pick:/etc/getty ttya3 9600 viewpoint-90     # PCP 0 line 3
06:2:pick:/etc/getty ttya4 9600 viewpoint-90     # PCP 0 line 4
```

You can also copy parts of individual entries from the old **/etc/inittab** file into the new one. The most critical element of an entry is the first part, the ID number.

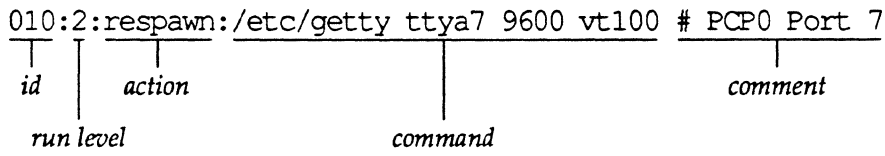


Figure 3-4: Components of an **/etc/inittab** Entry

If keyboarding an entry into the **/etc/inittab** file, take care to follow the rules governing ID numbers:

1. No two ID numbers can be alike.
2. There can be no space after an ID number.
3. ID numbers can be 1 to 4 characters in length.
4. An ID number for an **/etc/getty** command (login line) must never be changed in multiuser mode. It can be changed in single-user mode.

/etc/uxrc

Merge the entries of the new file into the old file. Systems with SMILE and DCS, will definitely want to integrate the new entries into the old **/etc/uxrc** files. If you find any entry in the new one that is not in the old one, consult the **uxrc**(4) manual page to see if you need or want it.

/etc/rc.pick

You will only have to merge entries from this file into the 4.00 file if you have the ICON/PICK Application Environment on your system. It is strongly suggested that this file be managed using the ICON/PICK administration facilities rather than integrating the contents of this file by hand.

/etc/smiledisks, /etc/smiledisks_*xy*, and /etc/smileprinters

Copy these files directly into the 4.00 directory.

Restore Data Files

To restore the data files you backed up with the **fdump** command, you will use the **sysadm restore** subcommand of Release 4.00's **sysadm** interactive "FILE MANAGEMENT" menu.

You will be using the "fdump INTERACTIVE restore option." The interactive restore allows you to **cd** to the directories on the backup tape and select only specific data file directories for extraction. This leaves the 3.*xx* system files safely off of the 4.00 system.

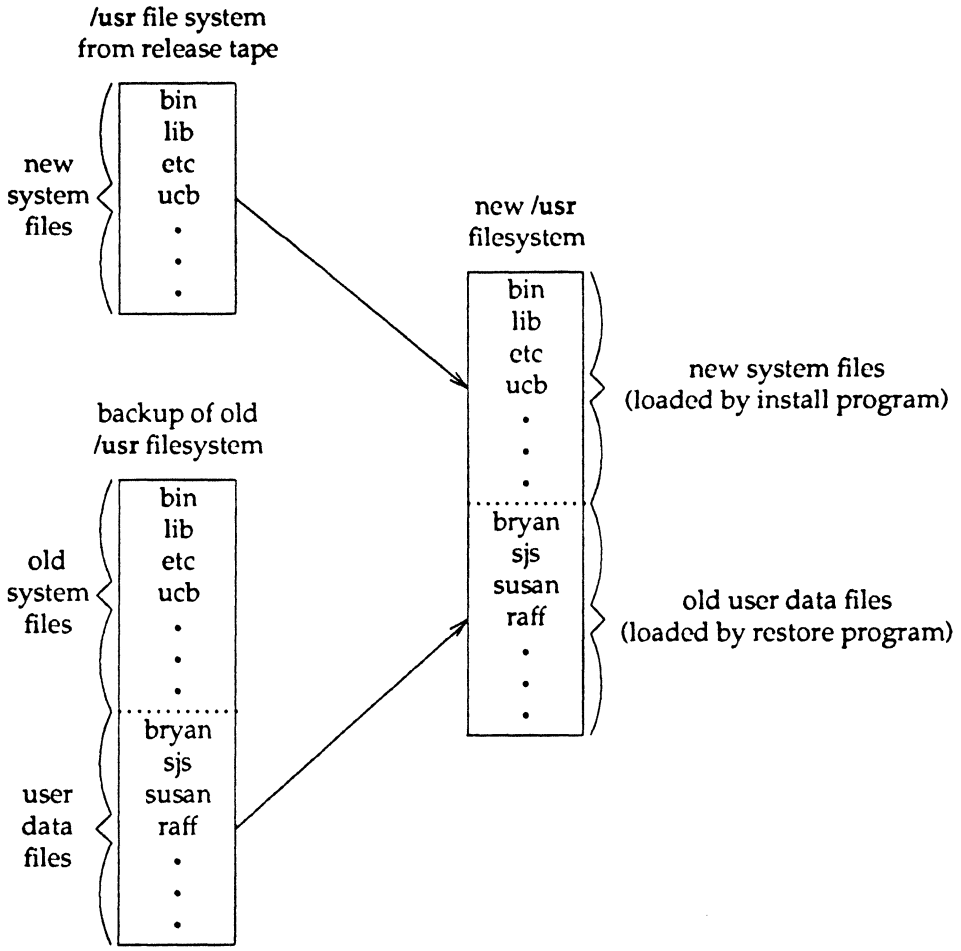


Figure 3-5: Merging Old Data Files with New System Files

The figures below list the system directories in the /usr file system and those of the / (root) file system. By using the interactive option of the restore program, we can mark only the user files directories for extraction, leaving the system file directories on the tape. If you mistakenly extract the system file directories, start the installation process over from **./sinstall**.

Directory	Contents
bin/	Basic commands and utilities
dev/	Device and special file entries
etc/	Administrative commands and utilities
lib/	Basic libraries and tools
mnt/	Standby mount point, often unused
stand/	Stand-alone utilities
swap/	Mount point for swap file system
tmp/	For temporary files, often a mount point
unix/	Backup copy of operating system kernel
usr/	Mount point for **/usr** file system

Figure 3-6: System Directories in the / (root) File System

Directory	Contents
adm/	System accounting and performance information
admin/	System administration tools (**sysadm**)
bin/	Commands and utilities
etc/	Supplemental administrative tools
ftp/	Virtual root for FTP network connections
hosts/	Networking hostname-to-remsh link directory
include/	Header files
include_gen/	Header files, NCR compatible mode (m68k only)
lbin/	Non-standard tools
lib/	Libraries and supporting tools
lib_gen/	Libraries, NCR compatible mode (m68k only)
local/	Local tools and libraries
mail/	Mail spooling directory
man/	Reference manual pages
news/	News spooling directory
oasys/	FACE tools
preserve/	**vi**/**ex** file preservation directory
pub/	Public reference files
skel/	Skeletal home directory entries
spool/	Spooling area (printer, uucp, et cetera)
src/	Source files (usually empty)
tmp/	Alternate temporary file directory
ucb/	BSD commands
vmsys/	More FACE tools

Figure 3-7: System Directories in the **/usr** File System

Restoration Procedure

To restore data files from the /usr file system, proceed as illustrated in the example below.

Step 1: Use the **restore** option to the **sysadm** command. You will see a display similar to the one on the next page. Select the correct drive to use and then select media format 4, "fdump INTERACTIVE restore."

```
# sysadm restore

Running subcommand 'store' from menu 'filemgmt',
FILE MANAGEMENT

Select which drive to use:

    1 ct0                11 mt/sc0q3f11r        20 mt/sc0q3f150t
    2 rct0               12 mt/sc0q3f11t        21 mt/sc0q3f150tr
    3 qic0               13 mt/sc0q3f11tr       22 mt/sc0q3f24
    4 rqic0              14 mt/sc0q3f120        23 mt/sc0q3f24r
    5 mt/sc0c1           15 mt/sc0q3f120r       24 mt/sc0q3f24t
    6 mt/sc0c1r          16 mt/sc0q3f120t       25 mt/sc0q3f24tr
    7 mt/sc0c1t          17 mt/sc0q3f120tr      26 mt/sc0q3r
    8 mt/sc0c1tr         18 mt/sc0q3f150        27 mt/sc0q3t
    9 mt/sc0q3           19 mt/sc0q3f150r       28 mt/sc0q3tr
   10 mt/sc0q3f11

Enter a number, a name, the initial part of a name, or ? for HELP, q to
QUIT: 26

Select media format:
1. tar
2. cpio
3. fdump
4. fdump INTERACTIVE restore
Enter a media format [?,q]: 4
```

Step 2: List the file directories on the backup tape by typing ls, as shown in the screen on the next page. Then mark your user file directories for extraction with the add command. All file directories that have been marked for extraction or that contain a file you have marked for extraction are indicated with an asterisk (*).

In addition to extracting user file directories, if you have created any custom 3.*xx* system files, you can cd to this system file directory and mark only that particular file for extraction. It will then be extracted along with the user directories when you type the extract command.

After you have extracted all data file directories and all custom system files, you will have completed the installation of this release. You may then resume normal operation.

```
.
.
.
restore > ls
adm/            include/        lost+found/     pub/            ucb/
admin/          include_gen/    mail/           skel/           vmsys/
bin/            lbin/           man/            spool/          victor/
etc/            lcr/            lib/            news/           src/
ftp/            lib_gen/        oasys/          susan/
hosts/          local/          preserve/       tmp/

restore > cd susan
restore > ls
datafile1          datafile2          datafile3          datafile4

restore > cd ..
restore > add susan
restore > ls
adm/            include/        lost+found/     pub/            ucb/
admin/          include_gen/    mail/           skel/           vmsys/
bin/            lbin/           man/            spool/          victor/
etc/            lcr/            lib/            news/           src/
ftp/            lib_gen/        oasys/         *susan/
hosts/          local/          preserve/       tmp/

restore > add lcr victor
restore > ls
adm/            include/        lost+found/     pub/            ucb/
admin/          include_gen/    mail/           skel/           vmsys/
bin/            lbin/           man/            spool/         *victor/
etc/           *lcr/            lib/            news/           src/
ftp/            lib_gen/        oasys/         *susan/
hosts/          local/          preserve/       tmp/
restore > extract

Insert the medium in the qic24 drive.  Press <RETURN> when ready.
[q] <RETURN>
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for "."? [yn] y
restore > q
Restoration procedure complete.
#
```

Chapter 4: Security Notes

Overview of Security Features

Release 4.00 introduces several new security enhancements to the ICON/UXV system. This section of the *Release Notes* provides details about some of them.

These features fall into two groups: standard features that become available automatically when you install Release 4.00, and optional features that you must install separately by using commands provided with Release 4.00. The first part of this chapter describes the standard features. Descriptions of the optional features, along with instructions for installing them, are provided in the second part.

You may also want to read further details about the editors' **.exrc** file and the **login**(1) command, both of which are important to the security of a system, in the "Compatibility Notes" section of these *Release Notes*.

Standard Security Features

/usr/spool/cron

The **/usr/spool/cron** directory that contains directories for **at**(1), **cron**(1M), and **crontab**(1) jobs is no longer accessible to users. (The directory mode will now be set to 700.) If you want to read your own file, you must use **crontab -l**.

login(1)

In the past, 10 login attempts were permitted before a line was dropped. There is now no limit on the number of tries, but the delay upon failure gets longer with each try.

login now checks for "last login" information and password aging is supported.

login.secure

If **login.secure** exists, a password is required to log in. If no password exists, it forces the user to select one. In addition, root logins are allowed only from the console.

After three failed attempts to log in, **login.secure** sleeps for 20 seconds and then exits. If the optional login log file, */usr/adm/loginlog*, exists, any three consecutive failed login attempts will be logged there.

login.secure also checks for "last login" information and supports password aging. **login.secure** must be moved to */bin/login* (and the old */bin/login* preserved) in order to use these features.

passmgmt(1M)

The **passmgmt**(1M) command does not check the system parameter MAXUID in this release. To avoid potential conflict with UIDs reserved for RFS, do not use a number larger than 60,000 as an argument to the **-u** option.

passwd(4)

Except for **root** and **setup** entries in the **/etc/passwd** file, passwords for default entries are locked on systems that have been restored (either partially or fully). This does not apply to systems on which you have done release upgrades.

Sticky Bit

The sticky bit now has meaning when set on a directory. Until now, removing a file (or directory) required the parent directory to be writable by the attempting process. Now the sticky bit is set on the parent directory. Therefore, if you want to remove a file or directory, you must be sure of two things: first, that the parent directory is writable, and second, that at least one of the following is true:

- the user must own the file,
- the user must own the parent directory,
- the file itself must be writable by the user, or
- the user must have superuser privileges.

The sticky bit on a directory is set by a regular user via the **chmod**(1) command or the **chmod**(2) system call.

uucp(1C)

The group ID (GID) for all **uucp** directories is now **uucp** instead of **sys**. If the GID on any other file is equal to 5 (**uucp**), change it to another ID.

Optional Security Features

The following features are available with Release 4.00. The shadow password file will not be installed along with the full release, but may be installed or removed as required by your applications. The enhanced **/bin/sh** is the default and will be installed by the Release 4.00 Upgrade Procedure. (It may be uninstalled if necessary.) Because these features enhance the security of your ICON/UXV Release 4.00 system, we strongly recommend that you use them.

loginlog(4)

To turn on the mechanism that logs unsuccessful attempts to access the system, the administrator must create the file **/usr/adm/loginlog**. If this file exists and five consecutive unsuccessful login attempts occur, all will be logged in **loginlog** and then **login** will sleep for 20 seconds before dropping the line. If a person makes fewer than five unsuccessful attempts, none of them will be logged.

If **loginlog** does not exist, five failed login attempts will still cause the system to sleep for 20 seconds and drop the line, but nothing will be logged.

The **loginlog** file is a text file that contains one entry for each unsuccessful attempt. Entries in **/usr/adm/loginlog** have the following format:

login name:tty specification:time

The *login name* field contains the login name used in the failed login attempt. The *tty specification* field contains the terminal location of the login attempt and *time* contains the approximate time of the login attempt.

The default status is for this file not to exist and for logging to be off. To enable logging, create the log file with read and write permission for **root** only.

Step 1: Reset the default file creation privileges in a separate shell level.

```
/bin/sh
umask 066
```

Step 2: Create the loginlog file.

```
> /usr/adm/loginlog
```

Step 3: Set the group to **sys**.

```
chgrp sys /usr/adm/loginlog
```

Step 4: Change the ownership of the file to **root**.

```
chown root /usr/adm/loginlog
```

Step 5: Return from the newly created shell level.

```
exit
```

This file may fill up quickly. To use this information and to prevent the file from getting too large, it is important to check and to clear the contents of the loginlog file occasionally.

Shadow Password File

To protect encrypted user passwords, an optional security feature allows a system administrator to move all password and aging information from the publicly readable password file **/etc/passwd** to an access-restricted file called the shadow password file. The shadow password file contains one entry per login. Each entry consists of the following information:

username	the user's login name (ID)
password	a 13-character encrypted password for the user and a *lock* string to indicate that the login is not accessible (or no string to show that there is no password for the login)
lastchanged	the number of days between January 1, 1970, and the date that the password was last modified
min	the minimum number of days required between password changes.
max	the maximum number of days the password is valid

Installing /etc/shadow

Initial conversion of a system's single password file, **/etc/passwd**, to the new scheme using two files (**/etc/passwd** and **/etc/shadow**), is done by running the privileged command **pwconv**(1M), which creates **/etc/shadow** with information from **/etc/passwd**. The command populates **/etc/shadow** with the user's login name, password, and password aging information.

Further updates of these password files should be done by the new command **passmgmt**(1M) and by the enhanced **passwd**(1) command. The **passwd** command updates the password and aging information in the appropriate password file. The **passmgmt** command is used to add or to change all other information in the password file(s). The **sysadm chgpasswd** command can also be used to update the password file(s).

The **pwconv** command may be run more than once. If the two files, **/etc/passwd** and **/etc/shadow**, should ever become inconsistent (for example, because someone manually changes one of the files) they may be made consistent by running **pwconv** again.

If password aging information does not exist in **/etc/passwd** for a given user, none will be added to **/etc/shadow** but the "last changed" information will be updated. This occurs only when **pwconv** creates **/etc/shadow** or adds an entry not previously in the shadow password file.

Backing out /etc/shadow

Certain applications may not work with the new security password file changes. A possible indication of this problem is that you will be unable to log in.

If you are running such an application, you may have to "back out" the shadow password change so that this application will run. The **pwunconv**(1M) command (**/usr/bin/pwunconv**) accomplishes the reverse of the **pwconv** command. This command converts an ICON/UXV system from the two-password file scheme back to the one-password file scheme. System administrators can run this command to solve compatibility problems caused by the introduction of the shadow password file.

Chapter 5: Software Notes

Software Notes

This section offers some additional information about ICON/UXV Release 4.00. Notes about commands, system calls, and files are listed alphabetically and are organized by the reference manual where those commands, system calls, and files appear. For example, the section "User Commands" contains notes about commands that are listed in the user's reference part (first half) of the *System Administrator's Reference Manual*. Any further problems may be resolved through the Customer Support Center according to the terms of your maintenance contract.

In this chapter, commands are listed as many times as there are concept to be conveyed. For example, **bc**(1) is listed three times.

User Commands

awk(1)

An **awk**(1) command link that contains pipes may not be processed in the same sequence shown in the script.

For example, the following **awk**(1) script

```
BEGIN {
        print "Header Goes Here";
}
{
        printf("%s\t%s\n", $1, $2) | "sort -ru" ;
}
END {
        printf("%s\n", "Trailer Goes Here");
}
```

will produce

```
Header Goes Here
Trailer Goes Here
output from sort
```

when executed.

bc(1)

When you enter the command line

bc *file1 file2*

and **bc** cannot open *file2*, the error message displayed says that **bc** cannot open *file1*.

Also, when you enter the command line

bc *file*

and **bc** cannot open *file*, the error message displayed contains garbled characters instead of the name of *file*:

bc(1)

bc does not handle the following two constructs in the same way:

```
(1)     if ( expr ) {
        ...
        }

(2)     if ( expr )
        {
        ...
        }
```

The first case produces what one would expect. The second case is equivalent to an **if** followed by an empty statement, and the compound statement always is executed. If nothing else, the second case should produce a syntax error, but it does not. It dumps core silently.

bc(1)

The **bc** command prints an incorrect remainder when dividing with a negative number.

```
$ bc
-1.2 % 1
9.8
<control-d>
```

Clearly, when dividing -1.2 by 1, the remainder that returns should not be 9.8.

bdiff(1)

The command **bdiff** causes a bus error when it is run.

cdc(1)

The **cdc**(1) command ends abnormally when you invoke it without the **-m** option on an SCCS file that does not have the **v** flag set.

cpio(1)

cpio overwrites files when invoked with the **-u** option. If an I/O error occurs while reading the archive or if the file system runs out of space, the file being created is either corrupted or truncated, and the original file is lost because it has been overwritten.

cpio(1)

To ensure that a file either in an archive (**cpio -i**) or being copied (**cpio -p**) does not exceed the **ulimit**, **cpio** calculates the **ulimit** for the current user. To do this, **cpio** takes the value returned by **ulimit**(2) (a long), and multiplies that by 512; the result is the file's size in bytes. If the **ulimit** was changed so much before the **cpio** command was invoked that an overflow occurs during the multiplication by 512, many of the files in the archive will be skipped for exceeding the **ulimit**.

To prevent this problem from occurring, keep the **ulimit** below the value that causes an overflow in a long variable when multiplied by 512 (for an ICON computer, this is about 2^{22}*512).

cpio(1)

Currently when **cpio** reaches end-of-media for a device, it assumes that the previous write successfully completed.

ct(1C)

When ct(1C) uses a port on which a **uugetty**(1M) is running, the terminal being called displays the message

```
Hit carriage return
```

Then the line is dropped; no connection is made. **ct** will only work on ports which have no **uugetty**.

cu(1C)

The first invocation of **cu** after a power-up may fail. If this happens, the error message `cannot access device` will be displayed. Later invocations of **cu** will succeed.

cu(1C)

If you do a **cu** from machine_A to machine_B, and then do a **cu** from machine_B to machine_C, the command **~%take** *file* will not transfer *file* to machine_A.

Transfer files over one link at a time by one of two methods.

- Log in to machine_A; **cu** to machine_B and then to machine_C as described above. Run the command **~%take** *file*; this will transfer *file* to machine_B. Then type **~.**. From machine_B, use the command **~%take** *file* to transfer *file* to machine_A.
- If possible, **cu** from machine_A to machine_C. You can successfully transfer files over a single link.

cu(1C)

Occasionally, even though all devices are available, **cu** fails and the following message is displayed:

```
NO DEVICES AVAILABLE
```

If this happens, execute **cu** again.

cu(1C)

If there is a command in **/etc/inittab** that is respawning too rapidly and it generates an error message while the console user does a **cu** to another system, the console user will find that every command typed is echoed twice on the console, although it is only executed once (as it should be).

cu(1C)

If, while the **cu**(1C) command is being used to call a remote system, a disconnect occurs during use of the ~![*cmd*] escape feature, all processes initiated during the escape will continue to run. Control will be returned to the process that initiated the **cu**.

Depending on the nature of the process(es), this problem may become apparent as a loss of characters on input and/or as unpredictable terminal behavior.

To return the system to a clean state, you must terminate all appropriate processes created during the escape. The easiest way to do this is by logging off, or, in the case of a multiple window terminal, by deleting the affected window.

Another way to return the system to a clean state is to send signal 1 (**SIGHUP**) to the appropriate processes by running the **kill**(1) command.

cut(1)

The **cut**(1) command succeeds (instead of returning an error) if CHRCLASS is set, and if a string without a newline is echoed and piped to **cut** as shown in the example below:

```
echo 'abcdef\c' | cut -c2
```

If CHRCLASS is set, **cut** returns

```
b
```

If CHRCLASS is not set, **cut** returns the following error message:

```
cut: ERROR: Line too long
```

This problem also occurs with the **-d** and **-f** options.

cut(1)

The command **cut** has difficulty cutting a single column. For example, if you give it the command **tty | cut -c8-8**, instead of printing "y" it prints nothing.

ed(1)

The error message for the **k** command should specify that only ASCII lower-case letters should be used to mark lines.

ipcs(1)

ipcs(1) always reports the number of processes attached to shared memory segments (NATTCH) as zero, even when running processes are currently attached to shared memory segments.

layers(1)

If the **layers** command detects an error on a line in a **layers** startup file, (i.e. the file used as the argument to **-f**), processing of the file will end at that line. The rest of the lines in the file will not be processed.

If this happens, exit layers, correct the error, and invoke **layers** again.

login(1)

After **exec login** is executed, the following warning message is produced intermittently:

```
no utmp entry ... execute from the lowest level shell
```

If the **who** command shows that someone is still logged in on the line listed, you must kill the **getty** and allow it to respawn. Otherwise no action is required.

You can avoid the problem by executing **exec su -** *xxxx* (where *xxxx* stands for a login), or by logging out (that is, by hanging up or executing an **exit** command) and calling back in.

login(1)

After overnight backups are made, the times on the inode associated with the *$HOME/.lastlogin* file are altered. This causes an annoying "Warning: .lastlogin modified since last login" message.

lp(1)

Any lines longer than 131 characters in files sent to printers that use the **prx** model interface program will be truncated to 131 characters.

lp(1)

The **lp**(1) command places a limit of 99 copies on any print request.

ls(1)

The **ls** command prints an incorrect value for the create time of a file if the user or group ID in **/etc/passwd** or **/etc/group** contains more than 16 characters.

mail(1)

A user who executes **mail** as **root** will not be able to save mail messages in the root directory unless a file for this purpose already exists.

For example, if the command **s /foo** is issued and **/foo** doesn't exist, the user will get an error that the file can't be created.

There are two possible workarounds for this. The first is to give a relative pathname for the file (i.e., **s foo**). This assumes that the user was in the root directory when mail was invoked. If this isn't the case, the user can escape to the shell and create the file by typing **!>/foo** at the mail prompt. Once the file has been created, the message can be saved.

mailx(1)

When the "reply" or "Reply" command is used, **mailx** attempts to build the return address by looking at the headers in the mail message. The headers in some messages are apparently created in such a way that **mailx** does not determine the return address correctly.

If the wrong return address results from using either of these two commands, use the **m** command to reply and manually type the return path to send mail.

mailx(1)

The command **mailx -f** *a/b*, where *a* is a nonexistent directory, causes **mailx** to pause for a long time, as it continuously tries to access the nonexistent directory. Eventually, **mailx** will time out.

The argument supplied to the **mailx -f** option must be a valid directory.

mailx(1)

The commands **mail** and **mailx** do not clean up their lock files.

nice(1)

The command **nice**(1) uses **/bin/sh** to produce a shell that will execute a user's command(s). The shell provided in the environment variable *SHELL* is ignored.

od(1)

If a file has an odd number of bytes, **od -c** reports a trailing null byte.

passwd(1)

The **sysadm** help file for setting a password does not warn of the eight-character limit on passwords. Thus, a user advised to add a numeric character to a password such as "abcdefgh" may then choose "abcdefgh9." This password will also be rejected, however, because the number does not appear within the first eight characters.

pg(1)

The regular expression used with the "+" option should not contain a "/" (slash). The command will assume that the slash in the regular expression terminates the expression, even if escaped with a "\" (back slash).

The following, for example, is invalid:

```
pg +/string\/tail/
```

pr(1)

When **pr** is used to print a file containing lines over 80-characters long (the column width), for example,

```
pr -n -i -w80 -l60 0f file | lp
```

the printer prints a new line and carriage return <CR> and then prints the rest of the line. This causes the printer to print more than the specified number of lines on a page. If more than four or five lines wrap around in this manner, the form feed issued after printing the page causes a blank page to appear in the listing.

ps(1)

If the */etc/ps_data* file is missing, only root can re-create it.

sdiff(1)

Do not **sdiff** a file that contains more than 196 characters in a single line. If you do, the **sdiff** output will lose the separator symbol (|) loop infinitely, or do both.

sh(1)

Using **sh**| |does not work correctly with built-in commands. For example, the command sequence

```
cd nowhere| |date
```

generates an error and does not execute **date**. This example can be executed without using| |.

```
cd nowhere
if [ "$?" != "0" ]
then
date
fi
```

sh(1)

The **-s** flag is set if **sh** is not executing commands from a file. The **-s** flag is not set if **sh** is executing commands from a file. For example,

sh -c ' echo $-'

will echo s, but

echo ' echo $-' > *filename*; **chmod +x** *filename*; **sh -c '** *filename*

will not echo s.

sh(1)

The **-t** option causes the shell to exit after reading and executing one command. The command must end with a newline (carriage return). Commands separated by one of the characters:

; & && ||

are read as one command by the shell. For example,

sh -ct "*command_1*; *command_2*"

will execute both *command_1* and *command_2*. If, however, the semicolon (;) is replaced with a newline:

sh -ct "*command_1*
command_2"

only *command_1* will be executed.

NOTE

A statement such as an `if` statement or a `while` statement is considered a single command in the shell. For a list of these compound statements, see **sh**(1) in the *System Administrator's Reference Manual.*

shl(1)

If you hang up while working in shell layers, /etc/utmp may not be cleaned up (the **who** command will show that you are still logged in, and the ps command will show a **getty** running on that line). If someone else then calls into that line, the **login** fails because the **utmp** entry cannot be found. To kill the **getty** corresponding to the affected line, follow this procedure:

Step 1: Execute **ps -ef** to find the process ID of the **getty** running on the line that has been reported, erroneously, as occupied.

Step 2: Execute **kill -9** *PID* to kill the **getty**. When the **getty** automatically respawns, the problem will be cleared.

shl(1)

When using the **shl** command, trapping a "hang-up" signal inside a layered shell suspends the layer. This renders the device unusable until the shell is killed. You receive a warning message that a layer is still running, and then you are returned to the ICON/UXV system shell.

If your terminal does not accept input commands because of the suspended layer, you must log in at another terminal. To kill the suspended shell, follow this procedure:

Step 1: Find the process number of the shell by executing the following command:

ps -eaf

Step 2: Look at the output of the command for the appropriate tty (for example, sxt001) and for the process number of the shell.

Step 3: Enter the following command:

kill -9 *PID*

where *PID* is the process number of the shell that you want to kill.

shl(1)

The shell layer does not respond properly if you are at the console in **shl** and you execute the following sequence of commands:

1. an **stty** command with a carriage return select style argument (for example, **stty cr3**)
2. an **echo** command
3. an **stty** command with a carriage-return select style argument (for example, **stty cr0**)

In particular, the layer's prompt does not return until the <BREAK> or <DEL> key is hit. Once the prompt appears, the output of any command executed will not be printed fully or at all until one or more carriage returns are entered.

Return to the **shl** control layer and use the **shl** delete command to delete the layer.

shl(1)

If you execute a background process that is not in a shell layer but that sends output to your terminal, and then you enter shell layers (**shl**(1)), some screen output from the original process may be lost. When you enter **shl**, the output from the original process temporarily stops printing on the screen. The screen output resumes when you exit **shl**(1). Loss of data, if it occurs, is noticeable when the screen output resumes.

This is not normal use of the system, and should be avoided.

shl(1)

When resuming a layer of **shl**, the prompt `resuming` *xyz* is often garbled.

spell(1)

The file */usr/lib/spell/hstop* was built from the source for **hlistb** instead of the source for **hstop**.

stty(1)

stty(1) uses 0377 as the undefined control character. For example, **stty erase '^-'** will set the erase character to 0377. This is undesirable in standard character sets where 0377 is a defined character (y-umlaut).

Also, if one types **stty erase** <*y-umlaut*>, **stty** produces the following output:

```
speed 4800 baud; -parity hupcl
erase <undef>; swtch = ^';
brkint -inpck -istrip icrnl onlcr tab3
echo echoe echok
```

Instead of using the undefined control character, you can use other eight-bit control characters. For example,

stty erase `echo '\0210'`

will set the erase character to 0210 (an 8-bit backspace on some terminals).

uucp(1C)

In the **/usr/lib/uucp/Permissions** file, read/write/execute permissions are defined for each **uucp** login. These permissions authorize remote hosts to read/write/execute local files using **uucp** and **uux** commands. Currently, a file cannot be made readable by giving it read permission in the **Permissions** file.

To make a file readable, authorize both read and write permissions for it.

uucp(1C)

uucp and **uux** are supposed to allow the use of a tilde (~) with a login name to designate your home directory (for example, ~mxb for mxb's home directory). This works in most cases, but **uux** does not handle tilde expansion in **/usr/lib/uucp/Permissions** file correctly.

Use the full pathname for the home directory in the **Permissions** file.

uucp(1C)

uucp logins, such as **uucp** and **nuucp**, should have distinct user ID numbers. In the **/usr/lib/uucp/Permissions** file, different read/write/execute permissions are defined for each **uucp** login. Login names in this file are used to associate permissions with the login. These permissions do not work correctly if more than one **uucp** login has the same user ID. For example, if **uucpa** and **uucpb** both have user ID *10*, and **uucpa** appears before **uucpb** in the /usr/lib/uucp/Permissions file, then regardless of whether **uucp** logs in as **uucpa** or **uucpb**, the permissions will be those of the first user ID (**uucpa**) that matches the user ID requested. If the user Logs in as **uucpb** (which has user ID *10*), the permissions will be those of **uucpa**.

To avoid this problem, use distinct user IDs for **uucp** logins.

uucp(1C)

In Chapter 9 of the *System Administrator's Guide*, ("Basic Networking"), the section "Supporting Data Base" contains the following erroneous example of the *time* field of the **Systems** file used by **uucp**:

```
Wk 1700-0800, Sa, Su
```

For **uucp** to work properly, the spaces should be removed. The corrected example reads:

```
Wk1700-0800,Sa,Su
```

Do not separate subfields with spaces in the *time* field of the Systems file.

uulog(1C)

If you specify the f or number option with **uulog** (see **uucp**(1C)), you do not receive a prompt after the execution of **uulog**. To regain access to your terminal, press the <BREAK> key.

uuto(1C)

The manual page for the **uuto**(1C) command describes the destination as **system!***user*, implying that only a single system may be specified.

The manual page should indicate that the destination may also be a list of systems known to **uucp** by presenting it in the following format:

system!...!system!*user*

uuxqt(1M)

Documentation for the **uux**(1C) command indicates that the sequence ~*xxx* may be used on the command line, and will be expanded to the full pathname of the login directory of user *xxx* on the remote system. At this time **uuxqt**(1M), which processes the request on the remote system, does not expand the ~*xxx* sequence. Additionally, **uuxqt** does not properly expand the ~/ sequence to the expected **/usr/spool/uucppublic/** directory.

The full pathname must be specified on the local system for all commands and files used in a **uux** command.

vi(1)

vi ignores a character corresponding to octal value 377 when it is typed as input to a file.

While this is not a problem for many terminals, it does affect terminals that support European code sets in which octal 377 represents a printable character. If you press the key corresponding to octal 377 (y-umlaut) while working in **vi** on such a terminal, there will be no effect.

Note that this problem occurs only when attempting to type the 377 character from a terminal (i.e. when standard input for **vi** is a terminal). **vi** accepts octal 377 as input if standard input has been redirected.

vi(1)

If when editing a file, you write a new file (e.g., **:w** *newfile*) and then type **ZZ**, the editing session will not be reflected in the original file.

To ensure that the original file is updated, use the **:w** command with the original file name before typing **ZZ** to end the session.

vi(1)

The **vi**(1) manual page makes no mention of the special notation that **vi** uses to display "unprintable" characters. When an ASCII control character sequence is typed as input to a file, **vi** will display it as an alphanumeric character preceded by a caret (^) For example, typing **control-a** as input in **vi** will result in ^A being displayed.

If eight-bit characters (which have no ASCII equivalents) are not classified as prinatble, **vi** displayus them using octal notation (see **ctype**(3C)).

Finally, the manual page doesn't mention that tilde (~) is a command in **vi** that will toggle the current character between lower and upper case.

vi(1)

Using a named buffer twice in a **vi** map sequence results in the following error message:

```
Can't put partial line inside macro
```

Use named buffers only once in map sequences.

vi(1)

When the **-r** or **-L** options of **vi** are used, **vi** searches for saved files in only two directories: **/tmp** and **/usr/preserve**. If a user has the following line in the editor initialization file (**$HOME/.exrc**):

```
directory=/usr/tmp
```

lost files could end up in **/usr/tmp** and would not be retrieved automatically.

The lost file(s) could be copied to **/tmp** manually. This would make them retrievable with the **vi** command.

wc(1)

When input to **wc** is not redirected from keyboard input (not redirected from a file), you must type two EOF sequences (**control-d**) to signal the end of input.

System Administrator Commands

Administrative Login Passwords

When you request information about administrative login passwords, the following message appears on the terminal:

```
For more information about passwords and their use,
read the SECURITY chapter of the O/O manual.
For more about assigning passwords, see the chapter on
SIMPLIFIED SYSTEM ADMINISTRATION.
```

The reference to the "Simplified System Administration" is incorrect. Refer instead to the paragraph "System Administration Menus" in Chapter 2 of the *Release Notes*.

ff(1M)

The **ff** command lists the options **U** and **S** in the usage message. These options don't exist.

finc(1M)

The following **finc** error message may be displayed during a backup to cartridge tape:

```
Error occurred, error #=??
```

The actual error number is echoed to the console. No further information is provided. Check the console for the error number, and look it up in the *System Administrator's Guide.*

frec(1M)

The **frec** command will abort the recovery if the files being recovered are not on the first reel of a multi-reel **volcopy** backup.

To work around this problem, use **volcopy**(1M) to read the entire backup into a temporary file system and then recover the files from the temporary file system.

fuser(1M)

The **fuser** manual page lists the following letter codes that you can use to indicate which processes you want to print:

c	current directory in use by process
p	parent directory in use by process
r	root directory in use by process

this list is no longer correct: the **p** letter code is no longer supported, and other options have been added. The correct list of letter codes is:

c	current directory in use by process
r	root directory in use by process
S	server process
U	receive descriptor in use by process
t	text busy process
s	process sleeping on receive descriptor

fuser(1M)

The command **fuser -k** *file1 file2* does not kill all processes using *file2*; It kills the processes using *file1* (as specified on the manual page), but does not kill the processes used by *file2* because the **-k** flag is turned off before those processes are found.

To kill all processes using both *file1* and *file2* use the following command line:

fuser -k *file1* **-k** *file2*

getty(1M)

If a **getty** is spawned from **/etc/inittab** on a non-existent terminal, it will fail after printing an appropriate error message. However, if this **getty** fails before the console **getty** opens the console terminal, it will be the first process to open it, and the console will become the controlling tty for the failed **getty**. This action prevents the console **getty** from getting the console as the controlling tty. As a result, you will be prevented from logging into the console with the following error message `Login incorrect`.

To correct this problem, log in to the contty or one of the ports, edit **/etc/inittab** to turn off the **getty** producing the error message, and re-boot the machine.

infocmp(1M)

The terminfo database contains descriptions of some terminals; the termcap database contains no such descriptions. You can create corresponding entries for the termcap database by using the **infocmp** command. See **infocmp**(1M) in the *System Administrator's Reference Manual.*

init(1M)

init cannot parse a shell metacharacter unless it is preceded by a backslash. If you use metacharacters in your specifications in **/etc/TIMEZONE**, be sure to precede each one with a backslash. For a list of shell metacharacters, see the **sh**(1) manual page.

init(1M)

Whenever **init** prints a message on the console to show a change in run-level or to indicate that a process is respawning rapidly, it resets the console terminal settings to the default values.

init(1M)

/etc/init is unable to spawn gettys if a string of the form:

```
TZ=CST6CDT; export TZ
```

is in **/etc/TIMEZONE**. When **/etc/init** parses **/etc/TIMEZONE** in order to set the environment variable **TZ**, it expects the statements above to be on two separate lines as follows:

```
TZ=CST6CDT
export TZ
```

init(1M)

When entered from a remote terminal, **init S** drops the remote line and changes the speed to 9600 baud.

The **init** command should be executed only from the console.

init(1M)

Executing **init s** from within **shl** layers causes inconsistent results. Sometimes the machine hangs after printing the following message:

```
INIT: SINGLE USER MODE
```

Other times, the system may not really change run states. Within **shl** layers, **init s** never does what it is supposed to do.

lp(1) Commands

To make the **disable** and **enable** commands non-executable by regular users, log in as **root** or **lp** and execute the following commands:

```
chmod -s /usr/bin/disable
chmod -s /usr/bin/enable
```

NOTE

It is no longer necessary to make the **cancel** command non-executable by regular users because it has been changed to allow them to cancel only their own jobs. Only **lp** and **root** may cancel any job.

The following commands have the effective group ID bit turned off in this release:

/usr/bin/cancel
/usr/bin/disable
/usr/bin/enable
/usr/bin/lp
/usr/bin/lpstat
/usr/lib/accept
/usr/lib/lpadmin
/usr/lib/lpmove
/usr/lib/lpsched
/usr/lib/lpshut
/usr/lib/reject

lp(1)

When a form or print wheel alert needs to be mounted on a printer, a message is sent to the system administrator, alerting him or her to the need to do so. This alert message includes a list of printers, the number of requests queued for each printer, and the total number of requests awaiting the form or print wheel. The message may be misleading because the number of requests queued for each printer may add up to a total larger than the number of requests queued. The total number of requests queued is always correct. The number of requests queued for each printer reflects the method by which the requests were originally submitted. Requests submitted to a class or to the special destination **any** will be counted for each printer on which they may be printed.

For example, assume there are three printers - **lp1**, **lp2**, and **lp3** - and that two of them (**lp1** and **lp2**) are members of class **LP** and that all three printers allow the form **checks** and the print wheel **accountant**. Three jobs are currently in the queue, as shown below:

```
# lpstat -o -l
any-1                   root              218   Feb  8 11:09
        assigned lp1, form checks, charset accountant
LP-2                    root              218   Feb  8 11:09
        assigned lp1, form checks, charset accountant
lp2-3                   root              218   Feb  8 11:12
        queued for lp2, form checks, charset accountant
```

An alert sent in this situation would appear as follows:

```
>From lp Mon Feb  8 11:17 EST 1988
Subject: Mount form checks

The form checks needs to be mounted
on the printer(s):
        lp1             (2 requests)
        lp2             (3 requests)
        lp3             (1 request)
3 print requests await this form.
Use any ribbon.
Use the accountant print wheel, if appropriate.
```

Printer **lp1** has two requests because the request for the special destination **any** and the request for the class **LP** may be printed on it. These two requests are also queued for printer **lp2**, in addition to a request specifically for **lp2**. Only one request (the one queued for the special destination **any**) is queued for printer **lp3**.

The number of requests listed after a printer name is equal to the total number of requests queued that may be printed on that printer.

NOTE

Mounting a form/print wheel stops the alert for that form/print wheel even though some requests may still exist that should cause an alert to occur.

Using the previous example, if the form is mounted on **lp1**, requests **LP-2** and **any-1** will print, but request **lp2-3** will not.

lp(1)

This note applies to three types of items that you can add to the LP print service: forms, printers, and print wheel alerts. The limitation of 40 items applies to each type of item individually; you can add up to 40 forms, 40 printers, or 40 print wheel alerts. (The 40-item maximum does not refer to a combination of forms, printers, and print wheel alerts.)

The LP print service allows you to add up to 40 new forms, 40 new printers, or 40 new print wheel alerts, in addition to the number of similar items already defined for your system at the time the print service was started. If you want to add more than 40 items (beyond the number defined for your system at the time the print service was enabled), you must stop and restart the print service. Because new items can be added to the print service while it is stopped, we suggest you do the following if you want to add over 40 new items to your print service during one session:

1. stop the print service
2. add as many new items as you like
3. restart the print service

For example, if you want to make a total of 60 forms available on a system for which 10 forms have already been defined, shut down the print service, add the additional 50 forms, and restart the print service.

If you attempt to add more than 40 new items while the print service is still running, you will be warned that the additional items cannot be used. The new items, however, will be stored on disk but the status produced by the **lpstat** command may not appear consistent with the service available to you until you shut down and restart the print service. For example, if you add more than 40 new printers and run `lpstat -v`, the command output will indicate that all new printers are available. In fact, the information about those printers will be stored on disk, but the printers themselves will not be available until you have stopped and restarted the print service.

To stop the print service, use the **/usr/lib/lpshut** command; to restart it, use the **/usr/lib/lpsched** command.

lp(1)

The LP error messages do not exactly match those in the "Error Messages" section (Appendix C) of the *System Administrator's Guide*. At the same time, the existing LP error messages are detailed enough to be self-explanatory.

lp(1)

The dial-out feature of the new LP Spooling Utilities, whereby a printer can be connected via a dial-up modem or a network, drops the connection after each print request has been printed. This allows the LP Spooling Utilities on one machine to share the printer with similar print services on other machines.

shutdown(1M)

Sometimes when you **shutdown** to single-user mode, the unmount of the **/usr** file system fails with the busy error. Manually unmount **/usr**.

SQFILE

Do not create a **/usr/lib/uucp/SQFILE** file. If this file is present on one or both machines trying to communicate using **uucp**, **uucp** will fail.

sysadm(1)

The **sysadm** help file for setting a password does not warn of the eight character limit on passwords. Someone who chooses a password "abcdefgh" is told that the password requires a numeric character, but the password is rejected when "abcdefgh9" is chosen because the number does not appear within the first eight characters.

sysadm(1)

There is no protection provided against multiple users using the same **sysadm** subcommand at the same time.

If you use **sysadm** for floppy diskette or for cartridge tape handling, you must be certain you have control of the appropriate drives. This is no different from using tape drives on larger ICON/UXV system machines. If the session involves changing administrative files, the problem is probably one of system management; specifically, only one person should be authorized to make changes to **uucp** and to **passwd** files.

sysadm(1) backup

The **sysadm backup** procedure sometimes prints a failure message before it prints one saying that the procedure succeeded. In this case, ignore the message saying that the backup succeeded.

sysadm(1) backup

During a **sysadm backup**, if a bad block is encountered while writing on a tape, it is possible that you will not be warned of the problem. Furthermore, when a **sysadm restore** is executed, the data from the point of the bad block throughout the remaining backup tapes may be unreadable.

You should make frequent individual file or directory backups, with periodic complete backups. It is strongly recommended that you verify a file or a directory that you back up. The file and directory storage commands are available in the **store** menu under the **filemgmt** menu of the System Administration Menus.

sysadm(1) portmgmt delete

After executing **sysadm portmgmt delete**, the **inittab** entry for the port is not returned to a usable state. For example, if you connect the modem to tty21 and you execute **portmgmt delete**, the **/etc/inittab** file entry for tty21 looks similar to the following line:

```
21:2:respawn:/etc/getty -t 60 tty21 1200H
```

To change the entry to a usable state, log in as **root**, edit the **/etc/inittab** file, and change the entry for tty21 to look like the following line:

```
21:2:off:/etc/getty tty21 1200
```

sysadm(1) portmgmt modify

If you are connecting a modem to a port that had a terminal connected to it, **sysadm portmgmt modify** may not start the **uugetty**. Before connecting a modem to the port, you should check to see if a **getty** is running on the port. Execute the following command and look for the process number of the port to which you want to connect the modem. You can identify the port by its tty number (for example, tty14 or tty22).

ps -eaf

After you have identified the process number, execute the following command.

kill -9 *PID*

where *PID* is the process number. You can now connect a modem to the port. The **portmgmt modify** command should execute properly.

sysadm(1) syspasswd

The command **sysadm**(1) **syspasswd** does not change the password of system logins. The command is used only to set initial passwords to system logins.

To change a system login password you must login, either as that ID or as **root**, and use the **passwd**(1) command.

sysadm(1) uucpmgmt

When setting up a bi-directional line for uucp, it looks for a "9600H" login in *gettydefs* rather than finding 9600UUCP. To correct this mistake, the *inittab* file can be edited to change the 9600H entry to 9600UUCP.

sysadm(1) uucpmgmt

sysadm uucpmgmt and **sysadm devicemgmt** use the last two digits of the **tty** name for the ID field in the **/etc/inittab** file. Thus, the same ID field is created for tty11 and tty111, which causes **init** a problem. This problem occurs only when you have 10 or more ports boards, and you can get around it by editing **/etc/inittab** manually to change the first field.

sysadm(1) uucpmgmt

sysadm uucpmgmt states that you can depress the <CR> key to select the default speed (contty). If you depress the <CR> key, an error message is returned saying that the default speed is not found in the **gettydefs** file. Instead of selecting the default speed, you need to enter a baud rate, for example, 300, 1200, or 9600.

umount(1M)

umount tries to map a mount point to a block special device or a remote resource using **/etc/fstab** instead of **/etc/mnttab**.

/etc/fstab lists defaults for mount (and other file system related commands), so it may list **/dev/disk1** as the device usually mounted at **/usr**. If a user manually mounts a different disk (say **/dev/disk7**) at **/usr**, then a `umount /usr` would incorrectly map to **/dev/disk1** as listed in **/etc/fstab** instead of **/dev/disk7**.

To avoid faulty mapping of the associated device, **unmount** file systems by their device/resource name.

uucheck(1M)

uucheck without any options prints nothing; you should always use the **-v** option with **uucheck**.

You cannot ask for different levels of debugging information with **uucheck -x**.

uucheck(1M), uucleanup(1M), Uutry(1M)

Most of the Basic Networking Utilities commands can be executed by users. The exceptions are **uucheck** and **uucleanup**, which require either an administrative (**uucp**) login or a **root** login.

uucheck, **uucleanup**, and **Uutry** are located in the **/usr/lib/uucp** directory, which is not in the search path for most logins, including those for **uucp** or **root**. Therefore, you must give the full pathname, or you must be in the **/usr/lib/uucp** directory to execute these three commands.

Another alternative is to link the command where it may be easily accessed, for example, **/usr/bin**.

uucico(1M)

It is possible to get **uucico** into a runaway state when you use it through the STARLAN network under an extremely heavy network load. For example, the process may accumulate too much time (2500 minutes of CPU time).

Suppose on machine_A you have a **uucico** to machine_B with suspiciously high CPU time. Log in to machine_B and execute **ps -ef**. Look at the output of the **ps** command for a **uucico** process talking to machine_A (that is, with a command line argument such as `-s machine_A`).

If there is such a process, the connection is still active. If there is not, the **uucico** on machine_A is in a runaway state.

If **uucico** is in a runaway state, take the following steps:

Step 1: Kill the **uucico** process with **kill -9** *PID*, where *PID* is the process ID of the looping **uucico**.

Step 2: Remove the associated lock file with

rm /usr/spool/locks/LCK..*machine*

where *machine* is the system where the **uucico** originated, for example, machine_A.

Uutry(1M)

When you are using the Basic Networking Utilities over a transport provider, and a remote system listens on an address different from that in the local **Systems** file, trying to **Uutry** to the remote system results in the following error message:

```
Connect failed: NO DEVICES AVAILABLE
```

This message does not necessarily imply that there are no available devices on the local system. However, it does mean that **Uutry** has failed after opening a device and before achieving a connection.

This failure could be caused by a variety of problems, including no devices available on the local system or the address in the **Systems** file being incorrect. To see the local devices that are in use, type **uustat -p**.

volcopy(1M)

While executing **volcopy**(1M), you have the option of hitting **DEL** to obtain a shell. On exiting the shell, **volcopy** fails, dumps core, and prints the error message: `bus error- core dumped`

volcopy(1M)

Even though **volcopy -y** is supposed to answer all the questions **volcopy** asks affirmatively, you still have to reply to the first question. **-y** is an undocumented option.

volcopy(1M)

volcopy prompts you for several questions; however, if **volcopy** is run in the background, the prompts do not appear on your screen even though volcopy is still waiting for responses from your terminal. Do not run **volcopy** in the background.

whodo(1)

The **whodo** command will fail if the */etc/ps_data* file is missing.

Programmer Commands, System Calls

/etc/TIMEZONE

To make use of the extended timezone functionality for internationalization, the timezone specifications in **/etc/TIMEZONE** must be enclosed in double quotation marks ("). This is necessary because semi-colons are part of the specification, and when the shell sees them, it interprets them as the end of a command. See **ctime**(3C), **cftime**(4), **timezone**(4), and **environ**(5) in the *Programmer's Reference Manual*.

curses(3x)

To facilitate debugging of **curses** applications, passing an invalid window pointer to a **curses** function will probably cause a core dump. This problem can be circumvented by testing the return code of the function that created the window pointer.

curses(3x)

keypad, meta, slk_clear, and **slk_refresh** change the timestamp of the tty and can change the output of the **scr_dump, scr_init,** and **scr_restore** functions. The command **rcsmerge** fails because the file */usr/lib/diff3* is not found.

terminfo(4)

If an 8-bit terminal needs an escape sequence that requires \0200, the sequence cannot be coded because \0200 in such a sequence is treated like a null character (\0).

Miscellany

binstall

If **binstall** has not been loaded onto HSMD drives, they function as if they had zero (0) space on them. A message to this effect will be provided in a future release.

console

If you are running **shl**(1) on the console, and you then run **shutdown -is**, the console hangs.

Exit **shl** before executing **shutdown**(1M).

lp(1) use of terminfo(4)

The LP print service needs a string capability, currently not defined in **terminfo**(4), that sets the length of a page. As a workaround, the string capability **u9** is being used. When adding a new type of printer to the **terminfo** database, define **u9** to be a control sequence that sets the page length to the value of the first assignment (in lines). For information on adding **terminfo** entries, see the "curses/terminfo" chapter in the *Programmer's Guide.*

Color Terminals Supported

The following color terminals and emulators are supported:

- AT&T 386 color console
- AT&T CTRM Terminal Emulator
- AT&T PC6300 with EMOTS Terminal Emulator
- Compucolor 2 and 8001
- Datamedia Color Scan 10
- Digital Equipment Corporation VT52 for the PC
- Hewlett Packard 2397 and 2397A
- Tektronix 4205

Other color terminals can be supported by creating the appropriate **terminfo**(4) entries.

Read/Write Permissions for Basic Networking Do Not Work

The read/write permissions for the Basic Networking Utilities do not work correctly. For a system to be able to read a directory, the target machine must also have granted the system write permissions (**/usr/lib/uucp/Permissions**). Suppose that your system has read permissions in root and write permissions in **/usr/tmp**. The following command fails:

uux "A!pr A!/etc/inittab > A!/usr/tmp/B.out"

However, if your system has read permissions in root and write permissions in **/usr/tmp** and **/etc**, the above command line succeeds.

Converting to getopts(1) by Hand

getoptcvt (see **getopts**(1)) adds about 30 lines of code to a shell script, so you may want to convert scripts by hand instead. Converting by hand probably will make the code cleaner and easier to understand. Also, you do not have to worry about parsing option-arguments that are also options.

Follow these guidelines to convert most scripts that currently use the **getopt**(1) command.

Step 1: Delete the old invocation line and the **if** statement that checks the exit code.

Step 2: Change the **for** loop to a **while** loop that invokes **getopt**(1).

Step 3: Change the patterns in the **case** statement from **-option** to single option letters.

Step 4: Delete the case for **--**.

Step 5: Add a case for **'?'**. This case may be used to print the usage message and to exit with a non-zero exit code. Note that the **?** is quoted because it is interpreted for filename expansion.

Step 6: Remove all **shift** commands within the **case** statement.

Step 7: Change **$ 2** to **$ OPTARG** for cases that require an option argument.

Step 8: Add the statement **shift `expr $OPTIND - 1`** after the **while** loop so the remaining arguments can be referenced as before. Here is an example of a script before and after conversion:

```
# before conversion
set -- `getopt abo: $*`
if [ $? != 0 ]
then
        echo $USAGE
        exit 2
fi
for i in $*
do
        case $i in
        -a | -b)FLAG=$i; shift;;
        -o)OARG=$2; shift 2;;
        --)shift; break;;
        esac
done
```

```
# after conversion
while getopts abo: i
do
            case $i in
            a | b)FLAG=$i;;
            o)OARG=$OPTARG;;
            ?)echo $USAGE
            exit 2;;
            esac
done
shift `expr $OPTIND - 1`
```

If you want your script to work on releases before ICON/UXV Release 4.00 (that is, using either **getopts** or **getopt**), convert it as the example below shows:

```
if [ "$OPTIND" = 1 ]
then
            while getopts abo: i
            do
            case $i in
            a | b)FLAG=$i;;
            o)OARG=$OPTARG;;
            ?)echo $USAGE
            exit 2;;
            esac
            done
            shift `expr $OPTIND - 1`
            echo $*
else
            set -- `getopt abo: $*`
            if [ $? != 0 ]
            then
            echo $USAGE
            exit 2
            fi
            for i in $*
            do
            case $i in
            -a | -b)FLAG=$i; shift;;
            -o)OARG=$2; shift 2;;
            --)shift; break;;
            esac
            done
            echo $*
fi
```

Toshiba MK156 Hard Drive

The Toshiba MK156 hard drive does not return the standard configuration information. The formatter, **dkfmt**, does not support this drive.

Basic Networking Utilities: Intelligent Modems

Features have been added to the **/usr/lib/uucp/Dialers** and **/usr/lib/uucp/Devices** files to prevent problems that occur when using System 75s, System 85s, Hayes-compatible modems, and other intelligent modems that do not keep Carrier Detect (CD) high all the time.

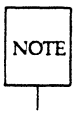
Devices Adding a **,M** to the second field of an entry in the **Devices** file will cause the O_NDELAY flag to be set when the device is opened. This prevents BNU software from blocking on the device while waiting for CD. The example below shows how to add the **,M** to a **Devices** file entry for a device connected to an automatic call unit for a Hayes modem.

```
ACU ttyll,M - 1200 hayes \T
```

Dialers Adding **\M** before the chat script in a **Dialers** file entry will set CLOCAL, preventing any change in the CD lead from resetting the state of the device. Once the conversation is established, **\m** will clear CLOCAL. This will allow BNU to again monitor changes in CD (for example, to notice if the line drops).

The example below shows how to add **\M** and **\m** to an entry for a Hayes modem in the **Dialers** file.

```
hayes  "=,-,"  ""  \M\dAT\r\c OK\r \EATDT\T\r\c CONNECT \m\c
```



For some devices, adding a **\p** after the **\M** may be necessary.

FACE/FMLI Notes

Conventions Used in this Section

The following conventions are used in these notes.

- Anything that can be displayed in a frame is referred to as an object (e.g. forms, menus, text). The message line, command line, banner line, and labeled function keys are also objects, but have no frames. When the ICON/UXV operating system, or a program running under ICON/UXV, takes over the entire screen, that application is an object.
- The names of all keys appear in capital letters: for example, LEFT-ARROW. The capitalized word refers to a single key on the keyboard. If a keyboard doesn't have that key, there is a control key sequence defined as an alternative.
- All FMLI descriptors are printed in italics. For example: *action*.
- All FACE or FMLI commands are printed in italics. For example: *refresh*

Hardware Notes

FACE is a rather large and complex FMLI application. When using FACE the minimum memory requirement is 2 Megabytes for single users and 4 Megabytes for multi-user mode.

The terminfo supplied with ICON/UXV Release 4.00 has been enhanced to support color. The new terminfo database contains entries to support the following color terminals and emulators:

- AT&T 6386 color console
- Softel PC52 Emulator
- Hewlett Packard® 2397A, and 2627A
- Tektronix® 4105, 4107, and 4200 series

You may also use one of the following monochrome terminals. Color commands will simply be ignored by these terminals.

- DEC VT100
- Hewlett Packard® 2621

Additional terminals will work with FMLI and FACE, provided a proper terminfo entry exists for that terminal. For information about writing or modifying terminfo entries, refer to chapter 10 of the *ICON/UXV Programmer's Guide*. (This guide can be ordered through Customer Support: 1-800-726-9687.

If your function keys, hard label keys, or alternate keystrokes do not work, check the terminfo entry for your terminal.

Hewlett Packard color terminals go to their hardware default settings when you enter the ICON/UXV system from an FMLI application. When you return to the application the colors are not reset to the colors defined in your application unless you execute the *reinit* command.

FACE Software Notes

Adding the System Administrator to FACE

Once FACE is installed, you will be able to add yourself as a new FACE user. Do not enter the FACE system to do this! Return to the System Administration Menu. Select the Package Management Menu, then the FACE Menu, and then usrmgmt. Respond to the prompt with your system login ID. Respond to the next question by typing "yes", which will cause the item System Administration to appear in your FACE Office Menu when you are in FACE. You will then be asked if you want to add another user. Answer "n", log out, and when you login the next time, you will be in the FACE Interface.

Whenever a user is added to FACE, if they already had a .profile, it will be saved as .profile.old.

Commands

The user has three print routines to choose from when they execute the *print* command. FACE has the **lp** command entered as the default for all three choices. If your system doesn't use **lp**, you will need to change these options to reflect the system default print device. If the system uses **lp** you should make sure that any necessary options are added to the default commands.

A maximum of five suspended applications are allowed through the user interface. If you have suspended applications, and then use the *frm-mgmt list* command, there is an inconsistency as to what appears in the frame. In some cases you will see the full pathname of the program suspended, and in others just the name of the suspended application.

If you execute the *security* command on a frame that is not a directory (such as a help frame), a blank Security Form appears.

The ordering of a file folder depends on the default settings of the *organize* command. If a menu is organized alphabetically by name, the *rename* command may change the order of the items in that file folder. If it does, there is a good chance that the cursor will no longer highlight the file that was just renamed.

If there is more than one frame on the screen, and you *move* one of them, and then *move* the Office Menu, there is a chance the Office Menu will lose a column of text.

If the user tries to *organize* a non-folder frame they will receive an error message stating `You do not have write permission to` *organize* this folder. The message is inappropriate, though it is correct in that you cannot
organize that frame.

If you *copy* a non-copyable file via the command line, giving all arguments, the name will appear in the destination folder, but the file isn't there. The item appears to be movable, copyable, and deletable, but only a name is moving around. Delete it to the WASTEBASKET, and let the system get rid of it for you.

If the user cancels a *goto* command that was executed from the command line, the screen labeled keys will display incorrectly until a *goto* command is completed.

General

The FACE user interface stores information about object-types in an ASCII file called .ott that is contained in each directory. This file describes all the objects in that directory. You should not edit the contents of .ott files. When you open a directory, if the .ott file doesn't exist FACE creates it. If the file exists, no check is made to verify its content. Thus, if it has a length of 0 or is corrupted, remove it.

It is possible that the MAIL icon will not get posted for as much as five minutes (e.g. if $MAILCHECK = 300) after the arrival of mail. It is also possible that the icon will not be cleared after mail is deleted until the same amount of time has passed.

Any command that sends messages to the terminal asynchronously will overwrite the FACE screen. If this occurs, the *refresh* command will correct the screen. For example, when the user selects one of the Mail Services, the ICON/UXV system command **mailx** is called, and takes over the entire screen. If an error is made by the user, it may not be discovered until after the user is returned to the FACE Interface. If this is the case, the **mailx** error message will overwrite the FACE screen.

If the user opens the folder of another user, and doesn't have write permission in that folder, an attempt to execute any command will cause the frame containing the folder to be updated.

You cannot open root (/) while in FACE.

Users should probably be reminded as often as possible that in order to complete a form, you press the SAVE function key. Pressing RETURN on the last field won't do the job.

Messages

Some messages on the MESSAGE LINE are by definition "shortterm", which means they disappear on the first keystroke after they are posted. This means that if a user presses the HELP key to get more information about a prompt on the MESSAGE LINE, when they are finished reading the help, the prompt message is gone. To see the prompt message again, it would be necessary to CANCEL the operation and then start it again. Examples are *move, copy,* and *reshape.*

Services

Using the Add Services feature under the Services Administration Menu may not work for every application. The Add Services Form does not allow arguments to be entered. Some applications may require arguments. Work-arounds include:

- Modifying the shell script that FACE generates
- Writing your own shell script and giving its name as the command.

See *Adding a Personal Service* in chapter 5, and *Changing a FACE Generated Shell Script* in chapter 6 of the *FACE User's Guide* for more information.

If anything except the menuname changes when a service is modified, a new <name>.ins script is written. When the new script is written, the old script is not removed. However, if menuname is the same, the old script will be overwritten.

Modify Services, under the Services Administration Menu, applies to personal services that you added from the Services Administration Menu. If you edited the shell script created by FACE when you added the service, you will need to edit it again if you have changed anything except the name the service has in the Services Menu, because FACE re-generates the shell script.

Avoid double quotes and blanks in a Service Menu Name.

When sending mail, the user is presented with a form where the login ID (mail name) of the recipient is to be typed. Unfortunately, this field will accept special characters (e.g. | ' " { } [ ] and escape sequences), some of which will cause **mailx** to loop forever. Control can be regained by pressing BREAK, DEL, or QUIT.

The Add and Modify Personal Services Form does no validation on terminal type.

WASTEBASKET

If the system prompts you that a file is scheduled for deletion, and you override the delete (keeping the file in the WASTEBASKET), from that point on, every time you login, you will be prompted that the file is scheduled for deletion.

There is at present no way to selectively allow the deletion of files scheduled to be deleted. The user must override the deletion and then open the WASTEBASKET and delete individual files and folders if they want selectivity.

If the user tries to delete an opened folder under WASTEBASKET, the message `Can't move an open object, close it first` appears. After the object is closed, trying to undelete it will generate a message telling you the object already exists in the WASTEBASKET, and you must rename this object to undelete it. FACE believes that this folder originally resided in the WASTEBASKET, and any attempt to undelete it will try to undelete it to the WASTEBASKET. If this situation occurs, use *move* or *copy* to get the file out of the WASTEBASKET.

FMLI Software Notes

Using FMLI

Once FMLI is installed, those who wish to use it should add the line

```
tput init
```

to their .profile to ensure that all of the necessary terminfo variables are set.

Commands

If an FMLI application initiates a call to a ICON/UXV system command (e.g., `action='unix_command'`), you will not be able to do other tasks until the command completes. If the ICON/UXV system command takes a considerable amount of time to execute, the application writer may want the command to execute in the background. Since FMLI does not recognize the shell background symbol "&", the *shell* built-in command must be used (e.g., `action='shell "unix_command &"'`). The application writer may also wish to explore the co-processing facility *coproc*(1F) which establishes a "pipe" between FMLI and another ICON/UXV process.

If the user cancels a *goto* command that was executed from the command line, the screen labeled keys will display incorrectly until a *goto* command is completed.

If there is more than one frame on the screen, and you *move* one of them, and then *move* the initial object, there is a chance the initial object will lose a column of text. To clean up the frame, execute the *refresh* command.

If you want to define PREV-FRM, NEXT-FRM, PREVPAGE, or NEXTPAGE on the second set of function keys, the labels must be spelled exactly as they are on the first set.

Forms

FMLI does not validate all of the fields in a form before saving it. FMLI validates the current field whenever its value changes. It also validates the current field when a *save* operation is performed. If necessary, one can validate all fields as part of the *done* descriptor, which is always evaluated when the SAVE key is pressed.

Choices for a form field can be specified using the *rmenu* descriptor. The value of *rmenu* is a list of items enclosed in brackets. There must be at least one white-space character that separates the brackets from the item list (e.g., `rmenu={ "item 1" "item 2" "item 3" }`).

The construct `rmenu={}` is illegal syntactically and will cause a core dump. The minimum requirement is `rmenu={ "" }`.

The *rmenu* descriptor works two ways. If there are three or fewer choices, a different choice will appear in the field each time the CHOICES key is pressed. If there are four or more choices, a menu will be presented and the item selected from it will be placed in the field. You can force a choices menu by the construct `rmenu=OPEN MENU menuname`.

There must be at least one active field in a form. If you open a form with only one field defined, and that field cannot be posted because it is inactive or *rows* or *columns* is negative or 0, **fmli** will core dump.

General

The FMLI Interpreter does not use EOF to exit a program. The command *exit* must be invoked. The assumption is that any application will be interactive, and at some point allow the user to select an item that evaluates to this command. Otherwise, the FMLI application will run forever. Specifically, this means you cannot direct the output of a file to **fmli**.

If a daemon process is started via a shell script that FMLI code invokes, the interpreter will wait for this process until the system clears up zombies. While waiting, the system appears to be locked.

In an FMLI definition a type cast may be used to avoid using the naming convention. The **fmli** invocation line requires the naming convention for all filenames, because no cast is recognized on the invocation line. For example, within FMLI you may define the *action* descriptor by

```
action=OPEN MENU mymenu
```

which will open the menu called mymenu. FMLI knows this is a menu because the cast MENU is used. On the invocation line this cast cannot be used, so the naming convention must be used instead.

```
fmli -i initfile -a aliasfile -c commandfile Menu.mymenu
```

Initialization

If you re-initialize the FMLI application with a new background color, text in the banner line will be shifted one character to the right. To avoid this problem, force the banner to be re-evaluated by including the banner descriptors in the new initialization file.

The *slk_bar* and *slk_text* descriptors do not work correctly. The background of the screen labeled keys will be the color defined for *highlight_bar_text* and the text on these keys will be the color defined for *highlight_bar*.

Keystrokes

The alternate keystroke sequence for HOME-DOWN (^fe) is supposed to position the selector on the bottom line of the page currently being displayed in the active frame. Instead, it will always return you to the bottom line of the first page.

The keystroke control-o is supposed to insert a line in a form with a scrollable field or an editable text object. It does, but the cursor stays on the line it was on. Thus, if you don't move the cursor down one line, you will be overwriting an existing line instead of entering text on the new line.

In a multi-page form, the TAB and BACKTAB keys work differently than DOWN-ARROW and UP-ARROW keys, as documented. Pressing the TAB key while at the bottom of a page, will cause the cursor to go to the top of the current page. Pressing BACKTAB at the top of a page, will cause the cursor to go to the bottom of the previous page.

Menu

In a menu, if the combined length of the name and description of an item is greater than 76 characters, the next item defined will not be posted.

Messages

Backquoted expressions that appear on a line by themselves are evaluated immediately when they are parsed. Thus, if a backquoted expression produces output to the message line, it will appear before the object being parsed is posted. This delay may or may not be significant and depends on the complexity of the object.

When the *checkworld* command is executed, the message line is cleared.

All messages on the MESSAGE LINE are by default "shortterm", which means they disappear on the first keystroke after they are posted. To force a message to be "permanent", use the *message* option "-p", which will display the message until another one is displayed. When that message clears, the "permanent" message will re-appear. To clear a "permanent" message, use the "-p" option and a null string.

Syntax

When creating a new form, menu, or text object, all quotes and backquotes must match. Quoting mismatches may cause the object to never appear, appear incorrectly, or in some circumstances cause **fmli** to core dump.

Text

The SCROLL-DOWN key will display the complete final page of a text object, even if much of it was already visible. The SCROLL-UP key will display the entire first page of a text object, even if most of it was already visible. The action of the SCROLL-DOWN key might be a surprise to the user if they are not also aware that the scroll down icon has disappeared, signaling that they are at the end of the text.

Chapter 6: Compatibility Notes

Compatibility Notes

Introduction

By providing ICON/UXV Release 4.00, Sanyo/Icon is continuing its commitment of moving forward by expanding the capabilities of ICON/UXV. During the development of the new ICON-specific features of Release 4.00, a greater level of attention was given to maintaining ICON/UXV compatibility with previous releases of ICON/UXV and with System V standards and practices than ever before. In addition, greater attention has been given to making Berkeley command extensions fully

awk(1)

These are the known incompatibilities between Release 3.30 **awk(oawk)** and Release 4.00 **awk.**

1. **awk** supports new keywords: `delete`, `do`, `func` (and `function`), **and** `return`. **awk** programs may now contain user-defined functions.
2. **awk** supports new built-in functions: `atan2`, `cos`, `sin`, `rand`, `srand`, `gsub`, `sub`, `match`, `close`, `system`.
3. **nawk** supports new pre-defined variables: `FNR`, `ARGC`, `ARGV`, `RSTART`, `RLENGTH`.
4. The `for (i in arrayname)` construct may produce array elements in an order different from before, depending on which version of **awk** you use.
5. The input field separator variable, `FS`, in **awk** specifies a regular expression, rather than just a single character. (The third argument to `split` also behaves this way.) Previously, **awk** (**oawk**) only used the first character in the string assigned to the input field separator if the string had more than one character.
6. Strings in **awk** may contain escape sequences like their C counterparts: `\b`, `\f`, `\n`, `\r`, `\t`, *\ddd*. Previously, the *\ddd* notation was not interpreted. Also, if a backslash was not part of an escape sequence, it was retained in the string. To be consistent with C, backslashes are dropped in **awk** if they are not part of escape sequences. For example, **oawk** prints **\c** for the command **print "\c"**, and **awk** prints **c**.
7. The precedence of operators has changed. **awk** firmly establishes their precedence, which matches C precedence, but which differs from their previous **oawk** precedence in some cases. Two examples of code that breaks are:

```
while ( n /= 10 > 1 ) ...
if (!"wk" ~ /bwk/) ...
```

The old version of **awk** (**oawk**) is provided to avoid compatibility problems.

cron(1M)

When **cron**(1M) completes a job, it sometimes sends a mail message to the user. This mail message had been sent by **root** but will now be sent with the ID of the user who started the cron job.

cron(1M)

The **/usr/spool/cron** directory that contains directories for **at**(1), **cron**(1M), and **crontab**(1) jobs will no longer be accessible to users. The directory mode will now be set to 700. If you want to read your own file, you must use **crontab -l**.

crypt(1)

The **crypt** command now takes options that begin with a dash (-), such as **-k**. These can no longer be given as keys. This change will only affect those who have the Security Administration Utilities package installed on their systems.

date(1)

The default timezone for **date** is now GMT.

Another change affecting **date** involves the **TZ** environment variable. Previously, if **TZ** was unset or null, the timezone used would be **EST** or **EDT**. Now the timezone used is **GMT**. For example,

```
TZ=' date
```

now prints

```
Thu May  1 21:29:08 GMT 1986
```

instead of

```
Thu May  1 17:29:08 EDT 1986.
```

date also recognizes the extended syntax in the **TZ** environment variable for alternate time zones. This change should not cause problems since the old syntax continues to be recognized. The changes involving the **TZ** environment variable occur because the **date** command uses the new **cftime**(3C) function to compute the date and time. The **cpio**(1), **ls**(1), **mount**(1M), **pr**(1), and **vi**(1) commands also use **cftime**, and their functionality in displaying times is affected by the **TZ** environment variable in the same manner.

date(1)

date(1) has been changed to take into account the new date for Daylight Savings Time.

ed(1)

The **ed** command now defaults to using the **/usr/tmp** directory to hold temporary files, instead of **/tmp**.

This change does not apply to the **vi**(1) or **ex**(1) programs.

You do not need to take any special action to use **ed** in single-user mode if the **/usr** file system is not mounted. When **ed** cannot put its temporary file in **/usr/tmp**, it tries **/tmp** instead.

/etc/shutdown

Previously, **/etc/shutdown** was mode 775. However, because only **root** can perform shutdowns, in this release of ICON/UXV the mode has been changed to 755.

ex(1), vi(1)

The structure of the **/usr/preserve** directory used by **vi** and **ex** has changed.

Instead of an editing session being saved as a file directly under the "**/usr/preserve**" directory, it is saved in a subdirectory with the name of the user whose session is saved. Only the same user can access the contents of the subdirectory.

In general you will not be able to recover a **vi** or **ex** session preserved before the upgrade to ICON/UXV Release 4.00. All sessions should have been recovered before upgrading.

ex(1), vi(1)

The **ex** and **vi** commands now exit with a return code equal to the number of errors encountered during the editing session. Before Release 4.00, no specified return code was used if errors were encountered.

Because all return codes must be between 0 and 255, if more than 255 errors are encountered the return code will not be accurate. If an integral multiple of 256 errors are found, then **ex** and **vi** exit with a zero return code.

ex(1), vi(1)

The **ex** and **vi** commands no longer set the eighth bit in the characters of the **%** expansion of the current filename.

When the percent sign (%) is used in a shell escape from **ex** or **vi** via the exclamation mark (!), the **%** is replaced with the name of the file being edited. Previously, each character in this replacement had the eighth bit set to 1 to quote it; now the eighth bit is left alone.

Generally, you can use older versions of the **ex** or **vi** commands on Release 4.00, but you cannot use the percent sign in a shell escape via the exclamation mark, even if the file being edited has no special characters in it.

file(1)

Additions to the **file** command's list of file types (found in the system file **/etc/magic**) make the command incompatible with an older **/etc/magic**.

ipcs(1)

The **ipcs**(1) command is distributed with the setgid mode bit turned on and the group of the file set to a privileged ID. This implementation allows a user to open and read privileged system files to obtain ipcs information. However, if a user specifies the **-C** or **-N** option, the user and group IDs of the process are reset to the user invoking the **ipcs**(1) command.

mail(1)

The following special characters are not valid in the mail forwarding line:

```
` ; & | ^ < > ( ) <CR>
```

If one of these special characters is encountered in the "forward to" line, the mail will be returned to the sender with the message

```
invalid address
```

mailx(1)

mailx changes the modification time of your mailbox (**/usr/mail/***login* so that the time stamp on the mailbox selects the last time new mail arrived, not the last time the mailbox was accessed and changed. In previous releases, the modification time was the last time the mailbox was accessed and changed.

mailx(1)

The **sendmail** variable that contains the delivery agent (by default, **rmail**) will no longer have the group ID set to mail. Instead, the program will have the group ID set to the effective group ID of the user who invokes **mailx**(1).

pr(1)

The **pr** command now correctly interprets the combined options **-m -k** as an error, where before one option would be ignored

Shell scripts that took advantage of the earlier fault in the **pr** command must be changed to use the correct option, invoking the **ipcs**(1) command.

sh(1)

A trailing colon in the shell PATH variable causes the current directory to be included in command searches. Previously, a trailing colon was ignored.

sh(1)

The **test** and **[...]** commands now use the effective user and group IDs instead of the real IDs (as previously), to determine permissible file access.

The only way to invoke the **test** command with different effective and real IDs is to invoke the command, or a shell script containing it, from a compiled program that has the set user (group) ID on execution permission. Otherwise, the effective IDs are the same as the real IDs, and this change will have no effect. If your program relies on the test operators to behave as they did previously, which is to have **test** use the real user and group IDs, you should change it to use the **setuid**(2) or **setgid**(2) system calls to set the effective ID to the real ID before invoking the **test** command or shell script.

sh(1)

The shell no longer treats the eighth bit in the characters of a command line argument specially; it also no longer strips the eighth bit from the characters of an error message.

If you have any program that sets the eighth bit of characters, it should be changed. You should use one of the standard shell quoting mechanisms, such as the backslash, instead of setting the eighth bit.

sh(1) 3.1

The shell **type** command displays backslashes before each character that was quoted initially.

This change is a result of the change described above, where the eighth bit is no longer used by the shell to quote characters.

sh(1)

The result of a parameter substitution in a command like

ls *"${a:=xyz abc} lmnop"*

is now correct.

In general, the parameter substitution

${ parameter:=word }

when used inside double quotes and when the *word* contains spaces, now works correctly. If you have programs that rely on the previous incorrect behavior, you should change them to reflect the correct behavior.

stty(1)

stty -tabs, which expands tabs to spaces, does not work with 8-bit characters. This is due to the fact that the terminal driver does not recognize 8-bit characters, so when it expands tabs to spaces, it generates an extra space for every 8-bit character preceding the tab character. For example: if **x** is an 8-bit character, the output of

echo 'xx\ta'

would be:

```
xx       a  (2 extra spaces)
```

instead of:

```
xx     a
```

For terminals that understand tabs properly, the following command should be executed:

stty tabs

who(1)

The **who -q** command now lists login names in space padded fields of equal size and no longer sorts entries.

If you have any programs that process the output of the **who -q** command, you should inspect them to see if they will still work with the new form of the output.

System Administrator Commands

dirname(1)

The **dirname** command now properly parses the names // and *//anything/* .

Previous to Release 4.00, the **dirname** command would return a dot (.) if given the argument // or *//anything/*. Now it correctly returns a slash, /.

You should change any shell scripts that rely on the previous incorrect behavior of the **dirname** command to reflect the correct operation.

/etc/cron

The owner of **/etc/cron** has been changed from **bin** to **root**, and the mode has been changed from 544 to 500.

init(1M)

The **init** command now reads an environment file when it first begins to run. It uses the contents of that file to retrieve default environment variable settings, and it passes these settings into the environment of all processes it spawns.

The **init** command now reads the file **/etc/TIMEZONE** to retrieve the **TZ** environment variable setting for the system it is running on. The retrieved **TZ** value, as well as a default **PATH** value, is then passed into the environment of every process spawned by **init**. Formerly, the only environment information passed to processes spawned by **init** was a value for **PATH**; all other environment variables were left uninitialized. Now, up to five environment variables can be specified in the **/etc/TIMEZONE** file. This provides the ability to specify system-wide default values for environment variables, including values that may be useful for international systems, such as default character sets.

login(1)

If you have installed the new security features on your system to discourage intruders, note that the encrypted password and password aging information formerly found in **/etc/passwd** has been moved to **/etc/shadow**. This file is readable only by its owner, and its owner is the same as the owner of **/etc/passwd** (**root**, by default). Users and administrators will still be able to change their passwords using the **passwd**(1) command. However, administrators must use the **passmgmt**(1M) command to change other fields in **/etc/passwd**.

Password and aging information is initially moved from **/etc/passwd** to **/etc/shadow** by running a new program, **pwconv**(1M). This program can only be executed by the superuser.

If you have an application or program that writes password and/or aging information into **/etc/passwd**, the program should be modified so that it uses **passmgmt**(1M) and **passwd**(1) to edit the password files. Until this modification can be made, the administrator with superuser privilege will have to run **pwconv**(1M) each time this program adds password information to **/etc/passwd**.

For more information about these changes, please see the "Security Notes" earlier in these *Release Notes*. Also see **passwd**(1)in the *User's Reference Manual, and* **passmgmt**(1M), **pwconv**(1M), and **pwunconv**(1M) in the *System Administrator's Reference Manual.*

lp(1)

You cannot change the status of printers or queues without the print service running (**/usr/lib/lpsched**).

The following commands will not work if the print service is not running:

enable disable /usr/lib/accept /usr/lib/reject /usr/lib/lpmove

The LP print service has been enhanced to allow all administrative commands to be run while the print service is running. This makes it easier to use the system and makes the system more responsive to changes and status requests. However, the enhancements give the responsibility for managing the system status to the background program, **/usr/lib/lpsched**, which must be running if printer or queue changes are to be made.

A related issue is that the **lpstat** command will not give a complete status report if the print service is not running.

To make changes (such as adding a printer), you no longer have to shut down the print service. Therefore you should leave the print service running. If it is currently not running, you can start it by entering the following command while logged in as **root** or **lp**:

```
$ /usr/lib/lpsched
```

lp(1)

Several **lp -o** options have been standardized in the Release 4.00 LP print service. A new, standard model has been introduced.

With the introduction of the new, standard model interface for printers, it is possible to provide uniform support for additional printing features, such as setting line and character pitches, page size, character sets, and optional banner page. These features will work on most printers that have been defined in the **terminfo** database. To allow custom interfaces to be upgraded fairly easily, and yet minimize the impact on existing software, most of these features are accessed through the **-o** *options* of the **lp** command.

This change will be a problem only if you have custom interface programs, and only if these programs make use of the following options or environment variables:

-o nobanner	(skip the banner page)
-o nofilebreak	(no inter-file page breaks)
-o cpi=	
-o lpi=	
-o length=	(set the page length)
-o width=	(set the page width)
-o stty=	(set special stty options)
$CHARSET	(identifies default character set)
$TERM	
$FILTER	

Also, if you have a model called "standard," it will be overwritten by the installed model.

If you have a custom interface that also uses these options or environment variables, see if the use is consistent with the standard use. If not, you should change the interface to use different option or variable names.

lp(1)

The **lpmove** command no longer changes the request ID.

In previous releases, when an lp request was moved to a new destination with the **/usr/lib/lpmove** command, the request ID was changed to reflect the new destination. For example, a request submitted for the printer *lp1* might get the request ID *lp1-43*. Then if the request was moved to the printer *lp2*, it would get the new ID of *lp2-43*. In this release this is no longer true: the request ID remains the same for the life of the request.

This change was made to minimize the impact on users; now users can use the original request ID to track the progress of their print request. You should check for shell scripts that depend on the old behavior of the **lpmove** command, and fix them to reflect the new behavior.

lp(1)

In this release, if a print request is submitted via RFS from a remote machine, the name of the user (as he or she is identified on the remote machine) is displayed on the banner page, preceded by the name of the remote machine, as in *machine!login*.

This change should affect only custom interface programs (models) that expect the user name argument to consist only of a single name, as it did in previous releases. Check your custom scripts, and change them to allow an optional machine name prefix to the name.

lp(1)

In this release the **lp** spooling directory structure has changed; many files have been moved and the formats have been changed.

The *System Administrator's Guide* (Chapter 7), has a section that lists the spooling directory structure. Printer and class configuration information, including the directory containing the interface scripts, have been moved to a subdirectory **admins/lp**. The model directory has not been changed, nor has the default destination file. The **logs** directory has been split into a **logs** directory at the top level and one at the **admins/lp** level. The content of the **logs** directory is completely different.

If you have scripts that depend on the structure of the spooling directory, it is likely they will have to be changed. Scripts that examine the logs will also have to be changed. The *System Administrator's Guide* details the layout of the request logs. Scripts that determine the names of the printers should look in:

/usr/spool/lp/admins/lp/printers

or, preferably, use the **lpstat -p** command.

NOTE

Note that a default crontab entry is supplied that will age the logs and clean out old logs. You may not need to update an old script if the new crontab entry is sufficient.

lp(1)

The printer interface scripts are invoked under the user ID and group ID of the user who submitted the request.

Previously, the interface scripts that control the printers were invoked under the **lp** user ID. This allowed these scripts to access LP files not generally accessible, but prevented the interface script from accessing protected user files, so that they could not be printed. Now the scripts are run under the user and group IDs of the user who submitted the request, so that the same set of files accessible to the user can be accessed (and printed) by the interface program.

A notable inclusion here is the **root** ID; if a request is submitted by the **root** user (or an alias such as **sysadm**), the interface program is run with **root** (superuser) permissions.

If you have custom interface scripts that depend on being run under the **lp** user ID, they will have to be changed. A typical problem would be a script that calls the **disable** command when a printer fault occurs; if the **disable** command is not being made available to regular users, this will fail. (Such scripts should be changed, anyway: they should not use the **disable** command, but should instead submit a fault message to the print service so it can alert the adminstrator.) The administrator who adds the scripts should have superuser permission.

PATH

The default PATH environment variable for root no longer searches the current directory.

prtvtoc(1M)

The output of the **prtvtoc** command has been enhanced.

Previously, **prtvtoc** showed the mount directory only if it was found in **/etc/fstab**. It now shows the current mount point for each mounted partition by looking in **/etc/mnttab** first. Also, **prtvtoc** now prints the number of sectors per cylinder and the end sector number of each partition.

uucp(1C)

System names must not contain unprintable characters or any of these special characters:

```
` ; & | ^ < > ( ) <CR> <TAB> <SPACE>
```

The **uuxqt** daemon will not perform remote execution requests for systems whose names contain any of these characters.

Programmer Commands, System Calls

acct(2)

The **acct** system call now sets **errno** to EACCESS when given an argument that is a directory instead of a file. (Previously, **acct** set **errno** to EISDIR in this situation.

brk(2)

If allocating memory causes a deadlock, the **brk** system call fails, setting **errno** to **EAGAIN**.

In previous releases it was possible for the ICON/UXV system to become deadlocked if free swap space or free main memory were not available. When a deadlock occurred, you could not "swap in" a runnable process because there was no room in main memory. It was also impossible to "swap out" a process to free the needed main memory because there was no room in the swap area. While rare, this situation occurred often enough to require adding checks in Release 2.1 to prevent it.

Several system calls now fail and set **errno** to indicate that a deadlock might have occurred. Figure 2-1 shows which system calls have been changed and what value is given to **errno** for each.

System Call	errno
brk	EAGAIN
exec	EAGAIN
fork	EAGAIN
plock	EAGAIN
shmat	ENOMEM
shmctl	ENOMEM

Figure 6-1: Deadlock Detecting System Calls

NOTE

Note that this new behavior should occur only when the system has nearly exhausted its memory capacity. If it occurs often, you should consider adding more main memory or increasing the size of the swap space.

exec(2)

If an **a.out** has been built and contains unresolved external references (for example, the **F_EXEC** flag is not set in the file header), **exec**(2) fails with **errno** set to **ENOEXEC**.

exec(2)

If allocating memory causes a deadlock, the **exec** system call can fail, setting **errno** to **EAGAIN**.

See discussion of deadlock detection in **brk**(2).

exec(2)

The **exec** system call no longer checks the **F_EXEC** bit in the **a.out** file header flags of a program before attempting to execute the program.

This change affects only those programs that were produced using the **-r** option in the **ld** program (also available through the **cc** program.) Such programs are often still executable, so the new behavior allows you to run them. Previously, the **exec** system call would fail with **errno** set to **ENOEXEC**.

fcntl(2)

The **flock** structure returned with the **F_GETLK** command of the **fcntl** system call has been changed. The **l_pid** element has been changed from type **int** to type **short**, and a new element, **l_sysid** of type **short**, has been added.

This change was made to accommodate cases where a file or record lock has been set by a process on a remote computer. The new structure now uniquely identifies a process.

Programs compiled under Release 3.3*x* that use record locking will have to be recompiled because **l_sysid, which occupies the space that used to be the high order 16 bits of the (old Release 3.34 32-bit) PID, can be non-zero for remote locks. This will cause problems if the application tries to kill that PID.**

fcntl(2)

The **fcntl** system call, on a file or record lock request, now sets **errno** to **ENOLCK** when the system runs out of lock resources, instead of setting it to **ENOSPC** or **EMFILE**, as before.

This change will affect only programs that currently check for **ENOSPC** or **EMFILE** to learn if the system has run out of lock resources.

Note that the effect will be seen only when you run the program and other programs have used up all the available locks.

fork(2)

The **fork** system call now has additional reasons for failing, but it still sets **errno** to **EAGAIN** in these cases.

This should not affect a program's attempt to catch cases where the **fork** system call fails because the same **errno** value is used. You should recognize, however, that the new paging system introduces new ways for **fork** to fail when system resources are running low.

mount(2)

The **mount** system call now correctly fails, with **errno** set to **ENOTDIR**, on an attempt to mount a special file on itself.

mount(2)

The **mount** system call now fails and sets **errno** to **EINVAL** if the file system's type is not recognized or if the **mflag** (previously **rwflag**) argument is not correct.

plock(2)

If locking a process results in a memory deadlock, the **plock** system call fails, setting **errno** to **EAGAIN**.

See the discussion of deadlock detection under the **brk**(2) section above.

shmctl(2)

If locking the shared memory region causes a deadlock, the **shmctl** system call now fails, setting **errno** to **ENOMEM**. For example, the following will fail if the attempt to lock the shared memory segment, identified by *shmid*, causes a deadlock:

shmctl(*shmid*, **SHM_LOCK**)

See the discussion of deadlock detection under the **brk**(2) section above.

shmctl(2)

The **shmctl** system call ignores an attempt to unlock a shared memory segment that is already unlocked, instead of failing, as before.

This change is not likely to cause a problem unless a program deliberately tries unlocking a shared memory segment without knowing if the segment is already locked in memory. If such a program looks for the **EINVAL** error return to indicate that the unlock attempt was not needed, it will no longer work as expected.

shmop(2)

If there is not enough memory to allocate page tables or if attaching the shared segment causes a deadlock, the **shmat** system call may fail, setting **errno** to **ENOMEM**.

The unavailability of additional memory for tables needed to manage the separate pages of the shared segment, or the possibility of a memory deadlock, may also cause the system call to fail. For a discussion of deadlock detection, see **brk**(2) above.

shmop(2)

The **shmat** system call now allows text as well as data segments to be shared.

Previously, a shared memory segment could only be attached to an address in the data segment of a process. With Release 4.00.0, a shared memory segment can be attached to any address in a process.

signal(2)

The **signal** system call now returns a pointer to a function of type **void** instead of a pointer to a function of type **int**, as before.

This change was made to bring ICON/UXV closer to conforming with the IEEE standard on the UNIX operating system. Since the function to which the return value of the **signal** system call points does not itself return a value, **void** is its correct type, not **int**.

signal(2)

The signal **SIGIOT** is being phased out to be replaced with the signal **SIGABRT**.

This change was made to bring ICON/UXV closer to conforming with the IEEE standard on the UNIX operating system.

Currently, both names are supported so source code is compatible. In the future the name **SIGIOT** will no longer be supported, so you should start changing your source code now. However, the value of **SIGIOT** and the value of **SIGABRT** are the same, which means that all compiled programs, including application packages you may have purchased, will continue to work, even in the future. For example, the **abort**(3C) library routine is now described as issuing the **SIGABRT** signal instead of the **SIGIOT** signal as before. You should therefore write new source code to expect the **SIGABRT** signal. However, since the values are the same, a program previously compiled to expect the **SIGIOT** signal from **abort** will continue to work when linked with the new **abort** routine.

signal(2)

The **signal** system call may fail, setting **errno** to **EINVAL**, if the *func* argument is invalid.

Previously the **signal** system call did not check its second argument, *func*, to ensure that it was one of **SIG_DFL**, **SIG_IGN**, or a valid function address.

umount(2)

On an attempt to unmount a special device whose major and minor numbers do not exist, the **umount** system call now sets **errno** to **EINVAL** instead of **ENXIO**.

There should not be many programs affected by this change, since special devices are usually mounted and unmounted using the **mount**(1M) and **umount**(1M) shell commands. You should see if any of your programs that use the **umount**(2) system call check **errno** for the value **ENXIO** when the system call fails. Any that check for **ENXIO** should be changed to check for **EINVAL**.

umount(2)

For **umount**, **EBUSY** is now returned if the device of the file system to be unmounted is the default pipe device.

The default pipe device is that section of hard disk used for unnamed pipes. If it is overridden and placed in a section of disk belonging to a file system that can be unmounted (for example, **/usr**), and you attempt to unmount that file system, the unmount will fail with the above error.

unlink

The **unlink** command fails without issuing an error message when you use it to unlink a busy text file.

ustat(2)

A new error return has been added. If the root inode of the mounted file system that you are doing the **ustat** on is NULL, **ENOENT** is set.

abort(3C)

The **abort** routine now issues the **SIGABRT** signal instead of the **SIGIOT** signal.

See **signal**(2) in the section on system calls.

abort(3C)

The **abort** routine no longer closes files when the **SIGABRT** (previously **SIGIOT**) signal is being caught or ignored. Previously, the **abort** routine would close all open files before issuing the **SIGIOT** signal that would normally cause the program to halt. If, however, the program had arranged to trap or ignore the **SIGIOT** signal, it would have to reopen the closed files before continuing.

With Release 4.00 the **abort** routine closes the files only if the program will halt on receiving the **SIGABRT** signal (which has the same value as the **SIGIOT** signal).

If you have a program that used the **abort** routine and trapped or ignored the **SIGIOT** signal, you should check to see if the new action by **abort** of keeping files open causes a problem.

ctime(3C)

The types of the argument **clock** in the **ctime**, **gmtime**, and **localtime** routines have been changed from "pointer to **long**" to "pointer to **time_t**." This is another change made to bring ICON/UXV closer to conforming with the IEEE standard on the UNIX operating system.

No source code changes are required for Release 4.00, but you should start changing source code now to ensure compatibility with future ICON/UXV releases. All previously compiled programs and application packages that use these routines will still work with this release.

puts(3S)

The **fputs** and **puts** routines now correctly return **EOF** if the attempt fails, instead of zero as before.

If you have a program that checks for a zero return from **puts** or **fputs** to indicate a write error, you should change it to check for **EOF**.

curses(3X)

To use the new **curses** features, use the Release 4.00 version of **curses**. You can link applications with object files based on Release 3.3*x* **curses/terminfo** with a Release 4.00 **libcurses.a** library; however, you cannot link applications with object files based on Release 4.00 **curses/terminfo** with the Release 3.3*x* **libcurses.a** library.

curses(3X)

cur_set no longer tries to simulate cursor modes that are undefined for the terminal being used.

curses(3X)

To maintain object mode compatibility with prior releases of **libcurses**, all calls to **newterm** and **initscr** must appear in the **.o** that was compiled in the earliest release.

curses(3X)

The terminal continues operating in 7-bit or 8-bit mode depending on the state of the environment when it is invoked. In Release 4.00.0 each window always began in 7-bit mode and was changed to 8-bit mode by calling **meta(win, TRUE)**.

curses(3X)

In Release 4.00 **pechochar( )** has two problems:

1. **pechochar( )** does not recognize a **wmove( )**
2. **pechochar( )** does not update the cursor position

You should use **waddch( )** followed by **prefresh( )** instead of **pechochar( )**.

regexp.h

regexp.h has been changed to allow for nested subexpressions.

If a regular expression has one or more occurrences of \(and no occurrences of \), the function *compile* will return with the statement **ERROR(42)** meaning \(, \) imbalance. Before, a check for matching parentheses was made only when a right parenthesis was encountered.

The change to allow nested subexpressions should not cause any problems. It is unlikely that someone would type in a regular expression with one or more instances of \(and no occurrences of \). Also, if one tried to reference an unbalanced subexpression with **\number**, **grep** and other commands would complain with a `digit out of range` message. If one tried to reference an unbalanced subexpression in a substitution replacement pattern in **ed** and **sed**, a null string would be substituted for the subexpression.

Miscellany

Sticky Bit

Added to this release is the new functionality of the sticky bit being settable by a user on a directory. If a directory is writable and the sticky bit is on, a user can remove a file in that directory only if one or more of the following is true:

1. the user owns the file

2. the user owns the directory
3. the file is writable by the user
4. the user is the superuser

In previous releases of ICON/UXV, the sticky bit had meaning only on files.

For information on how to set the sticky bit, see **chmod**(1) in the *User's Reference Manual.*

As part of this new feature, the public directories **/tmp** and **/usr/tmp** now have the sticky bit set. To remove a file or a directory in these publically readable and writable directories, the above rules apply. Formerly, anyone could remove any file or directory in **/tmp** and **/usr/tmp**.

/usr/lib/uucp/remote.unknown

The Basic Networking Utilities **remote.unknown** file has been changed from a shell script to a C Language executable program. When an unknown machine starts a conversation with the local machine, **remote.unknown** logs the conversation attempt. For more information on this file, see Procedure 9.3 in the *System Administrator's Guide.*

/usr/spool/uucp

Except for **root**, users will no longer be able to write in the **/usr/spool/uucp** directory or any directories under it. The directory mode will now be set to 755.

ICON/UXV 4.00 Release Notes
for M68k-based ICON Computers

Printed in USA

169-043-001 A