LBI-39040

ERICSSON

Mobile Communications

EDACS™ Jessica
PBX Gateway

PBX Interface User’s Manual

TABLE OF CONTENTS

This manual is published by **Ericsson GE Mobile Communications 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 GE Mobile Communications 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 GE Mobile Communications Inc.**

TABLE OF CONTENTS (Cont.)

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 VME bus backplane. The PI also includes hard and floppy disk storage units. In the VME backplane, slot 2 is left open and slot 3 is used for Primary Rate Interface (PRI)-48/-64. Slot 4 contains the Multisite Interface Controller (MIC).

This manual presents information on PI hardware, initial setup details such as loading application software and configuration data, user interface operation, and file maintenance and utility commands.

PI Controller, PRI-48/-64, and WanServer boards are detailed in separate manuals included with the system shipment.

Additional information for Jessica can be found 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)

2. PI HARDWARE

This section provides a general description of the hardware of the PI.

PI Controller

Using the 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, and four serial port interfaces for ASCII terminals. The PI does not use the IEEE 802..3 local area network interface during normal operation, and only one terminal interface is active.

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 user console. This board must be in slot 1.

P2 Adapter Board

The P2 Adapter board is a small circuit board that routes the PIC input/output (I/O) signals and grounds from its concentrated VME bus backplane connector (P2) to the 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 MVME147 is at the end of the SCSI bus.

712M I/O Transition Module

The 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 data terminal equipment (DTE) or data communications equipment (DCE). The I/O Transition Module has sockets for SCSI terminating resistors.

PI Diagnostic Terminal

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

3. INITIAL SETUP

3.1. POWER CORD INSTALLATION

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

110 Volt AC

The North American T1 PI/MUX 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 E1 PI/MUX 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 CONSOLE

1. Connect the console terminal to the female DB-25 connector labeled "SERIAL PORT 1/CONSOLE" on the MVME-712/M 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 console terminal. If a PC is used as the console terminal, then a DB-9 adapter may be needed. In either case, the serial cable for the console is wired straight-through.

2. Configure the console terminal for the communications parameters below.

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

3. If desired, perform the steps below.

   - Perform "Date" function at PI Console with current date and time.
   - Type "Timesav" at the PI console.

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

   - The red LEDs on the PI are not illuminated.
   - The amber "STATUS" LED on the PIC board is lit dimly or is flashing.
   - The green "RUN" LED on the PMI board is illuminated.
   - The green "RUN" LED on the PRI board is illuminated.
   - The console 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. APPLICATION LOADING

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

Type of Loading	Application	Expected Results
Application loading from floppy drive	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 console 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 console will indicate the copy of the following file to the PI hard drive: JESSICA.SX 6. At the prompt for the next disk, remove Jessica Disk 2 and press "Enter".	1. The Jessica executable files are copied to the hard disk. 2. The PRI-48/-64 ISDN board and the MIC are booted. 3. "Login" is displayed.
Application loading from hard drive	1. Engage Jessica system power.	1. The PRI-48/-64 ISDN board and the MIC are booted. 2. "Login" is displayed.

3.5. 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 telephone-originated 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 the next section for default configuration parameters.

NOTE

Steps 1 and 2 of the Procedure section in the table below should be performed on a PC with a text editor.

Table 1 - Configuration File Loading

Type of Loading	Procedure	Expected Results
Jessica loading of configuration data from the floppy disk drive.	1. Copy the CONFIG.E1 or CONFIG.T1 file on Jessica Disk 3 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 3 to match your application. SITE_ID range 1-32, must match PIM Controller site ID. Default is 16. HANG_TIME range 0-255 seconds should be set to a value equal to the System Manager Interconnect Hang Time and the GETC Special Call Hang Time. IP.DAT defines the IP address, subnet mask, and host name of the JESSICA_PI.	

Table 1 - Configuration File Loading (Cont.)

Type of Loading	Procedure	Expected Results
	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_CHANNEL_MASK range 0X0 to 0XFFFFFFFF hex with the MSB corresponds to channel 32 and the LSB corresponding to channel 1. This must match the channels equipped in the MUX and connected to PIM audio channels in the CEC/IMC. Leading zeroes can be left out. Example: 0XF enables MUX channels 1-4 for interconnect. T1 supports up to 23 channels, and E1 supports up to 30 channels. TRUNK_TYPE must be set to T1 or E1	
	3. Insert Jessica Disk 3 into the PI floppy drive. 4. Apply power to the PI or execute "reboot-h" if logged into the PI. 5. The PI console will indicate the copy of the following file to the PI hard drive: CONFIG.DAT 6. At the prompt for the next disk, remove Jessica Disk 3 and press "Enter". 7. Log in to the system. 8. After the shell prompt is displayed, type "config" to display the configuration parameters. 9. Verify that the parameters specified on the floppy disk are displayed with the "config" command.	1. The configuration file is copied from the floppy drive during the boot phase. 2. The following Jessica configuration parameters are displayed when the "config" command is executed: MUX_CHANNELS_MASK SITE_ID CONVERSATION_LIMIT HANG_TIME
Jessica loading of configuration data from the hard disk drive.	1. Engage Jessica system power or execute "reboot-h" if logged into the PI. 2. Log in to the system. 3. After the shell prompt is displayed, type "config" to display the configuration parameters. 4. Verify that the parameters specified on the hard disk are displayed with the "config" command.	1. The configuration file is read from the hard drive during the boot phase. 2. The following Jessica configuration parameters are displayed when the "config" command is executed: MUX_CHANNELS_MASK SITE_ID CONVERSATION_LIMIT HANG_TIME

3.6. PASSWORD PROTECTION -- LOCAL LOGIN

The console 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 console displays the following messages:

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

      pSH+>_
```

4. At this point, some or all of the PI utilities may be run, depending on the user's level of access (the account name which you entered at the "Login" prompt). All of these utilities are described in section 6.

3.7. CONFIGURATION DATA

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

All variable names and enumerated values are case-sensitive. Comment lines must begin with "#". The user sets the values in the first table below. Selecting E1 or T1 automatically sets the appropriate values for the entries in the second table.

Table 2 - 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	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

Table 2 - PI Configuration Parameters (Cont.)

Variable Name	Use	Allowed Values
TRUNK_CLOCK_SOURCE	Clock source for trunk framing.	RECOVER_MD110_TRUNK, RECOVER_MUX_TRUNK, INTERNAL_CLK, EXTERNAL_CLK T1, E1 default = RECOVER_MD110_TRUNK
MUX_TRUNK_FRAMING	Framing type selection for multiplexor trunk.	ESF, ESF_ZBTSI, SF, SF_SLC96 T1 default = ESF. E1 default = CRC4
MUX_TRUNK_CODING	Coding type selection for multiplexor trunk.	B8ZS, B7ZS, AMI, HDB3 T1 default = B8ZS. E1 default = HDB3
MD110_TRUNK_FRAMING	Framing type selection for MD110 trunk.	ESF, ESF_ZBTSI, SF, SF_SLC96 T1 default = ESF. E1 default = CRC4
MD110_TRUNK_CODING	Coding type selection for MD110 trunk.	B8ZS, B7ZS, AMI, HDB3 T1 default = B8ZS. E1 default = HDB3
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 T1 default = ATT5ESS_USER E1 default = MD110_E1_USER

Trunk parameter changes require a PI reboot.

These variables normally are not used in the CONFIG.DAT file.

4. PI USER INTERFACE OPERATION

The PI user interface is the diagnostic terminal that connects to the PIC. The following examples present several of the most commonly used commands. Explanations are offset with {}. The prompt is pSH+>.

{To get a list of all available commands}

```
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
```

{To get a list of all files and directories in current directory}

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

{To show current working directory}

```
pSH+> pwd
01.02/activity
```

{To list files and directories in current directory in long format with file size}

```
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
```

{To show complete log for July 24}

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

{To return to root directory}

```
pSH+> cd /
```

{To show current software versions for Jessica}

```
pSH+> ver

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

{To retrieve help about passwd command}

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

{To retrieve help about config command}

```
pSH+> help config
config - config [-fu] disp: f=full disp, u=update (not reentrant, not locked)
```

{To reload configuration from config.dat file on hard drive}

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

{To show 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
```

{To show current status}

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

{To disable new calls and allow current calls to complete}

```
pSH+> shutdn
Call Controls    = [CALLS_DISABLED]  >> [0] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
MSC Comm Status  = [COMM_ENABLED     ] [Link Up]
[4] Total Channels
[0] Active Channels
pSH+> Activity Recording Shutdown on DATE: August 4, 1993 TIME: 7:45:44 pm
```

{To enable new calls}

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

{To retrieve help about shutdn command}

```
pSH+> help shutdn
shutdn - shutdown [-i] system: i=immediate (not reentrant, not locked)
```

{To disable and immediately terminate all calls}

```
pSH+> shutdn -i
Call Controls    = [CALLS_TERMINATED]  >> [0] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
MSC 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
```

{To enable new calls}

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

{To show current date and time}

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

{To retrieve help about timesav command}

```
pSH+> help timesav
timesav - save system time in battery backed RTC (not reentrant, not locked)
```

{To retrieve help about lp command}

```
pSH+> help lp
lp     - print a file: lp [file_name] (not reentrant, not locked)
```

{To retrieve help about reboot command}

```
pSH+> help reboot
reboot - reboot system: reboot[-sh] (not reentrant, not locked)
```

{To perform a complete system reset}

```
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...
```

{To show current status}

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

5. ACTIVITY LOGGING

Activity logging is useful for billing purposes or tracking the general activity of radios. Activity is accessed through the daily activity file (shown as month, day, and ending with log, e.g., jun14.log). To view activity files, use the head and tail commands described in section 6.

This section presents information on non-verbose and verbose messages and disconnect reason codes.

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. Check DISK_LOG_STATES and PORT_LOG_STATES. If DISK_LOG_STATES is set to false, then non-verbose messages result. If DISK_LOG_STATES is set to true, then verbose messages are written to the log file. If PORT_LOG_STATES is set to true, then verbose call states are sent to the user console on port 2. The terminal must be connected to port 2 to view PORT_LOG_STATES. If the terminal is not connected to port 2, then the information will not appear on the terminal screen, but it will be written to disk. If PORT_LOG_STATES is set to false, non-verbose messages will result.

The following list contains fields that appear in non-verbose and verbose messages and explanations of what the fields represent.

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 ID. 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]:MSC, or another disconnect reason code
   Reason that the call terminated. [NORMAL_DISCONNECT] and [FORCED_DISCONNECT] are types of disconnect reason codes. These codes are discussed in the next section of the document. PBX and MSC designate whether the phone user or radio user initiated the disconnect.

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

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

8. PHONE[0000]
   Phone number dialed by the radio.

9. SITES: O[01] D[00]
   O is the originating site (where the call was placed from) and D is the disconnect site. In the originating site codes, 00 designates the originating site, i.e., inbound call, and in the disconnect site codes, 00 designates that the PI terminated the call; all other codes are EDACS site IDs.

10. DUR: 000.00
    Duration of the call.

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

12. ST: 00:00:00
    Time that the call was originally placed.

13. CON: 00:00:00
    Time that the call was actually connected (radio to phone or phone to radio).

14. DIS: 00:00:00
    Time that the call was terminated.

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

{**Non-Verbose Messages** - shows normal call activity; each call consists of three lines}

```
CALL:[00003] OUTBOUND DIGITAL LID[06533] [NORMAL_DISCONNECT]:MSC
MSC[01] PBX[01] PHONE[2001]   SITES: O[01] D[01]
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]:MSC
MSC[01] PBX[01] PHONE[2001]   SITES: O[01] D[01]
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]:MSC
MSC[01] PBX[01] PHONE[2001]   SITES: O[01] D[01]
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
MSC[01] PBX[01] PHONE[2001]   SITES: O[01] D[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
MSC[01] PBX[01] PHONE[2002]   SITES: O[01] D[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]:MSC
MSC[01] PBX[04] PHONE[]   SITES: O[00] D[01]
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]:MSC
MSC[01] PBX[01] PHONE[]   SITES: O[00] D[01]
DUR: 019.02 DATE: 08/20/93 ST: 09:29:13 CON: 00:00:00 DIS: 09:29:32
```

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/Radio Terminated
*******************************************************************************

CALL:[00084] OUTBOUND CLEAR   LID[02405] [NORMAL_DISCONNECT]:PBX
MSC[01] PBX[01] PHONE[92372004]   SITES: O[01]  D[00]
DUR: 027.72 DATE: 08/06/93 ST: 07:54:00 CON: 07:54:03 DIS: 07:54:28
Time->[223196]   Event->[CONSTRUCTION  ] From->[MSC] State->[OUTBOUND_ACTIVE  ]
Time->[223196]   Event->[CHANNEL_REQ   ] From->[MSC] State->[OUTBOUND_ACTIVE  ]
Time->[223201]   Event->[OUTBOUND_SETUP] From->[MSC] 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->[MSC] 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_MSC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISCONNECT_ACK] From->[MSC] State->[DISCONNECTING    ]
Time->[223311]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[12] Events

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

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

CALL:[00134]  INBOUND CLEAR   LID[03945] [NORMAL_DISCONNECT]:PBX
MSC[01] PBX[01] PHONE[]   SITES: O[00]  D[00]
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->[MSC] State->[SETUP_IN_PROGRESS]
Time->[243522]   Event->[ALERTING      ] From->[MSC] 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_MSC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISCONNECT_ACK] From->[MSC] State->[DISCONNECTING    ]
Time->[243530]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[11] Events

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

CALL:[00113]  INBOUND CLEAR   LID[02405] [NORMAL_DISCONNECT]:MSC
MSC[01] PBX[04] PHONE[]   SITES: O[00]  D[01]
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->[MSC] State->[SETUP_IN_PROGRESS]
Time->[229473]   Event->[ALERTING      ] From->[MSC] State->[CALL_PROGRESSING ]
Time->[229493]   Event->[CONNECT       ] From->[MSC] State->[CALL_ALERTING    ]
Time->[229493]   Event->[CONNECT_ACK   ] From->[PBX] State->[CALL_CONNECTING  ]
Time->[229557]   Event->[DISCONNECT    ] From->[MSC] State->[CALL_CONNECTED   ]
Time->[229557]   Event->[DISC_TO_ISDN  ] From->[PI ] State->[DISCONNECTING    ]
Time->[229557]   Event->[DISC_TO_MSC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[229557]   Event->[DISCONNECT_ACK] From->[MSC] State->[DISCONNECTING    ]
Time->[229558]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[229558]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[13] Events
```

```
*******************************************************************************
                        Group Inbound Call/Radio Terminated
*******************************************************************************

CALL:[00089]  INBOUND CLEAR   GID[00273] [NORMAL_DISCONNECT]:PBX
MSC[01] PBX[01] PHONE[]   SITES: O[00]  D[00]
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->[MSC] State->[SETUP_IN_PROGRESS]
Time->[225553]   Event->[ALERTING      ] From->[MSC] State->[CALL_PROGRESSING ]
Time->[225581]   Event->[CONNECT       ] From->[MSC] 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_MSC   ] From->[PI ] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISCONNECT_ACK] From->[PBX] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISCONNECT_ACK] From->[MSC] State->[DISCONNECTING    ]
Time->[225638]   Event->[DISCONNECTED  ] From->[PI ] State->[DISCONNECTED     ]
[13] Events

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

                         CALL STATES DISABLED

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

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

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

*******************************************************************************
                          Outbound Call/Phone Terminated
*******************************************************************************

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

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

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

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

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

*******************************************************************************
                        Group Inbound Call/Radio Terminated
*******************************************************************************

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

5.2. 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 multiplexer 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 or when the radio ID is not in the range 1-16382 or the group ID is not in the range 0-2047.

8. FORCED_DISCONNECT
   Occurs when active calls are terminated at the user console. Disabling calls at the user console 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 MSC during inbound call setup.

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

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

11. HANG_TIME_EXPIRED
    Occurs when the radio 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 data base.

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

6. FILE MAINTENANCE AND UTILITY COMMANDS

6.1. JESSICA-SPECIFIC PSOSYSTEM COMMANDS

This section of the document provides detailed information on Jessica-specific commands, their syntax, and examples.

6.1.1. 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 pSOSystem *clear* command. User preference between ***clr*** and ***clear*** depends on the terminal characteristics and network connection type.

OPTIONS:

None.

6.1.2. config

config -- displays the current configuration

USAGE:

config [-fu]

DESCRIPTION:

config displays the current configurations of Jessica.

OPTIONS:

- -f Full configuration list
- -u Update - Reread Config.dat

EXAMPLE:

```
pSH+> config
MUX_CHANNELS_MASK        0X3FF

SITE_ID                  13
CONVERSATION_LIMIT       10
HANG_TIME                30

pSH+> config -f
MUX_CHANNELS_MASK        0X3FF

SITE_ID                  13
CONVERSATION_LIMIT       10
HANG_TIME                30
```

```
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

ALL_DEBUG_ENABLED        FALSE
DISK_LOG_CALLS           TRUE
PORT_LOG_CALLS           TRUE
DISK_LOG_STATES          TRUE
--More--
PORT_LOG_STATES          TRUE
DISK_LOG_ISDN            FALSE
PORT_LOG_ISDN            FALSE
LOG_BASE_PATH            01.02/activity
ACTIV_QUEUE_TIMEOUT      0

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

6.1.3. 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 PI printer.

OPTIONS:

None.

6.1.4. 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 character (i.e., Enter or Return). To terminate the command, 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
S31400158FCD746865207379737374656D000014C9880F
S31400158FDC00158FBE00100A5A0000000000000095
S31200158FEB000000000000000000000000005E
S30C0014C9887265626F6F740003
S314001000004E56FFF8487800004EBB817000027212
S3140010000F18588F2D40FFFC4EBB8170000272688F
S3140010001E4EBB8170000272AE1D40FFFB4A2EFFD3
S3140010002DFB6712202EFFFC068000000210206ECB
S3140010003CFFFC21400200206EFFFC7032214000B5
--More--q
pSH+>
```

6.1.5. passwd

passwd -- performs password management.

USAGE:

passwd [login_name]

DESCRIPTION:

passwd changes (or installs) a password 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.

6.1.6. pmi

pmi -- opens/closes the pmi statistics file

USAGE:

pmi [-oc]

DESCRIPTION:

pmi is used to create the *pmistats.dat* file, which is a binary file.

OPTIONS:

- -o Opens the binary file
- -c Closes the binary file

EXAMPLE:

```
pSH+> pmi -o
pSH+> pmi -c
```

NOTES:

Do not display contents of console.

6.1.7. 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 be issued prior to reboot to assure that all user data flushes 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+>
```

6.1.8. rest

rest -- restarts the Jessica application

USAGE:

rest

DESCRIPTION:

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

OPTIONS:

None.

EXAMPLE:

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

6.1.9. 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 display on the terminal.

OPTIONS:

None.

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)
```

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.

6.1.10. 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+> help shut
shutdn - shutdown [-i] system: i=immediate (not reentrant, not locked)

pSH+> shutdn
Call Controls    = [CALLS_DISABLED]  >> [0] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
MSC Comm Status  = [COMM_ENABLED     ] [Link Up]
[4] Total Channels
[0] Active Channels
pSH+> Activity Recording Shutdown on DATE: August 4, 1993 TIME: 7:45:44 pm
```

```
pSH+> shutdn -i
Call Controls    = [CALLS_TERMINATED]  >> [0] Calls Active
ISDN Comm Status = [COMM_ENABLED     ] [No Trunk Alarms]
MSC 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
```

6.1.11. 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]
MSC Comm Status  = [COMM_ENABLED     ] [Link Up]
[10] Total Channels
[1] Active Channels
```

6.1.12. 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.
```

6.1.13. 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]
```

6.2. PSOSYSTEM COMMANDS

6.2.1. cat

cat -- concatenate and display

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 the 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*** can not be aborted. It may, however, be suspended and resumed using flow control characters ^S and ^Q.

6.2.2. cd

cd -- changes the working directory

USAGE:

cd [directory]

DESCRIPTION:

directory becomes the new working directory.

6.2.3. clear

clear -- clears the terminal screen

USAGE:

clear

DESCRIPTION:

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

See also PI-specific command ***clr***.

OPTIONS:

None.

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 subsequent 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.

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.

Note: This implementation of Jessica does not have linked files.

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*** copies the ***directory1*** directory within ***directory2*** (as a subdirectory), along with its files and subdirectories.

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.

6.2.6. 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 second. The year may be omitted; the current value is supplied as default.

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

NOTE:

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

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.

6.2.8. exit

exit -- exits the shell

USAGE:

exit

DESCRIPTION:

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

OPTIONS:

None.

6.2.9. getid

getid -- obtains user ID (uid) and group ID (gid)

USAGE:

getid

DESCRIPTION:

getid displays the user and group ID 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
```

6.2.10. 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.

EXAMPLE:

The following example provides the current priority of the PI's telnet server daemon.

```
pSH+> getpri tnpd
tnpd task priority = 50
```

6.2.11. 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...
```

6.2.12. help

help -- provides help about shell commands

USAGE:

help [command_name]

DESCRIPTION:

help prints to the console 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.

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)
```

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 available only 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.

6.2.13. 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 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.

6.2.14. 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 of *mv* 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.

6.2.15. 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.

6.2.16. pcmkfs

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

USAGE:

pcmkfs [-i] volume_name format

DESCRIPTION:

pcmkfs performs initialization (i.e., format) of 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 Call device driver initialization procedure.

NOTES:

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

EXAMPLE:

The following example correctly formats a diskette installed in the PI 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
```

6.2.17. pcmount

pcmount - mounts an MS-DOS file system.

USAGE:

pcmount volume_name [sync_mode]

DESCRIPTION:

pcmount mounts 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
```

6.2.18. 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/
```

6.2.19. 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
```

6.2.20. pwd

pwd -- displays the pathname of the current working volume

USAGE:

pwd

DESCRIPTION:

pwd prints the pathname of the working (current) directory.

OPTIONS:

None.

EXAMPLE:

```
pSH+> cd /log
pSH+> pwd
1.2/log
```

6.2.21. rm

rm -- removes (unlinks) 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 .*".

6.2.22. rmdir

rmdir -- removes (unlinks) directories

USAGE:

rmdir directory ...

DESCRIPTION:

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

OPTIONS:

None.

6.2.23. 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.

6.2.24. 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 beginning 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.
- -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.

6.2.25. 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.

6.2.26. 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 flush 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+> mount 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
```

6.3. ADVANCED PSOSYSTEM COMMANDS

NOTE

The following section discusses pSOSystem commands which generally 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 by Jessica-specific commands, 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 that the command provides 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).***

6.3.1. console

console -- redirects the console output to a telnet session

USAGE:

console [-r] [task_name]

DESCRIPTION:

console will redirect output going to the PI's system console 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 console input when this command is issued, the task's input redirection will take effect only after it returns from the waiting.

EXAMPLE:

Telnet into the PI and enter: pSH+> console -r DBGT

All output from the task “DBGT” will be redirected to the telnet session, and the task will receive its input from the telnet session.

NOTES:

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

6.3.2. 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.

6.3.3. 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.

EXAMPLE:

The following example kills the ftp server daemon on the PI.

```
pSH+> kill ftpd
```

6.3.4. 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.

6.3.5. 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.

6.3.6. 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 190.1.2.3
PING (190.1.2.3): 56 data bytes
190.1.2.3 is alive.
```

6.3.7. 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.

EXAMPLE:

The following example would resume the PI telnet server daemon, had it been previously suspended.

```
pSH+> resume tnpd
```

6.3.8. setenv

setenv -- sets environment variables

USAGE:

setenv variable_name value

DESCRIPTION:

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

OPTIONS:

None.

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
```

NOTES:

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

6.3.9. setid

setid -- set 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
```

6.3.10. 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.

EXAMPLE:

The following example adjusts the priority of the PI's telnet server daemon.

```
pSH+> getpri tnpd
tnpd task priority = 50
pSH+> setpri tnpd 100
pSH+> getpri tnpd
tnpd task priority = 100
```

6.3.11. sleep

sleep -- suspends execution for the specified interval

USAGE:

sleep time

DESCRIPTION:

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

OPTIONS:

None.

6.3.12. 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.

EXAMPLE:

The following example would suspend the PI telnet server daemon.

```
pSH+> suspend tnpd
```

Ericsson GE Mobile Communications Inc.
Mountain View Road • Lynchburg Virginia 24502

Printed in U.S.A.

APPENDIX A
TELNET FEATURES

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]

Open 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

Close a TELNET session and return to command mode.

quit

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

status

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

display [argument...]

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

? [command]

Get ***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 FEATURES

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.