PI User's Manual

EDACS® Jessica
PBX Gateway

ERICSSON

NOTICE!

This manual covers Ericsson and General Electric products manufactured and sold by Ericsson Inc.

NOTICE!

Repairs to this equipment should be made only by an authorized service technician or facility designated by the supplier. Any repairs, alterations or substitution of recommended parts made by the user to this equipment not approved by the manufacturer could void the user's authority to operate the equipment in addition to the manufacturer's warranty.

NOTICE!

The software contained in this device is copyrighted by Ericsson Inc. Unpublished rights are reserved under the copyright laws of the United States.

This manual is published by **Ericsson Inc.,** without any warranty. Improvements and changes to this manual necessitated by typographical errors, inaccuracies of current information, or improvements to programs and/or equipment, may be made by **Ericsson Inc.,** at any time and without notice. Such changes will be incorporated into new editions of this manual. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of **Ericsson Inc.**

Copyright © June 1994, Ericsson Mobile Communications Inc.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

This page intentionally left blank.

1. INTRODUCTION

This manual is a guide describing how to configure and use the Jessica Private Branch Exchange (PBX) Gateway's PBX Interface (PI). The PI is a multiprocessor system consisting of a general purpose microcomputer board and multiple intelligent serial communications controllers. These boards communicate over an industry standard VMEbus backplane. The PI also includes hard and floppy disk storage units.

This manual presents information on PI hardware, initial setup details such as loading application software and configuration data, activity logging, and user commands. Appendix A presents the Telnet command and Appendix B discusses the File Transfer Protocol command.

Additional information for Jessica is available in the following publications:

- LBI 39000, EDACS Jessica PBX Gateway System Manual
- LBI-39001, EDACS Jessica PBX Gateway Operator's Manual
- LBI-39039, EDACS Jessica PBX Gateway MD110 Configuration Manual
- LBI-39080, EDACS Jessica PBX Gateway Operator's Manual (Quick Reference Guide)

This page intentionally left blank.

2. PI HARDWARE

This section provides a general description of the hardware of the PI. The figure below shows the architecture of the PI.

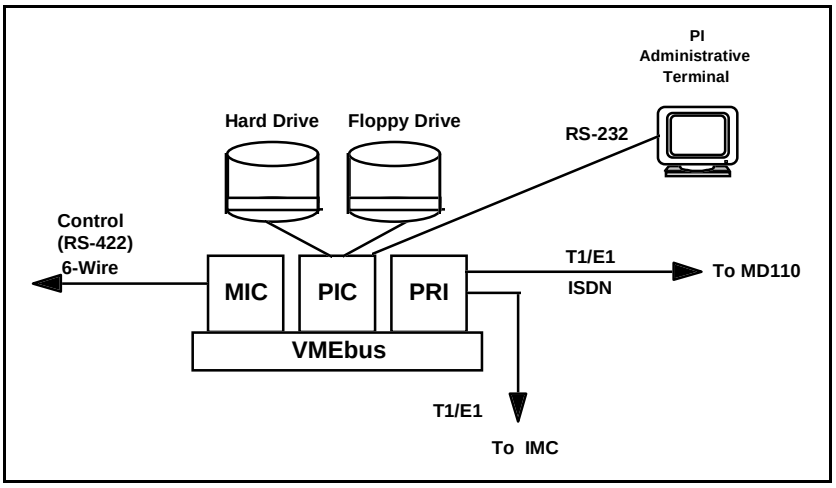

Figure 1 - PI Architecture

PI Component Description

The PI is a multiprocessor system consisting of a general purpose microcomputer board (PIC) in slot 1 and multiple microprocessor-based intelligent serial communications controllers which communicate over an industry standard VMEbus backplane. Slot 2 is left open, slot 3 is used for the Multisite Interface Controller (MIC), and slot 4 is used for a Primary Rate Interface (PRI) PRI-48 or a PRI-64. The PI also includes mass storage devices.

PI Controller

Using a 68030 microprocessor, the PI Controller (PIC) is a general purpose computing board that provides typical computer peripheral interfaces for the PI. These include disk facilities through a Small Computer Systems Interface (SCSI) bus, a Centronics parallel printer connection, IEEE 802.3 Ethernet connector, and four serial port interfaces. PI administration can occur over the IEEE 802.3 local area network interface or over serial port 1. Serial port 2 is used as a debug terminal and serial port 3 is unused. Serial port 4 is used for the System Manager.

In addition to servicing the PI peripherals, the PIC is the central point through which the PRI and MIC boards pass messages. Also, during the startup phase, the PIC reads the configuration files and loads application software and configuration parameters onto other processor boards in the system. Finally, the PIC processes commands from the PI terminal. This board must be in slot 1.

P2 Adapter Board

The P2 Adapter board is a small circuit board that routes the PIC I/O signals and grounds from its concentrated VMEbus backplane connector (P2) to the TVME-712M transition module. The board plugs directly onto the rear of the backplane and has two mass termination connectors. Two ribbon cables carry the I/O signals from these connectors to the transition module. Also, the P2 has sockets for SCSI terminating resistors if the SCSI interface of the TVME-147 is at the end of the SCSI bus.

TVME-712M I/O Transition Module

The TVME-712M is a separate circuit board which receives the PIC I/O lines from the P2 Adapter Assembly ribbon cables and routes them to the appropriate industry standard connector on its front panel. The I/O Transition Module has four DB-25 connectors for serial I/O, a 50-pin SCSI port connector, a DB-15 connector for Ethernet, and a Centronics compatible printer connector. Jumpers on the I/O Transition Module allow the serial ports to be configured as DTE or DCE. The I/O Transition Module has sockets for SCSI terminating resistors.

Multisite Interface Controller (MIC)

The MIC provides an RS-422 communication link between the PI and the PIM in the IMC using LAPB.

Primary Rate Interface (PRI)

The PRI-64 Integrated Services Data Network (ISDN) card is a VMEbus-compliant E1 interface card optimized for data formatting and transmission in digital switching applications. This card supports two E1 spans: one from the IMC carrying IMC audio channel information, the other from the MD110 ISDN interface. On-board dual port RAM is used to facilitate communication with the PI MIC card via the VMEbus.

The PI-MD110 E1 interface supports ISDN primary rate signaling (30B + D, i.e., 30 audio connections plus a control channel). The PRI card is supplied equipped with all necessary ISDN software to control call functions as defined in layers 1 through 3 of the ISDN specification, i.e., I.430, Q.921, and Q.931. The PRI card supplies layer 3A to provide a comprehensive communication mechanism with the PI application software.

For the North American, Japanese, and Korean markets, the PRI-48 ISDN card (T1) is used, with the corresponding decrease in B channels from 30 to 23.

PI Administrative Terminal

The administrative terminal allows the user to control system operation, view the system configuration, and view the activity logs stored on the fixed disk.

3. INITIAL SETUP

3.1. POWER CORD INSTALLATION

Information for power cord installation for the PI cabinet is presented below.

110 Volt AC

The North American, Japanese, and Korean T1 PI power supply feed cord is wired as follows for connection to an AC power outlet:

Black	Hot
White	Neutral
Green	Earth Ground

220 Volt AC

The European, Asian, and South American E1 PI power supply feed cord is wired as follows for connection to an AC power outlet:

Brown	Live/Hot
Blue	Neutral
Green/Yellow	Earth Ground

3.2. PI ADMINISTRATIVE TERMINAL

1. Connect the terminal to the female DB-25 connector labeled "PORT 1/CONSOLE" on the TVME-712M transition module located in the rear of the PI enclosure. Connect the other end of this cable to the host port or equivalent in the terminal. If a PC is used as the administrative terminal, then a DB-9 adapter may be needed. In either case, the serial cable for the terminal is wired straight-through.

2. Configure the terminal for the communications parameters below.

   - 9600 Baud
   - No Parity
   - 8 Data Bits
   - 1 Stop Bit

3. If desired, perform the steps below.

   - Enter the ***date*** command at the PI terminal.
   - Type ***timesav***.

4. When the bootup sequence is completed, the following will be the steady-state conditions:

   - The red LEDs on all PI boards are not illuminated.
   - The amber "STATUS" LED on the PIC board is lit dimly or is flashing.
   - The green "RUN" LED on the MIC board is illuminated.
   - The green "RUN" LED on the PRI board is illuminated.
   - The terminal displays the message "Login."

5. If any of these conditions are not met, then double-check the connections and try to restart the PI by switching off the power, waiting 15 seconds, and switching on the power again. If normal operation is not achieved, then reinstall the software.

3.3. PI SOFTWARE INSTALLATION

The PI Operating System is ROM resident. The PI application software is installed through a floppy disk interface.

3.4. CONFIGURATION FILE LOADING

This section describes the installation of the Jessica configuration from either a floppy disk or a hard drive.

The Jessica configuration file may specify:

- Site ID
- Hang Time (used for half-duplex interconnect calls)
- Conversation Time Limit (used for all interconnect calls)
- Trunk Type (T1 or E1)
- Mask of Available MUX Audio Channels

If a configuration parameter is not specified, its default value will be used during system operation. Refer to section 4 for default configuration parameters.

Table 1 - CONFIG.DAT File Loading

Type of Loading	Procedure	Expected Results
Jessica loading of configuration data from the floppy disk drive. (Files are edited on a separate host.)	1. Copy the CONFIG.E1 or CONFIG.T1 file on Jessica Disk 4 to CONFIG.DAT on the same disk. Choose the appropriate file based upon Jessica being E1 or T1. 2. Edit the CONFIG.DAT file on Jessica Disk 4 to match your application. This is the minimal set to place a call. • SITE_ID range 1-32. Must match PIM Controller site ID. Default is 16. • HANG_TIME range 0-255 seconds. Set to a value equal to the System Manager Interconnect Hang Time and the GETC Special Call Hang Time. • CONVERSATION_LIMIT range 0-255 minutes. Must be set to a value higher than the System Manager Message Conversation Limit and the GETC Message Trunked Timer. • MUX_CHANNELS_MASK range 0X0 to 0XFFFFFFFF hex, with the MSB corresponding to channel 32 and the LSB corresponding to channel 1. • TRUNK_TYPE must be set to T1 or E1. Features -- See section 4.1 herein for instructions on how to tailor CONFIG.DAT to incorporate the features available in Jessica.	1. Edit files to create the appropriate configuration.
Editing of configuration parameters from the PI terminal or remotely using Telnet.	1. Configuration parameters may be edited using the commands below. • ***config -s*** sets parameters temporarily. A reboot erases the settings. • ***savecfg*** writes all parameters out to CONFIG.DAT. • ***config -u*** reads in parameters from CONFIG.DAT.	

3.5. APPLICATION LOADING

This section describes the installation of the Jessica application from either a floppy disk or a hard drive.

Jessica is supplied with 5 diskettes. The table below discusses loading the first 3 diskettes. Disk 4 deals with the configuration files which may need to be edited off-line. Disk 5 contains voice notification files which are only applicable when the voice notification feature is licensed. Review Table 1 before initiating new software installation. Take care regarding Disk 4, which has CONFIG.DAT, ALLOW.DAT, DISALLOW.DAT, PRIORITY.DAT, IP.DAT, ROUTES.DAT, PBXFEAT.DAT, and EXPORTS.DAT. For application updates, some of these files may have been tailored for the customer site. You may wish to skip steps involving Disk 4. If Disk 4 is installed, there is an automatic backup procedure which copies the old configuration files to the backup directory. You may recover the customer site-specific information from this backup directory.

Table 2 - Application Loading

Type of Loading	Application	Expected Results
Application loading from floppy drive (Required to install new Jessica software)	1. Insert Jessica Disk 1 into the PI. 2. Apply power to the PI or execute ***reboot - h*** if logged into the PI. 3. The PI terminal will indicate the copy of the following files to the PI hard drive: LOADER.SX PRI.SX 4. At the prompt for the next disk, insert Jessica Disk 2 into the PI floppy drive. 5. The PI terminal will indicate the copy of the following file to the PI hard drive: JESS_A.SX 6. At the prompt for the next disk, remove Jessica Disk 2 and press "Enter". 7. At the prompt for the next disk, insert Jessica Disk 3 into the PI floppy drive.	1. The Jessica executable and configuration files are copied to the hard disk.

Table 2 - Application Loading (Cont.)

Type of Loading	Application	Expected Results
	8. The PI terminal will indicate the copy of the following file to the PI hard drive: JESS_B.SX 9. At the prompt for the next disk, remove Jessica Disk 3 and press "Enter". 10. Insert Jessica Disk 4 into the PI floppy drive. 11. The PI terminal will indicate the copy of the following files to the PI hard drive: CONFIG.DAT, ALLOW.DAT, DISALLOW.DAT, PRIORITY.DAT, IP.DAT, ROUTES.DAT, PBXFEAT.DAT, and EXPORTS.DAT. 12. At the prompt for the next disk, remove Jessica Disk 4 and press "Enter". 13. Insert Jessica Disk 5 into the PI floppy drive. 14. The PI terminal will indicate the copy of the following files to the PI hard drive: NOT_LOG.DAT, NUM_RES.DAT, FC_ACPT.DAT, FC_RJCT.DAT, CN_TONE, DSTRB_E.DAT, DSTRB_D.DAT, REDIAL.DAT, VN_FWD.DAT, and RINGING.DAT. 15. After the files are loaded, remove the disk, and press “Enter” to continue booting. 16. Log in to the system. 17. After the shell prompt is displayed, type ***config*** to display the configuration parameters. 18. Verify that the parameters specified on the floppy disk are displayed with the ***config-f*** command (displays full configuration).	2. After the files on disk 5 are loaded, the PRI-48/-64 ISDN board and the MIC are booted. 3. "Login" is displayed.
Application loading from hard drive (Requires Jessica software to have been previously installed)	1. Engage Jessica system power.	1. The PIC, PRI-48/-64 ISDN board, and the MIC are booted. 2. "Login" is displayed.

3.6. PASSWORD PROTECTION -- LOGIN

Jessica allows local logins from the PI terminal and remote logins using Telnet or FTP over the LAN connection. The terminal may be used to log in as shown below.

1. At the "Login" prompt, enter your authorized user name. The name should be one of those listed below.

Login Name	Password
root	root
ege	ege
jessica	jessica
user	user
guest	guest
field	field

These are default passwords which may be changed.

2. The PI requests your password. The correct password must be entered to use any of the PI utilities.

3. The terminal displays the following messages:

```
Copyright (c) Integrated Systems, Inc., 1992
Welcome to pSOSystem . . .

      pSH+>_
```

4. Some commands may only be executed from the root level. These commands are ***comm***, ***clone***, ***passwd***, ***purge***, and ***trap***.

4. CONFIGURATION FILES

Configuration parameters are read from disk files on reboot or upon execution of the ***config -u*** command. If a parameter is not specified in the appropriate disk file, a default is used. Some parameters are also receivable from the System Manager interface. Additionally, the ***config*** command may be used to display the value of the parameters in system RAM and to set some configuration parameters. CONFIG.DAT, ALLOW.DAT, DISALLOW.DAT, IP.DAT, EXPORTS.DAT, IN_ALW.DAT, IN_DIS.DAT, ROUTES.DAT, PBXFEAT.DAT, and PRIORITY.DAT parameters are presented in the subsections that follow. All variable names and enumerated values are case-sensitive. Comment lines must begin with "#".

Please refer to LBI-39171, "EDACS Network Management Installation and Technical Reference Manual," for instructions on setting up the PI for use with the EDACS Network Manager.

4.1. CONFIG.DAT PARAMETERS

Configuration variable names, values, and comments are contained in the data file named CONFIG.DAT.

The user sets the values in the tables below. Selecting E1 or T1 automatically sets the appropriate values for the entries in Table 16. The PI's assigned EDACS site ID (SITE_ID) can be updated and used real-time with the ***config -s*** command.

Table 3 - PI Configuration Parameters

Variable Name	Use	Allowed Values
SITE_ID	Site ID for Jessica.	1 - 31 default = 16
CONVERSATION_LIMIT	Conversation limit time in minutes.	0 - 255 default = 5 minutes
HANG_TIME[1]	Radio channel hang time in seconds.	0 - 255 default = 30 seconds
MUX_CHANNELS_MASK	Specifies multiplexer channels that may be used. Each bit in the hex mask represents a channel. LSB = channel 1, MSB = channel 32.	0X0 - 0XFFFFFFFF default = 0XFFFFFFFF (all valid)
TRUNK_TYPE	Specifies the type of trunk used.	T1, E1 default = T1
OOS_TIMEOUT	Specifies the time in seconds during which the IMC must confirm to the PI that a radio is logged in. (The PI determines that a radio is not logged in if the IMC does not send back a confirmation in OOS_TIMEOUT seconds.)	2-9 default = 9

[1] PI hang time and conversation limit may need to be set greater than those of the site so that the site GETC will generate alerting tones to the radio before dropping a call. This setup provides the radio user the options of pushing the PTT button to continue the call or simply allowing the call to be dropped due to expiration of the hang time or conversation limit.

Table 4 - Site-Based Call Routing Parameters

Variable Name	Use	Allowed Values
SITE_ROUTING_PREFIX	Specifies the first digit of the routing code used by the MD110 for site-based routing. Following this digit is the 2-digit EDACS site ID.	0-9 default = 0
SITE_ROUTING_ENABLE	Specifies if site-based routing is enabled.	TRUE and FALSE default = FALSE Site-based routing is disabled.
SBR_FOR_3_DIGITS	Specifies if site-based routing digits are prepended for 3-digit numbers such as 911.	TRUE and FALSE default = FALSE Site-based routing digits are not prepended for 3-digit numbers.

Table 5 - Call Forwarding Parameters

Variable Name	Use	Allowed Values
FEATURE_CODE_PREFIX	Single digit used in outbound calls indicating that a feature code follows.	0-9 default = 0
OUTBOUND_ALERTING_LIMIT	Specifies, in seconds, the maximum alerting time for outbound calls. When this time expires, the call is terminated by the PI.	1-255 default = 30
INBOUND_ALERTING_LIMIT	Specifies, in seconds, the maximum alerting time for inbound calls. When this time expires, the call is terminated by the PI.	1-255 default = 30
FORWARD_NO_ANSWER_TIME	Specifies, in seconds, the inbound no-answer time before the call is forwarded.	1-255 default = 30
PHONE_PHONE_CONV_LIMIT	Specifies, in minutes, the phone-to-phone conversation limit. When an inbound call is forwarded to a phone, the IMC is no longer involved to impose the conversation limit. This parameter imposes a conversation limit.	0-255 default = 5 Note: 0 will immediately tear down the call.

Table 6 - Activity Parameters

Variable Name	Use	Allowed Values
ALL_DEBUG_ENABLED	If this is TRUE, the PI behaves as if DISK_LOG_CALLS, PORT_LOG_CALLS, DISK_LOG_STATES, and PORT_LOG_STATES were all set equal to TRUE.	TRUE and FALSE default = FALSE Each type of call logging must be enabled individually.
DISK_LOG_CALLS	Specifies if call activity is to be recorded in the daily log file.	TRUE and FALSE default = TRUE Call activity is recorded in the daily log file.
PORT_LOG_CALLS	Specifies if call activity is to be displayed on the debug port.	TRUE and FALSE default = FALSE Call activity is not displayed on the debug port.
DISK_LOG_STATES	Specifies if detailed call states and events are to be recorded in the daily log file.	TRUE and FALSE default = FALSE Call states are not recorded in the daily log file.
PORT_LOG_STATES	Specifies if detailed call states and events are to be displayed on the debug port.	TRUE and FALSE default = FALSE Call states are not displayed on the debug port.
DISK_LOG_ISDN	Specifies if ISDN records are to be recorded in the ISDN log file. (Each record is individually enabled/disabled by the ISDN Activity Parameters below.)	TRUE and FALSE default = FALSE ISDN records are not recorded in the ISDN log file.
PORT_LOG_ISDN	Specifies if ISDN records are to be displayed on the debug port.	TRUE and FALSE default = FALSE ISDN records are not displayed on the debug port.
LOG_BASE_PATH	Specifies the directory where activity files will be placed.	LOG_BASE_PATH must be a complete path to a valid directory without a trailing “/”. default = 01.02/activity.
ACTIV_QUEUE_TIMEOUT	Specifies how long the activity task will wait at its queue for activity records. When it times out, a “No Activity Timeout” message will be written to the daily log file.	(0 - 4,294,967,295) 1/100 second units. default = 0 Wait indefinitely; do not generate “No Activity Timeout” messages.

The parameters in Table 7 simply select which of the ISDN records to record. The actual writing of all ISDN records is controlled by DISK_LOG_ISDN and PORT_LOG_ISDN.

Table 7 - ISDN Activity Parameters

Variable Name	Use	Allowed Values
ISDN_BOOT_ACTIVITY	Specifies if PRI board boot records are to be recorded.	TRUE and FALSE default = FALSE Boot records are not recorded.
ISDN_RAW_ACTIVITY	Specifies if PRI board raw activity records are to be recorded.	TRUE and FALSE default = FALSE Raw activity records are not recorded.
ISDN_AUX_ACTIVITY	Specifies if PRI board auxiliary activity records are to be recorded.	TRUE and FALSE default = FALSE Auxiliary activity records are not recorded.
ISDN_CALL_ACTIVITY	Specifies if ISDN call records are to be recorded.	TRUE and FALSE default = FALSE ISDN call records are not recorded.
ISDN_STATUS_ACTIVITY	Specifies if PRI board status records are to be recorded.	TRUE and FALSE default = FALSE PRI board status records are not recorded.

Table 8 - Enhanced Activity Report Parameters

Only authorized Ericsson Inc. personnel should modify the variables below. Incorrect use of these variables may damage your system.

Variable Name	Use	Allowed Values
PMI_ACTIVITY	Controls recording of PMI interface activity.	TRUE and FALSE default = FALSE Activity is disabled.
SYS_MGR_ACTIVITY	Controls recording of System Manager interface activity.	TRUE and FALSE default = FALSE Activity is disabled.
RECORD_SM_DATABASE	Controls recording of System Manager database (full and incremental) activity.	TRUE and FALSE default = FALSE Activity is disabled.

Table 9 - Channel Assignment Parameters

Variable Name	Use	Allowed Values
ROTATING_ASSIGNMENTS	Specifies whether PI-IMC channel assignments are rotating (balanced loaded) or first available.	TRUE and FALSE default = FALSE First available channel is assigned.
ASSIGNMENT_ORDER	Specifies whether PI-IMC channel assignments are in ascending or descending order.	ASCENDING and DESCENDING default = ASCENDING

Table 10 - Call Validation Parameters

Variable Name	Use	Allowed Values
CALL_NUM_RESTRICTIONS	Specifies if toll call restrictions are to be applied.	TRUE and FALSE default = FALSE restrictions are not applied.

Table 11 - Caller ID Parameters

Variable Name	Use	Allowed Values
CALLER_ID	Specifies if the radio LID is to be inserted as the caller ID for outbound calls.	TRUE and FALSE default = FALSE

Table 12 - Telephone Caller ID-to-Radio Parameters

Variable Name	Use	Allowed Values
INB_CALLER_ID	When enabled, shows the last four digits of incoming number on the radio display. (The IMC must also be enabled for inbound caller ID for the radio to see the digits.) When disabled, shows the IMC channel's LID on the radio display.	TRUE and FALSE default = FALSE
INB_CALLER_RESTRICT	If enabled, unpermitted inbound callers will be denied, while permitted callers will be allowed and INB_CALLER_ID configuration is checked as shown in the algorithm in LBI-39000.	TRUE and FALSE default = FALSE

Table 13 - System Manager Interface Parameters

Variable Name	Use	Allowed Values
SYS_MGR_UPDATES	Specifies if the LID/GID configuration parameters (not database) received from the System Manager are to be used.	TRUE and FALSE default = TRUE
SM_BAUD_RATE	System Manager baud rate.	9600 or 19200 default = 19200
SM_PASSWORD	System Manager password (must be in uppercase letters).	12-character string default = JESSICA

Table 14 - Network Manager Interface Parameters

Variable Name	Use	Allowed Values
NODE_ID	Specifies the node number of the IMC to which the Jessica is connected (used by the Network Manager).	0-64 default = 0

Table 15 - Group Call Parameters

Variable Name	Use	Allowed Values
RING_GROUP_CALLS	Specifies whether inbound ringing in group calls should be on (TRUE) or off (FALSE).	TRUE and FALSE default = TRUE

Selecting E1 or T1 in Table 3 automatically sets the appropriate values for the entries in Table 16 below.

Table 16 - PI Trunk Parameters

Variable Name	Use	Allowed Values
TRUNK_CLOCK_SOURCE	Clock source for trunk framing.	RECOVER_MD110_TRUNK RECOVER_MUX_TRUNK INTERNAL_CLK EXTERNAL_CLK default = RECOVER_MD110_TRUNK
MUX_TRUNK_FRAMING	Framing type selection for multiplexer trunk.	ESF ESF_ZBTSI SF SF_SLC96 CRC4 default = ESF
MUX_TRUNK_CODING	Coding type selection for multiplexer trunk.	B8ZS B7ZS AMI HDB3 default = B8ZS
MD110_TRUNK_FRAMING	Framing type selection for MD110 trunk.	ESF ESF_ZBTSI SF SF_SLC96 default = ESF
MD110_TRUNK_CODING	Coding type selection for MD110 trunk.	B8ZS B7ZS AMI HDB3 default = B8ZS
Q931_CONNECTION_TYPE	Defines the Q.931 layer of PABX Interface as seen by the MD110.	ATT4ESS_USER ATT5ESS_USER NTI_USER ATT4ESS_NETWORK ATT5ESS_NETWORK NTI_NETWORK CEPT default = ATT5ESS_USER

4.2. ALLOW.DAT (TOLL CALL RESTRICTION)

Outbound call restrictions are stored in two different files which are read upon startup of the PI: ALLOW.DAT and DISALLOW.DAT. ALLOW.DAT and DISALLOW.DAT may be viewed and updated with the ***callres*** command. Please refer to the ***callres*** command in section 6 for more details on file syntax.

The allow file contains a table of phone numbers, each of which is followed by a field of 15 (0-14) Yes/No entries that define whether the number is allowed to be called by the class.

Class 15 is permitted to call any number, so there is no Y/N listing for the class in the allow file.

Y Means the call is permitted for a particular class.
N Means the call is denied in this instance, but it may be allowed later in the table.

The permission table (once it has been read in from a file and stored in the PI) works as follows when a call is placed:

```
While NOT End Of Allow Table
{
    if (called phone number = = Allow Number in table)
        If there is a Y under the user's class in the table then the call is permitted.
                The disallow table must now be checked to see if the number is one of the disallow entries (e.g.,
                1-900 numbers).
        else
                go to the next entry in the list and continue to search for matching phone numbers.
    else
        go to the next entry in the table.
}
```

The entire table is searched until no more matching entries are found. If no match is found when the end of the table is reached, then the call is disallowed.

Table 17 - Sample Entries in the Allow Table

#Called	0	1	2	3	4	5	6		14
#Number								...	
0	Y	Y	Y	Y	Y	Y	Y	...	Y
6???	Y	Y	Y	Y	Y	Y	Y	...	Y
7???	Y	Y	Y	Y	Y	Y	Y	...	Y
9*	N	N	N	N	N	N	N	...	Y
911	Y	Y	Y	Y	Y	Y	Y	...	Y
97??????	Y	Y	Y	Y	Y	Y	Y	...	Y
9800???????	Y	Y	Y	Y	Y	Y	Y	...	Y
99?????????	N	N	N	N	N	N	Y	...	Y
9922????	Y	Y	Y	Y	Y	Y	Y	...	Y
9948????	Y	Y	Y	Y	Y	Y	Y	...	Y

As an example, a user in class 14 makes a “911” call. The search finds a match at entry “9*”. Since “911” matches “9*” for Class 14, the table search returns “allowed” and the call is allowed. Users in class 14 can make any calls starting with a “9”.

Suppose users in classes 0 through 13 place “911” calls. The search of the table matches at “9*”. Since users in these classes cannot make calls starting with a “9” and followed by any string of digits, the search of the table continues to “911”, which is the next entry. The number called (911) and the number in the table (911) match so the table search returns and the call is allowed.

A user in class 0 calls “8564”. This call is not allowed since there is no entry in the table. Only numbers which are explicitly defined in the table are allowed.

4.3. DISALLOW.DAT (TOLL CALL RESTRICTION)

Outbound call restrictions are stored in two different files which are read upon startup of the PI: ALLOW.DAT and DISALLOW.DAT. ALLOW.DAT and DISALLOW.DAT may be viewed and updated with the ***callres*** command. Please refer to the ***callres*** command in section 6 for more details on file syntax.

The table of disallow entries contains phone numbers which may be disallowed for users in classes 0 to 14 . Class 15 users are not affected by the disallow table. Called numbers that the allow table permits must be checked to determine whether they are denied by the disallow table.

The disallow file contains a table of phone numbers, each of which is followed by a field of 15 (0-14) Yes/No entries that define whether the number is permitted or denied for the class.

Y Means the call is permitted for a class.
N Means the call is denied.

The disallow table (once it has been read in from a file and stored in the PI) works as follows when a call is placed and is shown to be permitted by the allow table:

```
While NOT End Of Disallow Table
{
    if (called phone number = = Disallow Number in table)
        If there is an N in the table entry, the call is denied.
        else
            If there is a Y under the user's class in the table,
            continue to scan the table to make sure there is not another entry which would deny the call.
    else
        Go to the next entry in the table.
}
```

If the end of the disallow table is reached and no phone numbers in the disallow table match the called phone number, then the call is allowed.

If a phone number in the disallow table matches the called phone number and there is an "N" under the user's class, then the call is denied.

Table 18 - Sample Entries in the Disallow Table

#Called Number	0	1	2	3	4	5	6	...	14
9900???????	N	N	N	N	N	N	N	...	Y

In the allow table from the preceding section, users in classes 6 and 14 are allowed to make calls of the form 99?????????.

As an example of how the disallow table works, suppose a user in class 6 makes a call "99001234567". The allow table shows that the call is permitted because the last entry in the table is "99?????????". Then, the disallow table is checked and "99001234567" matches "9900???????". Since class 6 is set to "N" in the disallow table the call is denied.

If a user in class 14 calls the same number, he would pass through the permission table and proceed to the disallow table. His call would be allowed, since his class has a "Y" in the disallow table.

4.4. IN_ALW.DAT

Inbound caller identification shows the last 4 digits of the calling party's number on the radio display. Its implementation is similar to toll call restriction. Inbound caller ID is set up to check the phone number against allow and disallow tables (IN_ALW.DAT and IN_DIS.DAT) to determine whether the number is permitted to be displayed or the user is allowed to make the call. The number must be explicitly allowed in the IN_ALW.DAT file, and not be denied in the IN_DIS.DAT file. Unlike the outbound ALLOW.DAT and DISALLOW.DAT tables, the inbound tables do not include 16 classes since there will only be one common "class" of telephone users.

NOTE

This feature requires specially modified radios. Contact your Ericsson Sales Representative for more information.

4.5. IN_DIS.DAT

Inbound caller identification shows the last 4 digits of the calling party's number on the radio display. Its implementation is similar to toll call restriction. Inbound caller ID is set up to check the phone number against allow and disallow tables (IN_ALW.DAT and IN_DIS.DAT) to determine whether the number is permitted to be displayed or the user is allowed to make the call. The number must be explicitly allowed in the IN_ALW.DAT file, and not be denied in the IN_DIS.DAT file. Unlike the outbound ALLOW.DAT and DISALLOW.DAT tables, the inbound tables do not include 16 classes since there will only be one common "class" of telephone users.

NOTE

This feature requires specially modified radios. Contact your Ericsson Sales Representative for more information.

4.6. IP.DAT PARAMETERS (REMOTE ACCESS SETUP)

The Ethernet parameters for each PI are stored in the disk file IP.DAT as shown below. A sample IP.DAT file is shown in Figure 2 below. In the figure, note that the first character (#) designates a comment line. The comment line indicator must be removed from the last three parameters shown in the figure and the system must be rebooted for the changes to take effect.

Table 19 - IP.DAT File Parameters

Parameter	Use	Allowed Values
IP_ADDRESS	IP address of the PI.	Valid IP address in Internet dotted-decimal notation.
HOST_NAME	Host name of the PI.	1-32 contiguous characters.
SUBNET_MASK	IP subnet mask for the PI. Specified as eight hexadecimal digits, in upper case.	0x00000000-0xFFFFFFFF

```
#---------------------------------------------------------------------------
# File: IP.DAT
# Description:
#   Template for the Jessica IP.DAT file. This file configures the
# unit's IP address, subnet mask, and host name string. You should
# modify these values to suit your addressing scheme.
#
# This file is processed during the application loading (boot) phase only.
# Modifications made after the unit is up and running will take affect
# during the next system boot.
#
# NOTES:
#   (1) All of the entries in this file are commented out!
#       i.e., Preceded by a # character.
#
#
#   $Id: IP.DAT,v 1.1 1994/08/10 15:04:27 jharker Exp $
#   $Log: IP.DAT,v $
# Revision 1.1  1994/08/10  15:04:27  jharker
# Initial revision
#
#
#---------------------------------------------------------------------------

#---------------------------------------------------------------------------
#---------------------------------------------------------------------------

#IP_ADDRESS  147.XXY.AA.DDD
#HOST_NAME   jess01
#SUBNET_MASK FFFFF000
```

Figure 2 - Sample IP.DAT File

4.7. EXPORTS.DAT PARAMETERS (NFS ACCESS SETUP)

The EXPORTS.DAT file applies to the NFS Server software feature. It defines which network clients are privileged to mount the Jessica system disk. If EXPORTS.DAT is missing, anyone may NFS mount the Jessica crate. If EXPORTS.DAT is present, only the hosts listed in EXPORT.DAT may mount the crate. Example exports provided on the template disk are commented out (i.e., preceded with a "#" character), and thus have no effect when the file is processed.

A log file (1.2/log/nfs.log) is generated on each system boot. The log summarizes processing of this file, indicating what has been exported to which user, and any errors encountered in processing the EXPORTS.DAT file.

Export entries are specified in the following form:

directory client_ip

Table 20 - EXPORTS.DAT File Parameters

Parameter	Meaning
directory	Directory structure(s) to be exported. Must be a fully specified, valid directory on the Jessica system disk (volume 1.2). Note that directories are case-sensitive. Specifying the Jessica root directory (1.2/) indicates that the entire volume may be mounted by the host specified by ***client_ip***.
client_ip	Internet address of the NFS client permitted to mount ***directory***, specified in dotted-decimal notation. If no address is specified, it indicates that **any** client may mount the directory.

Examples of valid export entries in the EXPORTS.DAT file are shown below.

```
#########################################################################
# In the following example, hosts "fallwell" and "hagee" may NFS mount
# the entire Jessica system disk (volume 1.2).
# Host "robertson" may only mount the Jessica activity directory.
# Any host may mount the log directory.
#########################################################################

1.2/            147.XXX.YY.ZZZ      # host name - fallwell
1.2/            147.XXX.YY.ZZY      # host name - hagee
1.2/ activity   147.XXX.YY.ZZX      # host name - robertson
1.2/log                             # any client may mount
```

4.8. ROUTES.DAT PARAMETERS (NFS ACCESS SETUP)

The ROUTES.DAT configuration file specifies network routing paths for the NFS feature to be established at system boot. Typically, this file will be required only if Jessica is connected to a wide area network, where routers exist between the unit and other customer host facilities. The template file contains a few example routes, which are commented out (i.e., preceded with a "#" character). If no network routing is required, this file may be omitted, or left unaltered. Table 21 lists the parameters specified in ROUTES.DAT.

A log file (1.2/log/routes.log) is generated on each system boot. This file contains a summary of successful routing additions, as well as any errors detected during processing of the ROUTES.DAT file.

Table 21 - ROUTES.DAT File Parameters

Parameter	Meaning
type	Keyword for the type of route being added, either ***host*** or ***network***.
destination_ip	IP address of the destination host or network, in Internet dotted-decimal notation.
gateway	Keyword indicating that the next field is that of the gateway node.
gateway_ip	IP address of the gateway to be used to communicate with the host address specified by destination_ip.

Parameter specification is not case-sensitive, and is parsed (i.e., net is equivalent to network). White space and/or tabs may separate the parameters. Trailing comments (#) are allowed.

Network routes should be entered in a logical order. That is, if there are multiple gateways between the Jessica and a destination, the most direct route(s) should be specified first.

Network routes may be manually added and deleted using the ***route*** command discussed in section 6. If you are unsure of proper routing, use the ***route*** command to experimentally determine the proper, or most efficient, parameters, and then add these to the ROUTES.DAT file.

Proper routing is intimately related to the IP address and subnet mask specified in the IP.DAT configuration file. Keep these parameters in mind when adjusting ROUTES.DAT contents.

The following are examples of routing entries in the file ROUTES.DAT. In this example, packets destined for host 147.XXX.A.B will be routed to gateway 147.XXX.CD.AAA for forwarding. All packets destined for hosts on network 147.BBB.O.O will be routed to gateway 147.CCC.CD.B for forwarding. Similarly, destinations on network 147.ABC.D.E will be routed through 147.DDD.CD.E. Network routes are specified in the following form:

```
type      destination_ip  gateway    gateway_ip
host      147.XXX.A.B gateway 147.XXX.CD.AAA    # Example of a host route
network 147.BBB.O.O gateway 147.CCC.CD.B      # Example of a network route
net       147.ABC.D.E GATE    147.DDD.CD.E      # Example of parsing
```

4.9. PBXFEAT.DAT (COMMON CALL FORWARDING)

Call forwarding configuration variable names, values, and comments are contained in the data file named PBXFEAT.DAT. These variables are presented in the table below.

Table 22 - PBXFEAT.DAT Parameters for Call Forwarding

Variable Name	Use	Allowed Values
COMMON_FORWARD_1	Common forwarding point one for call forwarding.	Valid phone digits string.
COMMON_FORWARD_2	Common forwarding point two for call forwarding.	Valid phone digits string.
COMMON_FORWARD_3	Common forwarding point three for call forwarding.	Valid phone digits string.

4.10. PRIORITY.DAT (PRIORITY SERVICE CHANNELS)

Priority Service Channels are created and stored in the PRIORITY.DAT file. Create PRIORITY.DAT on the Jessica hard disk as indicated in the table below. Level 4 is intentionally left out in this example so that the system will default to 0 channels at level 4.

Table 23 - PRIORITY.DAT Example

#Priority	No. Channels
PRIORITY_7	1
PRIORITY_6	1
PRIORITY_5	0
PRIORITY_3	0
PRIORITY_2	3
PRIORITY_1	1
PRIORITY_0	0

This page intentionally left blank.

5. ACTIVITY LOGGING

Activity logging is useful for billing purposes or tracking the general activity of radios. To view activity files, use the ***head*** and ***tail*** commands described in section 6.

This section presents information on activity logging, non-verbose and verbose messages, call states, call events, and disconnect reason codes.

Activity in the PI is recorded based on function as shown in the logs listed below.

Table 24 - Activity Logs

"today's date".log.	Call activity, database uploads, System Manager connections, and database activity, if enabled, are in the log.
isdn.log.	Only ISDN-related entries are in the log.
pmistats.log	Only PMI-related information exists here.
sysmgr.log	Contains System Manager link entries.
wan.log	Only WAN-related entries are in the log.

The PI stops recording data when a threshold of 1 Megabyte of disk space left is reached. Recording begins again when at least 1 Megabyte is available. A message is entered in the error log and printed to the terminal when this event occurs.

5.1. NON-VERBOSE AND VERBOSE MESSAGES

Non-verbose and verbose messages are set through the CONFIG.DAT file. To view the type of message set up, type ***config -f*** to show the full configuration. If DISK_LOG_STATES is set to false, then non-verbose messages are written to the daily log file. If DISK_LOG_STATES is set to true, then verbose messages are written to the log file.

PORT_LOG_CALLS and PORT_LOG_STATES send the call information to port 2 as well and do not affect port 1 operation. Port 2 has no flow control, so overflow information will not be maintained. If PORT_LOG_STATES is set to false, non-verbose messages will be sent to port 2.

The following list contains fields that appear in non-verbose and verbose messages and explanations of what the fields represent. The fields in items 17 through 20 appear only in forwarded calls. Note that in forwarding chains intermediate steps are not logged. The fields presented in items 21 through 25 appear only in verbose messages. Examples of the field placement in activity records follow the fields.

1. CALL:[00000]
   Call number assigned internally by the PI (unique for each call). The field resets to 0 each time that the PI is rebooted.

2. OUTBOUND or INBOUND
   Type of call. An outbound call is a radio-originated call to a phone. An inbound call is a phone-originated call to a radio.

3. DIGITAL or CLEAR
   Type of voice. Digital refers to voice transmission in a digital format. Clear refers to voice transmission in an analog format.

4. LID [00000] or GID [0000]
   EDACS radio identification. LID is an individual radio or console ID (LIDs are 5-digit numbers). GID is a radio group ID (GIDs are 4-digit numbers).

5. [NORMAL_DISCONNECT]:PBX, [FORCED_DISCONNECT]:IMC, [NORMAL_DISCONNECT]:FOR, and [FORCED_DISCONNECT]:PI
   Reason that the call terminated. [NORMAL_DISCONNECT] and [FORCED_DISCONNECT] are types of disconnect reason codes. These codes are discussed in section 5.4. PBX and IMC designate whether the phone user or radio user initiated the disconnect, respectively. FOR designates that the phone the call is forwarded to initiated the disconnect. PI designates that the PI initiated the disconnect, e.g., timeout or user denied.

6. IMC[00]
   Channel between the PI and the IMC.

7. PBX[00]
   Channel between the PI and the MD110.

8. PHONE[000000000000000000]
   Phone number dialed by the radio. The width of this field will vary. The field will contain no leading zeros.

9. SITES: O[00] D[00]
   O is the originating site (where the call was placed from) and D is the disconnect site. If the call was originated at or disconnected by the PI or PBX, the Jessica site ID is used; all other codes are EDACS site IDs.

10. PRI[G/R]
    G is the granted priority level and R is the requested priority level.

11. TOLL[00]
    Toll call class as specified by the System Manager (range is 0 to 15).

12. DUR: 000.00
    Duration (second. hundredths of second) of the call.

13. DATE: 00/00/00
    Date (day/month/year) on which the call was made.

14. ST: 00:00:00
    Time (hour: minute: second) that the call was originally placed.

15. CON: 00:00:00
    Time (hour: minute: second) that the call setup (radio to phone or phone to radio) was completed.

16. DIS: 00:00:00
    Time (hour: minute: second) that the call was terminated.

Forwarded calls contain an additional line of information with the parameters shown below.

17. PHONE FORWARD, COMMON FORWARD, or RADIO FORWARD
Designates the type of forwarding.

18. PBX2[00]
Channel between the PI and MD110 used for the forwarded call.

19. PHONE2[00]
Phone number that was forwarded to. This field is present only on phone forwarding.

20. DIGITAL or CLEAR and LID [00000] or GID [0000]
These fields are defined. Please see the appropriate non-verbose and verbose message field descriptions above. This field is present only on radio forwarding.

21. Time
Measured in quarter-seconds.

22. Event
Activity that is occurring.

23. From
Source of the event. Includes the same parameters used in _DISCONNECT (refer to No. 5 of this list).

24. State
State before event occurred.

25. Events
The maximum number of recorded events is 20. Any excess events are discarded.

Examples of non-verbose and verbose messages are presented below. Explanations are offset with {}.

EXAMPLE 1

{**Non-Verbose Messages** -- shows normal call activity; each call consists of three to four lines -- four lines are used when forwarding}

```
CALL:[00003] OUTBOUND DIGITAL LID[06533] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[01] PHONE[2001]   SITES: O[01] D[01] PRI[0/0] TOLL[00]
DUR: 011.50 DATE: 08/20/93 ST: 09:21:37 CON: 09:21:43 DIS: 09:21:49

CALL:[00004] OUTBOUND DIGITAL LID[06533] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[01] PHONE[2001]   SITES: O[01] D[01] PRI[0/0] TOLL[00]
DUR: 022.20 DATE: 08/20/93 ST: 09:21:56 CON: 09:22:01 DIS: 09:22:18

CALL:[00005] OUTBOUND DIGITAL LID[06533] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[01] PHONE[2001]   SITES: O[01] D[01] PRI[0/0] TOLL[00]
DUR: 042.70 DATE: 08/20/93 ST: 09:22:30 CON: 09:22:35 DIS: 09:23:13

CALL:[00006] OUTBOUND CLEAR   LID[06533] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[2001]   SITES: O[01] D[00] PRI[0/0] TOLL[00]
DUR: 014.69 DATE: 08/20/93 ST: 09:23:22 CON: 09:23:26 DIS: 09:23:37
```

```
CALL:[00007] OUTBOUND CLEAR   LID[06533] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[2002]   SITES: O[01] D[00] PRI[0/0] TOLL[00]
DUR: 019.47 DATE: 08/20/93 ST: 09:23:46 CON: 09:23:50 DIS: 09:24:06

CALL:[00008]  INBOUND CLEAR   LID[06183] [FORCED_DISCONNECT]:IMC
IMC[01] PBX[04] PHONE[]   SITES: O[00] D[01] PRI[0/0] TOLL[00]
DUR: 019.21 DATE: 08/20/93 ST: 09:27:08 CON: 00:00:00 DIS: 09:27:27

CALL:[00009]  INBOUND CLEAR   LID[06183] [FORCED_DISCONNECT]:IMC
IMC[01] PBX[01] PHONE[]   SITES: O[00] D[01] PRI[0/0] TOLL[00]
DUR: 019.02 DATE: 08/20/93 ST: 09:29:13 CON: 00:00:00 DIS: 09:29:32

CALL:[00008]  INBOUND CLEAR   LID[06044] [NORMAL_DISCONNECT]:PBX
IMC[00] PBX[03] PHONE[] SITES: O[17] D[17]  PRI[2/2] TOLL[07]
PHONE FORWARD   PBX2[01] PHONE2[2002]
DUR: 010.54 DATE: 11/08/94 ST: 11:28:45 CON: 11:28:54 DIS: 11:28:56

CALL:[00015]  INBOUND CLEAR   LID[06044] [NORMAL_DISCONNECT]:PBX
IMC[00] PBX[02] PHONE[] SITES: O[17] D[17]  PRI[0/2] TOLL[07]
COMMON FORWARD  PBX2[00] PHONE2[2100]
DUR: 018.11 DATE: 11/08/94 ST: 11:30:42 CON: 00:00:00 DIS: 11:31:00

CALL:[00018]  INBOUND CLEAR   LID[06044] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[03] PHONE[] SITES: O[17] D[17]  PRI[2/2] TOLL[07]
RADIO FORWARD   CLEAR   LID[06367]
DUR: 029.26 DATE: 11/08/94 ST: 11:32:47 CON: 00:00:00 DIS: 11:33:16
```

EXAMPLE 2

In the verbose messages below, the information following the parameters already defined for non-verbose and verbose messages presents internal events for the PI's active call state machine.

{Verbose Messages -- additional information which can be logged for each call; this is enabled through the configuration of Jessica}

```
################################################################################

                        CALL STATES ACTIVITY LOGGING ENABLED

################################################################################

********************************************************************************
                          Outbound Call/Telephone Terminated
********************************************************************************

CALL:[00084] OUTBOUND CLEAR   LID[02405] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[92372004]   SITES: O[01]  D[00] PRI[2/6] TOLL[10]
DUR: 027.72 DATE: 08/06/93 ST: 07:54:00 CON: 07:54:03 DIS: 07:54:28
Time->[223196]   Event->[CONSTRUCTION  ] From->[IMC] State->[OUTBOUND_ACTIVE  ]
Time->[223196]   Event->[CHANNEL_REQ   ] From->[IMC] State->[OUTBOUND_ACTIVE  ]
Time->[223201]   Event->[OUTBOUND_SETUP] From->[IMC] State->[AWAITING_SETUP   ]
Time->[223203]   Event->[ALERTING      ] From->[PBX] State->[SETUP_IN_PROGRESS]
Time->[223210]   Event->[CONNECT       ] From->[PBX] State->[CALL_ALERTING    ]
Time->[223210]   Event->[CONNECT_ACK   ] From->[IMC] State->[CALL_CONNECTING  ]
Time->[223311]   Event->[DISCONNECT    ] From->[PBX] State->[CALL_CONNECTED   ]
Time->[223311]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[12] Events


********************************************************************************
                          Outbound Call/Radio Terminated
********************************************************************************
CALL:[00111] OUTBOUND CLEAR   LID[06078] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[00] PHONE[2004]   SITES: O[01]  D[01] PRI[1/5] TOLL[03]
DUR: 006.20 DATE: 08/06/93 ST: 08:17:04 CON: 00:00:00 DIS: 08:17:10
Time->[228968]   Event->[CONSTRUCTION  ] From->[IMC] State->[OUTBOUND_ACTIVE  ]
Time->[228968]   Event->[CHANNEL_REQ   ] From->[IMC] State->[OUTBOUND_ACTIVE  ]
Time->[228972]   Event->[OUTBOUND_SETUP] From->[IMC] State->[AWAITING_SETUP   ]
Time->[228974]   Event->[ALERTING      ] From->[PBX] State->[SETUP_IN_PROGRESS]
Time->[228994]   Event->[DISCONNECT    ] From->[IMC] State->[CALL_ALERTING    ]
Time->[228994]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[228994]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[228994]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[228995]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[228995]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[10] Events
```

```
*******************************************************************************
                    Individual Inbound Call/Telephone Terminated
*******************************************************************************

CALL:[00134]  INBOUND CLEAR   LID[03945] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[]   SITES: O[00]  D[00] PRI[0/1] TOLL[15]
DUR: 002.13 DATE: 08/06/93 ST: 09:15:14 CON: 00:00:00 DIS: 09:15:16
Time->[243521]   Event->[CONSTRUCTION  ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[243521]   Event->[REF_REQUEST   ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[243521]   Event->[INBOUND_SETUP ] From->[PBX] State->[AWAITING_SETUP   ]
Time->[243521]   Event->[PROGRESSING   ] From->[IMC] State->[SETUP_IN_PROGRESS]
Time->[243522]   Event->[ALERTING      ] From->[IMC] State->[CALL_PROGRESSING ]
Time->[243530]   Event->[DISCONNECT    ] From->[PBX] State->[CALL_ALERTING    ]
Time->[243530]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[11] Events


*******************************************************************************
                      Individual Inbound Call/Radio Terminated
*******************************************************************************

CALL:[00113]  INBOUND CLEAR   LID[02405] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[04] PHONE[]   SITES: O[00]  D[01] PRI[6/6] TOLL[10]
DUR: 020.33 DATE: 08/06/93 ST: 08:19:05 CON: 08:19:10 DIS: 08:19:25
Time->[229472]   Event->[CONSTRUCTION  ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[229472]   Event->[REF_REQUEST   ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[229472]   Event->[INBOUND_SETUP ] From->[PBX] State->[AWAITING_SETUP   ]
Time->[229472]   Event->[PROGRESSING   ] From->[IMC] State->[SETUP_IN_PROGRESS]
Time->[229473]   Event->[ALERTING      ] From->[IMC] State->[CALL_PROGRESSING ]
Time->[229493]   Event->[CONNECT       ] From->[IMC] State->[CALL_ALERTING    ]
Time->[229493]   Event->[CONNECT_ACK   ] From->[PBX] State->[CALL_CONNECTING  ]
Time->[229557]   Event->[DISCONNECT    ] From->[IMC] State->[CALL_CONNECTED   ]
Time->[229557]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[229557]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[229557]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[229558]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[229558]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[13] Events
```

```
*******************************************************************************
                        Group Inbound Call/Telephone Terminated
*******************************************************************************

CALL:[00089]  INBOUND CLEAR   GID[00273] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[]   SITES: O[00]  D[00] PRI[0/0]
DUR: 020.63 DATE: 08/06/93 ST: 08:03:25 CON: 08:03:32 DIS: 08:03:46
Time->[225552]   Event->[CONSTRUCTION  ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[225552]   Event->[REF_REQUEST   ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[225552]   Event->[INBOUND_SETUP ] From->[PBX] State->[AWAITING_SETUP   ]
Time->[225552]   Event->[PROGRESSING   ] From->[IMC] State->[SETUP_IN_PROGRESS]
Time->[225553]   Event->[ALERTING      ] From->[IMC] State->[CALL_PROGRESSING ]
Time->[225581]   Event->[CONNECT       ] From->[IMC] State->[CALL_ALERTING    ]
Time->[225581]   Event->[CONNECT_ACK   ] From->[PBX] State->[CALL_CONNECTING  ]
Time->[225638]   Event->[DISCONNECT    ] From->[PBX] State->[CALL_CONNECTED   ]
Time->[225638]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[13] Events


*******************************************************************************
            Individual Inbound Call Radio Forwarded/Telephone Terminated
*******************************************************************************

CALL:[00018]  INBOUND CLEAR   LID[06044] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[03] PHONE[] SITES: O[17] D[17]  PRI[2/2] TOLL[07]
RADIO FORWARD   CLEAR   LID[06367]
DUR: 029.26 DATE: 11/08/94 ST: 11:32:47 CON: 00:00:00 DIS: 11:33:16
Time->[1667771]   Event->[CONSTRUCTION  ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[1667771]   Event->[REF_REQUEST   ] From->[PBX] State->[INBOUND_ACTIVE   ]
Time->[1667771]   Event->[INBOUND_SETUP ] From->[PBX] State->[AWAITING_SETUP   ]
Time->[1667771]   Event->[FORWARD_RADIO ] From->[PI ] State->[FORWARDING       ]
Time->[1667771]   Event->[PROGRESSING   ] From->[IMC] State->[SETUP_IN_PROGRESS]
Time->[1667772]   Event->[ALERTING      ] From->[IMC] State->[CALL_PROGRESSING ]
Time->[1667793]   Event->[FWD_NO_ANSWER ] From->[PI ] State->[CALL_ALERTING    ]
Time->[1667793]   Event->[DISC_TO_IMC   ] From->[PI ] State->[FORWARDING_DROP  ]
Time->[1667793]   Event->[DISCONNECT_ACK] From->[IMC] State->[FORWARDING_DROP  ]
Time->[1667793]   Event->[FORWARD_RADIO ] From->[PI ] State->[FORWARDING       ]
Time->[1667793]   Event->[PROGRESSING   ] From->[IMC] State->[SETUP_IN_PROGRESS]
Time->[1667793]   Event->[ALERTING      ] From->[IMC] State->[CALL_PROGRESSING ]
Time->[1667888]   Event->[DISCONNECT    ] From->[PBX] State->[CALL_ALERTING    ]
Time->[1667888]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[1667888]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[1667888]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[1667888]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[1667888]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[18] Events
```

```
*******************************************************************************
                      Radio-Initiated Feature Code/PI Terminated
*******************************************************************************

CALL:[00005] OUTBOUND CLEAR   LID[06044] [FEATURE_SUCCESS_DISCONNECT]:PI
IMC[01] PBX[00] PHONE[0022002] SITES: O[01] D[00]  PRI[2/2] TOLL[07]
DUR: 005.88 DATE: 11/08/94 ST: 11:27:12 CON: 00:00:00 DIS: 11:27:18
Time->[1666431]   Event->[CONSTRUCTION  ] From->[IMC] State->[OUTBOUND_ACTIVE  ]
Time->[1666431]   Event->[CHANNEL_REQ   ] From->[IMC] State->[OUTBOUND_ACTIVE  ]
Time->[1666434]   Event->[OUTBOUND_SETUP] From->[IMC] State->[AWAITING_SETUP   ]
Time->[1666434]   Event->[FCODE_SUCCESS ] From->[PI ] State->[AWAITING_SETUP   ]
Time->[1666455]   Event->[FEEDBACK_LIMIT] From->[PI ] State->[FEATURE_FEEDBACK ]
Time->[1666455]   Event->[DISC_TO_IMC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[1666455]   Event->[DISCONNECT_ACK] From->[IMC] State->[DISCONNECTING    ]
Time->[1666455]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[8] Events
```

EXAMPLE 3

```
###############################################################################

                                CALL STATES DISABLED

###############################################################################

*******************************************************************************
                          Outbound Call/Telephone Terminated
*******************************************************************************

CALL:[00084] OUTBOUND CLEAR   LID[02405] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[92372004]   SITES: O[01]  D[00] PRI[5/6] TOLL[10]
DUR: 027.72 DATE: 08/06/93 ST: 07:54:00 CON: 07:54:03 DIS: 07:54:28

*******************************************************************************
                           Outbound Call/Radio Terminated
*******************************************************************************

CALL:[00111] OUTBOUND CLEAR   LID[06078] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[00] PHONE[2004]   SITES: O[01]  D[01] PRI[0/5] TOLL[03]
DUR: 006.20 DATE: 08/06/93 ST: 08:17:04 CON: 00:00:00 DIS: 08:17:10

*******************************************************************************
                      Individual Inbound Call/Telephone Terminated
*******************************************************************************

CALL:[00134]  INBOUND CLEAR   LID[03945] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[]   SITES: O[00]  D[00] PRI[1/1] TOLL[15]
DUR: 002.13 DATE: 08/06/93 ST: 09:15:14 CON: 00:00:00 DIS: 09:15:16
```

```
*******************************************************************************
                    Individual Inbound Call/Radio Terminated
*******************************************************************************

CALL:[00113]  INBOUND CLEAR   LID[02405] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[04] PHONE[]   SITES: O[00]  D[01] PRI[3/6] TOLL[10]
DUR: 020.33 DATE: 08/06/93 ST: 08:19:05 CON: 08:19:10 DIS: 08:19:25

*******************************************************************************
                    Group Inbound Call/Telephone Terminated
*******************************************************************************

CALL:[00089]  INBOUND CLEAR   GID[00273] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[]   SITES: O[00]  D[00] PRI[0/0]
DUR: 020.63 DATE: 08/06/93 ST: 08:03:25 CON: 08:03:32 DIS: 08:03:46

*******************************************************************************
          Individual Inbound Call Phone Forwarded/Callee Terminated
*******************************************************************************

CALL:[00014]  INBOUND CLEAR   LID[06183] [NORMAL_DISCONNECT]:FOR
IMC[00] PBX[01] PHONE[]   SITES: O[16]  D[16] PRI[0/1] TOLL[11]
PHONE FORWARD  PBX2[02] PHONE2[2002]
DUR: 010.50 DATE: 11/02/94 ST: 09:11:07 CON: 09:11:14 DIS: 09:11:18

*******************************************************************************
            Individual Inbound Call Common Forwarded/Caller Terminated
*******************************************************************************

CALL:[00022]  INBOUND CLEAR   LID[06183] [NORMAL_DISCONNECT]:PBX
IMC[00] PBX[04] PHONE[]   SITES: O[16]  D[16] PRI[0/1] TOLL[11]
COMMON FORWARD PBX2[00] PHONE2[2100]
DUR: 014.49 DATE: 11/02/94 ST: 09:16:44 CON: 00:00:00 DIS: 09:16:58

*******************************************************************************
             Individual Inbound Call Radio Forwarded/PI Terminated
*******************************************************************************

CALL:[00025]  INBOUND CLEAR   LID[06183] [NORMAL_DISCONNECT]:PI
IMC[01] PBX[01] PHONE[]   SITES: O[16]  D[00] PRI[0/1] TOLL[11]
RADIO FORWARD  CLEAR  LID[06044]
DUR: 010.26 DATE: 11/02/94 ST: 09:17:49 CON: 00:00:00 DIS: 09:17:49
```

5.2. CALL EVENTS

Call events that appear in the PI activity records are presented below

1. CONSTRUCTION
   Initiation of a new call.

2. REF_REQUEST
   PBX side is requesting a new call reference.

3. CHANNEL_REQ
   IMC side is requesting a new call reference.

4. INBOUND_ SETUP
   Setup for an inbound call.

5. OUTBOUND_SETUP
   Setup for an outbound call.

6. PROGRESSING
   Call is progressing.

7. ALERTING
   Telephone or radio is alerting (ringing).

8. CONNECT
   Called party has answered the call.

9. CONNECT_ACK
   Calling party acknowledges the connection.

10. DISCONNECT
    Request to terminate the call.

11. DISC_TO_ISDN
    PI instructs the PBX side to terminate the call.

12. DISC_TO_IMC
    PI instructs the IMC side to terminate the call.

13. DISC_TO_FOR
    PI instructs that a forwarded call be terminated.

14. DISCONNECT_ACK
    Protocol stack acknowledges the disconnect.

15. TERMINATED
    Call has been terminated by the ***shutdown -i*** command.

16. PROC_TIMEOUT
    PI times out waiting for the next event.

17. ALERT_TIMEOUT
    PI times out waiting for connect. This parameter is controlled by OUTBOUND_ALERTING_LIMIT or INBOUND_ALERTING_LIMIT, depending on call direction.

18. CONV_LIMIT
    Call has exceeded the conversation limit. This parameter only applies to phone and common forwarded calls (see PHONE_PHONE_CONV_LIMIT). The EDACS site controls the conversation limit for regular calls.

19. DISCONNECTED
    Call has been disconnected.

20. CHAN_REMOVED
    Channel that the call is using has been taken out of service. (This can occur if the PI-IMC control link is broken.)

21. FWD_NO_ANSWER
    Called party has not answered within the limit, call will be forwarded.

22. FORWARD_RADIO
    Call is being forwarded to a radio.

23. FORWARD_PHONE
    Call is being forwarded to a phone.

24. FORWARD_COMMON
    Call is being forwarded to a common number.

25. FCODE_SUCCESS
    Feature code was successfully processed.

26. FCODE_FAILURE
    Feature code was invalid.

27. FEEDBACK_LIMIT
    Call has been listening to audio feedback longer than the maximum time. Call will be terminated.

28. NUM_RESTRICT
    User attempted to call a number that is not allowed for his or her toll call class.

29. INVALID_EVENT
    PI received an unexpected event. Call will be terminated.

30. DISC_TO_VOICE
    PI instructs that a voice test call will be terminated.

5.3. CALL STATES

Various call states that appear in the PI activity records are listed below.

1. INACTIVE_CALLREF
   Call that has not been started.

2. INBOUND_ACTIVE
   Start of an inbound call.

3. OUTBOUND_ACTIVE
   Start of an outbound call.

4. AWAITING_SETUP
   Waiting for setup from calling party.

5. SETUP_IN_PROGRESS
   Waiting for progress from called party.

6. CALL_PROGRESSING
   Waiting for alerting or connect from called party.

7. CALL_ALERTING
   Both parties are ringing, waiting for connect.

8. CALL_CONNECTING
   Received connect from called party, waiting for connect acknowledgment from calling party.

9. CALL_CONNECTED
   Received connect acknowledgment from calling party, audio path connected.

10. DISCONNECTING
    Received disconnect, terminating the call.

11. BUSY_DISCONNECT
    Sending busy tone to the radio.

12. DISCONNECTED
    Termination of the call.

13. FORWARDING_DROP
    Dropping the current call to bring up a forwarded call.

14. FORWARDING
    Call is being forwarded.

15. FEATURE_FEEDBACK
    Sending audio feedback to radio user.

16. VN_FORWARDING
    Call is being forwarded/voice notifications messaging is enabled.

17. VN_NOANSWER
    Call has not been answered/voice notification messaging is enabled.

5.4. DISCONNECT REASON CODES

The following list contains disconnect reason codes that appear in the PI call activity records and the circumstances that surround the generation of these codes.

1. NORMAL_DISCONNECT
   Occurs when either the phone hangs up or the radio drops the call. Normal end of call.

2. CALLEE_BUSY_DISCONNECT
   Occurs when ISDN trunk returns a Disconnect, Clear, or Clear_with_Restart following a Call_Request. Usually occurs if the called phone is off-hook.

   Also occurs if the called LID or GID is already in an interconnect call.

3. NO_CHANNEL_AVAILABLE
   Occurs when no PI-IMC channel is available at the beginning of an inbound or outbound interconnect call.

4. INVALID_ID_SELECTED
   Occurs when an invalid radio ID digit sequence was entered at the calling phone.

5. TIMEOUT_NO_ANSWER
   Occurs when the called party does not answer during the 30-second period allowed for ringing.

6. PROCESSING_TIMEOUT
   Occurs when a call does not proceed through normal steps and reach the alerting state.

   For outbound calls this can occur if the dialed digits are not received within 3 seconds, or if the ISDN trunk does not return progress or alerting within 10 seconds.

   For inbound calls this can occur if confirmation is not received from all sites within 10 seconds.

7. USER_DENIED_DISCONNECT
   Occurs when the EDACS site denies the call, the radio ID is not in the range 1-16382, the group ID is not in the range 0-2047, the user is not valid on the Jessica site, or LID/GID is not inbound or outbound interconnect enabled.

8. FORCED_DISCONNECT
   Occurs when active calls are terminated at the user terminal. Disabling calls at the user terminal will force a disconnect of those calls that have not progressed to the point where audio has connected.

   Also occurs if a call queued or system busy confirmation status is received from the IMC during inbound call setup.

9. FAILURE_DISCONNECT
   Occurs if the PI-MD110 or PI-IMC trunks are in an alarm condition.
   Also can indicate mismatched call states between the PI and the MD110, or occur as result of an internal software failure, a full ISDN, or PMI request queue.

10. CHANNEL_REMOVED
    Occurs when status of a PI-IMC channel changes to disabled while the channel is in use. Will occur if the control link between the PI and the PIM is disabled.

11. HANG_TIME_EXPIRED
    Occurs when the radio user fails to key within the defined channel hang time.

12. NO_PHONE_DIGITS
    Occurs when valid phone digits are not received by the PI during the setup of an outbound call.

13. SYSTEM_BUSY_DISCONNECT
    Occurs when call cannot be added to the PMI's active call database.

14. CONVERSATION_LIMIT_EXCEEDED
    Occurs when the PI's conversation time limit is exceeded during an active call.

15. DO_NOT_DISTURB_DISCONNECT
    Occurs when a radio user has Do Not Disturb enabled without a forwarded destination.

16. FORWARD_ERROR_DISCONNECT
    Occurs when a forwarding destination is not valid.

17. FEATURE_SUCCESS_DISCONNECT
    Occurs when the PI times out while sending feature feedback tones on a successful feature code.

18. FEATURE_FAILURE_DISCONNECT
    Occurs when the PI times out while sending feature feedback tones on an unsuccessful feature code.

19. NUMBER_RESTRICT_DISCONNECT
    Occurs when a radio user attempts to call a phone number that is not allowed by the user’s toll call class.

6. COMMANDS AND SYNTAX

This section of the document provides detailed information on specific commands, their syntax, and examples. The commands should be typed at the prompt `pSH+>`. Please note that the commands are case-sensitive.

Command redirection to a file is a new capability of Jessica. The symbol ">" redirects the output to a file (command name > file_name) and overwrites an existing file_name**.** The symbol ">>" appends output to a file (command name >> file_name).

NOTE

Remote sessions (FTP, Telnet, and NFS) require XTERM (VT100) windows when executed from SUN systems.

The configuration parameter (PMI_ACTIVITY) in CONFIG.DAT controls PMI activity.

6.1. SYSTEM ADMINISTRATION COMMANDS

The following commands support system administration functions of the Jessica PI.

6.1.1. callres

callres -- displays the current called number restrictions in use by the PI.

USAGE:

callres

DESCRIPTION:

callres displays the current called number restrictions in use by the PI. The PI's internal allow and disallow tables are shown. The ***-u*** option allows the user to update the called number restrictions real-time from the ALLOW.DAT and DISALLOW.DAT files defined below.

OPTIONS:

- u Update called number restrictions real-time in the PI. The two called number restriction files will be read in and tables generated which will be referenced every time a call is placed. This information is read in initially when the PI is booted.

- t ALLOW.DAT and DISALLOW.DAT will be parsed for syntax errors. If syntax errors are found, a text message will be printed to the terminal. The option will prompt for the used to enter "**t**" to test a phone number and "**q**" to exit.

- ***callres*** with no options allows the user to view the call restrictions table which is stored internally in the PI. The format of the table is shown in Figure 3.

```
                                    Allow table
                                              11111
Number                  User Class  012345678901234
--------------------                --------------- 
123456789012345678                  YNYNYNYNYNYNY


                                   Disallow Table
                                              11111
Number                  User Class  012345678901234
--------------------                --------------- 
123456789012345678                  YNYNYNYNYNYNY
```

Figure 3 - Output from callres

***callres* Files**

Call restrictions are stored in two different files which are read upon startup of the PI. The allow file (ALLOW.DAT) contains numbers that the users are selectively permitted to call. The disallow file (DISALLOW.DAT) contains numbers that the users are selectively disallowed to call. This setup is beneficial if the super-user wants to restrict the calls of a certain class of users. For example, the users might be able to call any number except those starting with 1900. The disallow table makes such situations easier to set up. See ALLOW.DAT (section 4.2) and DISALLOW.DAT (section 4.3).

File Format

The pound character (#) indicates that the line contains comment information. Blank lines, spaces, and tabs are allowed in the files.

The beginning of a line contains the phone number. There may be spaces before the number, but there must be at least one space after it.

The phone numbers may contain the characters [0 - 9] | ? | *.

- ? Substitutes for exactly one character in a phone number.
  9? indicates a 9 followed by any one digit.
- * Substitutes for one or more characters in a phone number.
  9* indicates a 9 followed by any one or more digit(s).

The maximum characters for a number entry is 18.

Any combination of the characters above is allowed except * may occur only once in an entry and can not be followed by any character.

Following the phone number is a series of 15 letters, Y or N. These letters indicate whether the call is explicitly permitted for a certain class of user. There may be 0 or more spaces between these letters. The first entry is class 0 and the last entry is class 14. The information must fit on one line.

Warnings will be printed for files containing syntax errors and files not conforming to the file format. The internal PI tables will not be built for these files.

6.1.2. config

config -- displays the current configuration

USAGE:

config -fu [-g SM|CM|IMC|UI|PI|ISDN|ACT] [-s name value]

DESCRIPTION:

config displays the current configurations of Jessica.

By default, the ***config*** command displays only partial configuration information.

OPTIONS:

- f Displays full configuration list
- u Update -- reads in configuration from the configuration file (CONFIG.DAT)
- s Sets configuration parameter to the assigned value
- g Displays configuration parameters by group
  - SM System Management
  - CM Call Management
  - IMC Integrated Multisite Controller
  - ISDN Integrated Services Data Network (IMC/MD110 link parameters)
  - ACT Activity Recording

These options may not be combined.

EXAMPLES:

The example below reloads the configuration from the CONFIG.DAT file on the hard drive.

```
pSH+> config -u
Config Data Reloaded
```

The example below shows the full configuration.

```
pSH+> config -f


MUX_CHANNELS_MASK = 0XF

TRUNK_TYPE = T1
TRUNK_CLOCK_SOURCE = RECOVER_MD110_TRUNK
MUX_TRUNK_FRAMING = ESF
MUX_TRUNK_CODING = B8ZS
MD110_TRUNK_FRAMING = ESF
MD110_TRUNK_CODING = B8ZS
Q931_CONNECTION_TYPE = ATT5ESS_USER
ISDN_BOOT_ACTIVITY = FALSE
ISDN_RAW_ACTIVITY = FALSE
ISDN_AUX_ACTIVITY = FALSE
ISDN_CALL_ACTIVITY = FALSE
ISDN_STATUS_ACTIVITY = FALSE
```

```
SITE_ID = 14
CONVERSATION_LIMIT = 5
HANG_TIME = 20

VERBOSE_USER_DEBUG = FALSE

ALL_DEBUG_ENABLED = FALSE
DISK_LOG_CALLS = TRUE
PORT_LOG_CALLS = TRUE
DISK_LOG_STATES = TRUE
PORT_LOG_STATES = TRUE
DISK_LOG_ISDN = FALSE
PORT_LOG_ISDN = FALSE
LOG_BASE_PATH = 01.02/activity
ACTIV_QUEUE_TIMEOUT = 0
```

The example below displays configuration parameters for the ISDN group.

```
pSH+> config -g ISDN
TRUNK_TYPE                T1
TRUNK_CLOCK_SOURCE        RECOVER_MD110_TRUNK
MUX_TRUNK_FRAMING         ESF
MUX_TRUNK_CODING          B8ZS
MD110_TRUNK_FRAMING       ESF
MD110_TRUNK_CODING        B8ZS
Q931_CONNECTION_TYPE      ATT5ESS_USER
ISDN_BOOT_ACTIVITY        FALSE
ISDN_RAW_ACTIVITY         FALSE
ISDN_AUX_ACTIVITY         FALSE
ISDN_CALL_ACTIVITY        FALSE
ISDN_STATUS_ACTIVITY      FALSE
CALLER_ID                 FALSE
```

6.1.3. date

date -- displays or sets the date

USAGE:

date [yyyymmddhhmm [.ss]]

DESCRIPTION:

If no argument is given, ***date*** displays the current date and time. Otherwise, the current date will be set. ***date*** should only be used to display the current time and date.

yyyy is the four digits of the year; the first ***mm*** is the month number; ***dd*** is the day number in the month; ***hh*** is the hour number (24-hour system); the second ***mm*** is the minute number; ***.ss*** (optional) specifies the seconds. The year may be omitted; the current value is supplied as the default.

The command ***timesav*** should be used to set the system's time and date.

NOTE:

The date and time must be set after the PI is set up to assure proper operation.

EXAMPLE:

The example below shows how to display the current date and time.

```
pSH+> date
19:46:30 Aug 4 1993
```

6.1.4. dbv

dbv -- displays the current LID/GID attributes set by the System Manager

USAGE:

dbv

DESCRIPTION:

dbv displays the current LID/GID attributes set by the System Manager. The ***-l*** option allows the user to view the LID attributes as contained in the LID/GID database. The ***-g*** option allows the user to view the GID attributes as contained in the LID/GID database. The attributes contain LID/GID database and feature code settings.

OPTIONS:

- l Allows the user to view the LID/GID database and feature code settings for the LID chosen.
- g Allows the user to view the LID/GID database and feature code settings for the GID chosen.

EXAMPLES:

```
pSH+> dbv -l 6525
Valid
Outbound Enabled
Inbound Enabled
Priority 7
Toll Call Class 0
Phone Forwarding 2020
Do Not Disturb Disabled

pSH+> dbv -g 273
Valid
Inbound Enabled
Priority 7
```

6.1.5. disc

disc -- ends a call started by the ***voice*** command

USAGE:

disc

DESCRIPTION:

disc simply ends a call begun by the ***voice*** command. Calls started with the ***voice*** command must be ended either by using the ***disc*** command or by the radio callee.

OPTIONS:

None.

EXAMPLE:

```
pSH+> disc
```

6.1.6. passwd

passwd -- performs password management

USAGE:

passwd [login_name]

DESCRIPTION:

passwd changes (or installs) a password (maximum of 8 characters; any characters beyond the 8 maximum will cause the compare to fail) associated with the user's username (your own by default). When changing a password, ***passwd*** prompts for the old password and then for the new one. You must supply both, and the new password must be typed twice to preclude mistakes. Only the owner of the name or the super-user may change a password; the owner must prove he knows the old password. The super-user can change any password, and is the account authorized to install a new user.

OPTIONS:

None.

NOTES:

1. Login and Password entries are case-sensitive.
2. To delete an account, (1) copy the file 1.2/etc/passwd to floppy disk, (2) edit the file with any ASCII text editor (simply delete the line which contains the user account to be removed), (3) convert the file on floppy back to UNIX format (i.e., LF terminates each line, rather than the CR/LF), and if necessary, (4) copy the file from floppy back to 1.2/etc. Changes will be effective upon the next system reboot. Note that the filename 1.2/etc/passwd is case-sensitive. You may need to use the ***mv*** command to rename the file to lower case if it is copied from a floppy disk.

EXAMPLE:

```
pSH+> passwd
Old password:
New password:
Retype new passwd:
Changed password for user
```

6.1.7. product

product -- used to view and modify product feature encryption licensing

USAGE:

product [-lsw]

DESCRIPTION:

product is used to view and modify product feature encryption licensing.

OPTIONS:

- l lists the feature encryption licenses installed on this unit.
- s sets which licenses are installed on this unit. Note that this option requires special privileges and can only be executed by authorized Ericsson Inc. personnel.
- w lists the encryption licenses available on this unit.

6.1.8. reboot

reboot -- reboots the system

USAGE:

reboot [-hs]

DESCRIPTION:

reboot performs an immediate and complete system reboot.

OPTIONS:

- h Performs immediate full system reboot (hard).
- s Restarts Jessica application already in memory.

NOTES:

The ***sync*** command should be issued prior to reboot to assure that all user data are flushed to the hard disk.

EXAMPLE:

```
pSH+> reboot -h


<OS> Beginning bootstrap loader: DATE: August 4, 1993 TIME: 8:18:42 pm

<OS> Scanning hard disk for LOADER.SX...
<OS> Loading file LOADER.SX from the hard disk...
<OS> Load complete. 70735 bytes loaded.
<OS> Hard disk has been unmounted.
<OS> Transferring program control to LOADER module...
<LOADER> Verifying system disk integrity...
<LOADER> Disk verification complete. No errors detected.

<LOADER> Installing Jessica Application: DATE: August 4, 1993 TIME: 8:19:08 pm

<LOADER> Scanning hard disk for 01.02/loads/JESSICA.SX...
<LOADER> Loading file JESSICA.SX from the hard disk...
<LOADER> Load complete. 391572 bytes loaded.
<LOADER> Hard disk has been unmounted.
<LOADER> Transferring program control to JESSICA...


Jessica System Initializing: DATE: August 4, 1993 TIME: 8:20:29 pm
<Jessica> Pri-48 found at [F000FFF0]
<Jessica> Pri-64 found at [FFFF200D]
<Jessica> Scanning the hard disk for PRI.SX...
<Jessica> Loading file PRI.SX at memory address [0x006D3910]...
<Jessica> Done. 115429 bytes loaded
Login:
Password:

Copyright (c) Integrated Systems, Inc., 1992.
Welcome to pSOSystem...

pSH+>
```

6.1.9. report

report -- displays the number of calls made, total duration of calls (in seconds), and average duration of calls (in seconds) for a range of LIDs (if the **-l** option is specified), GIDs (if the **-g** option is specified), or both, in an activity log.

USAGE:

report [-lgdio] activity.log [id#1] [id#2]

activity.log is the path to the activity log to be used in generating the report.

id#1 is the lower end of the range of radio LIDs (or GIDs if **-g** is specified) to be included in the report.

id#2 is the upper end of the range of radio LIDs (or GIDs if **-g** is specified) to be included in the report.

NOTES:

- If neither *id#1* nor *id#2* are entered as arguments, it is assumed that all LIDs or GIDs are to be included in the report. If only *id#1* is entered, only that LID or GID will be included in the report.
- Options and arguments may be interspersed. There is no limitation on the order of the options, but the arguments must come in the order *activity.log, id#1, id#2.*

DESCRIPTION:

report displays the number of calls made, total duration of calls (in seconds), and average duration of calls (in seconds) for a range of LIDs (if the **-l** option is specified), GIDs (if the **-g** option is specified), or both, in an activity log. The user can tell the report command to print information on the individual calls included in the report by including the **-d** option. The user may also specify that the report be generated using inbound (phone-generated) calls, outbound (radio-generated) calls or both, through use of the **-i** and **-o** options.

OPTIONS:

- l Specifies that individual calls are to be included in the report (LIDs are used).
- g Specifies that group calls are to be included in the report (GIDs are used).
- d Specifies that a detailed report will be given. The **-d** option outputs detailed information about the individual calls that are included in the report before it outputs the report statistics. After each screen of output, the user is prompted to enter any key to see the next screen of output, or **q** to quit (as in the ***more*** command). The information output for each call is identical to the information contained in non-verbose activity log entries. For more information on non-verbose activity log entries, see section 5.
- o Specifies that outbound calls are to be included in the report. Outbound calls are calls from a radio to a phone.
- i Specifies that inbound calls are to be included in the report. Inbound calls are calls from a phone to a radio.

NOTES:

- If neither **-l** nor **-g** are specified, both individual and group calls are included in the report.
- If neither **-i** nor **-o** are specified, both inbound and outbound calls are included in the report.

EXAMPLES:

The example below includes all calls in a non-detailed report.

```
pSH+> report activity/jul16.log
number of calls: 56  duration: 560.00sec  average duration: 10.00sec
```

The following example includes all individual calls in a non-detailed report.

```
pSH+> report -l activity/jan01.log
number of calls: 47  duration: 481.75sec  average duration: 10.25sec
```

The example below includes all group calls in a detailed report.

```
pSH+> report -gd activity/sep16.log
CALL:[00013]  INBOUND CLEAR   GID[00117] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[]   SITES: O[00]  D[00] PRI[0/0]
DUR: 020.63 DATE 9/16/94   ST: 08:03:25 CON: 08:03:32 DIS: 08:03:46

CALL:[00226]  INBOUND CLEAR   GID[00023] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[01] PHONE[]   SITES: O[01]  D[01] PRI[0/0]
RADIO FORWARD   CLEAR     GID[0099]
DUR: 026.89 DATE 9/16/94   ST: 16:45:12 CON: 16:45:21 DIS: 16:45:39

CALL:[00250]  INBOUND CLEAR   GID[00117] [NORMAL_DISCONNECT]:PBX
IMC[01] PBX[03] PHONE[]   SITES: O[00]  D[00] PRI[0/0]
DUR: 051.23 DATE 9/16/94   ST: 17:12:01 CON: 17:12:07 DIS: 17:12:52

      number of calls: 3  duration: 98.75sec  average duration: 32.92sec
```

The following example includes individual inbound calls to radios with LIDs ranging from 00007 to 00010 in a non-detailed report.

```
pSH+> report -li activity/feb02.log 7 10
number of calls: 6  duration: 60.72sec  average duration: 10.12sec
```

The example below includes outbound calls from a radio with LID 00092 in a detailed report.

```
pSH+> report -odl activity/feb02.log 92
CALL:[00002] OUTBOUND DIGITAL LID[00092] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[03] PHONE[2446]   SITES: O[01]  D[01] PRI[0/1]
DUR: 062.14 DATE 02/02/95  ST: 09:12:07 CON: 09:12:13 DIS: 09:13:09

CALL:[00045] OUTBOUND DIGITAL LID[00092] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[01] PHONE[2001]   SITES: O[01]  D[00] PRI[1/1]
DUR: 006.20 DATE 02/02/95  ST: 10:15:45 CON: 00:00:00 DIS: 10:15:51

CALL:[000130] OUTBOUND DIGITAL LID[00092] [NORMAL_DISCONNECT]:IMC
IMC[01] PBX[01] PHONE[2001]   SITES: O[01]  D[00] PRI[1/1]
DUR: 078.91 DATE 02/02/95  ST: 11:45:28 CON: 11:45:33 DIS: 11:46:52

      Number of calls: 3  duration: 147.25sec  average duration: 49.08sec
```

6.1.10. restart

restart -- restarts the Jessica application

USAGE:

restart

DESCRIPTION:

restart performs a restart of the Jessica application so that new calls are allowed.

OPTIONS:

None.

EXAMPLE:

```
pSH+> restart
Call Controls    = [CALLS_ENABLED]  >> [0] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
IMC Comm Status  = [COMM_ENABLED     ] [Link Up]
[10] Total Channels
[0] Active Channels
```

6.1.11. savecfg

savecfg -- saves current settings

USAGE:

savecfg

DESCRIPTION:

savecfg saves all of the current settings (everything seen when you type ***config -f***), including parameters which have been changed from the command line or the System Manager interface, into the CONFIG.DAT file. During this save, ***savecfg*** overwrites comments that were in the previous configuration settings. A backup of CONFIG.DAT called CONFIG.BKP is made. The user is queried whether he wants to perform this action.

OPTIONS:

None.

EXAMPLE:

```
pSH+> savecfg
savecfg:01.02/cnfg/CONFIG.DAT copied to 01.02/cnfg/CONFIG.BKP  1331 bytes copied
savecfg:01.02/cnfg/CONFIG.DMP copied to 01.02/cnfg/CONFIG.DAT  1331 bytes copied
```

6.1.12. shutdn

shutdn -- shuts down Jessica

USAGE:

shutdn [-i]

DESCRIPTION:

shutdn performs the shutdown of Jessica so that new calls are denied.

OPTIONS:

- Performs orderly shutdown, existing calls remain unaffected.
- -i Performs immediate shutdown, existing calls are dropped.

EXAMPLE:

```
pSH+> shutdn -i
Call Controls    = [CALLS_TERMINATED]  >> [0] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
IMC Comm Status  = [COMM_ENABLED     ] [Link Up]
[4] Total Channels
[0] Active Channels
Activity Recording Shutdown on DATE: August 4, 1993 TIME: 7:46:03 pm
```

6.1.13. snap

snap -- provides a snapshot of the LID/GID, Priority, Toll Class, IMC Channel, PBX Channel, originating site, call number, call state, and phone number (outbound calls only) for all active calls.

USAGE:

snap

DESCRIPTION:

snap provides a snapshot of the LID/GID, Priority, Toll Class, IMC Channel, PBX Channel, originating site, call number, call state, and phone number (outbound calls only) for all active calls.

The output for ***snap*** will be formatted in a table as follows:

Number of call references:*xx*

LID/ GID	PBX Chan	IMC Chan	IMC PRI	Toll Class	Orig Site	Call #	Call State	Phone # Outbound

Figure 4 - Output for *snap* Command

OPTIONS:

None.

EXAMPLE:

```
pSH+> snap
Number of call references: 2
LID/  PBX  IMC  IMC Toll  Orig Call # Call              Phone #
GID   Chan Chan PRI Class Site        State             Outbound
----- ---- ---- --- ----- ---- ------ ----------------- ------------------
6065  2    1    0   11    11   82     CALL_CONNECTED    95281131
6367  1    2    0   11    11   81     CALL_CONNECTED    2020
```

6.1.14. status

status -- displays the current system status

USAGE:

status

DESCRIPTION:

status is used to view current status.

OPTIONS:

None.

EXAMPLE:

```
pSH+> status
Call Controls    = [CALLS_ENABLED]  >> [1] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
IMC Comm Status  = [COMM_ENABLED     ] [Link Up]
[10] Total Channels
[1] Active Channels
```

6.1.15. timesav

timesav -- saves the current date and time

USAGE:

timesav

DESCRIPTION:

timesav is used to save the current date and time in a battery-backed clock.

OPTIONS:

None.

EXAMPLE:

```
pSH+> timesav
Time saved in battery backed real time clock.
```

6.1.16. version

version -- displays software versions

USAGE:

version

DESCRIPTION:

version displays the current software revisions of the Jessica application and operating system components.

OPTIONS:

None.

EXAMPLE:

```
pSH+> version

Jessica Release:     [V01.01]
OS Software Version: [V01.02]
```

6.1.17. voice

voice -- allows the user to play back or update voice notification messages over a radio

USAGE:

voice -bcdeflnortu radio_lid
-h

radio_lid is the logical ID of the radio on which the message will be played.

DESCRIPTION:

voice is used to modify and play voice notification messages over a particular radio.

Playing a message:

The message played is specified by the option entered. Enter the LID of the radio on which the message will be played as an argument. When the ***voice*** command is entered, the radio starts ringing. Key the radio to begin message play. To disconnect the call, enter the ***disconnect*** command, or disconnect the call from the radio as for a regular call. ***Since there can be only one active voice notification test call at a time, the call must be disconnected from the radio or with the disconnect command before another call can be placed.***

Updating messages:

Locate the new voice notification message in the file 1.02/cnfg/TMPVOICE.DAT file. The message can be tested using the procedure described above (the -u option is used to play the message in 1.02/cnfg/TMPVOICE.DAT). To substitute the new voice notification message for one of the existing voice notification messages, enter the user interface ***cp*** or ***mv*** commands to replace the existing voice file with the file in TMPVOICE.DAT. ***The system must be rebooted for the new message to take effect.***

OPTIONS:

- b Play bad feature code message (filename FC_RJCT.DAT)
- c Play system congested message (filename CN_TONE.DAT)
- d Play do not disturb disabled message (filename DSTRB_D.DAT)
- e Play do not disturb enabled message (filename DSTRB_E.DAT)
- f Play call forwarding message (filename FORWARD.DAT)
- l Play user not logged on message (filename NOT_LOG.DAT)
- n Play number restricted message (filename NUM_RES.DAT)
- o Play feature code OK message (filename FC_ACPT.DAT)
- r Play redial: no last number stored (filename REDIAL.DAT)
- t Play ringing tone (filename RINGING.DAT)
- u Play user-created message (filename TMPVOICE.DAT)
- h Print help screen

NOTES:

Only one message may be played at a time.

EXAMPLES:

The example below plays the number restricted message on the radio with LID 6872.

```
pSH+> voice -n 6872
Proceeding with voice notification test call.
```

The example below substitutes the message stored in 01.02/config/TEMPVOICE.DAT for the “feature code OK message.”

```
pSH+> voice -u -o
```

6.2. FILE MAINTENANCE AND UTILITY COMMANDS

The following commands provide the maintenance and utility functions for the Jessica file system. These commands apply to both the system hard disk (volume 1.2) and floppy disk (volume 1.1) drives.

6.2.1. cat

cat -- concatenates and displays

USAGE:

cat [-benstv] filename...

DESCRIPTION:

cat reads each ***filename*** in sequence and displays it on the standard output.

OPTIONS:

- b Numbers the lines as ***-n*** but omits the line numbers from blank lines.
- e Displays non-printable characters as ***-v*** and in addition displays a $ character at the end of each line.
- n Precedes each line output with its line number.
- s Substitutes a single blank line for multiple adjacent blank lines.
- t Displays non-printable characters as ***-v*** and in addition displays TAB characters such as ^I (CTRL-I).
- v Displays non-printable characters (with the exception of TAB and NEWLINE characters) so that they are visible. Control characters print such as ^X for CTRL-X; the DEL character (octal 0177) prints as “^?”. Non-ASCII characters (with the high bit set) are displayed as M-x, where M- stands for "meta" and x is the character specified by the seven low order bits.

NOTES:

Using ***cat*** to redirect output of a file to the same file, such as ***cat filename1*** > ***filename1*** or ***cat filename1*** >> ***filename1***, does not work. This type of operation should be avoided at all times since it may cause the system to enter an indeterminate state.

Once started, ***cat*** cannot be aborted. It may, however, be suspended and resumed using flow control characters ^S and ^Q.

EXAMPLE:

```
pSH+> cat mmmdd.log
Starting Activity Logging : DATE: July 24, 1993 TIME: 1:45:59 pm  {when logging
started}
```

6.2.2. cd

cd -- changes working directory

USAGE:

cd [directory]

DESCRIPTION:

directory becomes the new working directory.

OPTIONS:

None.

EXAMPLE:

```
pSH+> cd /
```

6.2.3. clone

clone -- copies system files (wildcards supported)

USAGE:

clone -q *full_path/sourcefile destination_full_path*

DESCRIPTION:

clone is a file copy command similar to ***cp***, which supports the use of wildcard characters. Directories may not be copied with ***clone***.

clone may copy to a PC mounted file system but it can not copy from one. When copying files to a PC mounted file system, PC file naming conventions must be used. File name extensions following the dot (.) are limited to three characters.

Multiple files can be written to the destination directory since wildcards are allowed. If there is a collision (i.e., there is a file in the destination directory with the same name as the file being copied) the default is for ***clone*** to query the user whether to overwrite the file. The user responds with:

Y or y -- copies the source file overwriting the file in the destination directory
N or n -- does not overwrite the file
Q or q -- quits clone
Any other response results in the file not being deleted.

OPTIONS:

- q Quiet. Suppresses the query for whether matching destination files should be overwritten.

Source file names may include multiple question marks (?) and/or wildcards (*).
- ? Question marks substitute for one character in a file name.
- * Wildcards substitute for none to many characters in a filename.

EXAMPLE:

Directory 01.02/files contains:

misc.dat	1234.dat
1235.dat	demo

"demo" is a directory.

Directory 01.02 contains:
1234.dat

User enters:
```
clone 01.02/files/*.dat  01.02
```

Terminal displays:
```
clone:01.02/1234.dat exists. Overwrite?n
clone:01.02/files/demo is a directory
```

The files misc.dat, 1234.dat, and 1235.dat match. misc.dat and 1235.dat are copied to the 01.02 directory. The user is queried if 1234.dat should be removed. "n" is the response and so it is not removed.

User enters:
```
clone -q 01.02/files/*.dat  01.02
```

Terminal displays:
```
clone:01.02/files/demo is a directory
```

The files misc.dat, 1234.dat, and 1235.dat match. misc.dat, 1234.dat, and 1235.dat are copied to the 01.02 directory. 1234.dat is overwritten without inquiry.

It is not necessary to enter the volume name for the hard drive. 01.02/ and / have the same meaning. When copying to a floppy, however, the user must provide the volume name of the floppy.

6.2.4. cmp

cmp -- performs a byte-by-byte comparison of two files

USAGE:

cmp [-ls] filename1 filename2 [skip1] [skip2]

DESCRIPTION:

cmp compares ***filename1*** and ***filename2***. With no options, ***cmp*** makes no comment if the files are the same. If they differ, it reports the byte and line number at which the difference occurred, or, that on file is an initial subsequence of the other. ***skip1*** and ***skip2*** are initial byte offsets into ***filename1*** and ***filename2***, respectively. These offsets may be either octal or decimal, where a leading 0 denotes octal.

OPTIONS:

- l Prints the byte number (in decimal) and the differing bytes (in octal) for all differences between the two files.
- s Silent. Prints nothing for differing files.

EXAMPLE:

```
pSH+> cmp oct06.log oct07.log
oct06.log oct07.log differ: char 11, line 1
```

6.2.5. cp

cp -- copies files

USAGE:

cp [-i] filename1 filename2
cp -rR [-i] directory1 directory2
cp [-irR] filename ... directory

DESCRIPTION:

cp copies the contents of ***filename1*** onto ***filename2***. If ***filename1*** is a symbolic link, or a duplicate hard link, the contents of the file that the link refers to are copied; links are not preserved.

In the second form, ***cp*** recursively copies ***directory1***, along with its contents and subdirectories, to ***directory2***. If ***directory2*** does not exist, ***cp*** creates it and duplicates the files and subdirectories of ***directory1*** within it. If ***directory2*** does exist, ***cp*** makes a copy of the ***directory1*** directory (along with its files and subdirectories) within ***directory2*** (as a subdirectory).

In the third form, each ***filename*** is copied to the indicated ***directory***; the basename of the copy corresponds to that of the original. The destination ***directory*** must already exist for the copy to succeed.

OPTIONS:

- i Interactive. Prompt for confirmation whenever the copy would overwrite an existing file. A "y" in answer to the prompt confirms that the copy should proceed. Any other answer prevents ***cp*** from overwriting the file.
- r or R Recursive. If any of the source files are directories, copy the directory along with its files (including subdirectories and their files). The destination must be a directory.

NOTES:

cp refuses to copy a file onto itself.

The wildcard character "*" is not supported.

EXAMPLE:

```
pSH+> cp aug03.exp /temp
pSH+>

pSH+> cp -i aug03.exp /temp
overwrite /temp/aug03.exp? y
pSH+>
```

The following example copies a log file to a floppy disk.

```
pSH+> pcmount 1.1
pSH+> cp aug 31.log     1.1/aug31.log
pSH+> umount 1.1
```

6.2.6. df

df -- displays file system usage

USAGE:

df

DESCRIPTION:

df displays the amount of disk space occupied by currently mounted file systems, the amount of used and available space, and the amount of the file system's total capacity that has been used.

OPTIONS:

None.

EXAMPLE:

```
pSH+> pcmount 1.1
pSH+> df
Filesystem    kbytes      used     avail  capacity  Mounted on
01.01           1440      1244       196   86.389%  01.01/
01.02        1211728      7133   1204595    0.589%  /
```

6.2.7. du

du -- displays the number of disk blocks used per directory or file

USAGE:

du [-sa] [filename ...]

DESCRIPTION:

du provides the number of 512-byte disk blocks contained in all the files and, recursively, directories within each specified directory of file ***filename***. If ***filename*** is missing, "." (the current directory) is used. If no option is selected, entries are generated only for each directory.

OPTIONS:

- s Only displays the grand total for each of the specified filenames.
- a Generates an entry for each file.

EXAMPLE:

```
pSH+> du
2       ./etc
378     ./tmp
1026    ./bin
0       ./usr
0       ./var
0       ./export
0       ./mnt
9763    ./loads
495     ./cnfg
680     ./backup
4329    ./activity
183     ./log
381     ./temp
0       ./cdr
0       ./rar
18688   .
pSH+> du -s oct07.log
7       oct07.log
```

6.2.8. head

head -- displays the first few lines of specified files

USAGE:

head [-n] [filename ...]

DESCRIPTION:

head copies the first ***n*** lines of each ***filename*** to the standard output. If no ***filename*** is given, ***head*** copies lines from the standard input. The default value of ***n*** is 10 lines.

When more than one file is specified, the start of each file will appear as follows:

```
==> filename <==
```

Thus, a common way to display a set of short files, identifying each one, is as follows:

```
pSH+> head -1 filename1 filename2 ...
```

EXAMPLE:

```
pSH+> head -3 junk1 junk2 junk3

==> junk1 <==

Line 1 of junk1...
Line 2 of junk1...
Line 3 of junk1...

==> junk2 <==

Line 1 of junk2...
Line 2 of junk2...
Line 3 of junk2...

==> junk3 <==

Line 1 of junk3...
Line 2 of junk3...
Line 3 of junk3...
```

6.2.9. ls

ls -- lists the contents of a directory

USAGE:

ls [-aACdfFgilqrRs1] filename

DESCRIPTION:

For each ***filename*** that is a directory, ***ls*** lists the contents of the directory; for each ***filename*** that is a file, ***ls*** repeats its name and any other information requested. By default, the output is sorted alphabetically. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents.

OPTIONS:

- a Lists all entries. In the absence of this option, entries having names beginning with a "." are not listed.
- A Same as ***-a***, except that "." and ".." are not listed.
- C Forces multi-column output, with entries sorted down the columns; for ***ls***, this is the default when output is to a terminal.
- d If the argument is a directory, lists only its name (not its contents); often used with ***-l*** to obtain the status of a directory.
- f Forces each argument to be interpreted as a directory and lists the name found in each slot. This option turns off ***-l***, ***-s***, and ***-r***, and turns on ***-a***; the order is the order in which entries appear in the directory.
- F Marks directories with a trailing slash (/) and executable files with a trailing asterisk (*).
- g For ***ls***, shows the group ownership of the file in a long output.
- i For each file, prints the ***i***-number in the first column of the report.
- l Lists in long format, providing mode, owner, size in bytes, and time of last modification for each file. If the time of last modification is greater than six months ago, it is shown in the format "month date year"; files modified within six months show "month date time".
- q Displays non-graphic characters in filenames as the character "?"; for ***ls***, this is the default when output is to a terminal.
- r Reverses the order of sort to receive reverse alphabetic or oldest first, as appropriate.
- R Recursively lists the subdirectories encountered.
- s Provides the size of each file, including any indirect blocks used to map the file, in kilobytes.
- 1 Forces single-column output.

NOTE

Avoid entering "Print Scrn" before the ***ls*** command since it may cause some systems to reboot, temporarily suspending service.

EXAMPLES:

```
pSH+> ls
aug02.log       jul24.log       jul27.log       jul30.log
aug03.log       jul25.log       jul28.log       pmistats.dat
aug04.log       jul26.log       jul29.log


pSH+> ls -ls
total 545
  98 -rwxrwxrwx   1 root         115299 Aug 02 1993 23:57 aug02.log
  55 -rwxrwxrwx   1 root          93477 Aug 03 1993 22:24 aug03.log
 111 -rwxrwxrwx   1 root         187501 Aug 04 1993 19:20 aug04.log
   2 -rwxrwxrwx   1 root            983 Jul 24 1993 22:29 jul24.log
  10 -rwxrwxrwx   1 root           5000 Jul 25 1993 23:46 jul25.log
  85 -rwxrwxrwx   1 root          43039 Jul 26 1993 22:47 jul26.log
   8 -rwxrwxrwx   1 root         200623 Jul 27 1993 22:51 jul27.log
  37 -rwxrwxrwx   1 root          18499 Jul 28 1993 23:07 jul28.log
 102 -rwxrwxrwx   1 root          52060 Jul 29 1993 22:35 jul29.log
  36 -rwxrwxrwx   1 root          18235 Jul 30 1993 15:47 jul30.log
   1 -rwxrwxrwx   1 root            110 Jul 27 1993 19:31 pmistats.dat
```

6.2.10. mkdir

mkdir -- makes a directory

USAGE:

mkdir [-p] dir_name ...

DESCRIPTION:

mkdir creates the directory ***dir_name***.

OPTIONS:

- p Allows missing parent directories to be created as needed.

EXAMPLE:

```
pSH+> ls
BITMAP.SYS      activity        cnfg            log             tmp
FLIST.SYS       backup          etc             mnt             usr
LOADER.BCU      bin             export          rar             var
LOADER.SX       cdr             loads           temp
pSH+> mkdir exp_dir
pSH+> ls
BITMAP.SYS      activity        cnfg            loads           temp
FLIST.SYS       backup          etc             log             tmp
LOADER.BCU      bin             exp_dir         mnt             usr
LOADER.SX       cdr             export          rar             var
```

6.2.11. more

more -- browses or pages through a text file

USAGE:

more [-l lines] file1 ... filen

DESCRIPTION:

more is a filter that displays the contents of a text file on the terminal, one screen at a time. It pauses after each screen, and prints `--More--` at the bottom of the screen. To continue browsing the file, enter the CR (<Return>) character (i.e., Enter or Return). To terminate ***more***, enter the character "q", followed by a CR.

OPTIONS:

- l Displays the file in increments of line_count lines (default is 23 lines).

EXAMPLE:

```
pSH+> more -l 15 /loads/JESSICA.SX
S00600004844521B
S31400158F8C4572696373736F6E2047453A204A65C0
S31400158F9B7373696361204953444E2043544953F8
S31400158FAA3A204D564D452D3134372C20414D3734
S30900158FB939393000F7
S31400158FBE7265626F6F743A207265626F6F7420F9
S31400158FCD7468652073797374656D000014C9880F
S31400158FDC00158FBE00100A5A0000000000000095
S31200158FEB000000000000000000000000005E
S30C0014C9887265626F6F740003
S314001000004E56FFF8487800004EBB817000027212
S3140010000F18588F2D40FFFC4EBB8170000272688F
S3140010001E4EBB8170000272AE1D40FFFB4A2EFFD3
S3140010002DFB6712202EFFFC068000000210206ECB
S3140010003CFFFC21400200206EFFFC7032214000B5
--More--q
pSH+>
```

6.2.12. mv

mv -- moves or renames files

USAGE:

mv [-fi] filename1 filename2
mv [-fi] directory1 directory2
mv [-fi] filename ... directory

DESCRIPTION:

mv moves files and directories around in the file system. A side effect of ***mv*** is to rename a file or directory. The three major forms of ***mv*** are shown in the usage synopsis above.

The first form of ***mv*** moves (changes the name of) ***filename1*** to ***filename2.*** If ***filename2*** already exists, it is removed before ***filename1*** is moved.

The second form of ***mv*** moves (changes the name of) ***directory1*** to ***directory2***, only if ***directory2*** does not already exist; if it does, the third form applies.

The third form of ***mv*** moves one or more *filename*(s) (may also be directories) with their original names into the last ***directory*** in the list.

OPTIONS:

- f Force. Overrides any mode restrictions and the **-i** option. The **-f** option also suppresses any warning messages about modes which would potentially restrict overwriting.
- i Interactive mode. ***mv*** displays the name of the file or directory followed by a question mark whenever a move would replace an existing file or directory. If a line starting with "y" is typed, ***mv*** moves the specified file or directory, otherwise ***mv*** does nothing with that file or directory.

NOTES:

mv refuses to move a file or directory onto itself.

mv will not move a directory from one file system to another.

EXAMPLES:

```
pSH+> mv oct19.exp /temp
pSH+>

pSH+> mv -i oct19.exp /temp/oct19.exp2
overwrite /temp/oct19.exp2? y
pSH+>
```

6.2.13. purge

purge -- purges system files

USAGE:

purge -q [-d YYYY MM DD HH] full_path/filename

DESCRIPTION:

purge is a file deletion utility which supports wildcard characters (*) in *full_path/filename*. *rm* can only remove files one by one in pSOS. Directories may not be removed with ***purge***.

OPTIONS:

- q Quiet. User confirmation for file deletion is suppressed.
- [-d YYYY MM DD HH] Files which come before this date and match *full_path/filename* are deleted.

File names may include multiple question marks (?) and/or wildcards (*).

- ? Question marks substitute for one character in a file name.
- * Wildcards substitute for none to many characters in a filename.

NOTES:

When ***purge*** is used from a remote login (i.e., Telnet), some connection types may require that <RETURN> be entered twice when prompted for file deletion confirmation. ***purge*** may be aborted during a confirmation prompt by entering **q**, (rather than **y** or **n**), followed by <RETURN>.

purge does not work on a PC mounted file system.

As a default, users are queried as to whether they want files deleted. The user responds as follows:

Y or y -- deletes the file
N or n -- does not delete the file
Q or q -- quits purge
Any other response results in the file not being deleted.

EXAMPLE:

Directory 01.02/files contains:

misc.dat	1234.dat
1235.dat	demo

"demo" is a directory.

```
pSH+> purge 01.02/files/*.dat
purge:remove 01.02/files/misc.dat?y
Purge:remove 01.02/files/1234.dat?y
purge:remove 01.02/files/1235.dat?n
purge:01.02/files/demo is a directory
```

The files misc.dat, 1234.dat, and 1235.dat match. The user is asked if these should be deleted. misc.dat and 1234.dat are deleted since the user responded "y". 1235.dat is not deleted since the user responded "n".

```
pSH+> purge -q 01.02/files/*.dat
purge:01.02/files/demo is a directory
```

The files misc.dat, 1234.dat, and 1235.dat match. The files are deleted without inquiry.

Note: It is not necessary to enter the volume name for the hard drive. 01.02/ and / have the same meaning.

6.2.14. rm

rm -- removes (unlink) files

USAGE:

rm [-fir] filename

DESCRIPTION:

rm removes (directory entries for) one or more files. If an entry was the last link to the file, the contents of that file are lost.

OPTIONS:

- f Forces files to be removed without displaying permissions, asking questions, or reporting errors.
- i Asks whether to delete each file, and, under **-r**, whether to examine each directory. Sometimes called the "interactive option."
- r Recursively deletes the contents of a directory, its subdirectories, and the directory itself.

NOTES:

Removing the file ".." is forbidden to avoid inadvertently causing a problem such as "rm -r .*".

EXAMPLE:

```
pSH+> rm -i sep29.exp
rm: remove sep29.exp? y
pSH+>
```

6.2.15. rmdir

rmdir -- removes (unlinks) directories

USAGE:

rmdir directory

DESCRIPTION:

rmdir removes each named ***directory***. ***rmdir*** only removes empty directories.

OPTIONS:

None.

EXAMPLE:

```
pSH+> ls
BITMAP.SYS      FLIST.SYS       config.log      sep29.exp       temp1
pSH+> rmdir temp1
pSH+> ls
BITMAP.SYS      FLIST.SYS       config.log      sep29.exp
```

6.2.16. sync

sync -- forces changed blocks to disk

USAGE:

sync

DESCRIPTION:

sync brings a mounted volume up to date. It does this by writing to the volume all modified file information for open files, and flushing cache buffers containing physical blocks that have been modified.

This call is superfluous under immediate write synchronization mode, and is not allowed on an NFS mounted volume.

OPTIONS:

None.

EXAMPLE:

```
pSH+> sync
pSH+>
```

6.2.17. tail

tail -- displays the last part of a file

USAGE:

tail +|-number [lc] filename

DESCRIPTION:

tail copies ***filename*** to the standard output beginning at a designated place.

OPTIONS:

Options are not specified separately with their own "-" signs.

- -number Begins copying at distance number from the end of the file. number is counted in units of lines or characters, according to the appended option **-l** or **-c**. When no units are specified, counting is by lines. If number is not specified, the value 10 is used.
- l number is counted in units of lines.
- c number is counted in units of characters.

EXAMPLES:

```
pSH+> tail -5l wan.log
WAN Server Boot Complete

WAN_ACTIVITY_RECORD at Fri Oct 21 14:14:17 1994
WAN Port [0] On Line

pSH+> tail -50c wan.log
at Fri Oct 21 14:14:17 1994
WAN Port [0] On Line
```

6.2.18. touch

touch -- updates the access and modification times of a file

USAGE:

touch [-cf] filename

DESCRIPTION:

touch sets the access and modification times of each ***filename*** argument to the current time. ***filename*** is created if it does not exist (default).

OPTIONS:

- c Does not create filename if it does not exist.
- f Attempts to force the ***touch*** in spite of read and write permissions on filename.

EXAMPLE:

```
pSH+> touch -c sep29.exp
```

6.3. FLOPPY DISK COMMANDS

The following commands are specifically associated with the floppy disk drive of the PI.

6.3.1. pcmkfs

pcmkfs -- initializes a volume for an MS-DOS file system

USAGE:

pcmkfs [-i] volume_name format

DESCRIPTION:

pcmkfs initializes (i.e., formats) the volume ***volume_name*** for the MS-DOS disk type specified by ***format***; where ***format*** is one of the following:

1 = 360 Kbyte (5-1/4" double density)
2 = 1.2 Mbyte (5-1/4" high density)
3 = 720 Kbyte (3-1/2" double density)
4 = 1.4 Mbyte (3-1/2" high density)

OPTIONS:

- i Calls device driver initialization procedure.

NOTES:

The PI supports ***format*** specification 4 only; 1.4 Mbyte (3-1/2" high density)

EXAMPLE:

The following example would correctly format a diskette installed in the floppy disk drive.

```
pSH+> pcmkfs 1.1 4
Warning: this operation will destroy all data on the specified volume.
Do you wish to continue (y/n)? y
```

6.3.2. pcmount

pcmount -- mounts an MS-DOS file system.

USAGE:

pcmount volume_name [sync_mode]

DESCRIPTION:

pcmount will mount an MS-DOS volume ***volume_name***. A volume must be mounted before any file operations can be carried out on it. ***sync_mode*** specifies one of the following file system synchronization methods for the volume:

0 = Immediate write synchronization mode.
1 = Control write synchronization mode.
2 = Delayed write synchronization mode (default).

OPTIONS:

None.

NOTES:

The ***volume_name*** for the PI floppy disk drive is 1.1. A diskette should be inserted in the floppy drive prior to using ***pcmount***.

The ***umount*** command should be used prior to removing the diskette.

EXAMPLE:

The following example would mount a diskette in the PI floppy disk drive.

```
pSH+> pcmount 1.1
```

6.3.3. umount

umount -- unmounts the file systems

USAGE:

umount volume

DESCRIPTION:

umount unmounts a previously mounted file system ***volume***. Unmounting a file system causes it to be synchronized (all memory-resident data will be flushed to the device).

OPTIONS:

None.

EXAMPLE:

The following example mounts the PI floppy disk drive, changes directory to it, and lists the contents. The current directory is restored to the hard disk drive, and the floppy is unmounted.

```
pSH+> pcmount 1.1
pSH+> cd 1.1/
pSH+> ls
LOADER.SX SYSTEM.TXT
pSH+> cd 1.2/
pSH+> umount 1.1
pSH+> cd 1.1/
1.1/: no such file or directory
```

6.4. NETWORKING COMMANDS

The following commands provide statistical and diagnostics services associated with the TCP/IP networking capability of the PI.

6.4.1. netstat

netstat -- displays network statistics

USAGE:

netstat topic [-as]

DESCRIPTION:

netstat displays a variety of statistical information regarding network activity. This command can be especially valuable in characterizing network performance.

topic specifies the network statistical entity of interest, which may be one of the following:

if	Interface group statistics
icmp	Internet Control Message Protocol (ICMP) group statistics
ip	Internet Protocol (IP) group statistics
tcp	Transmission Control Protocol (TCP) group statistics
udp	User Datagram Protocol (UDP) group statistics

OPTIONS:

- a Displays all information available within the statistics group, including any “special” information.
- s Displays only “special” information available with the statistics group.

EXAMPLE:

```
pSH+> netstat tcp -s
tcpRtoAlgorithm = [4]: Van Jocobson
tcpRtoMin       = 1000       tcpRtoMax       = 64000
tcpMaxConn      = -1         tcpActiveOpens  = 11
tcpPassiveOpens = 7          tcpAttemptFails = 57
tcpEstabResets  = 0          tcpCurrEstab    = 3
tcpInSegs       = 19650      tcpOutSegs      = 15834
tcpInErrs       = 1          tcpOutRsts      = 0
tcpRetransSegs  = 0
```

6.4.2. ping

ping -- sends ICMP ECHO_REQUEST packets to network hosts

USAGE:

ping [-s] host_address [timeout]

DESCRIPTION:

ping utilizes the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified host, or network gateway. ECHO_REQUEST datagrams, or "pings," have an IP and ICMP header, followed by a timeval structure, and then an arbitrary number of bytes to pad out the packet. If ***host_address*** responds, ***ping*** will print out a message indicating that the host is alive, then exit. Otherwise, after ***timeout*** seconds, it will print out a message indicating that no answer was received, then exit. The default value of ***timeout*** is 10 seconds.

If the ***-s*** option is specified, ***ping*** sends one datagram per second, and prints one line of output for every ECHO_RESPONSE it receives. No output is produced if there is no response from ***host_address***. The default datagram size is 64 bytes (8 byte ICMP header + 56 data bytes).

When using ***ping*** for fault isolation, first “ping” the local host (127.0.0.1) to verify that the local network interface is running.

host_address must be specified in Internet dotted-decimal notation.

OPTIONS:

- s Sends one “ping” per second to host_address.
- timeout Maximum time to wait, in seconds, for a response from host_address.

EXAMPLE:

```
pSH+> ping 190.A.B.C
PING (190.A.B.C): 56 data bytes
190.A.B.C is alive.
```

6.4.3. route

route -- examines/modifies network routes

USAGE:

route [-f] [-sh] add|delete [host|net] destination gateway

DESCRIPTION:

route manually manipulates the network routing tables normally maintained by the system routing daemon, the configuration file ROUTES.DAT, or through default routes, and redirects messages from routers. ***route*** allows the super-user to operate directly on the routing table for the specific host or network indicated by ***destination***. The ***gateway*** argument indicates the network gateway to which packets should be addressed.

The ***add*** command instructs ***route*** to add a route to ***destination***. ***delete*** deletes a route. ***destination*** and ***gateway*** must be specified in Internet dotted-decimal notation. Any user may display the current routes using the **-s** or **-h** options. Only the super-user "root" may add or delete routes.

Routes to a particular host must be distinguished from those to a network. The optional keywords ***net*** and ***host*** force the ***destination*** to be interpreted as a network or a host, respectively. If neither the ***net*** or ***host*** keywords are supplied, the route is presumed to be to a host.

ERROR MESSAGES:

Permission denied. -- Attempt by non-super-user to add or delete a route entry.

Invalid arguments. -- Incorrect route parameters were entered.

The route specified cannot be found. -- (1) Attempting to delete a route with an incorrect host or net keyword. The host/net specification must match the route type. (2) Attempting to delete a non-existent route.

Network is unreachable. -- An attempt to add a route failed because the gateway listed was not on a directly connected network. Give the next-hop gateway instead.

Internal routing table out of space. -- An add operation was attempted, but the system was unable to allocate memory to create the new entry.

OPTIONS:

- f Flushes the network routing table.
- s Displays routes in Internet dotted-decimal notation.
- h Display routes in hexadecimal notation.
- host Specifies destination as an IP host address (default).
- net Specifies destination as an IP network address.

EXAMPLES:

```
pSH+> route -s
Destination:     Next hop:       IF:     Type:           Subnet Mask:
127.A.A.B        127.A.A.B       7       DIRECT          0xFF000000
147.AAA.CB.O     147.AAA.BJ.CDE  1       DIRECT          0xFFFFF000

pSH+> route add host 147.XXX.A.B  147.XXX.CF.BDG
pSH+> route add net  147.XXX.O  147.XXX.CB.B

pSH+> route -s
Destination:     Next hop:       IF:     Type:           Subnet Mask:
127.A.A.B        127.A.A.B       7       DIRECT          0xFF000000
147.XXX.CB.0     147.AAA.BJ.CDE  1       DIRECT          0xFFFFF000
147.XXX.A.B      147.XXX.CF.BDG  1       INDIRECT        0xFFFFF000
147.XXX.O.O      147.XXX.CB.B    1       INDIRECT        0xFFFFF000
```

6.4.4. tcpcon

tcpcon -- examines the TCP connection table

USAGE:

tcpcon -alrhH [-i ip_addr] [-p port_num] [-c conn_state]

DESCRIPTION:

tcpcon allows the user to examine the current TCP connection table entries. Several options are available which allow the table to be searched for only particular parameters of interest. In addition, the table search can be specified for either the local or remote (network) side of a connection. In the absence of other qualifiers, the default is to search both sides of a connection for any parameter specification(s).

Note that TCP connections are transient in nature. Table entries are present only as long as the connection in question exits. Also, recall that TCP implements a pseudo three-way handshaking to communicate over and shut down a connection. This is mentioned since the TCP connection table may appear quite dynamic in nature.

OPTIONS:

- a Examines all current TCP connection table entries. Overrides any other options.
- l Searches the local connection side for a specific IP address (**-i**) and/or TCP port number (**-p**).
- r Searches the remote connection side for a specific IP address (**-i**) and/or TCP port number (**-p**).
- h Provides help. Supplies a brief syntax and options explanation.
- H Provides extended help. Furnishes a list of connection state values which may be used with **-c**.
- i Searches for connection(s) involving the IP address specified by ***ip_addr***.
- p Searches for connection(s) involving the TCP port number specified by ***port_num***.
- c Searches for connections(s) which are currently in the state specified by ***conn_state***.

conn_state is a decimal value representing a specific TCP connection state. The possible values of ***conn_state*** are provided below. The intricacies of TCP connection state transitions are beyond the scope of this document. The explanations given are limited for the sake of brevity.

- 1 Closed. A Transmission Control Block (TCB) has been allocated, but is not currently in use.
- 2 Listen. A server daemon associated with the local port is awaiting connection(s) from clients.
- 3 Sync sent. A sync has been sent, and the connection side is awaiting acknowledgment.
- 4 Sync received. A sync has been received, or a sync has been sent without receiving an acknowledgment.
- 5 Established. Both sides are ready to exchange data and acknowledgments.
- 6 Final wait 1. One side of the connection has requested to close it , and is waiting for a response.
- 7 Final wait 2. One side of the connection has received a request to close it, and is ready to proceed with the shutdown.
- 8 Close wait. A connection shutdown is in progress for reasons other than a close request.
- 9 Last acknowledge. Awaiting final acknowledgment in closing down the connection.
- 10 Closing. Both sides of the connection have agreed to shut down.
- 11 Time wait. A graceful shutdown of the connection has been completed.
- 12 Delete TCB. A TCB allocated for the connection is being returned to the system pool.

EXAMPLE:

```
pSH+> tcpcon -H
usage: tcpcon -alrhH [-i ip_addr] [-p port_num] [-c conn_state]

conn_state is a decimal value representing a specific TCP connection state.
The possible values of conn_state are as follows:
         1 = Closed
         2 = Listen
         3 = Sync Sent
         4 = Sync Received
         5 = Established
         6 = Final Wait 1
         7 = Final Wait 2
         8 = Closed Wait
         9 = Last Ack
        10 = Closing
        11 = Time Wait
        12 = Delete TCB
```

6.5. UTILITY COMMANDS

The following commands provide general purpose utilities.

6.5.1. clear

clear -- clears the terminal screen

USAGE:

clear

DESCRIPTION:

clear attempts to clear the current terminal screen. It is an alternative to the ***clr*** command. User preference between ***clr*** and ***clear*** depends on the terminal characteristics and network connection type.

See also ***clr***.

OPTIONS:

None.

EXAMPLE:

```
pSH+> clear
```

6.5.2. clr

clr -- clears the terminal screen

USAGE:

clr

DESCRIPTION:

clr clears a terminal display, such as a VT100, or xterm connection. It is an alternative to the ***clear*** command. User preference between ***clr*** and ***clear*** depends on the terminal characteristics and network connection type.

OPTIONS:

None.

EXAMPLE:

```
pSH+> clr
```

6.5.3. exit

exit -- exits the shell

USAGE:

exit

DESCRIPTION:

exit exits (i.e., logs out) the user from the shell.

OPTIONS:

None.

EXAMPLE:

```
pSH+> exit
Connection closed by foreign host.
```

6.5.4. help

help -- provides help about shell commands

USAGE:

help [command_name]

DESCRIPTION:

help prints to the terminal information about shell commands. If no ***command_name*** is given, ***help*** prints out a list of available commands. If a valid ***command_name*** is given, help prints out information about that command.

OPTIONS:

None.

NOTES:

help on an individual command provides two additional types of information: whether the command is reentrant, and whether it is currently locked. If a command is indicated as reentrant, it may be used simultaneously by multiple shell users. A command indicated as not being reentrant is only available to one user at a time. Lock status indicates if a non-reentrant command is currently in use by another user. If a user attempts to execute a command that is currently locked, a message indicating "`Command not reentrant`" will be displayed.

EXAMPLE:

```
pSH+> help
ftp   telnet
cat   cmp     echo    help     mkfs     pcmount  pwd     setenv  suspend  umount
cd    cp      getid   kill     mount    ping     resume  setid   sync
clear date    getpri  ls       mv       popd     rm      setpri  tail
console du    head    mkdir    pcmkfs   pushd    rmdir   sleep   touch
config  reboot        shutdn   timesav  more     lp       clr
pmi   restart status  version  passwd   scsi

pSH+> help cat
cat      - concatenate and display (reentrant, not locked)
```

The example below retrieves help about the ***passwd*** command.

```
pSH+> help passwd
passwd - password management: passwd [login_name] (not reentrant, not locked)
```

6.5.5. lp

lp -- prints a text file

USAGE:

lp file_name

DESCRIPTION:

lp queues the text file specified by ***file_name*** to be output on the optional PI printer. ***file_name*** may be either an absolute or relative file specification on the PI system disk. ***lp*** does not support printing files from a floppy disk.

OPTIONS:

None.

EXAMPLE:

```
pSH+> lp sep29.exp
```

6.6. STATISTICS AND DIAGNOSTICS COMMANDS

The following commands provide statistical and diagnostics services associated with Jessica and its operation within an EDACS infrastructure.

6.6.1. scsi

scsi -- queries SCSI bus for active devices

USAGE:

scsi -n

DESCRIPTION:

scsi queries the SCSI bus for an active device, where ***-n*** is the SCSI bus address ID of the target device. Device information, such as the vendor, model number, and storage capacity are displayed on the terminal. ***n*** = 0 implies that all seven (1-7) SCSI addresses should be queried for devices.

OPTIONS:

None.

NOTES:

For removable storage devices (i.e., floppy disk or tape), no information will be available unless the media (i.e., diskette) is installed in the target device.

EXAMPLES:

The following example queries all seven SCSI IDs for active devices.

```
pSH+> scsi
SCSI ID: 1      No information available.
SCSI ID: 2      LUN: 0  Removable:  NO  System volume: 1.2
Blocks:  166243         Block Size: 512 bytes
Vendor:  CONNER         Model: CP30080E-85Mb
Type:    Direct access (disk)
SCSI ID: 3      No information available.
SCSI ID: 4      No information available.
SCSI ID: 5      No information available.
SCSI ID: 6      No information available.
SCSI ID: 7      No information available.
```

The following example queries a specific SCSI ID for an active device -- in this case, the PI system hard disk.

```
pSH+> scsi -2
SCSI ID: 2      LUN: 0     Removable:  NO  System volume: 1.2
Blocks:  166243               Block Size: 512 bytes
Vendor:  CONNER               Model: CP30080E-85Mb
Type:    Direct access (disk)
```

6.7. ENVIRONMENT COMMANDS

The following commands are associated with subtle characteristics of a user's operational environment. These commands are seldom, if ever, used.

6.7.1. getid

getid -- obtains user ID and group ID

USAGE:

getid

DESCRIPTION:

getid displays the user ID (uid) and group ID (gid) of the shell user. Values should reflect "/etc/passwd" assignments on a remote Unit Host. The command may be used for privilege validation when the Jessica PI is being accessed from a remote host.

OPTIONS:

None.

EXAMPLE:

```
pSH+> getid
uid: 20, gid: 100
```

6.7.2. popd

popd -- pops the directory stack

USAGE:

popd

DESCRIPTION:

popd pops the directory stack, and changes the current working directory to the new top directory.

OPTIONS:

None.

EXAMPLE:

```
pSH+> pushd activity
pSH+> pwd
1.2/activity
pSH+> popd
pSH+> pwd
1.2/
```

6.7.3. pushd

pushd -- pushes the current directory onto the directory stack

USAGE:

pushd directory

DESCRIPTION:

pushd pushes ***directory*** onto the directory stack, and changes the current working directory to that directory.

OPTIONS:

None.

EXAMPLE:

```
pSH+> pwd
1.2/
pSH+> pushd activity
pSH+> pwd
1.2/activity
```

6.7.4. setenv

setenv -- sets environment variables

USAGE:

setenv variable_name value

DESCRIPTION:

setenv changes a shell's variables to a new value. If used without any arguments, ***setenv*** prints a list of the shell variables and their current values.

OPTIONS:

None.

NOTES:

Currently, the only variable that can be changed is TERM.

EXAMPLE:

```
pSH+> setenv
CVOL=1.2
CDIR=/
SOFLIST=5
LOGNAME=root
IND=0
OUTD=0
TERM=sun
pSH+> setenv TERM vt100
CVOL=1.2
CDIR=/
SOFLIST=5
LOGNAME=root
IND=0
OUTD=0
TERM=vt100
```

6.7.5. setid

setid -- sets user ID and group ID

USAGE:

setid uid gid

DESCRIPTION:

setid changes the shell user's user ID to ***uid***, and group ID to ***gid***.

OPTIONS:

None.

EXAMPLE:

```
pSH+> getid
uid: 20, gid: 100
pSH+> setid 20 169
pSH+> getid
uid: 20, gid: 169
```

6.8. NETWORK MANAGER COMMANDS

NOTE

Release 3.0 of Jessica integrates the Billing Correlation Unit (BCU) with the PI. With this integration, the BCS user interface (discussed in LBI-38967) becomes available for use. In addition, the ***stats*** command (discussed in LBI-38967) and its options become available, but the command is referred to as the ***traf*** command instead of the ***stats*** command.

6.8.1. comm

comm -- allows the root user to view, add an entry to, delete an entry from, or modify an entry in the SNMPv1 Get-Set Community Profile Table.

USAGE:

comm -a -i address -c name -v view
-d -i address -c name
-m -i address [-c name] [-v view]
-h
-r

address is the IP address of the entry to be added/deleted/modified. It must be a class A, B, or C network address. Class D and class E network addresses are not supported. Network class versus IP address is as follows:

Class A:	0.0.0.0	through	127.255.255.255
Class B:	128.0.0.0	through	191.255.255.255
Class C:	192.0.0.0	through	223.255.255.255
Class D:	224.0.0.0	through	239.255.255.255 (Multicast address)
	224.0.0.0	is RESERVED.	
	244.0.0.1	permanently assigned to the `all hosts group'.	
Class E:	240.0.0.0	through	255.255.255.255 (Reserved)

name is the community name of the entry to be added/deleted/modified. It must consist of no more than 31 printable non-whitespace characters and cannot start with the "-" character.

view is the MIB view of the entry being added/modified. It can take on the following three values: "read," "write," or "none."

DESCRIPTION:

comm allows the root user to view, add an entry to, delete an entry from, or modify an entry in the SNMPv1 Get-Set Community Profile Table stored in the PI's 1.2/cnfg/COMCNFG.DAT file. The root user can also restore the Get-Set Community Profile to its default state. There can be a maximum of 10 entries in the table. A table entry contains a specific machine's IP address, get-set community name, and MIB view.

OPTIONS:

- -a Adds an entry with the specified IP address, community name, and MIB view to the SNMPv1 Get-Set Community Profile Table.
- -c Specifies that the next argument is a community name.
- -d Deletes an entry with the specified IP address from the SNMPv1 Get-Set Community Profile Table. If a community name is provided, both the IP address and community name will be matched before an entry is deleted.
- -h Displays help information for this command.
- -i Specifies that the next argument is an IP address.
- -m Modifies the community name and/or MIB view of the specified entry in the SNMPv1 Get-Set Community Profile Table.
- -r Restores the SNMPv1 Get-Set Community Profile Table to its default state.
- -v Specifies that the next argument is a MIB **v**iew ("read," "write," or "none").

EXAMPLES:

The following example adds an entry to the SNMPv1 Get-Set Community Profile Table with IP address 147.XXX.CG.DH, community name "toe_jam," and write MIB view.

```
pSH+> comm -a -i 147.XXX.CG.DH -c toe_jam -v write
```

The following example deletes an entry from the SNMPv1 Get-Set Community Profile Table with IP address 147.XXX.CJ.AA and community name "delete_me."

```
pSH+> comm -d -i 147.XXX.CJ.AA -c delete_me
```

The following example modifies an entry in the SNMPv1 Get-Set Community Profile Table with IP address 147.XXX.CL.AAA, and changes the community name to "fred_garvin" and the access to read.

```
pSH+> comm -m -i 147.XXX.CL.AAA -c fred_garvin -v read
```

The following example restores the PI's SNMPv1 Get-Set Community Profile Table to its default state.

```
pSH+> comm -r
```

6.8.2. trap

trap -- allows the root user to view, add an entry to, or delete an entry from the SNMPv1 Trap Destination Table.

USAGE:

trap -a -i address
-d -i address
-r
-e
-p

address is the IP address of the entry to be added/deleted/modified. It must be a class A, B, or C network address. Class D and class E network addresses are not supported. Network class versus IP address is as follows:

Class A:	0.0.0.0	through	127.255.255.255
Class B:	128.0.0.0	through	191.255.255.255
Class C:	192.0.0.0	through	223.255.255.255
Class D:	224.0.0.0	through	239.255.255.255 (Multicast address)
	224.0.0.0	is RESERVED.	
	244.0.0.1	permanently assigned to the "all hosts group."	
Class E:	240.0.0.0	through	255.255.255.255 (Reserved)

DESCRIPTION:

The ***trap*** command allows the root user to view, add an entry to, or delete an entry from the SNMPv1 Trap Destination Table stored in the PI's 1.2/cnfg/TRAPCNFG.DAT file. In addition, the root user can enable/prevent the PI from sending authentity traps. There can be a maximum of 10 entries in the table. A table entry contains a specific machine's IP address.

OPTIONS:

- -a Adds an entry with the specified IP address to the PI's SNMPv1 Trap Destination Table.
- -d Deletes an entry with the specified IP address from the PI's SNMPv1 Trap Destination Table.
- -i Specifies that the next argument is an IP address.
- -e Enables the PI to send authentity traps.
- -p Prevents the PI from sending authentity traps.
- -r Restores the PI's SNMPv1 Trap Destination Table to its default state.

EXAMPLES:

The following example adds an entry to the PI's SNMPv1 Trap Destination Table with IP address 147.XXX.CG.LK.

```
pSH+> trap -a -i 147.XXX.CG.LK
```

The following example deletes an entry from the PI's SNMPv1 Trap Destination Table with IP address 147.XXX.CM.AA.

```
pSH+> trap -d -i 147.XXX.CM.AA
```

The following example restores the PI's SNMPv1 Trap Destination Table to its default state.

```
pSH+> trap -r
```

The example below enables the PI to send authentity traps.

```
pSH+> trap -e
```

The following example prevents the PI from sending authentity traps.

```
pSH+> trap -p
```

6.8.3. xtrap

xtrap -- allows the user to send traps to the instances in the trapDestinationTable for testing.

USAGE:

xtrap -a
-c
-d *ifIndex*
-e *value*
-f
-h
-i *ip_addr*
-r
-u *ifIndex*
-w

DESCRIPTION:

The ***xtrap*** command allows the user to send traps to the instances in the trapDestinationTable for testing purposes. Both generic and enterprise-specific traps can be sent. The ***-h*** option displays a help screen which shows the options and the usage of the command. To send generic traps, the ***-cwaud*** options are used (***-c*** is coldStart, ***-w*** is warmStart, ***-a*** is authenticationFailure, ***-u*** is linkUp, and ***-d*** is linkDown). For the linkUp and linkDown generic traps, an interface index is also needed. For sending rising and falling threshold alarm traps, the ***-r*** and ***-f*** options are used. To send enterprise-specific traps, the ***-e*** option along with its specific value is used. The ***-i*** option allows traps to be sent to one IP address only (this option may be combined with others).

OPTIONS:

- -a Sends a generic authenticationFailure trap.
- -c Sends a generic coldStart trap.
- -d ***ifIndex*** Sends a generic linkDown trap on ***ifIndex***.
- -e ***value*** Sends an enterprise-specific trap of value ***value***.
- -f Sends a falling threshold alarm trap.
- -h Displays the xtrap usage.
- -i ***ip_addr*** Sends a trap to only ***ip_addr*** only (e.g., xtrap -e 10000 -i 147.XXX.DD.FC).
- -r Sends a rising threshold alarm trap.
- -u ***ifIndex*** Sends a generic linkUp trap on ***ifIndex***.
- -w Sends a generic warmStart trap.

EXAMPLES:

The example below sends a generic coldStart trap.

```
pSH+> xtrap -c
```

The following example sends a generic authenticationFailure trap to IP address 147.XXX.DD.FC.

```
pSH+> xtrap -a -i 147.XXX.DD.FC
```

6.9. ADVANCED COMMANDS

NOTE

The following section discusses commands which, in general, should not be used on the PI. Reasons for avoiding their use are summarized below. The commands are provided for factory troubleshooting.

1. Similar functions are provided which are uniquely tailored to the functional requirements of Jessica.

2. Their function is superfluous to the operations of the PI. That is, the function of the command either is not applicable or is provided under a different context. For example, the ***mkfs*** and ***mount*** commands are somewhat meaningless since the system disk of the PI is maintained under application control.

3. Their incorrect use may result in degraded or catastrophic system behavior (e.g., incorrect use of the ***kill*** command may result in the PI's crashing).

6.9.1. console

console -- redirects the terminal output to a Telnet session

USAGE:

console [-r] [task_name]

DESCRIPTION:

console redirects output going to the PI's system terminal to a Telnet session. The default is to redirect all output to the Telnet session. If a ***task_name*** is given, only the output from that task will be redirected to the Telnet session.

OPTIONS:

- r Redirects input from the Telnet session. Note that if a task is currently waiting for terminal input when this command is issued, the task's input redirection will take effect only after it returns from the waiting.

NOTES:

There is no graceful way to undo ***console*** redirection. Use of the ***console*** command should be avoided.

6.9.2. echo

echo -- echoes arguments to the standard output

USAGE:

echo [-n] [argument ...]

DESCRIPTION:

echo writes its arguments on the standard output. ***argument***(s) must be separated by SPACE characters or TAB characters, and terminated by a NEWLINE character.

OPTIONS:

- n Does not add the NEWLINE to the output.

6.9.3. getpri

getpri -- displays the priority of a task

USAGE:

getpri task_name | -task_id

DESCRIPTION:

getpri displays the priority of a task named ***task_name***, or with a task ID of ***task_id***.

OPTIONS:

None.

6.9.4. kill

kill -- terminates a task

USAGE:

kill task_name | -task_id

DESCRIPTION:

kill terminates a task named ***task_name***, or with an ID of ***task_id***. It does this by calling t_restart with a second argument of **1**. The task must be designed to read this second argument and perform its own resource cleanup, then terminate.

OPTIONS:

None.

6.9.5. mkfs

mkfs -- makes a file system (volume initialization)

USAGE:

mkfs [-i] volume_name label size num_of_fds

DESCRIPTION:

mkfs initializes a file system volume ***volume_name*** and labels it with ***label***. Its size will be ***size*** and the number of file descriptors will be ***num_of_fds***.

OPTIONS:

- i Calls device driver initialization procedure.

6.9.6. mount

mount -- mounts a file system volume

USAGE:

mount volume_name [sync_mode]

DESCRIPTION:

mount mounts a pHILE+ formatted volume ***volume_name***. A volume must be mounted before any file operations can be carried out on it. ***sync_mode*** specifies one of the following file system synchronization methods for the volume:

0 = Immediate write synchronization mode.
1 = Control write synchronization mode.
2 = Delayed write synchronization mode (default).

Permanent (i.e., non-removable media) volumes need only be mounted once. Removable volumes must be mounted and unmounted as required.

OPTIONS:

None.

6.9.7. resume

resume -- resumes a suspended task

USAGE:

resume task_name | -task_id

DESCRIPTION:

resume will resume a task named ***task_name***, or with an ID of ***task_id***, that was previously suspended.

OPTIONS:

None.

6.9.8. setpri

setpri -- sets task priority

USAGE:

setpri task_name | -task_id new_priority

DESCRIPTION:

setpri sets the priority of a task named ***task_name***, or with an ID of ***task_id***, to ***new_priority***.

OPTIONS:

None.

6.9.9. sleep

sleep -- suspends execution for the specified interval

USAGE:

sleep time

DESCRIPTION:

sleep suspends execution for ***time*** seconds.

OPTIONS:

None.

6.9.10. suspend

suspend -- suspends a task

USAGE:

suspend task_name | -task_id

DESCRIPTION:

suspend suspends a task named ***task_name***, or with an ID of ***task_id***.

OPTIONS:

None.

APPENDIX A
TELNET COMMAND

telnet - user interface to a remote system using the TELNET protocol

USAGE:

telnet [host [port]]

OPTIONS:

host - Specify the IP address of the remote host, in Internet dotted-decimal notation.
port - Specify which port number on the remote host to establish the connection to.

DESCRIPTION

telnet communicates with another host using the TELNET protocol. If ***telnet*** is invoked without arguments, it enters command mode, indicated by its prompt (**telnet**>). In this mode, it accepts and executes the commands listed below. If it is invoked with arguments, it performs an ***open*** command (see below) with those arguments.

Once a connection has been opened, ***telnet*** enters "character at a time" input mode. Text typed is immediately sent to the remote host for processing.

If the ***localchars*** toggle is TRUE , the user's **quit**, **intr**, and **flush** characters are trapped locally, and sent as TELNET protocol sequences to the remote side. There are options (see toggle autoflush and toggle autosynch below) which cause this action to flush subsequent output to the terminal (until the remote host acknowledges the TELNET sequence) and flush previous terminal input (in the case of quit and intr).

While connected to a remote host, ***telnet*** command mode may by entered by typing the telnet "escape character" (initially `^**]**, (control-right-bracket)). When in command mode, the normal terminal editing conventions are available.

TELNET COMMANDS

The following commands are available. Only enough of each command to uniquely identify it need be typed (this is also true for arguments to the mode, set, toggle, and display commands).

open host [port]

Opens a connection to the named ***host***. If no ***port*** number is specified, ***telnet*** will attempt to contact a TELNET server at the default port. The ***host*** specification must be an Internet address specified in dotted-decimal notation.

close

Closes a TELNET session and return to command mode.

quit

Closes any open TELNET session and exit ***telnet***. An EOF (in command mode) will also close a session and exit.

status

Shows the current status of ***telnet***. This includes the peer one is connected to, as well as the current mode.

display [argument...]

Displays all, or some, of the set and toggle values (see below).

? [command]

Gets ***help***. With no arguments, ***telnet*** prints a help summary. If a ***command*** is specified, ***telnet*** will print the help information for that command only.

send <arguments>

Sends one or more special character sequences to the remote host. The following are the <***arguments***> which may be specified (more than one argument may be specified at a time):

escape

Sends the current telnet escape character (initially `^]).

synch

Sends the TELNET SYNCH sequence. This sequence causes the remote system to discard all previously typed (but not yet read) input. This sequence is sent as TCP urgent data (and may not work if the remote system is a 4.2 BSD system -- if it does not work, a lower case "r" may be echoed on the terminal).

brk

Sends the TELNET BRK (Break) sequence, which may have significance to the remote system.

ip

Sends the TELNET IP (Interrupt Process) sequence,

ao

Sends the TELNET AO (Abort Output) sequence, which should cause the remote system to flush all output from the remote system to the user's terminal.

ayt

Sends the TELNET AYT (Are You There) sequence, to which the remote system may or may not choose to respond.

ec

Sends the TELNET EC (Erase Character) sequence, which should cause the remote system to erase the last character entered.

el

Sends the TELNET EL (Erase Line) sequence, which should cause the remote system to erase the line currently being entered.

ga

Sends the TELNET GA (Go Ahead) sequence, which likely has no significance to the remote system.

nop

Sends the TELNET NOP (No Operation) sequence.

?

Prints out help information for the send command.

set <argument> <value>

Sets any one of a number of ***telnet*** variables to a specific ***value***. The special value "***off***" turns off the function associated with the variable The values of variables may be interrogated with the display command. The ***argument*** (variables) which may be specified are as follows:

escape

This is the telnet escape character (initially "^[") which causes entry into telnet command mode (when connected to a remote system).

interrupt

If telnet is in localchars mode (see toggle localchars below) and the interrupt character is typed, a TELNET IP sequence (see send ip above) is sent to the remote host. The initial value for the interrupt character is taken to be the terminal's intr character.

quit

If telnet is in localchars mode (see toggle localchars below) and the quit character is typed, a TELNET BRK sequence (see send brk above) is sent to the remote host. The initial value for the quit character is taken to be the terminal's quit character.

flushoutput

If telnet is in localchars mode (see toggle localchars below) and the flushoutput character is typed, a TELNET AO sequence (see send ao above) is sent to the remote host. The initial value for the flush character is taken to be the terminal's flush character.

erase

If telnet is in localchars mode (see toggle localchars below), then when this character is typed, a TELNET EC sequence (see send ec above) is sent to the remote system. The initial value for the erase character is taken to be the terminal's erase character.

kill

If telnet is in localchars mode (see toggle localchars below), then when this character is typed, a TELNET EL sequence (see send el above) is sent to the remote system. The initial value for the kill character is taken to be the terminal's kill character.

toggle <arguments> ...

Toggle (between TRUE and FALSE) various flags that control how telnet responds to events. More than one argument may be specified. The state of these flags may be interrogated with the display command. Valid arguments are as follows:

localchars

If this is TRUE, then the flush, interrupt, quit, erase, and kill characters (see set above) are recognized locally, and transformed into (hopefully) appropriate TELNET control sequences (respectively ao, ip, brk, ec, and el; see send above). The initial value for this toggle is FALSE.

autoflush

If autoflush and localchars are both TRUE, then when the ao, intr, or quit characters are recognized (and transformed into TELNET sequences; see set above for details), telnet refuses to display any data on the user's terminal until the remote system acknowledges (via a TELNET Timing Mark option) that it has processed those TELNET sequences.

autosynch

If autosynch and localchars are both TRUE, then when either the intr or quit characters are typed (see set above for descriptions of the intr and quit characters), the resulting TELNET sequence sent is followed by the TELNET SYNCH sequence. This procedure should cause the remote system to begin throwing away all previously typed input until both of the TELNET sequences have been read and acted upon. The initial value of this toggle is FALSE.

crmod

Toggle RETURN mode. When this mode is enabled, most RETURN characters received from the remote host will be mapped into a RETURN followed by a LINEFEED. This mode does not affect those characters typed by the user, it affects only those received from the remote host. This mode is not very useful unless the remote host only sends RETURN, but never LINEFEED. The initial value for this toggle is FALSE.

options

Toggles the display of some internal telnet protocol processing (having to do with TELNET options). The initial value for this toggle is FALSE.

netdata

Toggles the display of all network data (in hexadecimal format). The initial value for this toggle is FALSE.

?

Displays the legal toggle commands.

NOTES

After exiting telnet, the first character typed is always lost.

Attempting to open a non-existent or non-responsive remote host may cause the shell daemon to become suspended. In this event, the local console will become inoperative until the next system reboot. However, the PI will continue to operate at its current system mode setting. Users should "ping" the desired remote host prior to attempting establishment of a telnet session.

The telnet implementation provided under pSOSystem does not support the "line-by-line" mode.

There is no adequate way for dealing with flow control.

APPENDIX B
FILE TRANSFER PROTOCOL COMMAND

ftp - file transfer program

USAGE:

ftp [host_address]

DESCRIPTION

ARPANET standard File Transfer Protocol (FTP) can browse to transfer files between the PI and a remote network site. This is done with the command ftp [host_address], where host_address refers to the Internet Protocol (IP) address of the remote host in dotted decimal notation. This appendix describes the FTP client services available under pSOSystem, which are a subset of ARPANET FTP.

When the client host with which ***ftp*** is to communicate is specified on the command line, ***ftp*** immediately attempts to establish a connection to an FTP server on that host; otherwise, ***ftp*** enters its command interpreter and awaits instructions from the user. When ***ftp*** is awaiting commands from the user, it displays the prompt "**tp>**".

OPTIONS:

host_address - Internet address of the remote host in dotted-decimal notation.

FTP COMMANDS

! [command]

Runs command as a shell command on the local machine.

account [passwd]

Supplies a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no ***passwd*** argument is included, the user will be prompted for an account password in a non-echoing input mode.

append local-file [remote-file]

Appends a local file to a file on the remote machine. If ***remote-file*** is left unspecified, the ***local-file*** name is used in naming the remote file. File transfer uses the current settings for "representation type," "file structure," and "transfer mode."

ascii

Sets the "representation type" to "network ASCII." This is the default type.

bell

Sounds a bell after each file transfer command is completed.

binary

Sets the "representation type" to "image."

bye

Terminates the FTP session with the remote server and exit ***ftp***. An EOF will also terminate the session and exit.

cd remote-directory

Changes the working directory on the remote machine to ***remote-directory***.

cdup

Changes the remote machine working directory to the parent of the current remote machine working directory.

close

Terminates the FTP session with the remote server, and returns to the command interpreter. Any defined macros are erased.

cr

Toggles RETURN stripping during "network ASCII" type file retrieval. Records are denoted by a RETURN/LINEFEED sequence during "network ASCII" type file transfer. When ***cr*** is on (the default), RETURN characters are stripped from this sequence to conform with the UNIX system single LINEFEED record delimiter. Records on non-UNIX-system remote hosts may contain single LINEFEED characters; when an "network ASCII" type transfer is made, these LINEFEED characters may be distinguished from a record delimiter only when ***cr*** is off.

delete remote-file

Deletes the file ***remote-file*** on the remote machine.

dir [remote-directory] [local-file]

Prints a listing of the directory contents in the directory, ***remote-directory***, and, optionally, places the output in ***local-file***. If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or ***local-file*** is "-", output is sent to the terminal.

disconnect

A synonym for close.

get remote-file [local-file]

Retrieves the file ***remote-file*** and stores it on the local machine. If the local file name, ***local-file***, is not specified, it is given the same name it has on the remote machine, subject to alteration by the current case, ntrans, and nmap settings. The current settings for "representation type," "file structure," and "transfer mode" are used while transferring the file.

glob

Toggles filename expansion, or "globbing," for mdelete, mget, and mput. If globbing is turned off, filenames are taken literally.

Globbing for mput is accomplished as in csh(1). For mdelete and mget, each remote file name is expanded separately on the remote machine, and the lists are not merged.

Expansion of a directory name is likely to be radically different from expansion of the name of an ordinary file: the exact result depends on the remote operating system and FTP server, and can be previewed by typing "**mls remote-files -**".

mget and mput are not meant to transfer entire directory subtrees of files. Entire directory subtrees may be transferred by transferring a tar(1) archive of the subtree (using a "representation type" of "image" as set by the binary command).

hash

Toggles hash sign (#) printing for each data block transferred.

help [command]

Prints an informative message about the meaning of command. If no ***command*** argument is given, ***ftp*** prints a list of the known commands.

lcd [directory]

Changes the working directory to ***directory*** on the local machine. If ***directory*** is not specified, the user's local home directory is used.

ls [remote-directory] [local-file]

Prints an abbreviated listing of the contents of a directory, ***remote-directory***, on the remote machine and, optionally, places the output in ***local-file***. If ***remote-directory*** is left unspecified, the current working directory is used. If no local file is specified, or if ***local-file*** is "-", the output is sent to the terminal.

mdelete [remote-files]

Deletes the ***remote-files*** on the remote machine.

mdir remote-files local-file

Similar to ***dir***, except multiple remote files may be specified. If interactive prompting is on, ***ftp*** will prompt the user to verify that the last argument is indeed the target ***local-file*** for receiving ***mdir*** output.

mget remote-files

Expands the ***remote-files*** on the remote machine and performs a ***get*** for each file name thus produced. See glob for details on the filename expansion. Resulting file names will then be processed according to case, ntrans, and nmap settings. Files are transferred into the local working directory, which can be changed with "**lcd directory**"; new local directories can be created with "**! mkdir directory**".

mkdir directory-name

Makes a directory, ***directory-name***, on the remote machine.

mls remote-files local-file

Similar to **ls**(1V), except multiple remote files may be specified. If interactive prompting is on, ***ftp*** will prompt the user to verify that the last argument is indeed the target local file for receiving ***mls*** output.

mode [mode-name]

Sets the "transfer mode" to ***mode-name***. The only valid ***mode-name*** is stream, which corresponds to the default "stream" mode.

mput directory-name

Expands wild cards in the list of local files given as arguments and performs a ***put*** for each file in the resulting list. See glob for details on filename expansion.

nlist [remote-directory] [local-file]

Prints an abbreviated listing of the contents of a directory on the remote machine. If ***remote-directory*** is left unspecified, the current working directory is used. If no local file is specified, or if ***local-file*** is "-", the output is sent to the terminal.

open host [port]

Establishes a connection to the specified host FTP server. An optional ***port*** number may be supplied, in which case, ***ftp*** will attempt to contact an FTP server at that port. If the ***auto-login*** option is on (default), ***ftp*** will also attempt to automatically log the user in to the FTP server (see below).

prompt

Toggles interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. By default, prompting is turned on. If prompting is turned off, any ***mget*** or ***mput*** will transfer all files, and any ***mdelete*** will delete all files.

put local-file [remote-file]

Stores a local file, ***local-file***, on the remote machine. If ***remote-file*** is left unspecified, the local file name is used in naming the remote file. File transfer uses the current settings for "representation type," "file structure," and "transfer mode."

pwd

Prints the name of the current working directory on the remote machine.

quit

A synonym for bye.

quote arg1 arg2 ...

Sends the arguments specified, verbatim, to the remote FTP server. A single FTP reply code is expected in return.

recv remote-file [local-file]

A synonym for ***get***.

remotehelp [command-name]

Requests ***help*** from the remote FTP server. If a ***command-name*** is specified it is supplied to the server as well.

rename from to

Renames the file ***from*** on the remote machine to have the name ***to***.

reset

Clears the reply queue. This command re-synchronizes command/reply sequencing with the remote FTP server. Resynchronization may be necessary following a violation of the FTP protocol by the remote server.

rmdir directory-name

Deletes a directory, ***directory-name***, on the remote machine.

runique

Toggles storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a get or ***mget*** command, a ".1" is appended to the name. If the resulting name matches another existing file, a ".2" is appended to the original name. If this process continues up to ".99," an error message is printed, and the transfer does not take place. The generated unique filename will be reported. Note: ***runique*** will not affect local files generated from a shell command (see below). The default value is off.

send local-file [remote-file]

A synonym for ***put***.

sendport

Toggles the use of PORT commands. By default, ***ftp*** will attempt to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails, ***ftp*** will use the default data port. When the use of PORT commands is disabled, no attempt will be made to use PORT commands for each data transfer. This is useful when connected to certain FTP implementations that ignore PORT commands but incorrectly indicate that they have been accepted.

status

Shows the current status of ***ftp***.

sunique

Toggles storing of files on remote machine under unique file names. The remote FTP server must support the STOU command for successful completion. The remote server will report the unique name. Default value is off.

tenex

Sets the "representation type" to that needed to talk to TENEX machines.

type [type-name]

Sets the "representation type" to t***ype-name*** The valid ***type-name***(s) are ascii for "network ASCII," binary or image for "image," and tenex for "local byte size" with a byte size of 8 (used to talk to TENEX machines). If no type is specified, the current type is printed. The default type is "network ASCII."

user user-name [password] [account]

Identify yourself to the remote FTP server. If the ***password*** is not specified and the server requires it, ***ftp*** will prompt the user for the ***password*** (after disabling local echo). If an ***account*** field is not specified, and the FTP server requires it, the user will be prompted for the ***account***. If an account field is specified, an account command will be relayed to the remote server after the login sequence is completed (provided that the remote server did not require it for logging in). Unless ***ftp*** is invoked with "auto-login" disabled, this process is performed automatically on initial connection to the FTP server.

verbose

Toggles verbose mode. In ***verbose*** mode, all responses from the FTP server are displayed to the user. In addition, if verbose mode is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose mode is on if ftp's commands are coming from a terminal, and off otherwise.

? [command]

A synonym for ***help***.

Command arguments which have embedded spaces may be quoted with quotation (") marks.

If any command argument which is not indicated as being optional is not specified, ***ftp*** will prompt for that argument.

ABORTING A FILE TRANSFER

The normal abort sequence, CTRL-C will not work during a transfer.

FILE NAMING CONVENTIONS

Local files specified as arguments to ftp commands are processed according to the following rules:

1. If the file name “-” is specified, the standard input (for reading) or standard output (for writing) is used.

2. Failing the checks above, if "globbing" is enabled, local file names are expanded according to the rules used in the csh(1); see the glob command. If the ftp command expects a single local file (for example, put), only the first filename generated by the "globbing" operation is used.

3. For mget commands and get commands with unspecified local file names, the local filename is the remote filename, which may be altered by a case, ntrans, or nmap setting. The resulting filename may then be altered if runique is on.

4. For mput commands and put commands with unspecified remote file names, the remote filename is the local filename, which may be altered by an ntrans or nmap setting. The resulting filename may then be altered by the remote server if sunique is on.

FILE TRANSFER PARAMETERS

The FTP specification designates many parameters which may affect a file transfer.

The "representation type" may be one of "network ASCII," "EBCDIC," "image," or "local byte size" with a specified byte size (for PDP-10s and PDP-20s mostly. The "network ASCII" and "EBCDIC" types have a further subtype which specifies whether vertical format control (NEWLINE characters, form feeds, etc.) are to be passed through ("non-print"), provided in TELNET format ("TELNET format controls"), or provided in ASA (FORTRAN) ("carriage control (ASA)") format. ftp supports the "network ASCII" (subtype "non-print" only) and "image" types, plus "local byte size" with a byte size of 8 for communicating with TENEX machines.

The "file structure" may be one of "file" (no record structure), "record," or "page." ftp supports only the default value, which is "file."

The "transfer mode" may be one of "stream," "block," or "compressed." ftp supports only the default value, which is "stream."

NOTES

Correct execution of many commands depends upon proper behavior by the remote server.

An error in the treatment of carriage returns in the 4.2 BSD code handling transfers with a "representation type" of "network ASCII" has been corrected. This correction may result in incorrect transfers of binary files to and from 4.2 BSD servers using a "representation type" of "network ASCII". Avoid this problem by using the "image" type.

This page intentionally left blank.

.

Ericsson Inc.
Private Radio Systems
Mountain View Road
Lynchburg, Virginia 24502
1-800-528-7711 (Outside USA, 804-528-7711)

Printed in U.S.A