

Distributed Systems Network



DSN/DS HP 3000 to HP 1000
Reference Manual for HP 3000 Users



HP Distributed Systems Network

DSN/DS HP 3000 TO HP 1000

Reference Manual for HP 3000 Users



**HEWLETT
PACKARD**

**INFORMATION NETWORKS DIVISION
19420 HOMESTEAD ROAD, CUPERTINO, CALIFORNIA 95014**

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page. Within the manual, changes are marked with a vertical bar in the margin.

First Edition January 1978
Second Edition January 1982

PRINTING HISTORY

New editions incorporate all update material since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover changes only when a new edition is published. If minor corrections and updates are incorporated, the manual is reprinted but neither the date on the title page and back cover nor the edition changes.

First Edition January 1978
Second Edition January 1982

This manual explains how an HP3000 user can communicate with HP 1000 computer systems using the Hewlett-Packard Distributed Systems Network, DSN/DS.

Before using DS/3000 you should be familiar with the basic operating principles of the HP 3000 Series II/III/30/33/40/44 and 64 computer systems as described in the following manuals:

- HP 3000 Computer System, Commands Reference Manual, HP Part Number, 30000-90009
- DS/3000 Reference Manual, HP Part Number 32190-90001
- Communications Handbook, HP Part Number 30000-90105
- HP 3000 Computer System, Intrinsic Reference Manual, HP Part Number 30000-90010
- HP 3000 Computer System, System Manager/System Supervisor Reference Manual, HP Part Number 30000-90014
- HP 3000 Computer System, Console Operator's Guide HP Part Number 32002-90004

The HP 1000 user should refer to the following manuals for how to use DS/1000-IV.

- DS/1000-IV User's Manual, HP Part Number 91750-90002.
- DS/1000-IV Network Manager's Manual, HP Part Numbers 91750-90010, and 91750-90011.
- Getting Started with DS/1000-IV, HP Part Number 91750-90004 91750-90004
- DS/1000-IV Quick Reference Guide, HP Part Number 91750-90005
- Communications Handbook, HP Part Number 30000-90105

CONVENTIONS USED IN THIS MANUAL

NOTATION DESCRIPTION:

[] An element inside brackets is optional. Several elements stacked inside a pair of brackets means the user may select any one or none of these elements.

Example: $\begin{bmatrix} A \\ B \end{bmatrix}$ User may select A or B or neither.
 $[X | Y | Z]$ User may select X, Y, or none.

{ } When several elements are stacked within braces the user must select one of these elements.

Example $\begin{Bmatrix} A \\ B \\ C \end{Bmatrix}$ User must select A or B or C.
 $\{ X | Y | Z \}$ User must select X or Y or Z.

italics Lowercase *italics* denote a parameter which must be replaced with a user-supplied variable.

Example: CALL *name*
name is one to 15 alphanumeric characters.

underline Dialogue: Where it is necessary to distinguish user input from computer output, the user input is underlined.

Example: NEW NAME? ALPHA

superscript C Control characters are indicated by a superscript C.

Example: C
Y

return return in italics indicates a carriage return.

linefeed linefeed in italics indicates a linefeed.

... A horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.

CONTENTS

Section 1	Introduction	1-1
Section 2	Establishing the Communications Link with HP 1000 Systems	2-1
	Opening a Line	2-1
	Line Opening Failures	2-3
Section 3	Executing Remote Operator Commands	3-1
	The REMOTE Command	3-1
	Executable RTE Operator Commands	3-4
	Limitations	3-5
Section 4	Executing Remote EXEC Calls	4-1
	DEXEC Call Elements	4-3
	Functional Return	4-4
	The READ Call	4-5
	The WRITE Call	4-8
	The I/O Control Call	4-11
	The Program Schedule Call	4-14
	The Time Request Call	4-16
	The Execution Time Call (Initial Offset)	4-17
	The I/O Status Call	4-19
Section 5	Remote File Access	5-1
	Programmatic Access	5-1
	The DAPOS Intrinsic	5-4
	The DLOS Intrinsic	5-6
	The DCONT Intrinsic	5-8
	The DCRET Intrinsic	5-11
	The DLOCF Intrinsic	5-13
	The DNAME Intrinsic	5-15
	The DOPEN Intrinsic	5-16
	The DPOSN Intrinsic	5-19
	The DPURG Intrinsic	5-20
	The DREAD Intrinsic	5-21
	The DTSAT Intrinsic	5-22
	The DWIND Intrinsic	5-23
	The DWRIT Intrinsic	5-24
	Example	5-25
Section 6	Program-to-Program Communications	6-1
	The POPEN Intrinsic	6-5
	The PREAD Intrinsic	6-7
	The PWRITE Intrinsic	6-9
	The PCONTROL Intrinsic	6-11
	The PCLOSE Intrinsic	6-13
	The GET Intrinsic	6-14
	The ACCEPT Intrinsic	6-16
	The REJECT Intrinsic	6-18
	The PCHECK Intrinsic	6-19
	Examples	6-20

CONTENTS (Continued)

Appendix A Error Codes and Messages	A-1
Communication Link Errors	A-1
DSLINe Syntax Errors	A-2
DSN/DS Functional Errors	A-3
:DSCONTROL Informatory messages	A-5
:DSCONTROL Error messages	A-5
Remote FMP Access Errors	A-7
DEXEC Errors	A-10
Appendix B DSCONTROL Console Command	B-1
Appendix C System Verification Test	C-1
Appendix D HP Character Set	D-1
Appendix E Applications	E-1
Line Buffer Configuration Limits	E-1
Opening a line from the 3000	E-2
Opening a Line From the 1000	E-2
Session	E-2
Running FMGR on the HP 1000	E-3
The "MO"ve Command	E-4
PTOP Testing using DS	E-4

ILLUSTRATIONS

Figure	Title	Page
1-1	Overview of HP 3000 to HP 1000 Communications	1-4
1-2	Multiple User Example	1-5
1-3	Multiple User, Multiple Line Example	1-6
1-4	DS Line Buffer Example	1-7
3-1	The REMOTE Command	3-2
6-1	POPEN Activity	6-6
6-2	PREAD Activity	6-8
6-3	PWRITE Activity	6-10
6-4	PCONTROL Activity	6-12
6-5	Pack Example	6-28

TABLES

Table	Title	Page
4-1	DEXEC Calls	4-2
5-1	RFA and FMP Cross Referencing	5-3
6-1	Master Program-to-Program Intrinsic	6-2
6-2	Slave Program-to-Program Intrinsic	6-3



The Hewlett-Packard Distributed Systems Network is a combination of hardware and software products that makes it possible for Hewlett-Packard computer systems to communicate with one another.

This manual describes how an HP 3000 user can communicate with HP 1000 computers by establishing a communications link using DS/3000. Since each of these computers perform functionally different data processing tasks, their association in a network provides you with the benefits of their combined capabilities. You can use the processing capability of both machines and pass data back and forth between them.

The DSLINE command is used to establish the communications link for a user and is described in Section 2 of this manual.

Once the link is established, you can:

- Use the Remote Operator (RTE) Commands to the HP 1000 System. This is described in Section 3.
- Use the Remote EXEC Calls to the HP 1000 System. This is described in Section 4.
- Perform Remote File Access to the HP 1000 System, using RFA intrinsics. This is described in Section 5.
- Perform Program-to-Program Communication between the HP 3000 and HP 1000 system, using Program-to-Program intrinsics. This is described in Section 6.

Figure 1-1 is an overview of the types of HP 3000 to HP 1000 communication available to a DS/3000 user. Using DS/3000 you can communicate with HP 1000 systems in four ways:

Introduction

	HP 3000	HP 1000
REMOTE OPERATOR COMMANDS	From a terminal issue a REMOTE command	Execute an Operator Command
REMOTE EXEC CALLS	Write and execute an SPL or FORTRAN program using remote EXEC calls	Execute an EXEC call
REMOTE FILE ACCESS	Write and execute an SPL or FORTRAN program using Remote File Access (RFA) intrinsics	Access a disc file
PROGRAM TO PROGRAM COMMUNICATION	Write and execute a master or slave program in any language using Program-to-Program intrinsics	Write and execute a master or slave Program in FORTRAN, PASCAL, or assembly language using Program-to-Program intrinsics

In Program-to-Program communication the programs can reside on either system:

Master Program >-----> Slave Program

or

Slave Program <-----< Master Program

HP 3000

HP 1000

The communications link between the HP 3000 and HP 1000 is established over a hardwired communications line, or via a pair of synchronous modems. The HP 1000 is established and consists of coaxial cables connected directly to the communications I/O interfaces of the two computers.

The communications link between the HP 3000 and HP 1000 is established over a hardwired communications line, or via a pair of synchronous modems. For a hardwired connection, there are two options available. The Hardwired Serial Interface (HSI), or the Intelligent Network Processor (INP), to Programmable Serial Interface (PSI).

The Hardwired Serial Interface (HSI) connection consists of the HP 30360A HSI interface card for the HP 3000, and the 12889A HSI card for the HP 1000. Each HSI card occupies one card slot in the Multiplexor channel of the HP 3000 and can accommodate up to four connections to HP 1000's. Only one of the four lines can be used at any given time, and the HP 3000 console operator has control over which line is active. The HP 3000 Series II and III can accommodate a maximum of 2 HSI cards. The HSI is not supported by the HP 3000 Series 30/33/40/44 and 64 Computer Systems.

The INP to PSI connection utilizes microprocessor-based interface cards in both the HP 3000 and HP 1000, and offloads the task of DS transmission management from the CPU. The connection is established by using the HP 30010A INP in the HP 3000 Series II and III, or by using the 30020B in the HP 3000 Series 30/33/40/44 and 64 Computer Systems, and the 12834A PSI card in the HP 1000. The maximum distance allowed when using this connection is one kilometer.

For most applications, the INP to PSI connection offers a much better method of HP 3000 to HP 1000 communication. The HSI cards impose substantial overhead on the CPU's (both 3000 and 1000) due to the combination of a fast transfer rate (2.5 megabits/sec) and a small interface buffer size (1 word). The INP to PSI connection operates at up to 56.8 kilobits/sec, but offers 1000 bytes of buffering on the INP, and 1400 bytes of buffering on the PSI. The microprocessors on each card handle the communications protocol and handshaking, thus freeing the CPU for more important tasks.

The modem connection between the HP 3000 and the HP 1000 is accomplished through the use of the 3055A SSLC I/O card in the HP 3000 Series 30/33/40/44 and 64, and the 12793A BISYNC MODEM PSI card in the HP 1000. Refer to figure 1-2 for a summary of the hardware requirements for an HP 3000 to HP 1000 connection.

If the HP 3000 system is able to obtain access to the specified line, it responds to each DSLINE command by displaying a DS line number at your log-on terminal. This line number is roughly analogous to the file number returned by the MPE File System FOPEN intrinsic in that it is an arbitrary number that uniquely identifies, within your local session, your current access to a particular communications line. DS line numbers are meaningful only if you have more than one line open concurrently within a single local session. In that case you are assigned a separate DS line number for each line you have opened and you subsequently use these line numbers to specify which line you wish to use.

In Figure 1-3, user Y has established two communications links concurrently from within a single local session. He is using DS line #1 to communicate with system B, and DS line #2 to communicate with system C.

At the same time, user X is sharing DS line #1 to communicate with system B.

When you transmit data over the communications line, keep in mind that the line buffer size is fixed at various sizes, as shown in Figure 1-4.

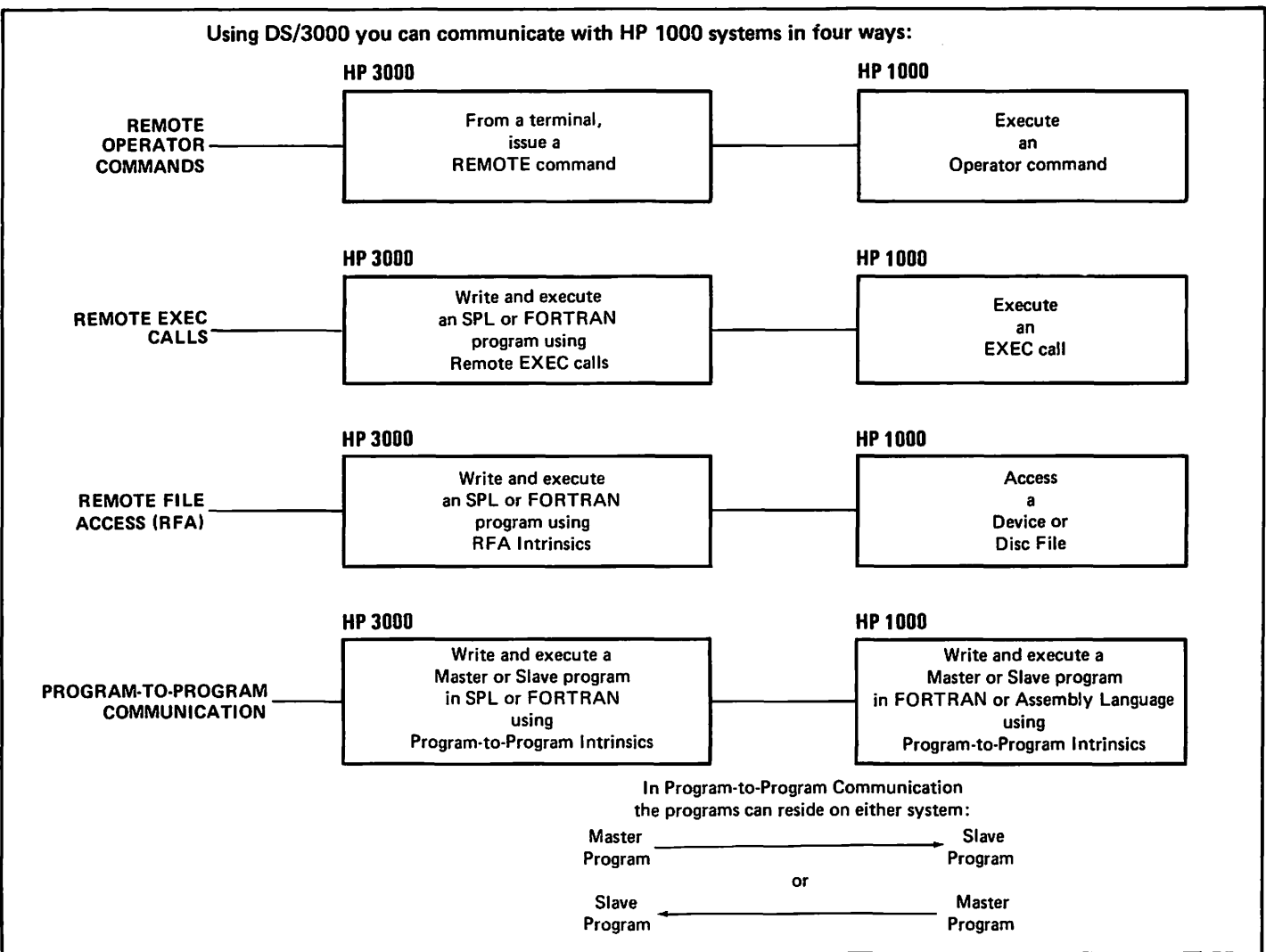
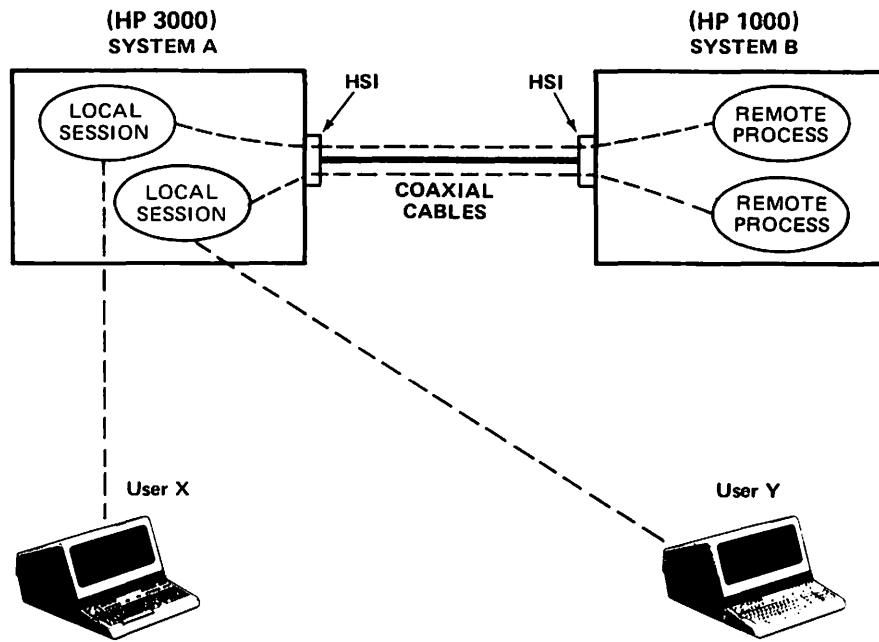


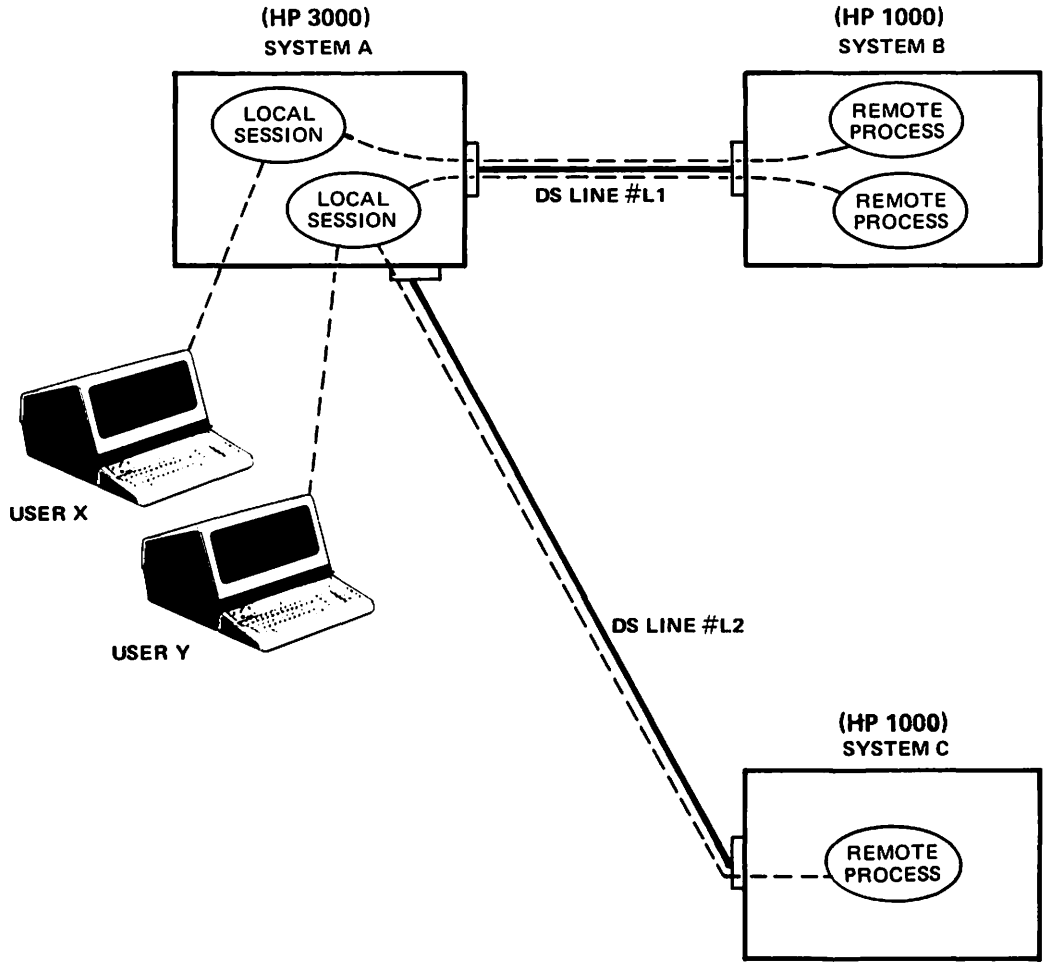
Figure 1-1. Overview of HP 3000 to HP 1000 Communication



In this example, User X initiates a local session in System A, obtains access to the hardwired communications line that connects System A to System B, and initiates a remote session in System B. User Y subsequently initiates a local session in System A, obtains access to the same communications line, and initiates a remote session in System B. The request by User Y for the particular communications line is granted by DS/3000 because neither user asked for exclusive access to the line.

Figure 1-2. Multiple User Example.

Introduction



In this example, User X initiates a local session in System A and remote processing in System B through DS line #L1. At the same time, User Y initiates a local session in System A, and remote processing in both System B and System C. User Y communicates with System B through DS Line #L1 and with System C through DS Line #L2.

Figure 1-3. Multiple User, Multiple Line Example.

Assume that you have initiated transmission of a block of data 500 words in length, from the HP 3000 to a remote HP 1000. If the buffer size is fixed at 256 words, a block of data is actually sent in two separate physical transmissions as follows:

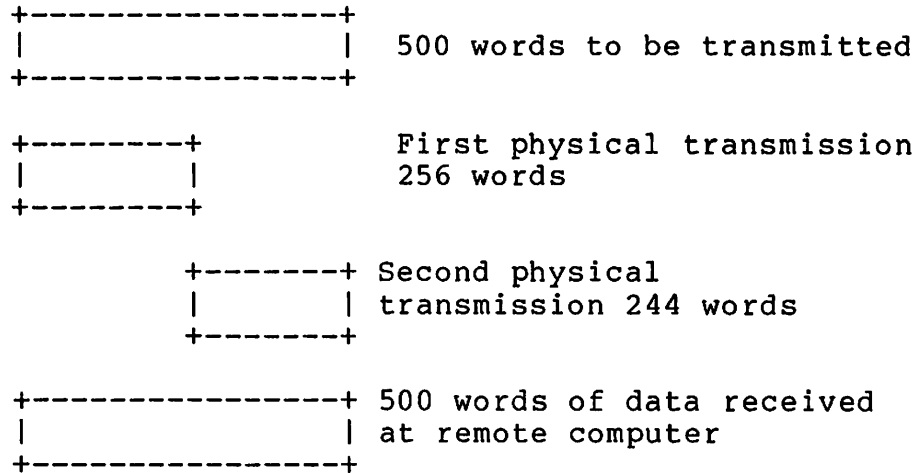


Figure 1-4. DS Line Buffer Example

Before you can send data from one system to another, you need to establish the communications link, as described in Section 2.

Once the link is established, you can select one, or a combination of communication capabilities, as described in Sections 3 through 6.



ESTABLISHING THE COMMUNICATIONS LINK WITH HP 1000 SYSTEMS

SECTION

II

As an HP 3000 user you need only issue a DSLINE user command from your terminal to establish a link with an HP 1000 system.

However, before you can issue this command successfully, the hardware connections must be made, and the DS/3000 and DS/1000-IV software enabled. To do this:

- The HP customer engineer installs the hardware connection between the systems.
- The HP 1000 network manager generates a DS/1000-IV network and runs the program DINIT and DSLIN. Instructions are provided in the Network Manager's Manual, Part Numbers 91750-90010, and 91750-90011.
- The HP 3000 system supervisor configures the HP 3000 system to include DS/3000. Instructions are provided in the System Manager/System Supervisor Manual, Part Number 30000-90014.
- The HP 3000 system operator initiates the :DSCONTROL command to enable the DS subsystem on the communications link. Instructions are provided in Appendix B of this manual.

Now you can log on to a terminal and issue the user DSLINE command to establish communications with the HP 1000 system, or systems.

Opening A Line

The format of the DSLINE command is described in detail in the DSN/DS Reference Manual and Appendix E. For purposes of communicating with an HP 1000, the command parameters identified here:

Before the DSLINE command can be used a :DSCONTROLxx;OPEN must be typed into a system console.

```
:DSLINE
  [;LINEBUF=buffersize]
  [;LOCID=local-id-sequence]
  [;REMID=remote-id-sequence]
  [;PHNUM=telephone-number]
  [;EXCLUSIVE]
```


line number is analagous to the file number returned to you by the MPE file system when you open a file programmatically using the FOPEN intrinsic. With your local session it uniquely identifies the particular line that you have opened. This becomes significant only if you open more than one communications line during a session.

Now that you have acquired access to a communications line between the HP 3000 and HP 1000, you can proceed to log on to an RTE session, then you can:

- execute Operator Commands on the HP 1000 system
- use Remote EXEC calls to the HP 1000 system
- access files on the remote HP 1000 system
- perform Program-to-Program Communication between systems

The processing that takes place in the HP 1000 system is referred to as remote processing. There are two distinct processing systems in progress concurrently; the local session in the HP 3000 and remote processing in the HP 1000 system. It is important to keep in mind that within the HP 3000 your local session operates under the capabilities and security restrictions for your log-on account, while within the HP 1000 system remote processing operates under the capabilities and security restrictions defined for the 1000 session to which you are attached.

Line Opening Failures

If the DSLINE command is not executed successfully, the error message may help identify the difficulty. Refer to Appendix A.

Things to consider if you have experienced a line opening failure:

- Insure :DSCONTROL ldev,OPEN was accomplished.
- Insure :RU,DINIT has been executed on the RTE system. And, if :DSCONTROL ldev,OPEN was previously done the

COMMUNICATIONS LINK UP

message should have been displayed on the HP 1000 system console.

Establishing the Link

- If previous line activity resulted in an error do :DSLLINE ldev;CLOSE and then try again to open the line with a :DSLLINE ldev
- In case of continuing problems consider any of the following:

- Enabling the line from the HP 1000 by :RU,DSMOD and using the /L command to re-enable the communication used to communicate with the 3000.

- Try to establish remote session from the HP 1000 to the HP 3000 instead with:

```
:RU,RMOTE  
#HELLO ...
```

- Do :BYE on the HP 3000 session

```
:DSCONTROL ldev,SHUT  
:DSCONTROL ldev,OPEN  
:HELLO ...
```

The :DSCONTROL ldev,OPEN allocates the software and establishes the logical links between software modules. This command sends an ENQ across the link. If an ACK0 is received from the HP 1000 the message

DS LINE NUMBER = #Lx

is printed, where x is a number.

On the HP 1000, after the DINIT program is executed, an initialization message is sent establishing the buffer size. The initialization reply from the HP 3000 results in satisfactory completion, and a printing of the message

COMMUNICATION LINK *UP*.

EXECUTING REMOTE OPERATOR COMMANDS

SECTION

III

Once you have obtained access to a physical communications line using the DSLINE command, the REMOTE command can be used to execute operator commands on the HP 1000 system.

Within the local session you have access to all I/O devices and disc files in your local HP 3000, subject to the usual MPE file security. This is a normal MPE interactive session in every respect; you enter MPE commands and use the various language and utility subsystems exactly as you would if the communications link were not present. This local session is running under the username, accountname, and groupname specified in the HELLO command that you used when you logged on. All user capabilities and file access available to you within the local session are determined by those log-on parameters.

You may use the :REMOTE HELLO user.group/password to log on to a 1000 account (if the 1000 has session monitor). This is a non-interactive session. It establishes the environment at the 1000 (Session LU table, capability, and file cartridge access).

Within remote processing you have access to the RTE operating system in the remote HP 1000 by entering RTE operator system commands exactly as you would if you were operating at the HP 1000 and the communications link were not present. The RTE operator commands are executed in the remote HP 1000; however, the output resulting from the executed commands appears at your log-on terminal.

NOTE

Only RTE systems commands can be executed. There is no copy of FMGR for your session.

The REMOTE Command

The format of the REMOTE command is shown in figure 3-1. Because the REMOTE command can initiate processing for you in a remote HP 1000, the parameters in that command specify information pertaining to the operating environment of the remote HP 1000, not your local HP 3000.

Executing Remote Operator Commands

:REMOTE [lnumber] command

lnumber The logical line number returned by DS when the communications line was opened. The line number is necessary only if you have more than one communications line open simultaneously. If it is omitted, the line most recently opened is referenced by default.

command The HP 1000 operator command sent to the remote processor. It must follow the command syntax of the remote processor verbatim, and the first character must not be a numeric or comma.

Figure 3-1. The REMOTE Command

Executing Remote Operator Commands

Different versions of RTE support different operator commands. Consult the documentation provided with your RTE system to determine the exact format of your RTE commands.

The following example shows how to log on to an HP 3000, open the DS line to the HP 1000, and issue an RTE operator command to examine the status of an I/O device on the HP 1000 system.

```
carriage return
:HELLO FIELD.SUPPORT,PUB]
HP 3000 / MPE III B.00.02 MON, MAR 2, 1981, 10:01 AM

WELCOME TO THE HP 3000
:DSLINE HDS2

DSLINE NUMBER = #L3

:REMOTE EQ,2
16 DVR00 0 B U 0 0
: ...
```

In the following example, two DS lines are opened, and a remote command is issued over one of the lines:

```
carriage return
:HELLO FIELD.SUPPORT,PUB]
HP 3000 / MPE III B.00.02 MON, MAR 2, 1981, 10:01 AM

WELCOME TO THE HP 3000
:DSLINE HDS2

DSLINE NUMBER = #L3

:DSLINE HDS1

DSLINE NUMBER = #L4

:REMOTE #L3 EQ,2

16 DVR00 0 B U 0 0
```

The preceding examples allow you to execute only a single remote command. After the remote command has been executed, control returns to your local session, as signified by the colon prompt character.

If you want to execute a whole series of remote commands in the remote session, you can simply enter the REMOTE command and press carriage return. REMOTE may also be followed by the line number. The HP 1000 system prompts you for each RTE operator command by displaying a pound sign, #, instead of the usual colon prompt.

Executing Remote Operator Commands

This is shown in the example below:

```
:REMOTE (or :REMOTE #L3)
#EQ,2
16 DVR00 0 B U 0 0
#LU,1
LU #1 = E2
#
```

When you want to return to the local session from remote processing, simply enter a colon in response to the # prompt.

```
#:
```

When you respond to the remote # prompt with a colon, control returns to the MPE Command Interpreter of your local HP 3000 which then prompts you for local commands with the colon prompt character. Note that the remote colon does not terminate remote processing; you can resume processing in a remote process by again issuing a REMOTE command.

Executable RTE Operator Commands

While in a remote process on the HP 1000 system, you can use the RTE operator commands listed below:

AB	Abort current batch program
BL	Set buffer limits
BR	Set break flag in the ID segment of the named program
DN	Declare I/O device unavailable
EQ	Examine status of I/O device
GO	Restart program out of suspension
IT	Set time intervals for programs
LU	Examine or alter device logical unit assignments
OF	Turn programs off, an abort
ON	Turn programs on
PR	Change priority of programs
RU	Start a program immediately
RT	Release disc tracks of a program
SS	Suspend programs
ST	Examine the status of programs
TI	Display the current time (RTE-IVB and RTE-6/VM)
TM	Display the current time (RTE-L, RTE-XL, RTE-A.1)
TO	Examine or alter time-out parameter of an I/O device
UP	Declare I/O device available



EXECUTING REMOTE EXEC CALLS

SECTION

IV

In Section 3 the remote operator commands are described as a way of communicating with a remote program. You can also interface with remote systems programmatically by executing remote EXEC calls from an SPL or FORTRAN program. These are called DEXEC calls in DS/3000.

By using DEXEC calls you can direct the RTE System Executive program to:

- Transfer information to or from a remote I/O device.
- Perform I/O operations on remote I/O devices.
- Schedule dormant remote programs for execution.
- Obtain the current time from the remote real-time clock.
- Schedule remote programs for a set time of execution.
- Obtain status information about a remote I/O device.

To help determine what you can accomplish with DEXEC calls, a summary is presented in table 4-1, listing each DEXEC call with its instruction code and the function performed by each call. For further information on EXEC calls, refer to the RTE Software System Programming and Operating Manual.

These programmatic DEXEC calls can be used from an HP 3000 to a HP 1000. They require the following commands to be executed:

- The DSCONTROL command to open the line at the 3000.
- Execution of an RU,DINIT to open the line at the 1000.
- Execution of a DSLINE command within the session.

Executing Remote Exec Calls (DEXEC Calls)

Table 4-1. DEXEC Calls

Instruction Code	Call:	Function:
1	Read	Reads a record from a remote I/O non-disc device.
2	Write	Writes a record on a remote I/O non-disc device.
3	I/O Control	Performs an I/O control operation on a remote I/O device.
10	Program Schedule	Schedules dormant remote programs for execution.
11	Time Request	Obtains the current time from the remote real-time clock.
12	Execution Time	Schedules remote programs for a set time of execution.
13	I/O Status	Obtains I/O status information about a local I/O device.

DEXEC Call Elements

From SPL, each DEXEC call consists of:

- A functional return, the returned values of the HP 1000 A- and B- registers.
- An assignment operator, a colon followed immediately by by equal sign.
- The double-word procedure name DEXEC.
- A parameter list.

The format of a typical DEXEC call is:

```
abreg := DEXEC ( destination, instructioncode,  
                controlword, buffer, bufferlength, opt1,  
                opt2 ) ;
```

The following DEXEC call formats include superscripts that describe the types of parameters and whether they must be passed by value instead of by reference. The superscripts have the following meanings:

BA	byte array
BP	byte pointer
D	double word, by reference
DA	double word array
DV	double word, by value
I	integer by reference
IA	integer array
IV	integer by value
L	logical by reference
LA	logical array
LV	logical by value
O-V	option variable
R	real by reference

The superscript O-V is shown for some DEXEC calls to denote option variable, meaning that the DEXEC call contains optional parameters.

Functional Return

The contents of the A- and B- registers immediately after the execution of the DEXEC (EXEC) call at the HP 1000 are returned in the double word variable abreg. Besides normal return information, four-character ASCII error codes such as I002, and SC05 may be returned in abreg. The meaning of these error codes is described in Appendix A of this manual.

The following describes some possible outcomes :

- < (CCL) --> Error detected by 3000. eg (1) contains error code (numeric).
- Completed at communications level, but 1000 detected error CCE returned by intrinsic. ASCII error code is in abreg.
- Completed alright at both ends. Registers contain status information, not error information.

ERROR CHECKING ROUTINE

```
LOGICAL PROCEDURE ALPHNUM(TESTWORD);
VALUE TESTWORD; INTEGER TESTWORD;
BEGIN
    ALPHNUM := FALSE;
    IF ("0" <= TESTWORD,(0:8) <= "Z") THEN
    IF ("0" <= TESTWORD,(0:8) <= "Z") THEN
    ALPHNUM := TRUE;
    END; << ALPHNUM >>
```

note: This function returns true only if both bytes of the parameter are alphabetic.

```
PROCEDURE CHECKDEXEC;
BEGIN
    COMMENT
    CHECK FOR DEXEC ERROR FROM REMOTE RTE NODE;
    IF < THEN
        BEGIN <<CONDITION CODE LESS ERROR>>
        MOVE MSG := "** DEXEC ERROR
        ASCII(REG(1),10,MSG(15)); <<CONVERT REG VALUE>>
        PRINT(MSG1,-21,%40);
        GOOD := FALSE;
        END
    ELSE IF ALPHNUM(REG(0)) AND ALPHNUM(REG(1)) THEN
        BEGIN <<REG ERROR IS ALPHANUMERIC>>
        MOVE MSG := "** DEXEC ERROR ";
        MOVE MSG1(8) := REG,(2);
        PRINT(MSG1,-20,%40);
        GOOD := FALSE;
        END;
    END; << CHECKDEXEC >>
```

READ

Reads a record from a remote device.

Instruction code 1

D	LA	IV	I
abreg := DEXEC(destination, instructioncode, controlword,			
IA	IV	IV	IV
buffer, bufferlength, opt1, opt2);			

The READ call, instruction code 1, transfers information from a remote I/O device to a buffer.

Functional Return

Upon completion of the call, the doubleword abreg contains the following information:

Word 1, displayed in the A-register, shows

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AV		EQT			STATUS										

STATUS Status information. Refer to the RTE
Bits (8:8) manual for description.

EQT Equipment type code. When this octal number
Bits (2:6) is linked with DVR it identifies the software
driver routine of the device.

Note: This is not supported to a disc.

AV Availability indicator:
Bits (0:2) 00 available for use, up
 01 disabled, down
 10 busy
 11 waiting for DMA

Word 2, displayed in the B-register, shows the positive number of words or characters actually read, depending on which was specified in the parameters of the procedure call.

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string destination or logical unit. The last word is used by the system to store the DSLINE number.

DEXEC Calls: Read

instructioncode integer by value (required). The instruction code is 1 for a READ call.

controlword integer (required). An integer specifying the logical control unit of the remote input device and control information as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0					X	A	K	V	M	LU					

LU Logical unit number of the remote input device. Note: You cannot specify a disc LU number. Use RFA commands for disc I/O.
Bits (10:6)

M M-bit. 0 = ASCII
Bit (9:1) 1 = binary

V V-bit.
Bit (8:1)
V=1: If M=1 the length of the punched tape input is determined by the word count in the first non-zero character read.
V=0: If M=1 the length of the punched tape input is determined by bufferlength.

K K-bit. 0 = no echo on keyboard
Bit (7:1) input. 1 = echo keyboard input.

A A-bit. Not effective on some devices.
Bit (6:1)
A=1 and M=0 designates punching without printing on a teleprinter.
A=0 designates that the M-bit determines the transfer mode.

X X-bit. When paper tape is used, X, M, and V indicate an honesty mode. Exactly what is punched on the tape is read.
Bit (5:1)
X=1, M=1, V=1: Absolute binary tape format.
X=1, M=1, V=0: Leader is not skipped.

X=1, M=0: ASCII format, leader is not skipped, bit 8 is stripped, a line feed terminates a record.

Bits (0:5) Must be 0.

The control word may have different meanings for different devices. Check the RTE Driver Manual for specific information pertaining to your device.

- buffer integer array (required). An array that will contain the information read from the device. You must insure the buffer is large enough to hold the amount of data specified by bufferlength.
- bufferlength integer (required). An integer specifying the number of words or characters to be read. If this value is positive it signifies the length in words; if it is negative it signifies the characters. The maximum buffer length in the HP 1000 is 512 words (1023 characters).
- opt1 and opt2 integer by value (optional). Integer parameters which may be optional, depending on the input device. Unlike a local RTE EXEC call, these parameters cannot be used to pass control buffers.

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

- CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.
- CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.
- CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

WRITE

Writes a record on a remote I/O device. Instruction code 2.

```

D          LA          IV          IV
abreg := DEXEC( destination, instructioncode, controlword,
              BA      IV      IV      IV      O-V
              buffer, bufferlength, opt1, opt2 );
```

The WRITE call, instruction code 2, transfers information from a buffer to a remote I/O device.

Functional Return

Upon completion of the call, the doubleword abreg, contains the following information:

Word 1 (displayed in the A-register) shows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AV	EQT						STATUS								

STATUS Status information. Refer to the RTE Bits (8:8) manual for description.

EQT Equipment type code. When this octal Bits (2:6) number is linked with DVR it identifies the software driver routine of the device. Note: This is not supported to a disc.

AV Availability indicator:
Bits (0:2) 00 available for use, up
 01 disabled, down
 10 busy
 11 waiting for DMA

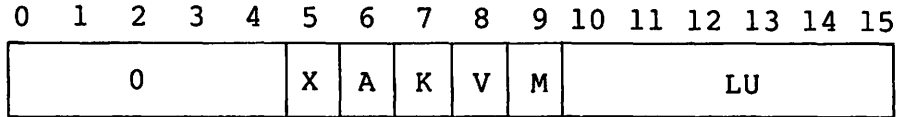
Word 2, displayed in the B-register, shows the positive number of words or characters actually read, depending on which was specified in the call.

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string destination or logical unit. The last word is used by the system to store the DSLINE number.

instructioncode integer by value (required). The instruction code is 2 for a WRITE call.

controlword integer by value (required). An integer specifying the logical control unit of the remote input device and control information as follows:



LU Logical unit number of the remote Bits (10:6) output device.

M M-bit. 0 = ASCII and 1 = binary. Bit (9:1)

V V-bit. Bit (8:1) V=1: If M=1 the length of the punched tape input is determined by the word count in the first non-zero character read.

V=0: If M=1 the length of the punched tape input is determined by bufferlength.

K K-bit. Bit (7:1) 0 = no echo on keyboard input. 1 = echo on keyboard input.

A A-bit. Not effective on some devices. Bit (6:1)

A=1 and M=0 designates punching without printing on a teleprinter.

A=0 designates that the M-bit determines the transfer mode.

X X-bit. When paper tape is used, X, Bit (5:1) M, and V indicate an honesty mode. Exactly what is punched on the tape is read.

X=1, M=1, V=1: Absolute binary tape format.

X=1, M=1, V=0: Leader is not skipped.

DEXEC Calls: Write

X=1, M=0: ASCII format, leader is not skipped, bit 8 is stripped, line feed terminates a record.

Bits (0:5) Must be 0.

The control word may have different meanings for different devices. Check the RTE Driver Manual for specific information pertaining to your device.

- buffer integer array (required). An array that will contain the information read from the device. You must insure the buffer is large enough to hold the amount of data specified by bufferlength.
- bufferlength integer by value (required). An integer specifying the number of words or characters to be read. If this value is positive it signifies the length in words; if it is negative it signifies the characters. The maximum buffer length in the HP 1000 is 512 words (1023 characters).
- opt1 and opt2 integer by value (optional). Integer parameters which may be optional, depending on the input device. These parameters cannot be used to pass control buffers.

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

- CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.
- CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.
- CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

I/O CONTROL

Performs an I/O control operation on a remote I/O device.

Instruction Code 3

D	LA	IV	IV
abreg := DEXEC(destination, instructioncode, controlword,			
	IV	O-V	
param);			

The I/O control call, instruction code 3, carries out various operations such as backspace, rewind, end of file, on a remote I/O device.

Functional Return

The call returns the A-register and B-register in a double word. The A-Register contains the status of the device, EQT word 5. The B-Register is meaningless.

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string destination or logical unit. The last word is used by the system to store the DSLINE number.

instructioncode integer by value (required). The instruction code is 3 for an I/O control call.

controlword integer by value (required). An integer specifying the logical unit of the remote I/O device and control information as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Reserved					Function Code					Logical Unit No.					

DEXEC Calls: I/O Control

Function Code: (1)	Purpose: (2)
00	Clear device.
01	Write end-of-file. (MT/CTU)
02	Backspace one record. (MT/CTU)
03	Forward space one record. (MT/CTU)
04	Rewind. (MT/CTU)
05	Rewind standby. (MT) Rewind. (CTU)
06	Dynamic status. (MT/CTU)
07	Set end-of-tape; leader skipped on next input request. (PT)
10	Generate leader. (PT) Write end-of-file if not just previously written or not at load point. (CTU)
11	Skip line. (LP)
12	Write 3-inch gap. (MT)
13	Forward space file. (MT/CTU)
14	Backspace file. (MT/CTU)
15	Conditional form feed. (LP)
20	Enable terminal. Allow terminal to schedule its program when any key is struck.
21	Disable terminal. Inhibit scheduling of program of terminal.
22	Set time-out interval parameter of terminal.
23	Ignore further action requests on terminal until <ul style="list-style-type: none">• the device queue is empty, or• an input request is encountered, or• a restore control request is received.
24	Restore output processing for terminal.
26	Write end-of-data. (CTU)
27	Locate file. (CTU)

NOTES

- (1) Octal value.
(2) Device abbreviations:

CTU cartridge tape unit
LP line printer
MT magnetic tape
PT paper tape

param integer by value (optional). A parameter required by the following controlwords:

control-
word:

Meaning:

11 Skip param lines on the line printer.

A negative value causes a skip to the top of the next page. On the teleprinter, a negative value is the same as a positive value.

param	teleprinter	line printer
+ n	space n lines	space n lines
- n	space n lines	top of form lines
0	no linefeed	no linefeed

22 Set the time-out interval parameter of the terminal.

27 Find the file number param, less than 256, on the CTU.

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.

CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

PROGRAM SCHEDULE

Schedules dormant remote programs for execution.

Instruction code 10

```

      D           LA           IV           IA   IV
abreg := DEXEC( destination, instructioncode, name, opt1,
              IV  IV   IV   IV   O-V
              opt2, opt3, opt4, opt5 );
```

The remote program schedule call, instruction code 10, designates a dormant remote program for execution at the remote system and optionally transfers up to five parameters to the program.

Functional Return

If the program to be scheduled is dormant, it is scheduled and a zero is returned to the calling program in the A-Register.

If the program to be scheduled is not dormant, it is not scheduled by this call, and its status, which is some non-zero value, is returned to the calling program in the A-Register.

A-register:	Meaning:
1	Scheduled.
2	I/O suspend.
3	Program in wait state.
4	Unavailable memory suspend.
5	Disc allocation suspend.
6	Operator suspend, or program suspend.

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string designating the DSLINE formal designator or logical unit. The last word is used to store the DSLINE number.

instructioncode integer by value (required). The instruction code is 10 for a program schedule call.

name integer array (required). A five character name array of the program to schedule.

opt1 - opt5 integer by value (optional). Up to five binary parameters which may be passed to the scheduled program.

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.

CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

TIME REQUEST

Obtains the current time from the remote real-time clock. Instruction code 11

D	LA	IV	IA
abreg:=DEXEC (destination, instructioncode, time);			

The time request call, instruction code 11, returns to your program the current time.

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string designating the DSLINE formal designator or logical unit. The last word is used to store the DSLINE number.

instructioncode integer by value (required). The instruction code is 11 for a time request call.

time integer array (required). A five word value array in which the current time is returned.

TIME (1) = 10s of milliseconds
TIME (2) = seconds
TIME (3) = minutes
TIME (4) = hours
TIME (5) = day of year

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.

CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

EXECUTION TIME (Initial Offset)

Schedules remote programs for a set time of execution after an initial offset time. Instruction code 12

```
      D          LA          IV          IA
abreg:=DEXEC (destination, instructioncode, prog,
              IV          IV          IV
              resolution, multiple, offset);
```

The execution time call schedules a program for execution at specified time intervals, starting after an initial offset time. The requested program is placed in the time list.

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string designating the DSLINE formal designator or the logical unit. The last word is used to store the DSLINE number.

instructioncode integer by value (required). The instructioncode is 12 for an execution time call.

prog integer array (required). A five character array naming the remote program to put in the time list.

resolution integer by value (required). The resolution code represents the units of time which, when used with the multiple value, result in the total time intervals. Values for the resolution code are:

1	tens of milliseconds
2	seconds
3	minutes
4	hours

multiple integer by value (required). A number from 0 to 4095, which is used with resolution to give the actual time interval for scheduling. If the execution multiple is set to zero, the program is run only once; if set to 1, it is run twice; and so forth.

DEXEC Calls: Execution Time (Initial Offset)

offset integer by value (required). The offset code is used with the resolution code for the initial time offset. It is a negative value representing the amount of time to wait before running the program for the first time.

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.

CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.

I/O STATUS

Obtains I/O status information about a remote I/O device.

Instruction code 13

```

      D          LA          IV
abreg := DEXEC( destination, instructioncode,
              IV          L          L
              controlword, status1, status2 );
    
```

Parameters

destination logical array (required). A five word array that contains in its first four words an ASCII string designation or logical unit. The last word is used to store the DSLINE number.

instructioncode integer by value (required). The instruction code is 13 for an I/O status call.

controlword integer by value (required). An integer specifying the logical unit of the remote I/O device and control information as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Reserved					0					LU					

LU logical unit number of the remote device.
Bits (10:6)

Bits (5:5) must be 0.

status1 logical (required). The status returned for EQT word 5.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AV		EQT					STATUS								

AV availability indicator

- 0 available for use
- 1 disabled, down
- 2 busy, concurrently in operation
- 3 waiting for an available DMA channel

EQT equipment type code. When this number is linked with DVR it identifies the software driver routine of the device

DEXEC Calls: I/O status

00 to 07, paper tape devices or system control devices (in octal):

00 teleprinter, or system keyboard control device
01 photo-reader
02 paper tape punch

05 subchannel 0: interactive keyboard device
subchannel 1,2: HP minicartridge device
07 multipoint driver

10 to 17, unit record devices:

10 plotter
11 card reader

12 line printer

15 mark-sense card reader

20 to 37, magnetic tape/mass storage devices:

31 7900 moving head disc
32 7906/7920/7925 moving head disc
33 flexible disc

40 to 77, instruments.

STATUS the actual physical or simulated status at the end of each operation.

status2

logical (optional). The status returned for EQT word 4.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	B	P	S	T	Unit Number					Channel Number					

- D = 1 means DMA is required.
- B = 1 means automatic output buffering is used.
- P = 1 means driver is to process a power fail.
- S = 1 means driver is to process a time out.
- T = 1 means the device is timed out. The system sets T to zero before each I/O request.

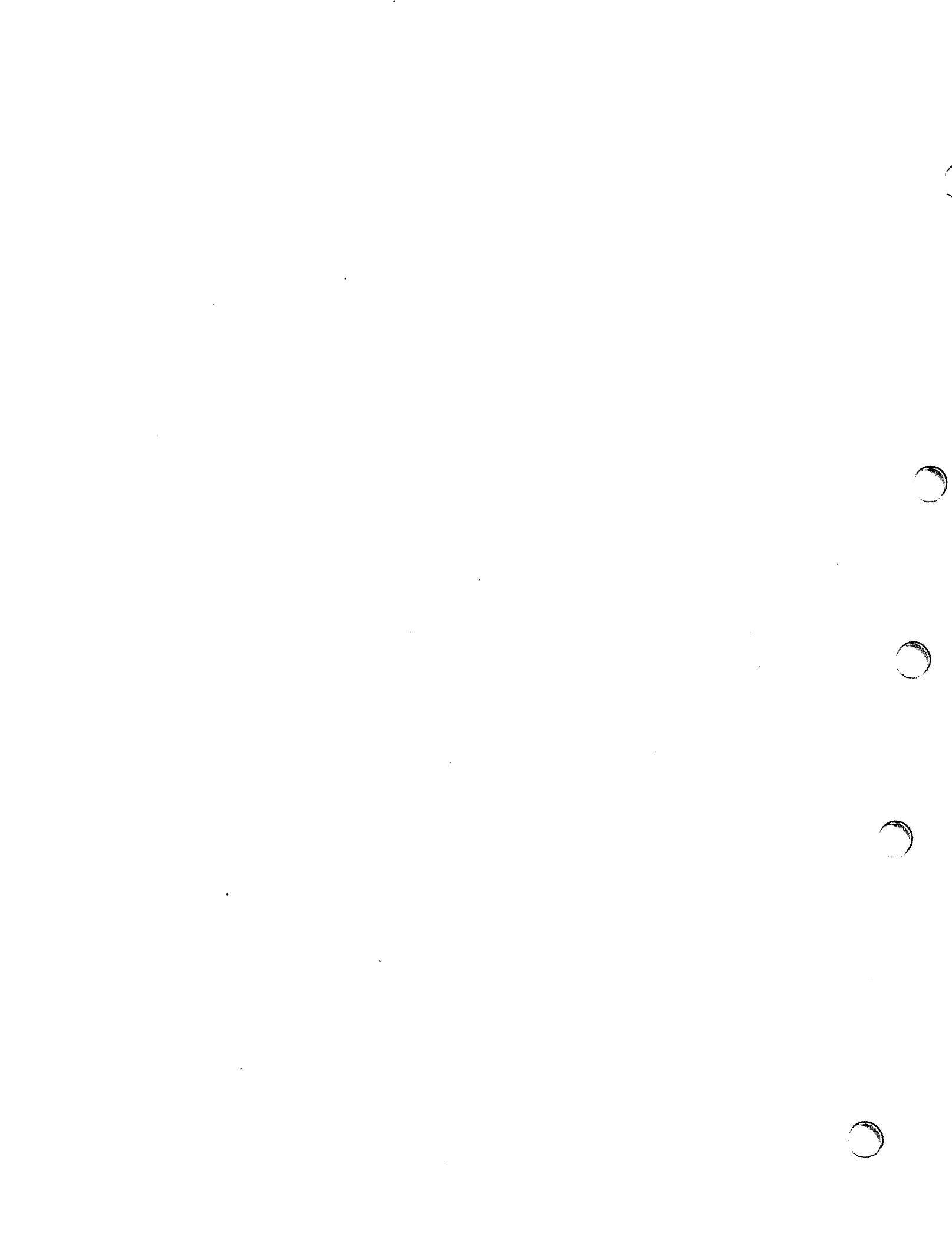
Unit Number is the last sub-channel addressed.

Channel Number is the I/O select code for device. Lower number if a multiboard interface.

Error Conditions

Each DEXEC call should be followed with a test of the condition code, whose values indicate

- CCE satisfactory completion. The abreg may contain information. Refer to the DEXEC call.
- CCL Error conditions on the HP 3000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.
- CCG Error condition on the HP 1000 end. Refer to Appendix A for specific error codes. The binary value is contained in the abreg.



Through the use of the DS/3000 Remote File Access capability, SPL, PASCAL or FORTRAN programs running in your local session can:

- access any of the disc files of the remote HP 1000 (which are available to your session) as though they reside at your local HP 3000 site.

The remainder of this section discusses how you can code portions of a program on your local HP 3000 using various file calls, intrinsics.

Programmatic Access

Once a communications link has been established between your HP 3000 and remote HP 1000, you can use the Remote File Access (RFA) intrinsics to access files available to your session, residing at the remote HP 1000 site. Remote File Access provides the means to create, open, read, write, close, and control the files on the remote HP 1000 system.

The following table (Table 5-1) lists the Remote File Access intrinsics, the corresponding RTE File Manager (FMP) call, and a corresponding description of each RFA intrinsic call. Refer to the Batch-Spool Monitor Reference Manual for further information on FMP calls.

Remote File Access

These RFA intrinsics may be used from the HP 3000 to an adjacent HP 1000. It requires that the line be opened by the commands:

- :DSCONTROL ldev;OPEN console comand on HP 3000 to enable DS/3000
- :RU,DINIT On the HP 1000 to schedule various monitors and to enable the line.
- :DSLINe ldev In session on HP 3000 to the logical use of a DS line to the HP 1000.

DSLINe NUMBER =#L3

Description of Error Conditions and Parameters

CCE	indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter.
CCL	indicates problems at the HP 3000 end. Refer to Appendix A for summary of the <u>error</u> code.
CCG	indicates problems at HP 1000 end. Refer to Appendix A for summary of the <u>error</u> codes. Refer to the appropriate Batch Spool Monitor Reference Manual for more detailed information.
error	integer (required). The error codes returned as a result of remote file access.

The following is a procedure used to check for errors after an RFA call:

```
PROCEDURE CHECKRFA;  
  BEGIN  
  COMMENT  
  CHECK FOR FILE ERROR FROM REMOTE RTE NODE;  
  IF < OR ERROR<0 THEN  
  BEGIN  
  MOVE MSG:= "*** RTE FILE ERROR ";  
  ASCII(ERROR,10,MSG(18));  
  PRINT(MSG1,-24,%40);  
  GOOD :=FALSE;  
  END;  
  END; << CHECKRFA >>
```

Table 5-1. RFA and FMP Cross Referencing

RFA	FMP	DESCRIPTION
DAPOS	APOSN	Positions disc files to a record. Record address usually obtained through DLOCF.
DCLOS	CLOSE	Close DCB. Make file available to others. Can also truncate file.
DCONT	FCONT	Perform I/O control function on non-disc device file, a type 0 file.
DCRET	CREAT	Create a file.
DLOCF	LOCF	Return information on opened file.
DNAME	NAMF	Close DCB and rename file.
DOPEN	OPEN	Open file to calling program.
DPOSN	POSNT	Skip records forwards or backwards.
DPURG	PURGE	Purge file. Cannot be opened.
DREAD	READF	Read a record to user buffer.
DSTAT	FSTAT	Return status of all mounted cartridges.
DWIND	RWNDF	Rewind type 0 files, including magnetic tape. Set to record 1 in disc file.
DWRIT	WRITEF	Write a record from the buffer of a user to file.

DAPOS

Sets the address of the next record for sequential RTE disc file.

A	I	IV	IV	IV	O-V
DAPOS (DCB, error, record, blocknum, wordnum);					

Parameters

- DCB** array (required). The Data Control Block (DCB), a 4-word array used for each RTE file opened. This is used by the system and should not be modified.
- error** integer (required). Error codes returned as the result of remote file access.
- record** integer by value (required). Record number of the next record in sequentially numbered file. The first record is numbered starting with 1.
- blocknum** integer by value (optional). Relative block number of block containing the next record, from the DLOCF intrinsic. This parameter must be included for correct operation on files with variable length records.
- wordnum** integer by value (optional). A word offset, within a block, for the beginning of a new record, from DLOCF intrinsic. This parameter must be included for correct operation on files with variable length records.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

- | | |
|-----|--|
| CCE | indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter |
| CCL | indicates problems at the HP 3000 end. Refer to Appendix A for information on <u>error</u> code. |
| CCG | indicates problems at HP 1000 end. Refer to Appendix A for summary of <u>error</u> codes. Refer to appropriate Batch Spool Monitor Reference Manual for more detailed information. |

NOTE

The DAPOS intrinsic allows random access of sequential files.

DCLOS

Closes the DCB and makes the RTE file available to other callers.

```
          A      I      IV      O-V  
DCLOS ( DCB, error, truncate );
```

Parameters

DCB array (required). The Data Control Block (DCB), a four word array used for each file opened. This is used by the system and should not be modified.

error integer (required). The error codes returned as a result of remote file access.

truncate integer by value (optional). The number of blocks in the file to truncate:

- + N number of blocks to be deleted from the end of the file.
- N delete all extents and retain the main file.
- 0 the standard close.

See also the warning note below.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

Refer to the appropriate Batch Spool Monitor Reference Manual for more detailed information.

NOTE

The DCLOS intrinsic closes the Data Control Block, and makes the RTE file available to other callers. It also gives the user the option of truncating the file size when:

- The file is opened exclusively.
- The current position is in the main file, not in the extent.
- The security codes match at open.
- The number of blocks to be truncated is less than or equal to the number of blocks in the file.

WARNING

If the number of blocks specified in truncate is equal to the number of blocks in the file, the file is purged.

DCONT

Sends standard RTE I/O control request to RTE type 0, non-disc, file.

```
          A      I      IV      IV      O-V
DCONT ( DCB, error, controll, control2 );
```

Parameters

DCB array (required). The Data control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.

error integer (required). Error codes returned as a result of remote file access.

controll integer by value (required). Control word specifying an action.

control 1: Meaning: (See note below for abbreviations.)

Octal

- 00 Clear device.
- 01 Write end-of-file (MT/CTU)
- 03 Forward space one record (MT/CTU)
- 04 Rewind (MT/CTU)
- 05 Rewind standby (MT)
- 06 Dynamic status (MT/CTU)
- 07 Set end-of-tape; leader skipped on next input request.
- 10 Generate leader (PT). Write end of file unless just written, or file is at load point (CTU).
- 11 Skip line (LP).
- 12 Write 3-inch gap (MT).
- 13 Forward space file (MT/CTU).
- 14 Backspace file (MT/CTU).
- 15 Conditional form feed (LP).
- 20 Enable terminal. Allow terminal to schedule its program when any key is struck.
- 21 Disable terminal. Inhibit scheduling of the program of the terminal.

- 23 Ignore further action requests on terminal until:
 - the device queue is empty, or
 - an input request is encountered, or
 - a restore control request is received.
- 24 Restore output processing for terminal.
- 26 Write end of data (CTU)
- 27 Locate file (CTU)

Device abbreviations are

CTU cartridge tape unit
 LP line printer
 MT magnetic tape
 PT paper tape

control2 integer by value (optional). Specifies the number of lines to be spaced. Required when controll = 11, 22, or 27.

controll control2
 Value: Meaning:

- 11 Skip control2 lines on the line printer. A negative value causes a skip to the top of the next page on line printers; a negative value is the same as a positive value on the teleprinter.

<u>con- trol2</u>	Teleprinter	Line printer
+ N	Space N lines	Space N lines.
- N	Space N lines	Top of form.
0	No linefeed.	No line feed.

- 22 Set terminal time-out interval parameter.
- 27 Find the number parameter, less than 256, on the CTU.

The DCONT intrinsic has no effect on disc files.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

Refer to the appropriate Batch Spool Monitor Reference Manual for detailed information.

Closes the DCB, if open, and creates the named RTE file on the specified disc with the specified number of blocks.

```

      A      I      A      A      IV      IV      IA  O-V
DCRET ( DCB, error, name, size, type, security, crn );
```

Parameters

- DCB array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.

- error integer (required). Number of sectors or error codes returned as a result of remote files access.

- name array (required). RTE file name of up to six characters.

- size array (integer). Two word integer array. Word 1 specifies the number of blocks desired. If the number of blocks is less than 0, the whole disc is used. Word 2, used only for type 2 files, contains record length in words.

1 The number of blocks desired.
2 The record length in words. Type 2 files.

- type integer by value (required). The file type.

type:	Meaning:
1	128-word record, random access.
2	User specified record length, random access.
3	3 or greater random record length, sequential access. EOF written at beginning of file.
4	Source program.
5	Relocatable program.
6	RTE load module.
7	Absolute program.
> 7	User defined.

- security integer by value (optional). The security code, two characters. Refer to the Operating System Manual for details.

- crn array. Required during the first create. A five word array where:

Remote File Access: DCRET

crn [0] cartridge number
crn [1]-[4] ASCII representation of the DS device
 class or logical unit number of DSLINE.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes. See the appropriate Batch Spool monitor manual for more detailed information.

NOTE

The RTE file created by the DCRET intrinsic is left open exclusively to the caller on successful completion of the intrinsic call.

This intrinsic does not create a type-0 file. These are device files, such as line printers or card readers. Type-0 files may be created only by FMGR operator commands at the HP1000.

DLOCF

Remotely accesses a DCB maintained in RTE to obtain location and status information about the next record.

A	I	I	I	I	I	I	
DLOCF	(DCB,	error,	recnum,	blocknum,	wordnum,	size,	lu,
	I	I	O-V				
	type,	recsize);					

Parameters

- DCB** array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.
- error** integer (required). The error codes as a result of remote file access.
- recnum** integer (required). Returns record number of next sequential record. Record numbers begin with 1.
- blocknum** integer (optional). Returns the block number of the current block.
- wordnum** integer (optional). Returns a word offset, within the current block, at which the next record will begin.
- size** integer (optional). Returns the number of sectors in the main file.
- lu** integer (optional). Returns the logical unit number of the file on disc, or of the non-disc device for type 0 files.
- type** integer (optional). Returns the file type as indicated in DCB.
- recsize** integer (optional). Returns the record size in words. For type 0 files a read/write code is returned as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
W	Reserved.														R

R Read (Bit 15) = 1
W Write (Bit 0) = 1

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

- CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter
- CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.
- CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes. Refer appropriate Batch Spool monitor manual for more detailed information.

DNAME

Closes the DCB, if open, and renames the specified RTE file.

```
      A      I      A      A      IV      A      O-V
DNAME( DCB, error, name, newname, security, crn );
```

Parameters

DCB array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.

error integer (required). Error codes returned as a result of remote file access.

name array (required). Current RTE file name up to six characters.

newname array (required). New RTE file name up to six characters.

security integer by value (optional). Two character security code of the old file. Remains unchanged.

crn array. Required if file not currently open to user.

 crn [0] cartridge number. If specified the search is limited to just that cartridge. If not specified all cartridges are searched, but only the first file encountered with the old name is changed.

 crn [1]-[4] ASCII representation of the DS device class or logical unit number of DSLINE.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes. See the appropriate Batch Spool Monitor Reference Manual for detailed information.

DOPEN

Closes the DCB, if open, and opens the named RTE file.

A	I	A	IV	IV	A	O-V
DOPEN (DCB, error, name, optn, security, crn);						

Parameters

- DCB** array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.
- error** integer (required). The file type or error codes returned as a result of remote file access.
- name** array (required). RTE file name assigned to file of up to six characters.
- optn** integer by value (optional). The option code is defined as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Reserved = *				X	*	K	V	M	*			F	T	U	E
												Function Code		1	

E E-bit.
Bit (15:1) 0 = exclusive open.
1 = non-exclusive open.

Exclusive allows one user at a time.
Non-exclusive allows up to seven users at a time.

U U-bit.
Bit (14:1) 0 = non-update mode.
1 = update mode.

Update causes block to be read before it may be modified, thus preserving subsequent records.

T T-bit.
Bit (13:1) 0 = file type defined at creation.
1 = file type is forced to Type 1.

F	F-bit.
Bit (12:1)	0 = function code defined at creation. 1 = function code defined in bits (5:5) of optn.
M	M-bit.
Bit (9:1)	0 = ASCII data. 1 = Binary data.
V	V-bit.
Bit (8:1)	0 If M=0 (ASCII) print column 1 on line printer. 0 If M=1 (binary) length of punched tape input determined by DCB buffer length specified in DOPEN. 1 If M=0 (ASCII) column 1 is for carriage control on the line printer. 1 If M=1 (binary) length of punched tape input determined by word count in first non-zero character read.
K	K-bit.
Bit (7:1)	0 = keyboard input not printed. 1 = keyboard input printed as received. 0 = If M=0 (ASCII) output printed not punched. 0 = If M=1 (binary) output punched not printed. 1 = If M=0 (ASCII) output punched not printed. 1 = If M=1 (binary) output punched not printed.

Remote File Access: DOPEN

X
Bit (5:1) X-bit.

0 = If M = 1 and V = 1, absolute binary tape format; any leader, special control characters, etc., are skipped.

1 = If M=1 and V=0 for input, number of words in word count read, leader not skipped; for output, record terminator not punched.

1 = If M=0 and V=0, ASCII tape format.

security integer by value (optional). Two character or binary security code of the file.

crn array (required during first open). A five word array where:

crn [0] cartridge number
crn [1]-[4] ASCII representation of the DS device class or logical unit number of DSLINE

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

See the appropriate Batch Spool monitor manual for more detailed information.

NOTE

The parameter optn for:

- A type 2 file must be modified in update mode.
- A type 3, and greater, file is in a non-update mode, and writes an EOF after each record.

Skips forward or backward a specified number of records.

```

      A      I      IV      IV      O-V
DPOSN ( DCB, error, record, flag );
```

Parameters

- DCB** array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.
- error** integer (required). Error codes returned as a result of remote file access.
- record** integer by value (required). The number of records to be skipped from the current file pointer position:
- + N number of records forward
 - N number of records backward
 - 0 no operation
- flag** integer by value (optional). The absolute or relative position of the record:
- 0 = record indicates the relative position. Default value.
 - 1 = record indicates absolute record number within the file.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

- CCE** indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter
- CCL** indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.
- CCG** indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

See the appropriate Batch Spool Monitor Reference Manual for more detailed information.

DPURG

Closes the DCB, if open, and deletes the named RTE file and all its extents.

A	I	A	IV	A	O-V
DPURG (DCB, error, name, security, crn);					

Parameters

DCB array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.

error integer (required). Error codes returned as a result of remote file access.

name array (required). RTE file name of up to six characters that identifies the file to be purged.

security integer by value (optional). Two character security code which must match security.

crn array (required if file not presently open to user). A five word array where:

crn [0] cartridge number

crn [1]-[4] ASCII representation of the DS device class or logical unit number of DSLINE.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

See the appropriate Batch Spool Monitor Reference Manual for detailed information.

DREAD

Reads from the RTE file currently open to DCB, to the user buffer.

```
      A      I      A      IV      I      IV      O-V
DREAD ( DCB, error, buffer, size, length, recnum );
```

Parameters

- DCB array (required). The Data Control Block (DCB), which is a four word array used for each RTE file opened. This is used by the system and should not be modified.
- error integer (required). The error codes returned from accessing remote file.
- buffer array (required). The buffer address to which data is read.
- size integer by value (required). A read request buffer length, that specifies the number of words transferred. The buffer is limited to 512 words.
- length integer (optional). Number of words transferred to buffer. If length = -1 an end of file was read.
- recnum integer by value (optional). Record number for random access type 1 and type 2 files:
- If recnum = 0 transfer starts at current pointer position (default).
 If recnum > 0 transfer starts at absolute record position.
 If recnum < 0 transfer starts at current position.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

- CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter
- CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.
- CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.
- See the appropriate Batch Spool monitor manual for more detailed information.

DSTAT

Returns information on all cartridge labels in the RTE system.

```
          A      I      A
DSTAT ( status, error, dsline );
```

Parameters

status array (required). A 125-word buffer for directory information.

 status [0] logical unit number (first disc).
 status [1] last track number.
 status [2] ID segment address of the locking program,
 if not locked.

error integer (required). Error codes returned as a result of remote file access.

dsline array (required). ASCII representation of the DS line name or logical unit number (4-word array).

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

 See the appropriate Batch Spool Monitor Reference Manual for more detailed information.

Rewinds RTE device type 0 files and resets RTE disc files to the beginning of the first record in the file.

```
      A      I  
DWIND ( DCB, error);
```

Parameters

DCB array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.

error integer (required). Error codes returned as a result of remote file access.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

See the appropriate Batch Spool Monitor Reference Manual for more detailed information.

DWRIT

Writes a record on the RTE file currently open to the DCB.

```
      A      I      A      IV      IV      O-V  
DWRIT( DCB, error, buffer, size, recnum );
```

Parameters

DCB array (required). The Data Control Block (DCB), a four word array used for each RTE file opened. This is used by the system and should not be modified.

error integer (required). Error codes returned as a result of remote file access.

buffer array (required). Address of a buffer from which data is written to record.

size integer by value. Length of buffer in words. The buffer is limited to 512 words.

recnum integer by value (optional). Record number for random access type 1 and type 2 files:

 If recnum = 0 transfer starts at current pointer position. The default value is zero.

 If recnum > 0 transfer starts at absolute record position.

 If recnum < 0 transfer starts at current position.

Error Conditions

Follow each call to an RFA intrinsic with a test of condition codes.

CCE indicates satisfactory completion. DCRET and DOPEN provide information in the error parameter

CCL indicates problems at the HP 3000 end. Refer to Appendix A for information on the error code.

CCG indicates problems at HP 1000 end. Refer to Appendix A for summary of the error codes.

 See the appropriate Batch Spool Monitor Reference Manual for more detailed information.

Example

The following SPL/3000 program shows how remote files can be accessed using the remote file intrinsic calls to pass files between the HP 3000 and the remote HP 1000.

```
$CONTROL USLINIT
$TITLE "GET A FILE FROM RTE"
BEGIN
COMMENT
```

```
THIS PROGRAM TRANSFERS ONE ASCII RTE FILE TO MPE.
BECAUSE THE RTE AND MPE EDITORS DIFFER IN THEIR
TEXT FORMATS, THE RTE FILE IS TRANSFERED AS
72-CHARACTER UNNUMBERED RECORDS.
```

```
RUN FROM MPE WITH
```

```
    :RUN GET
```

```
GET ASKS FOR THE NAME OF AN RTE FILE.  IF THE
FILE CANNOT BE OPENED, GET PROMPTS FOR A NEW
NAME.  ENTERING "/E" WILL TERMINATE GET.  AFTER
IT OPENS THE RTE FILE, GET ASKS FOR THE NAME OF
A NEW MPE FILE.  IF GET CANNOT CREATE THE
INDICATED FILE, IT PROMPTS FOR A NEW NAME UNTIL
IT CREATES THE FILE OR "/E" IS ENTERED.  THE
TRANSFER TAKES PLACE AFTER THE MPE FILE IS
CREATED.  WHEN IT IS DONE, GET PRINTS THE
NUMBER OF RECORDS TRANSFERED AND TERMINATES;
```

```
INTEGER FILENUM,          << MPE FILE NUMBER>>
    ERROR,                << ERROR CODE FOR FILE
                           INTRINSICS AND RFA CALLS>>
    I,                    << COUNTER >>
    NORECORDS := 0,       << NUMBER OF RECORDS
                           TRANSFERED >>
    LGTH;                 << RECORD LENGTH (WORDS)
                           >>
LOGICAL MOREDATA:=TRUE,  << MORE DATA ? >>
    OPEN;                 << FILE OPEN ? >>
```

```
LABEL LASTEND;
```

```
INTEGER ARRAY DCB (0:4), << DATA CONTROL BLOCK >>
    BUFFER (0:32),      << READ / WRITE
                           BUFFER >>
    FILENAME (0:13),   << RTE FILE NAME >>
    CRN (0:4);         << RTE CARTRIDGE NO
                           AND LINE NAME >>
```

```
<< MESSAGE BUFFER TO CONSOLE >>
BYTE ARRAY MSG (0:59); ARRAY MSF1 (*)=MSG;
```

Remote File Access Examples: SPL

```
<< USED FOR NUMBER CONVERSION >>
BYTE ARRAY NUMBER (0:5); ARRAY NUMBERI(*)=NUMBER;
```

```
<<EQUIVALENT BYTE ARRAYS >>
BYTE ARRAY FILENAMEB (*)= FILENAME;
```

```
<< FILE INTRINSICS >>
INTRINSIC FCLOSE, FOPEN, FWRITE;
```

```
<< SYSTEM INTRINSICS >>
INTRINSIC PRINT'FILE'INFO, PRINT, READ, ASCII,
BINARY;
```

```
<< RFA PROCEDURES >>
INTRINSIC (D3KINTRN) DCLOS, DOPEN, DREAD;
```

```
PROCEDURE REPORTRTE;
```

```
BEGIN << REPORT ERROR FROM RTE RFA CALL >>
MOVE MSG := "*** RFA ERROR ";
ASCII(ERROR, 10, MSG(14));
PRINT(MSG1,-20,%40);
END;
```

```
PROCEDURE REPORTMPE;
BEGIN <<REPORT ERROR FROM MPE FILE CALL >>
PRINT'FILE'INFO(FILENUM);
END;
```

```
<<BEGINNING OF MAIN PROGRAM>>
```

```
<< OPEN RTE FILE >>
OPEN := FALSE;
WHILE NOT OPEN DO
  BEGIN
    MOVE MSG := "/GET: NAME OF RTE FILE:";
    PRINT(MSG1,-24,%320);
    MOVE FILENAMEB:=28(" ");
    READ(FILENAME,-6);
    IF FILENAME(0)="/E" THEN GO TO LASTEND;
    MOVE MSG :="/GET: CARTRIDGE NUMBER:";
    PRINT(MSG1,-23,%320);
    MOVE NUMBER :=6(" ");
    READ(NUMBERI,-6);
    CRN(0) := BINARY (NUMBER,6);
    MOVE MSG := "/GET: WHAT IS THE DS LINE NAME?";
    PRINT( MSG1, -31, %320);
    MOVE CRN(1):= 4(" ");
    READ(CRN(1),-8);
    DOPEN(DCB(0),ERROR,FILENAME(0),%160,0,CRN);
    IF < OR ERROR < 0 THEN
      REPORTRTE
    ELSE
      OPEN:=TRUE;
  END;
```

```

<<CREATE MPE FILE>>
OPEN := FALSE;
WHILE NOT DO
  BEGIN
    MOVE MSG := "/GET: NAME OF MPE FILE:";
    PRINT(MSG1, -23, %320);
    MOVE FILENAMEB := 28(" ");
    READ(FILENAME, -28);
    IF FILENAME(0) := "/E" THEN GO TO LASTEND;
    FILENAME := FOPEN(FILENAMEB, 4, %344, 36);
    IF < THEN
      REPORTMPE
    ELSE
      OPEN := TRUE;
  END;

```

```

<<TRANSFER DATA>>
MOREDATA := TRUE;
WHILE MOREDATA DO
  BEGIN
    DREAD(DCB, ERROR, BUFFER(1), 36, LGTH);
    IF < OR ERROR < 0 THEN
      BEGIN
        REPORTRTE;
        MOREDATA := FALSE;
      END
    ELSE IF LGTH=-1 THEN
      MOREDATA := FALSE
    ELSE
      BEGIN
        FOR I:=LGTH+1 UNTIL 35 DO
          BUFFER(I) := " ";
          FWRITE(FILENUM, BUFFER(1), 36, 0);
          IF <> THEN
            BEGIN
              REPORTMPE;
              MOREDATA := FALSE;
            END
          ELSE
            NORECORDS := NORECORDS +1;
        END;
      END;
  END;
LASTEND: DCLOSE( DCB(0), ERROR);
FCLOSE(FILENUM, %11,0);
MOVE MSG:="/GET: RECORDS TRANSFERRED.");
ASCII(NORECORDS, 10, MSG(6));
PRINT(MSG1, -33, %40);
END.

```

Remote File Access Examples: FORTRAN

```
$CONTROL LABEL, LIST, MAP, CROSSREF, SOURCE, INIT
$CONTROL LOCATION
PROGRAM RFATR
C THIS PROGRAM CREATES A NEW FILE ON THE RTE.
C IT PROMPTS FOR INPUT, AND WRITES EACH RECORD
C INTO THE NEW FILE USING RFA INTRINSICS.
C UPON COMPLETION IT DISPLAYS THE CONTENTS,
C AND CLOSES, AND TRUNCATES THE FILE.
C
C
C     INTEGER BLOCKNUM, IERR, ISIZE, LENGTH
C     INTEGER OPTN, RECNM, SECURITY
C     INTEGER TRUNCATE, TYPE
C
C     LOGICAL BUFFER(40), DCB(4), NAME(3), SIZE(2)
C     LOGICAL CRTNM(5)
C
C     CHARACTER ANSW(6), BUFBYET(72), BUFCHAR*72
C     CHARACTER CRTBYTE(8)
C     INTEGER BUFINTG(2)
C     LOGICAL BUFPRNT(40)
C
C     EQUIVALENCE (BUFFER,BUFBYTE)
C     EQUIVALENCE (BUFFER,BUFINTEG)
C     EQUIVALENCE (BUFFER,BUFCHAR)
C     EQUIVALENCE (CRTNM(2),CRTBYTE)
C
C     SYSTEM INTRINSIC DCLOS, DCRET, DLOC, DOPEN
C     SYSTEM INTRINSIC DPURG, DREAD, DWRT
C     SYSTEM INTRINSIC FCHECK, FCLOSE, FOPEN
C     SYSTEM INTRINSIC FREAD
C
C ***** MAIN PROGRAM *****
C
C     WRITE(6,102)
C 102 FORMAT(" BUILD TR FILE ON REMOTE RTE.")
C     WRITE(6,112)
C 112 FORMAT(/,"+DSL LINE NUMBER = ? ")
C     DO 115 I=2,5
C 115 CRTNM(I)=%200400L
C     READ(5,*)CRTBYTE
C
C 120 WRITE(6,122)
C 122 FORMAT("+SPECIFY NEW FILE NAME = ? ")
C     DO 128 I=1,3
C 128 NAME(I)=%20040L
C     READ(5,124)NAME
C 124 FORMAT(3A2)
C
C     WRITE(6,126)
C 126 FORMAT("+SPECIFY FILE SECURITY CODE ",
C     *"(DEFAULT = 0) = ?")
C     READ(5,*)SECURITY
C
C     WRITE(6,134)
```

Remote File Access Examples: FORTRAN

```

134 FORMAT("+SPECIFY CARTRIDGE NUMBER = ?")
      READ(5,*)CRTNM(1)
C
C PREPARE TO DCRET
C
      TYPE = 4
C
      SIZE(1)=%100L
C
      CALL DCRET(DCB,IERR,NAME,SIZE,TYPE,SECURITY,
      *CRTNM)
      IF (.CC.) 1240,209,1250
209 IF (IERR) 1260,300,300
C
C OPEN TERMINAL AS FILE
C
300 IFN=FOPEN("$STDINX")
      IF(.CC.)309,305,309
305 WRITE(6,306)
306 FORMAT(" FOPEN $STDINX FAILED, CCL)
      CALL DEBUG
      STOP
C
C FILE NOW CREATED
C
309 LENGTH = 36
310 WRITE(6,312)
312 FORMAT(" WRITE COMMANDS, TERMINATE WITH "
      *"'STOP' ",/)
C
C LOOP
C
400 DO 410 I=1,36
410 BUFFER(I)-%200400L
C
C USE FREAD TO ALLOW ':' AND ','
C
      ICNT=FREAD(IFN,BUFFER,36)
      IF(.CC.) 1050,420,600
C
C TEST FOR '??'
C
420 IF(BUFINTG(1).EQ.%37477) GO TO 1100
C
C TEST FOR 'STOP'
C
      IF((BUFINTG(1).EQ.%51524).AND.(BUFINTG(2)
      *.EQ.%47520)) GO TO 600
C
      CALL DWGIT(DCB,IERR,BUFFER,ICNT)
      IF(.CC.) 1270,430,1280
C
430 IF (IERR) 1000,400,100
C
C END OF LOOP

```


Remote File Access Examples: FORTRAN

```
C
600 WRITE(6,606)
606 FORMAT(" GOT TO SHUT DOWN SEQUENCE")
C
      CALL DCLOS(DCB, IERR)
      IF(.CC.) 1300,620,1310
620 IF(IERR) 1200,675,675
C
C DCLOS COMPLETED.  CLOSE TERMINAL.
C
675 CALL FCLOSE(IFN,0,0)
      IF(.CC.)690,700,700
C
690 WRITE(6,692)
692 FORMAT(" FCLOSE FAILED, CCL)
      STOP
C
C READ, VERIFY, AND TRUNCATE FILE
C OPEN OPTN=2 TO ALLOW TRUNCATION
C
700 OPTN=2
      CALL DOPEN(DCB, IERR, NAME, OPTN, SECURITY,
      *CRTNM)
      IF(.CC.) 340,705,1350
705 IF (IERR)1360,1370,730
C
730 WRITE(6,732)IERR
732 FORMAT(" FILE TYPE = ",I6)
C
C VERIFY LOOP
C
      ISIZE=36
750 CALL DREAD(DCB, IERR, BUFFER, ISIZE)
      IF(.CC.) 1380, 755, 1390
C
755 IF(ISIZE.EQ.0) GO TO 800
      IF(IERR.EQ.-12) GO TO 800
      IF(IERR)1220,760,760
C
760 BUFPRNT(1)=%20040L
      DO 758 I=1,39
758 BUFPRNT(I+1)=BUFFER(I)
C
      WRITE(6,762)BUFPRNT
762 FORMAT(40A2)
C
      DO 766 I=1,40
766 BUFFER(I)=%20040L
      GO TO 750
C
C END VERIFY LOOP
C
C ENCOUNTERED END OF FILE
C
800 CALL DLOC( DCB, IERR, RECNM, BLOCKNUM)
```

```

      IF(.CC.) 1320, 803, 1330
C
  803 TRUNCATE=64-(BLOCKNUM+1)
      CALL DCLOSE( DCB, IERR, TRUNCATE)
      IF(.CC.) 1300, 805, 1310
  805 IF(IERR) 1230, 820, 820
C
  820 WRITE(6,822)
  822 FORMAT(" PROGRAM COMPLETED")
      STOP
C
C *** ERROR CONDITIONS ***
C
C DWRITE ERRORS
C
  1000 WRITE(6,1002)IERR
  1002 FORMAT(" DWRITE ERRORS, IERR = ",I6)
      GO TO 600
C
  1050 CALL FCHECK(IFN, IERR)
      WRITE(6,1052)IERR
  1052 FORMAT(" BAD READ FROM TERMINAL, IERR = ",
             *I6)
      GO TO 310
C
C
C
  1100 WRITE(6,1110)
  1110 FORMAT(" GO TO ?? PART ")
      WRITE(6,1120)
  1120 FORMAT(" THIS PROGRAM CREATES A FILE ",
             *"ON THE RTE.")
      WRITE(6,1130)
  1130 FORMAT(" IT WRITES THE COMMANDS, THEN",
             *" CLOSES THE FILE.")
      GO TO 310
C
C
  1200 WRITE(6,1202)IERR
  1202 FORMAT(" CLOSE ERROR, IERR = ",I6)
      STOP
C
  1220 WRITE(6,1222)IERR
  1222 FORMAT(" DREAD ERROR, IERR = ",I6)
      STOP
C
  1230 WRITE(6,1232)IERR
  1232 FORMAT(" DCLOS ERROR, IERR = ",I6)
      STOP
C
  1240 WRITE(6,1242)
  1242 FORMAT(" CCL ON DCRET")
      STOP
C
  1250 WRITE(6,1252)IERR

```

Remote File Access Examples: FORTRAN

```
1252 FORMAT(" CCG ON DCRET, IERR = ",I6)
      STOP
C
C DCRET ERROR
C
1260 IF(IERR.EQ.-2) GO TO 1400
      WRITE(6,1262)IERR
1262 FORMAT(" DCRET ERROR, IERR = ",I6)
      STOP
C
1270 WRITE(6,1272)
1272 FORMAT(" CCL ON DPURG")
      STOP
C
1280 WRITE(6,1282)IERR
1282 FORMAT(" CCG ON DPURG, IERR = ",I6)
      STOP
1290 WRITE(6,1292)IERR
1292 FORMAT(" DPURG FAILED, IERR = ",I6)
      STOP
C
1300 WRITE(6,1302)
1302 FORMAT(" CCL ON DCLOS")
      STOP
C
1310 WRITE(6,1312)IERR
1312 FORMAT(" CCG ON DCLOS, IERR = ",I6)
      STOP
C
1320 WRITE(6,1322)
1322 FORMAT(" CCL ON DLOCF")
      STOP
C
1330 WRITE(6,1332)IERR
1332 FORMAT(" CCG ON DLOCF, IERR = ",I6)
      STOP
C
1340 WRITE(6,1342)
1342 FORMAT(" CCL ON DOPEN")
      STOP
C
1350 WRITE(6,1352)IERR
1352 FORMAT(" CCG ON DOPEN, IERR = ",I6)
      STOP
C
1360 WRITE(6,1362)IERR
1362 FORMAT(" FAILED TO DOPEN FILE, IERR = ",I6)
      STOP
C
1370 WRITE(6,1372)
1372 FORMAT(" FAILED TO DOPEN FILE, IERR = 0")
      STOP
C
1380 WRITE(6,1382)
1382 FORMAT(" CCL ON DREAD")
```

```
      STOP
C
1390 WRITE(6,1392)IERR
1392 FORMAT(" CCG ON DREAD, IERR = ",I6)
      STOP
C
C DUPLICATE FILE NAME IN DCRET
C
1400 WRITE(6,1402)
1402 FORMAT("+DUPLICATE FILE NAME, WISH TO ",
*" PURGE (Y/N) = ?")
      READ(5,*)ANSW(1)
      IF(ANSW(1).EQ."N") GO TO 120
      IF(ANSW(1).EQ."Y") GO TO 1410
      GO TO 1400
C
1410 CALL DPURG(DCB, IER, NAM, SECURITY, CRTNM)
      IF (.CC.) 1270, 1412, 1280
C
1412 IF(IERR) 1290, 1420, 1420
C
1420 CALL DCRET(DCB, IERR, NAME, SIZE, TYPE,
*SECURITY, CRTNM)
      IF(.CC.) 1240, 1430, 1250
C
1430 IF(IERR) 1260,300,300
C
      END
```



PROGRAM-TO-PROGRAM COMMUNICATIONS

SECTION

VI

The preceding chapters describe how to establish communications links between HP 3000 and HP 1000 computers in order to form a communications network, and how to execute programs in any HP 3000 or HP 1000 from a single log-on terminal. Furthermore, the programs running within any HP 3000 or HP 1000 in the DS network can, under the proper circumstances, access the hardware or software resources available at either end.

However, for most remote processing applications it is essential that separate user programs be able to run simultaneously in separate computers within the network, and be able to communicate efficiently with one another. The DS/3000 program-to-program (PTOP) intrinsics described in this chapter provide that capability.

The structure of two programs that are communicating with one another in this manner is not symmetrical. One of them, referred to as the master program, is always in control and is the one that initiates all activity between the two programs. The other, referred to as the slave program, is always responding to requests received from the master. Those intrinsics used within a master program are summarized in Table 6-1 and those used within a slave program are summarized in Table 6-2.

The master program can reside on either the HP 3000 or HP 1000; the slave program resides on the other system. The program, either slave or master, residing on the HP 3000 system can be written in SPL or FORTRAN. The program, slave or master, residing on the HP 1000 system can be written in FORTRAN, PASCAL or assembly language. The calling sequences of HP 1000 programs are described in the DS/1000-IV Programmer's Reference Manual.

Table 6-1. Master Program-to-Program Intrinsic

Intrinsic Name	Function
POPEN	Initiates and activates a slave process in a remote system using a specified remote program file.
PREAD	Sends a request to the remote slave program asking the slave to send a block of data back to the master.
PWRITE	Sends a block of data to the remote slave program.
PCONTROL	Transmits a tag field containing user-defined control information to the remote slave program and receives a tag field back from the slave.
PCLOSE	Terminates (kills) the remote slave process.
PCHECK	Returns an integer code specifying the completion status of the most recently executed master program-to-program intrinsic.

The master program-to-program intrinsics are very similar to the MPE process handling and file system intrinsics that are used for process-to-process communication within a single-system environment.

Table 6-2. Slave Program-to-Program Intrinsic

Intrinsic Name	Function
GET	Receives the next master request to be received by the remote slave program.
ACCEPT	Accepts and completes the request received by the preceding GET intrinsic call.
REJECT	Rejects the request received by the preceding GET intrinsic call.
PCHECK	Returns an integer code specifying the completion status of the most recently executed slave program-to-program intrinsic.

The HP 1000 program-to-program calls perform the same functions as their HP 3000 counterparts, but may have slightly different names of parameters. Refer to the DS/1000-IV Programmers Reference Manual for details.

Once a DS communications link exists between an HP 3000 and HP 1000, a user program (the master program) can create and activate a slave program. The POPEN intrinsic performs this function in place of the standard MPE CREATE and ACTIVATE intrinsics.

When the master and slave program are both executing the master program can send data (PWRITE), or control information (PCONTROL) directly to the slave program, or send a request (PREAD), or control request (PCONTROL) to the slave program asking that the slave send data and/or control information back to the master.

PTOP Communications

In the following descriptions of program-to-program intrinsics, superscripts are used to describe the types of parameters and whether they must be passed by value. The default is by reference. The superscripts have the following meanings:

Superscript:	Meaning:
BA	byte array
BP	byte pointer
D	double
DA	double array
DV	double by value
I	integer
IA	integer array
IV	integer by value
L	logical
LA	logical array
LV	logical by value
O-V	option variable
R	real

POPEN

Initiates program-to-program communication with a remote slave program.

```
I           BA           BA           IA           O-V
dsnum := POPEN( dsdevice, progname, itag );
```

The POPEN intrinsic activates a slave program in the remote system using a specified remote program, progname, and optionally transmits a tag field, itag, to that remote slave program.

The remote slave program must issue a GET intrinsic call followed by either an ACCEPT or REJECT call to complete the POPEN operation.

The remote slave program may transmit a tag field back to the master program as part of an ACCEPT or REJECT call. If the master program transmitted a tag field, then the returned tag field, if any, is available in itag. If the master program did not transmit a tag field, then the returned tag field, if any, is not accessible.

The remote slave program remains activated and both the communications link and the DS buffer remain intact even if the POPEN request is rejected by the remote slave program. The meaning of a POPEN reject by the remote slave program must be established as part of the user application design.

Functional Return

When the POPEN intrinsic is executed, it returns to the master program a number, dsnum, by which DS uniquely identifies the particular communications link.

Parameters

dsdevice byte array (required). Contains a string of ASCII characters terminated by a space. This string must be the device class name or logical device number used in the DSLINE or REMOTE HELLO command that opened the communications line you will be using.

progname byte array (required). Contains a string of up to five ASCII characters that names a remote slave program.

PTOP Communications: POPEN

itag integer array (optional). A 20-word array used for transmitting and receiving tag fields. The format of the tag field is defined as part of the user application. Default: A tag field of all zeros is sent; the returned tag field, if any, is not available to the master program.

Condition Codes

- CCE Request accepted by remote slave program.
- CCG Request rejected by remote slave program.
- CCL Request denied; an error occurred. Issue a PCHECK intrinsic call to determine what happened.

The POPEN activity described above is illustrated in figure 6-1

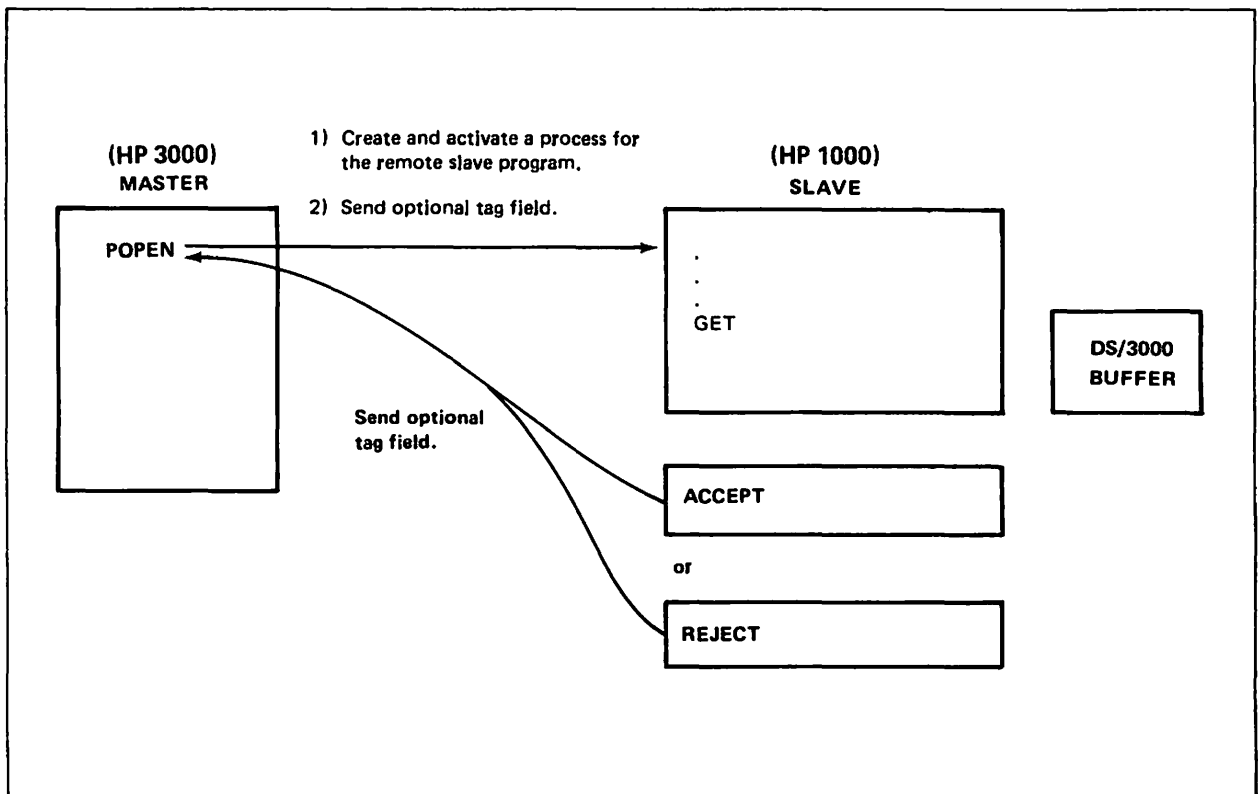


Figure 6-1. POPEN Activity

PREAD

Asks the remote slave program to send a block of data.

```
I           IV   IA   IV   IA   0-V
lgth := PREAD( dsnum, target, tcount, itag );
```

The PREAD intrinsic transmits a read request to the remote slave program and optionally transmits a tag field from itag to the remote slave program. The remote slave program must issue a GET intrinsic call followed by either an ACCEPT or REJECT call to complete the PREAD operation.

The ACCEPT call moves the requested block of data from the user's buffer in the remote system to target in the master program.

The REJECT call transmits no data, other than an optional tag field.

The remote slave program may transmit a tag field back to the master program as part of the ACCEPT or REJECT call. If the master program transmitted a tag field, then the returned tag field, if any, is available in itag. If the master program did not transmit a tag field, then the returned tag field, if any, is not accessible.

Functional Return

When the PREAD intrinsic is executed, it returns to the master program a number, lgth, specifying how many words of data were actually received into target.

Parameters

dsnum integer by value (required). The link identifier returned by the particular POPEN intrinsic call which initiated communication with the remote slave program.

target integer array (required). The array into which data received from the remote slave program will be deposited.

tcount integer by value (required). The positive number of words of data, up to 512, or the negative number of bytes.

itag integer array (optional). A 20-word array used for transmitting and receiving a tag field. The format of the tag field is defined by the master and slave programs of the user.

Condition Codes

- CCE Request accepted by the remote slave program.
- CCG Request denied by the remote slave program.
- CCL Request denied; an error occurred. Issue a PCHECK intrinsic call to determine what happened.

The PREAD activity is illustrated in figure 6-2.

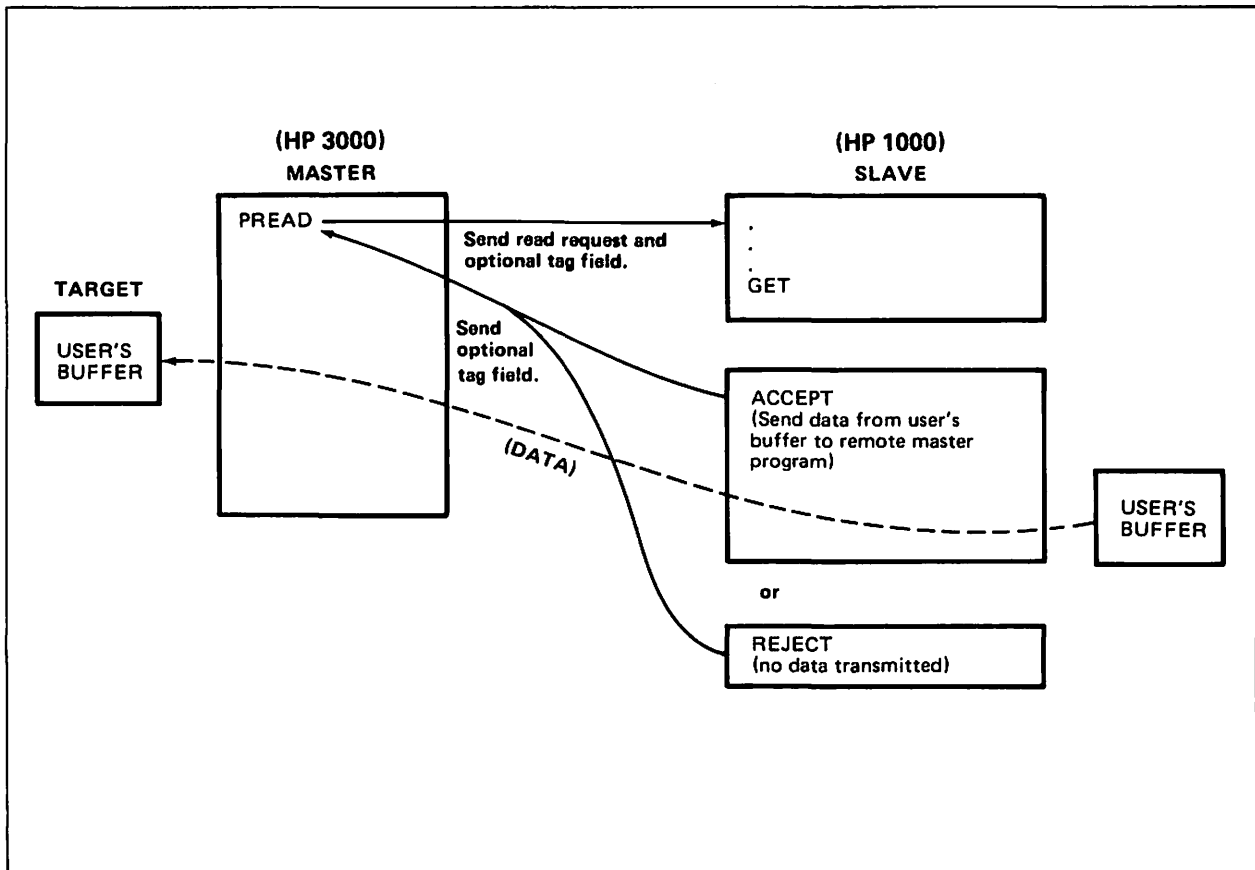


Figure 6-2. PREAD Activity

PWRITE

Sends a block of data to the remote slave program.

```
IV    IA    IV    IA    O-V
PWRITE( dsnun, target, tcount, itag );
```

The PWRITE intrinsic transmits a block of data from target to the remote slave program, and optionally transmits a tag field, itag, to the remote slave program. The number of words in the block is specified in tcount. The remote slave program must issue a GET intrinsic call followed by either an ACCEPT or REJECT call to complete the PWRITE operation.

The ACCEPT call moves the block of data from the slave program to the buffer of the user in the local HP 3000.

The REJECT call refuses the write request.

The remote slave program may transmit a tag field back to the master program as part of the ACCEPT or REJECT call. If the master program transmitted a tag field, then the returned tag field, if any, is available in itag. If the master program did not transmit a tag field, then any returned tag field is not accessible.

Parameters

- dsnun** integer by value (required). The link identifier returned by the particular POPEN intrinsic call which initiated communication with the remote slave program.
- target** integer array (required). The array from which data is transmitted to the remote slave program.
- tcount** integer by value (required). The positive number of words of data to be transmitted (up to 512) or the negative number of bytes.
- itag** integer array (optional). A twenty-word array used for transmitting and receiving a tag field. The format of the tag field is defined by the master and slave programs of the user.

Condition Codes

- CCE** Request accepted by remote slave program.
- CCG** Request denied by remote slave program.
- CCL** Request denied; an error occurred. Issue a PCHECK intrinsic call to determine what happened.

PTOP Communications: PWRITE

The PWRITE activity is illustrated in figure 6-3.

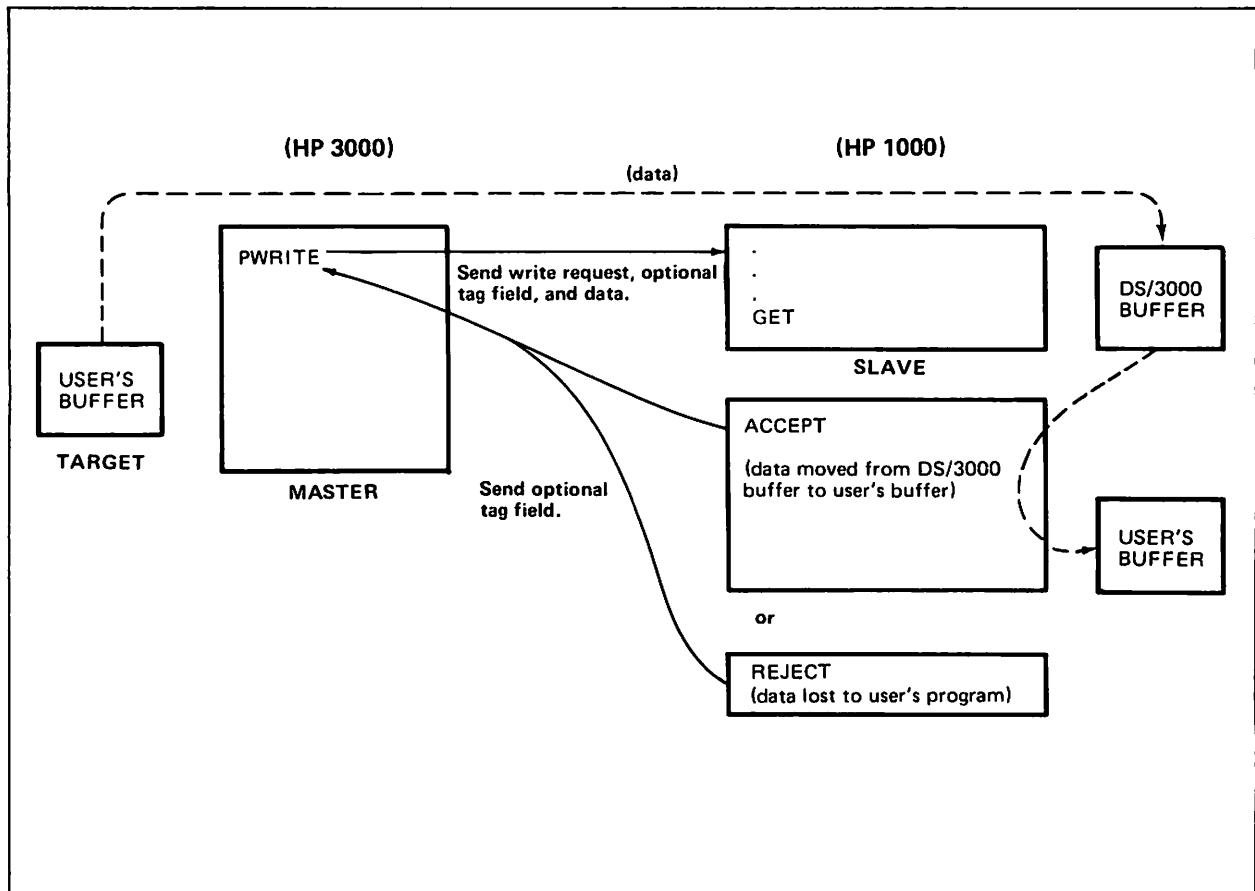


Figure 6-3. PWRITE activity.

PCONTROL

Exchanges tag fields with the remote slave program.

```
IV   IA   O-V  
PCONTROL( dsnum, itag );
```

The PCONTROL intrinsic optionally transmits a tag field to the remote slave program and accepts one in return. The remote slave program must issue a GET intrinsic call followed by either an ACCEPT or REJECT call to complete the PCONTROL operation. Both the ACCEPT and REJECT calls transmit a tag field back to the master program, available in itag.

Although this intrinsic was designed specifically for the exchanging of tag fields, you will notice that itag is an optional parameter. If the master program did not transmit a tag field, then the returned tag field, if any, is not accessible.

Parameters

dsnum integer by value (required). The link identifier returned by the particular POPEN intrinsic call which initiated communication with the remote slave program.

itag integer array (optional). A 20-word array used for transmitting and receiving a tag field. The format of the tag field is defined by the master and slave programs of the user.

Condition Codes

CCE Request accepted by remote slave program.

CCG Request denied by remote slave program.

CCL Request denied; an error occurred. Issue a PCHECK intrinsic call to determine what happened.

The PCONTROL activity is described in figure 6-4.

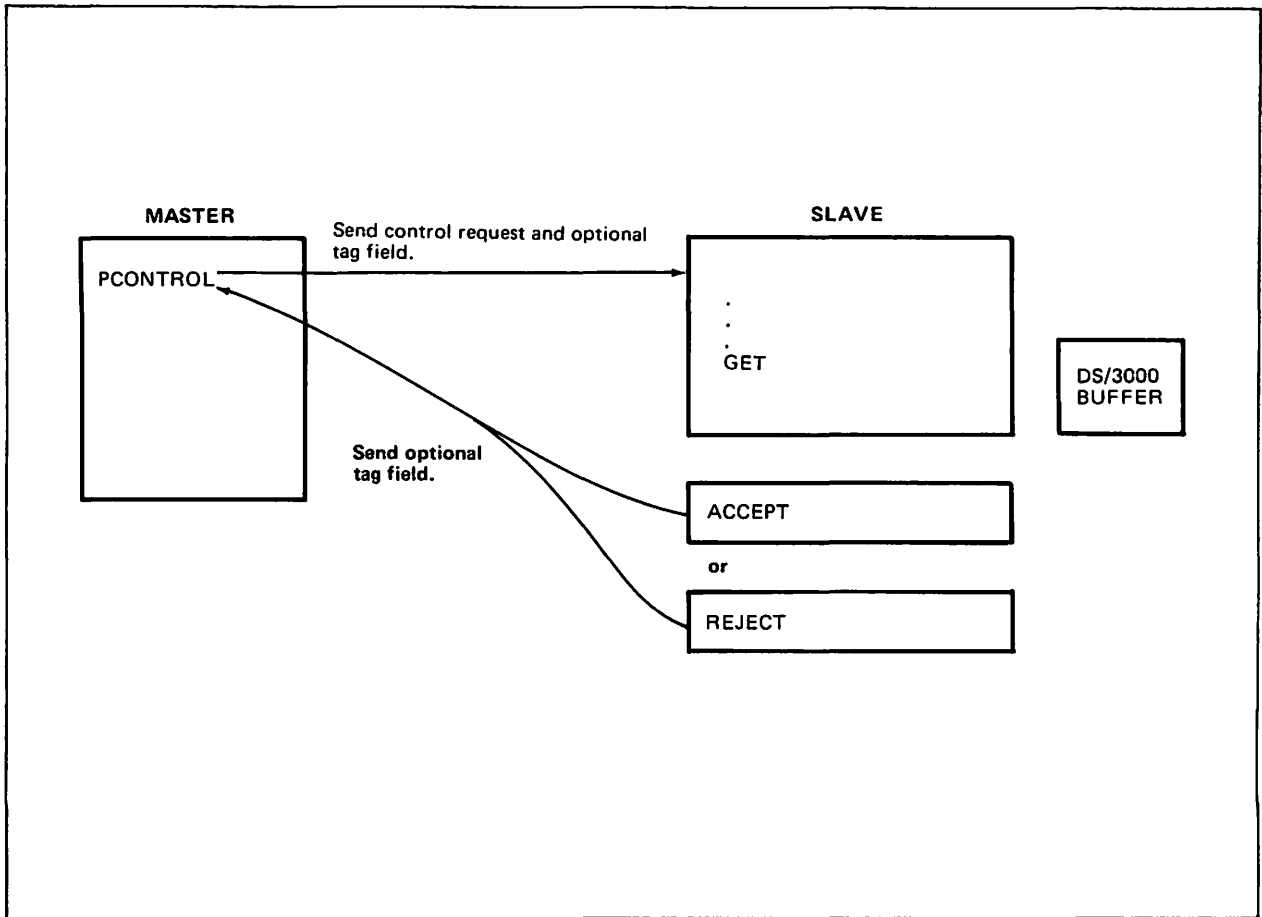


Figure 6-4. PCONTROL Activity.

PCLOSE

Terminates program-to-program communication with a remote slave program.

```
IV
PCLOSE( dsnum );
```

The PCLOSE intrinsic terminates the remote slave program associated with dsnum. The particular communications line itself remains open.

Parameter

dsnum integer by value (required). The line number returned by the particular POPEN intrinsic call which initiated communication with the remote slave program.

Condition Codes

CCE Successful completion.

CCG Not returned.

CCL Request denied; an error occurred. Issue a PCHECK intrinsic call to determine what happened.

NOTE

Although a master program normally signals the slave program to terminate by generating a code in the tag field, PCLOSE can be used to terminate the slave program unconditionally. However, make certain the HP 1000 slave program is not communicating with other master programs, since PCLOSE turns off the slave to other master programs also.

GET

Receives the next request from the remote master program.

```
I      IA  I      I      O-V
ifun := GET( itag, il, ionumber );
```

The GET intrinsic receives the next request from the remote master program and accepts an optional tag field, available in itag.

Functional Return

When the GET intrinsic is executed, it returns to the slave program a number, ifun, specifying the type of request received from the remote master program:

<u>ifun</u> :	Meaning:
0	An error occurred. This value is returned only when the condition code CCL is also returned. Issue a PCHECK intrinsic call with a <u>dsnum</u> parameter of zero to determine what happened.
1	POPEN request received.
2	PREAD request received.
3	PWRITE request received.
4	PCONTROL request received.
5	This value is returned only when the condition code CCG is returned. It indicates that a pending MPE File System I/O-without-wait request was completed instead of a DS remote I/O request. The parameter <u>ionumber</u> contains the file number associated with the completed I/O request.

The GET intrinsic call implicitly issue as an IOWAIT(0) intrinsic call. The parameter ifun value of 0 indicates that an IOWAIT error occurred. A ifun value of 5 occurs only if you are executing MPE File System intrinsic calls without wait in your program and the implicit IOWAIT(0) call completes a pending File System I/O request instead of the expected DS remote I/O request. In this case you will have to issue another GET call after processing the completed File System I/O request in order to receive the expected DS remote I/O request.

Parameters

itag integer array (optional). A 20-word array used for receiving a tag field. The format of the tag field is defined by the master and slave programs of the user.

il integer (optional). A word that has meaning only when a PREAD or PWRITE request is received from the remote master program.

For a PREAD request, il contains an integer specifying the number of words requested by the remote master program.

For a PWRITE request, il contains an integer specifying the number of words transmitted from the remote master program to the DS buffer.

ionumber integer (optional). A word that has meaning only when the condition code CCG and an ifun of 5 are returned. In that case ionumber contains the MPE File System number associated with the completed I/O without wait request. The default is that no file number is returned.

Condition Codes

CCE Request received successfully.

CCG The implicit IOWAIT(0) call issued by the GET intrinsic completed a pending MPE File System I/O without wait request instead of a DS remote I/O request. The parameter ionumber contains the file number associated with the completed file system request.

CCL An error occurred. Issue a PCHECK intrinsic call to determine what happened.

NOTE

You must not use IOWAIT(0) calls within a program containing DS GET calls. If you were to use an IOWAIT(0) call and it responded to a DS remote I/O request, your program would not be able to make any sense out of the information returned by the IOWAIT call.

ACCEPT

Accepts and completes the request received by the preceding GET intrinsic call and returns an optional tag field back to the remote master program.

```
          IA      IA      IV      O-V  
ACCEPT( itag, target, tcount );
```

The ACCEPT intrinsic accepts the request received by the most recent GET intrinsic call, completes the requested operation, and transmits an optional tag field back to the remote master program.

In the case of a POPEN request, the ACCEPT call transmits an optional tag field, itag, to the remote master program.

In the case of a PREAD request, the ACCEPT call transmits the specified number of words, tcount, from target to the remote master program and transmits an optional tag field itag to the remote master program.

In the case of a PWRITE request, the ACCEPT call moves the specified number of words, tcount, from the DS buffer to target and transmits an optional tag field, itag, to the remote master program.

In the case of a PCONTROL request, the ACCEPT call transmits an optional tag field, itag, to the remote master program.

Parameters

itag integer array (optional). A 20-word array used for transmitting a tag field. The format of the tag field is defined by the master and slave programs of the user.

target integer array (optional). An array for transmitting or receiving blocks of data.

For PREAD requests, this array contains the block of data to the remote master program.

For PWRITE requests, this array receives the block of data from the DS buffer.

For POPEN and PCONTROL requests, this array has no meaning and should be omitted.

tcount integer by value (optional). An integer specifying the number of words to be transmitted or received.

For PREAD requests, this parameter specifies how many words of data are to be transmitted from target to the remote master program.

For PWRITE requests, this parameter specifies how many words of data are to be moved from the DS buffer to target.

For POPEN and PCONTROL requests, this parameter has no meaning and should be omitted.

Condition Codes

CCE Request completed successfully.

CCG Not returned.

CCL An error occurred. Issue a PCHECK intrinsic call to determine what happened.

REJECT

Rejects the request received by the preceding GET intrinsic call and returns an optional tag field back to the remote master program.

```
IA    0-V  
REJECT( itag );
```

The REJECT intrinsic rejects the request received by the most recent GET intrinsic call and transmits an optional tag field, itag, back to the remote master program.

Parameters

itag integer array (optional). A 20-word array used for transmitting a tag field. The format of the tag field is defined by the master and slave programs of the user.

Condition Codes

CCE Response transmitted successfully to the remote master program.

CCG Not returned.

CCL An error occurred. Issue a PCHECK intrinsic call to determine what happened.

PCHECK

Returns an integer code specifying the completion status of the most recently executed program-to-program intrinsic.

```
I           IV
icode:=PCHECK( dsnum )
```

The PCHECK intrinsic can be used in both master and slave programs. It returns an integer value that specifies the completion status of the most recently executed DS program-to-program intrinsic.

Functional Return

When the PCHECK intrinsic is executed, it returns to the calling program a number, icode, that specifies the completion status of the most recently executed DS program-to-program intrinsic. The values of icode and the associated meanings are described in Appendix A.

Parameters

dsnum integer by value (required)

In a master program: The link identifier returned by the particular POPEN intrinsic that initiated communication with the remote slave program.

In a slave program: 0 (zero).

Condition Codes

CCE PCHECK request successfully completed.

CCG Not returned.

CCL PCHECK request denied because dsnum was invalid.

EXAMPLES

Following are examples of master program and slave program written in SPL for the HP 3000. These programs do not interact with each other; they are intended to interact with counterparts on the HP 1000 system.

Performance Considerations

When designing applications, PTOP solutions may provide greater throughput compared to RFA because of the ability to pack data into the buffer. The communications system overhead is comparable for a PTOP transfer and an RFA transfer, but packing records into the PTOP buffer may provide greater throughput. The actual size of the PTOP transfer depends upon the buffer size library used in the HP 1000 system, along with the configured line size of the communications link as set by the DSCONTROL command. The three HP 1000 libraries allow data buffers of approximately 256, 1024, and 4096 words.

When a 3000 master communicates with a 1000 slave, the maximum total data buffer is either 512, 1024 or 4096 words, depending upon which HP 1000 library is used. When a 1000 master communicates with a 3000 slave, it can transfer up to 4096 words.

PTOP Communications: EXAMPLES

Master Program on HP 3000

```
$CONTROL USLINIT
$TITLE "*** MASTER PROGRAM FOR RTE/MPE P-TO-P TEST"
BEGIN
COMMENT

    THIS MASTER PROGRAM, WHICH RUNS FROM MPE, DIRECTS
    AN RTE SLAVE.
    THE PROGRAM
        READS 500 WORDS
        WRITES 500 WORDS
        CONTROLS SLAVE TURN OFF BY SPECIFYING
        TAG(10)=0;

INTEGER DSNUM, ERROR, I, LENGTH;

LOGICAL OK;

BYTE ARRAY DSLINE(0:7), SLAVENAME(0:28)=29(" ");

INTEGER ARRAY    TAG(0:20),
                 MASTERNAME(0:4),
                 READTEXT(1:1000),
                 READBUFR(1:1000),
                 WRITETEXT(1:1000);

<<MESSAGE BUFFER>>
BYTE ARRAY MSG(0:59);    ARRAY MSG1(*)=MSG;

<<EQUIVALENCED ARRAYS>>
INTEGER ARRAY DSLINEI(*)=DSLIME;
INTEGER ARRAY SNAMEI(*)=SLAVENAME;

<<SYSTEM INTRINSICS>>
INTRINSIC ASCII, READ, PRINT;

<<DS/3000 P-TO-P INTRINSICS>>
INTRINSIC (D3KINTRN) POPEN, PREAD, PWRITE, PCONTROL,
PCHECK, PCLOSE;

PROCEDURE ERRORCHECK;
IF > THEN
    BEGIN
        MOVE MSG := "SLAVE REJECTED THE OPERATION";
        PRINT(MSG1,-29,%40);
        OK:=FALSE;
    END
ELSE IF < THEN
    BEGIN
        ERROR:=PCHECK(DSNUM);
        MOVE MSG:="P-TO-P ERROR           ";
        ASCII(ERROR,10,MSG(13));
        PRINT(MSG1,-18,%30);
        OK:=FALSE;
    END
```

```

        END
ELSE
    OK:=TRUE;

PROCEDURE FIXTAG;
    BEGIN
        MOVE MSG:="TAG:";
        MOVE MSG(4):=42(" ");
        FOR I:=1 UNTIL 6 DO
            ASCII(TAG(I),10,MSG(7*(I-1)+5));
        PRINT MSG1,-46,%60);
        MOVE TAG:=11(-1);
        MOVE TAG(1):=MASTERNAME,(4);
    END;

<< BEGINNING OF MAIN PROGRAM >>

<< GET DS LINE NAME >>
MOVE MSG:="WHAT IS THE DS LINE NAME? ";
PRINT(MSG1,-31,%320);
MOVE DSLINEI := 4(" ");
READ (DSLINEI,-8);

<<INITIALIZE READ/WRITE TEXT>>
FOR I:=1 UNTIL 1000 DO
    BEGIN
        READTEXT(I):=2*I;
        WRITETEXT(I):=3*I;
    END;

<<GET MASTER NAME>>
MOVE MSG:="WHAT IS THE MASTER'S NAME? ";
PRINT(MSG1,-27,%320);
MOVE MASTERNAME:=4(" ");
READ(MASTERNAME,-8);
MOVE TAG(1):=MASTERNAME,(4);
TAG(10):=-1;

<< GET SLAVE NAME >>
MOVE MSG := "WHAT IS THE SLAVE'S NAME? ";
PRINT(MSG1,-26,%320);
READ(SNAMEI,-28);

<< START UP SLAVE PROGRAM >>
MOVE MSG:="POPEN ISSUED";
PRINT(MSG1,-12,%40);
DSNUM:=POPEN(DSLINE,SLAVENAME,TAG(1));
ERRORCHECK;
IF OK=FALSE THEN GO TO LASTEND;
FIXTAG;

MOVE MSG:="PREAD ISSUED WITH LENGTH = 500";
PRINT(MSG1,-31,%40);
PREAD(DSNUM,READBUFR(1),500,TAG(1));
ERRORCHECK;

```

PTOP Communications: EXAMPLES

```
FOR I:=1 UNTIL 500 DO
  IF READBUFR(I) <> READTEXT(I) THEN
  BEGIN
    MOVE MSG:="BUFFER DIFFERS AT ELEMENT           ";
    ASCII(I,10,MSG(26));
    PRINT(MSG1,-32,%40);
    I:=500;
    END;
  FIXTAG;

MOVE MSG:="PWRITE ISSUED WITH LENGTH = 500";
PRINT(MSG1,-32,%40);
PWRITE(DSNUM,WRITETEXT(1),500,TAG(1));
ERRORCHECK;
FIXTAG;

MOVE MSG:="PCONTROL ISSUED TO TERMINATE SLAVE";
PRINT(MSG1,-34,%40);
TAG(10):=0;
PCONTROL(DSNUM,TAG(1));
ERRORCHECK;

PCLOSE(DSNUM);
LASTEND:END.
```

Slave Program on HP 3000

```

$CONTROL USLINIT, ADR, MAP, CODE, NOWARN
BEGIN
COMMENT
    THE NAME OF THIS PROGRAM IS SLAVEP(S).
    THIS PROGRAM IS TO BE COMPILED AND PREP'ED ON THE
    SLAVE HP 3000 SYSTEM.  IT WILL BE INITIATED FOR
    RUN BY THE MASTER.  THE FUNCTION OF THIS PROGRAM
    IS TO LOAD A DISK FILE WITH KNOWN TEST PATTERNS
    THAT WILL BE TRANSFERRED TO THE MASTER AND
    PRINTED ON THE MASTER'S LINE PRINTER;

INTEGER      DISK'FILNENUM, I, IL, IONUMBER, J;
LOGICAL ARRAY DISK'BUF(0:39);
LOGICAL ARRAY ITAG(0:19);

BYTE ARRAY MSG(0:79);
BYTE ARRAY TEST(0:4):="TEST ";

INTRINSIC FOPEN, DEBUG, PRINT, QUIT, READ, FWRITEDIR,
          FREADDIR, FCLOSE;

DEBUG;

MOVE MSG:="ISSUING A GET (REMOTE)";
PRINT(MSG,-22,0);
I:=GET;                                <<GET FOR POPEN>>
IF < THEN QUIT(25);
IF I=1 THEN
    BEGIN
        MOVE MSG:=("POPEN RCVD...ISSUING AN ACCEPT ",
            "(REMOTE)");
        PRINT(MSG,-39,0);
    END
ELSE
    QUIT(3);

MOVE ITAG:="ACCEPT SUCCESSFUL (REMOTE)";
ACCEPT;                                <<ACCEPT FOR POPEN>>

DISK'FILENUM:=FOPEN(TEST,4,%104,-80,,,,1,1,10D);
    IF <> THEN QUIT(1);

I:=GET;                                <<TEST REC FROM MASTER>>

IF <> THEN <> QUIT(2);
IF I=3 THEN                                <<PWRITE RECEIVED>>
BEGIN
    ACCEPT(,DISK'BUF);
    IF <> THEN QUIT(3);
END;
I:=-1;                                <<START WRITING TEST FILE>>
WHILE(I:=I+1)<5 DO
BEGIN

```

PTOP Communications: EXAMPLES

```
        FWRITEDIR(DISK'FILENUM,DISK'BUF,40,DOUBLE(I));
                                <<WRITE REC TO DISK>>
    IF <> THEN QUIT(4);
END;                                <<END WRITING TEST FILE>>

J:=-1;
WHILE (J:=J+1)<5 DO
BEGIN
    MOVE MSG:="ISSUING A GET (REMOTE)";
    PRINT(MSG,-22,0);
    I:=GET(ITAG,40,IONUMBER);
    IF < THEN QUIT(5);
    IF I=2 THEN
    BEGIN
        MOVE MSG:=("PREAD RCVD...ISSUING AN ",
                  "ACCEPT (REMOTE)");
        PRINT(MSG,-39,0);
    END
    ELSE
        QUIT(6);
    MOVE DISK'BUF;=%20040;
    MOVE DISK'BUF(1):=DISK'BUF(0),(39);
    FREADDIR(DISK'FILENUM,DISK'BUF,40,DOUBLE(J));
    IF <> THEN QUIT(7);
    ACCEPT(ITAG,DISK'BUF,40);
    IF <> THEN QUIT(8);
END;

FCLOSE(DISK'FILENUM,0,0);
END.
```

The following examples provide the PPACK and UNPACK SPL sub-program modules. They are used to convert the internal format for type REAL (floating point) for use between the HP 3000 and HP 1000. The master and slave programs pair generates square root values, passes the values between systems, and prints the comparison values. The subprograms can be added to a group SL, or they can be PREP'd with master program.

PPACK Convert 3000 to 1000 format
 PUNPACK Convert 1000 to 3000 format

Suggestions for use:

	(In SL)	(In Program)
Compile	:SPL PPACKS,,*LP :SPL PUNPACKS,,*LP	:FORTRAN PPACKS,,*LP :SPL PPACKS,,*LP :SPL PUNPACKS,,*LP
Load	:SEGMENTER -USL \$OLDPASS -BUILDSL SL.GROUP,20,1 -ADDSL SEG' -LISTSL -EXIT	:PREP \$OLDPASS,& \$NEWPASS,STACK=3000;PMAP :SAVE \$OLDPASS,PACK
Compile	:FORTRAN PACKS,,*LP :PREP \$OLDPASS,\$NEWPASS;& :STACK=3000;PMAP :SAVE \$OLDPASS,PACK	
Use:	:DSCONTROL 51,OPEN :DSLIN 51 :RUN PACK;LIB=G	:RUN PACK

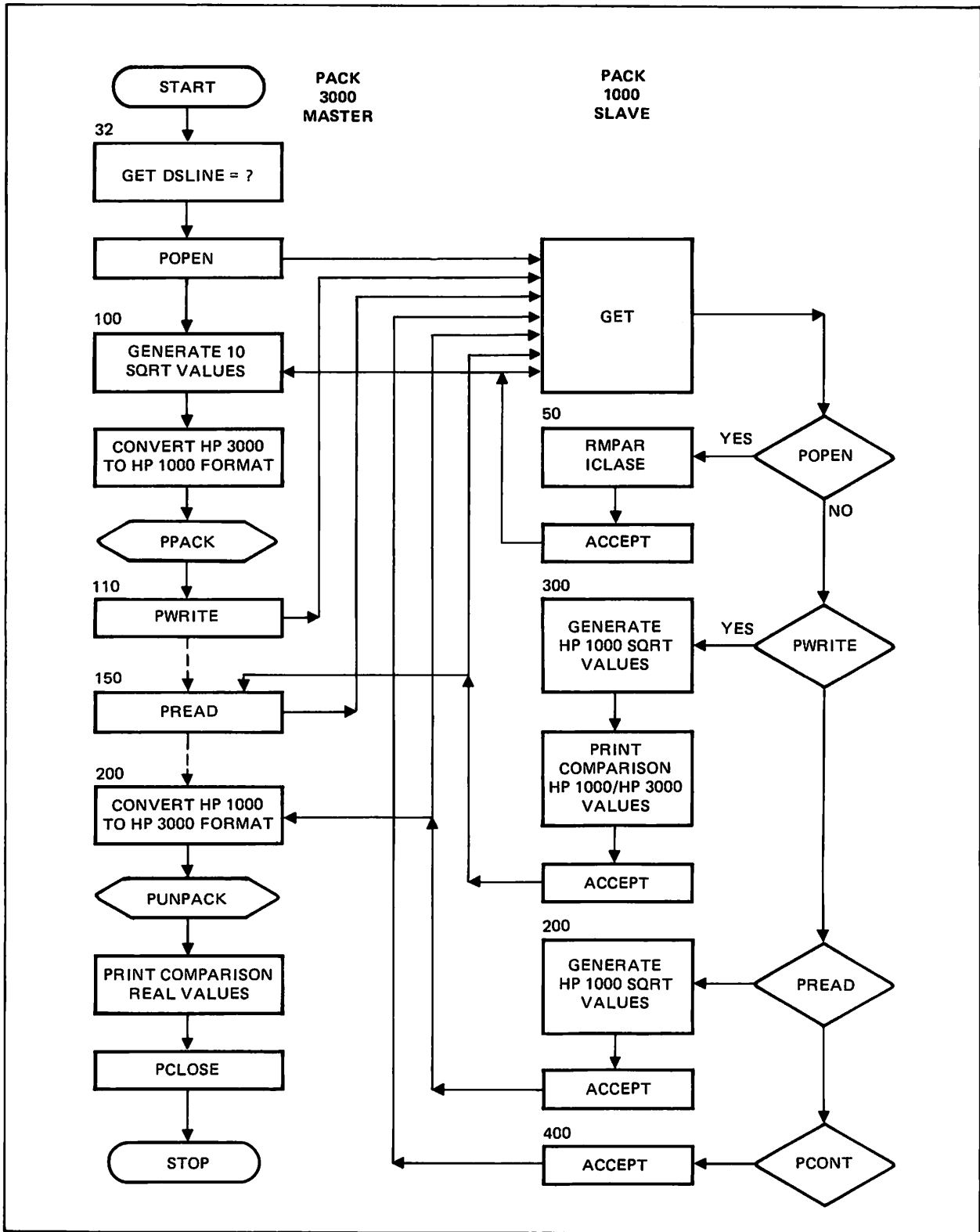


Figure 6-5 Pack Example

PPACK Subprogram

```

$CONTROL SUBPROGRAM
BEGIN
REAL PROCEDURE PPACK(REAL3000);
REAL REAL3000;
<<PPACK - PROCEDURE TO CONVERT HP3000 REAL
(FLOATING POINT) FORMAT TO HP2100 REAL
(FLOATING POINT) FORMAT.

REAL3000 - DATA IN HP3000 FORMAT (SINGLE PRECISION).
                                0123456789012345 0123456789012345
LREAL                                SEEEEEEEEEMMMMMM MMMMMMMMMMMMMMMMM

                                DATA IN HP1000 FORMAT
                                0123456789012345 0123456789012345
LTEMP                                SMMMMMMMMMMMMMMMM MMMMMMMMEEEEEEEES

S - SIGN                CONDITION CODE RETURNS
E - EXPONENT            CCL EXPONENT TOO SMALL (<127)
M - MANTISSA            CCE VALID CONVERSION
                        CCG EXPONENT TOO LARGE (>127)
                        >>

BEGIN
REAL TEMP:=0.0;                << FOR ZERO>>
REAL RREAL;
LOGICAL ARRAY LREAL(*)=RREAL;
LOGICAL ARRAY LTEMP(*)=TEMP;
INTEGER EXPONENT:=0;
LOGICAL STATUS=Q-1;
DEFINE CC=STATUS(6:2)#, CCE=2, CCL=1#, CCG=0#;
REAL:=REAL3000;
IF REAL3000=0.0                <<SPECIAL CASE, REAL3000=0>>
THEN CC:=CCE
ELSE BEGIN
    LTEMP.(0:1):=LREAL.(0:1);    <<SIGN BIT>>
    LTEMP.(1:1):=LOGICAL(%1);    <<LEADING 1>>
    LTEMP.(2:6):=LREAL.(10:6);   <<TRANSFER
                                MANTISSA>>
    LTEMP.(8:8):=LREAL(1).(8:8); <<GET TRUE
                                EXPONENT>>
    EXPONENT:=INTEGER(LREAL.(1:9)
                    -LOGICAL(%400));
    EXPONENT:=EXPONENT+1;        <<3000=1.MMM,
                                1000=.5MMM>>
                                <<TRANSFER
                                SIGN OF
                                EXPONENT>>
    LTEMP.((1).(15:1):=LOGICAL(EXPONENT.(0:1));
    IF EXPONENT > 127            <<TEST
                                MAGNITUDE TOO LARGE OR SMALL>>
    THEN CC:=CCG
    ELSE IF EXPONENT < -127

```

PTOP Communications: EXAMPLES

```

                                THEN CC:=CCL
                                ELSE CC:=CCE;
IF CC=CCE                        <<TRANSFER
                                EXPONENT>>
                                THEN
                                LTEMP(1).(8:7):=LOGICAL(\EXPONENT\)
                                ELSE LTEMP(1).(8:7):=LOGICAL(%177);
                                END;
PPACK:=TEMP;
END;
END.
```

```

$CONTROL SUBPROGRAM
BEGIN
REAL PROCEDURE PUNPACK (REAL1000);
  REAL REAL2100;
  << PUNPACK - PROCEDURE TO CONVERT 2100 REAL
  (FLOATING POINT) FORMAT TO HP 2100 REAL (FLOATING
  POINT) FORMAT.
  REAL 1000 - DATA IN HP2100 FORMAT (SINGLE
  PRECISION).

  LREAL =          0123456789012345 0123456789012345
                  SMMMMMMMMMMMMMM MMMMMMMMEEEEEEES

  PUNPACK - DATA IN HP 3000 FORMAT (SINGLE
  PRECISION).

  LTEMP =          0123456789012345 0123456789012345
                  SEEEEEEEEMMMMMMM MMMMMMMMMMMMMMMM

  S = SIGN
  E = EXPONENT
  M = MANTISSA                                >>

BEGIN
  REAL TEMP := 0.0; <<FOR ZERO>>
  REAL RREAL;
  LOGICAL ARRAY LREAL (*)=RREAL;
  LOGICAL ARRAY LTEMP (*)=TEMP;
  INTEGER EXPONENT:=0;

  RREAL:=REAL2100;
  IF RREAL = 0.0    <<SPECIAL CASE, ZERO>>
  THEN TEMP:=0.0
  ELSE BEGIN
      <<TRANSFER SIGN BIT>>
      LTEMP.(0:1):=LREAL.(0:1);
      <<TRANSFER MANTISSA>>
      LTEMP.(10:6):=LREAL.(2:6);
      LTEMP(1).0:8:=LREAL.(8:8);
      LTEMP(1).(8:8):=LREAL(1).(0:8);
      <<BIAS EXPONENT>>
      EXPONENT:=INTEGER(LREAL(1).(8:7)*(-1)^
      INTEGER(LREAL(1).(15:1))+400;
      EXPONENT:=EXPONENT-1;<<3000=1.MMMM,
      2100=.5MMMM>>
      LTEMP.(1:9):=LOGICAL(EXPONENT);
      END;
  PUNPACK:=TEMP;
  END;
END.

```

PTOP Communications: EXAMPLES

```

FTN4L,C,T,M
C   PACK 1000 SLAVE
    PROGRAM PACK1000
C
C   INTEGER IBUF21(50),IBUF30(50),ITAG(20),IP(5)
C
C   REAL R2100(10),R3000(10)
C
C   EQUIVALENCE (IBUF21,R2100),(IBUF30,R3000)
C
C   CALL PMPAR(IP)
    ICLASS=IP(1)
    LUTTY=1
20  IL=40
C   ***** GET *****
    CALL GET(ICLASS, IERR, IFUNC, ITAG, IL)
    IF (IERR) 900, 50, 910
50  GO TO (100, 200, 300, 400), IFUNC
C
C   ***** POPEN *****
C
100 CALL ACCEPT(ITAG, IERR)
    IF (IERR) 920, 20, 930
C
C   ***** PREAD *****
C
C   GENERATE 10 REAL VALUES IN R2100 ARRAY
C
200 WRITE(LUTTY,202)
202 FORMAT(" RECEIVED PREAD")
C
    DO 210 I=1,10
        R=FLOAT(I)
210  R2100(I)=SQRT(R)
C
    CALL ACCEPT(ITAG, IERR, IBUF21)
    IF (IERR) 920, 20, 930
C
C   ***** PWRITE *****
C
C   ACCEPT 10 VALUES FROM 3000 AND COMPARE WITH
C   LOCALLY GENERATED VALUES.
C
300 WRITE(LUTTY,302)
302 FORMAT(" RECEIVED PWRIT")
C
    DO 310 I=1,10
        R=FLOAT(I)
310  R2100(I)=SQRT(I)
C
    CALL ACCEPT(ITAG, IERR, IBUF30)
    IF(IERR) 920, 320, 930
C
320 WRITE(LUTTY,322)
322 FORMAT("          3000 VALUES    2100 VALUES 'SQRT 1",

```

```
      *" TO 10' ")
C
      DO 330 I=1,10
      WRITE(LUTTY,330)I,R3000(I),R2100(I)
330  FORMAT(I3,3X,R8.6,3X,R6.6)
      GO TO 20
C
C      ***** PCONTROL *****
C
400  WRITE(LUTTY,402)
402  FORMAT(" RECEIVED PCONT")
      CALL ACCEPT(ITAG,IERR)
      IF (IERR) 920,410,930
C
410  GO TO 20
C
C      **** ERRORS ****
C
900  WRITE(LUTTY,902)IERR
902  FORMAT(" IERR < 0 ON GET, IERR = ",I6)
      STOP
C
910  WRITE(LUTTY,912)IERR
912  FORMAT(" IERR > 0 ON GET, IERR = ", I6)
      STOP
C
C      **** ACCEPT ERRORS ****
C
920  WRITE(LUTTY,922)IERR
922  FORMAT(" IERR < 0 ON ACCEPT, IERR = ",I6)
      STOP
C
930  WRITE(LUTTY,932),IERR
932  FORMAT(" IERR > 0 ON ACCEPT, IERR = ",I6)
      STOP
      END
```

PTOP Communications: EXAMPLES

```

$CONTROL INIT,LABEL,SOURCE,MMAP,CROSSREF,LOCATION
PROGRAM PACK
C
C          THIS IS A MASTER PROGRAM FOR THE 3000
C          IT REQUIRES A SLAVE ON 2100 'PACK'
C          IT TRANSFERS FLOATING POINT VALUES BETWEEN
C          SYSTEMS
C
C          INTEGER DSNUM, ICODE, TCNT, LGTH
C          INTEGER ITAG(20), IBUF21(40), IBUF30 (40),
*PROGRAMME (3)
C          CHARACTER DSDEVICE*6, PROGBYTE*6
C          REAL R2100(10), R3000(10)
C          EQUIVALENCE (IBUF21,R2100), (IBUF30,R3000),
* (PROGRAMME,PROGBYTE)
C          SYSTEM INTRINSIC PCHECK, PCLOSE, PCONTROL, POPEN,
*PREAD, PWRITE
C          EXTERNAL PPACK, PUNPACK
C
C          ***** MAIN PROGRAM *****
C
C          WRITE(6,20)
20          FORMAT(
* " THIS PROGRAM DEMONSTRATES THE CONVERSION OF"/
* " FLOATING POINT FORMATS BETWEEN 2100 AND 3000."/
* " REQUIRES SLAVE 'PACK' ON RTE, AND PACK AND "/
* " UNPACK SL SEGMENTS IN GROUP LIB OR PREP'D "/
* " WITH PROG.")
C
C          WRITE(6,32)
32          FORMAT(2/,"+INPUT DSLINE = ? ")
C          READ (5,*)DSDEVICE
C
C          PROGBYTE="PACK "
C
C          DSNUM = POPEN(DSDEVICE, PROGBYTE, ITAG)
C          IF(.CC.)900,100,910
C
C          POPEN WAS OK
C          GENERATE FLOATING POINT VALUES
C
100          DO 110 I=1,10
C              R=FLOAT(I)
C              R3=SQRT(R)
C              R3000(I)=R3
C
C              CONVERT TO 2100 FORMAT
110          R2100(I)=PPACK(R3)
C
C          TCNT = 20
C
C              SEND TO 2100
C
C          CALL PWRITE(DSNUM, IBUF21, TCNT, ITAG)
C          IF(.CC.) 920, 150, 930
C

```

PTOP Communications: EXAMPLES

```

C          DATA WAS SENT OK
C          NOW RECEIVE DATA FROM 2100
C
150  LGTH=PREAD(DSNUM, IBUF21, TCNT, ITAG)
      IF(.CC.) 940, 200, 950
200  WRITE(6,202)
202  FORMAT("          2100 FLT. PT          3000 FLT. PT")
C
C          RECEIVE 2100 DATA AND UNPACK IT
C
      DO 220 I=1,10
      R30=PUNPACK(R2100(I))
C
C          COMPARE WITH 3000 DATA LOCALLY
C          GENERATED
C
220  WRITE(210)I,R30,R3000(I)
210  FORMAT(I6,3X,F8,6,5X,F8.6)
C
C          ALL DONE.  CLOSE SLAVE.
C
      CALL PCLOSE(DSNUM)
      IF(.CC.)960,300,970
C
300  WRITE(6,302)
302  FORMAT(" COMPLETED TASK.  TERMINATE.")
      STOP
C
C          ***** ERROR CONDITIONS *****
C
900  ICODE=PCHECK(DSNUM)
      WRITE(6,902)ICODE
902  FORMAT(" CCL ON GET, ICODE = ",I6)
      STOP
C
910  WRITE(6,912)
912  FORMAT(" GET REJECTED BY SLAVE.")
      STOP
C
920  ICODE=PCHECK(DSNUM)
      WRITE(6,922)ICODE
922  FORMAT(" CCL ON PWRIT, ICODE = ",I6)
      STOP
C
930  WRITE(6,932)
932  FORMAT(" CCG ON PWRIT.")
      STOP
C
940  ICODE=PCHECK(DSNUM)
      WRITE(6,942)ICODE
942  FORMAT(" CCL ON PREAD, ICODE= ",I6)
      STOP
C
950  WRITE(6,952)
952  FORMAT(" CCG ON PREAD, REJECTED BY SLAVE.")

```


PTOP Communications: EXAMPLES

```
C
 960  ICODE=PCHECK(DSNUM)
      WRITE(6,962)ICODE
 962  FORMAT(" CCL ON PCLOSE, ICODE=",I6)
      STOP
C
 970  WRITE(6,972)
 972  FORMAT(" CCG ON PCLOSE")
      STOP
C
      END
```

The following is a summary of the messages that may be encountered together with their meanings. The messages, as listed here have been grouped according to their associated activities. For this reason, some messages are listed under more than one heading.

Communication Link Errors

These errors may appear while using the DSLINE command, Remote File Access (RFA), or program-to-program intrinsics. The method of reporting depends on the operation in progress when an error occurs.

If you are executing commands, the error appears as a

60, nnn

If you are performing remote FMP access, the error is returned in the error parameter.

If you are using DEXEC calls, the error is returned as a four-character ASCII code in the A and B registers.

If you are using program-to-program operations, the error code is returned by the PCHECK intrinsic.

DS/3000 SUBSYSTEM

:DSLIME Syntax Errors

Error Code Message

These messages are sent to the terminal user to point out an error in syntax or to warn of the consequences of a request.

- 1300 REMOTE JOBS ARE NOT ALLOWED !. (CIERR 1300)
- 1301 DSLIME CANNOT CONTAIN BOTH OPEN AND CLOSE. (CIERR 1301)
- 1302 DSLIME REQUIRES AT LEAST ONE PARAMETER. (CIERR 1302)
- 1303 DSNUMBER SPECIFICATION MUST BE A NUMBER FROM 1 THRU 255. (CIERR 1303)
- 1304 DSLIME #L! DOES NOT IDENTIFY AN OPEN DS LINE. (CIERR 1304)
- 1305 EXPECTED LINEBUF, PHNUM, IOCID, REMID, OPEN, CLOSE, QUIET, COMP, NOCOMP, OR EXCLUSIVE. (CIERR 1305)
- 1306 MULTIPLE USE OF ! IS NOT ALLOWED. (CIERR 1306)
- 1307 THE SYNTAX FOR ! REQUIRES AN = SIGN FOLLOWED BY DATA. (CIERR 1307)
- 1308 PHNUM IS 1 TO 20 DIGITS AND DASHES. (CIERR 1308)
- 1309 ! LIST CAN CONTAIN ONLY ONE ELEMENT. (CIERR 1309)
- 1310 THE SPECIFIED LOGICAL DEVICE IS NOT OPEN. (CIERR 1310)
- 1311 THE FIRST CHARACTER OF AN ID SEQUENCE MUST BE A " OR A (CIERR 1311)
- 1312 THE ID SEQUENCE MUST TERMINATE WITH A). (CIERR 1312)
- 1313 THE ID SEQUENCE MUST TERMINATE WITH A ". (CIERR 1313)
- 1314 A NUMERIC ID SEQUENCE ELEMENT MUST BE 1 THRU 255 (OR %377). (CIERR 1314)
- 1315 LINEBUF MUST BE A NUMERIC VALUE FROM 304 THRU 4096. (CIERR 1315)
- 1316 UNABLE TO COMPLETE THE REMOTE COMMAND. (CIERR 1316)
- 1317 NOT A CURRENTLY AVAILABLE DSLIME. (CIERR 1317)
- 1318 USE OF EXCLUSIVE REQUIRES BOTH NS AND CS CAPABILITY. (CIERR 1318)
- 1319 THE DS LINE #L! IS IN USE BY A PROGRAM OR SUBSYSTEM AND CANNOT BE CLOSED. (CIERR 1319)
- 1320 EXPECTED A RESPONSE OF YES, Y, NO, OR N. (CIERR 1320)
- 1321 UNABLE TO OPEN THE DS LINE ON DEVICE !. (CIERR 1321)
- 1322 @ IS INVALID IN THIS CONTEXT. (CIERR 1322)
- 1323 A DSLIME OPEN REQUIRES A VALID DS DEVICE NAME AS THE FIRST PARAMETER. (CIERR 1323)
- 1393 COMPRESSION REQUEST NOT HONORED. REMOTE DOES NOT SUPPORT THIS FEATURE. (CIWARN 1393)
- 1394 COMPRESSION PARAMETER RESPECIFIES AND OVERRIDES PREVIOUS COMPRESSION PARAMETER. (CIWARN 1394)
- 1395 OPEN PARAMETERS ENTERED ON A CLOSE REQUEST ARE IGNORED (CIERR 1395)
- 1396 AN ID LIST MUST CONTAIN 255 OR LESS ELEMENTS. (CIWARN 1396)
- 1397 AN UNNECESSARY DELIMITER IS IGNORED. (CIWARN 1397)
- 1398 THERE ARE NO DS LINES OPEN. (CIWARN 1398)
- 1399 MULTIPLE USE OF ! IS REDUNDANT AND IGNORED. (CIWARN 1399)

DSN/DS Functional Errors

These messages report a functional problem within the system.

- 201 REMOTE DID NOT RESPOND WITH THE CORRECT REMOTE ID. (DSERR 201)
- 202 SPECIFIED PHONE NUMBER IS INVALID. (DSERR 202)
- 203 REMOTE ABORT/RESUME NOT VALID WHEN DOING PROGRAM-TO-PROGRAM COMMUNICATION. USE LOCAL ABORT/RESUME. (DSWARN 203)
- 204 UNABLE TO ALLOCATE AN EXTRA DATA SEGMENT FOR DS/3000. (DSERR 204)
- 205 UNABLE TO EXPAND THE DS/3000 EXTRA DATA SEGMENT. (DSERR 205)
- 206 SLAVE PTOP FUNCTION ISSUED FROM A MASTER PROGRAM. (DSERR 206)
- 207 SLAVE PTOP FUNCTION OUT OF SEQUENCE. (DSERR 207)
- 208 MASTER PTOP FUNCTION ISSUED BY A SLAVE PROGRAM. (DSERR 208)
- 209 SLAVE PROGRAM DOES NOT EXIST OR IS NOT PROGRAM FILE. (DSERR 209)
- 210 WARNING -- INVALID MAXDATA OR DLSIZE FOR A SLAVE PROGRAM. SYSTEM DEFAULTS ARE IN EFFECT. (DSWARN 210)
- 211 SLAVE ISSUED A REJECT TO A MASTER PTOP OPERATION. (DSWARN 211)
- 212 FILE NUMBER FROM IOWAIT NOT A DS LINE NUMBER. (DSWARN 212)
- 213 EXCLUSIVE USE OF A DS LINE REQUIRES BOTH ND AND CS CAPABILITY. (DSERR 213)
- 214 THE REQUESTED DS LINE HAS NOT BEEN OPEN WITH A USER :DSLIN COMMAND OR A REQUIRED :REMOTE HELLO HAS NOT BEEN DONE. (DSERR 214)
- 215 DSLIN CANNOT BE ISSUED BACK TO THE MASTER COMPUTER. (DSERR 215)
- 216 MESSAGE REJECTED BY THE REMOTE COMPUTER. (DSERR 216)
- 217 INSUFFICIENT AMOUNT OF USER STACK AVAILABLE. (DSERR 217)
- 218 INVALID PTOP FUNCTION REQUESTED. (DSERR 218)
- 219 MULTIPLE POPEN. ONLY ONE MASTER PTOP OPERATION CAN BE ACTIVE ON A DS LINE. (DSERR 219)
- 220 PROGRAM EXECUTING GET WAS NOT CREATED BY POPEN. (DSERR 220)
- 221 INVALID DS MESSAGE FORMAT. INTERNAL DS ERROR. (DSERR 221)
- 222 MASTER PTOP FUNCTION ISSUED PRIOR TO A POPEN. (DSERR 222)
- 223 REQUEST TO SEND MORE DATA THAN SPECIFIED IN POPEN. (DSERR 223)
- 224 FILE EQUATIONS FOR A REMOTE FILE CONSTITUTE A LOOP. (DSERR 224)
- 225 CANNOT ISSUE POPEN TO A SLAVE SESSION IN BREAK MODE. (DSERR 225)
- 226 COMMUNICATIONS HARDWARE HAS DETECTED AN ERROR. (DSERR 226)
- 227 CANNOT CURRENTLY GAIN ACCESS TO THE TRACE FILE. (DSERR 227)
- 228 COMMUNICATIONS INTERFACE ERROR. INTERNAL FAILURE. (DSERR 228)
- 229 COMMUNICATIONS INTERFACE ERROR. TRACE MALFUNCTION.

Error Codes and Messages: DS/3000 Subsystem

- (DSERR 239)
- 240 LOCAL COMMUNICATION LINE WAS NOT OPENED BY OPERATOR.
(DSERR 240)
- 241 DS LINE IN USE EXCLUSIVELY OR BY ANOTHER SUBSYSTEM.
(DSERR 241)
- 242 INTERNAL DS SOFTWARE ERROR ENCOUNTERED. (DSERR 242)
- 243 THE REMOTE COMPUTER IS NOT RESPONDING. (DSERR 243)
- 244 COMMUNICATIONS INTERFACE ERROR. THE REMOTE COMPUTER RESET
THE LINE. (DSERR 244)
- 245 COMMUNICATIONS INTERFACE ERROR. RECEIVE TIMEOUT. (DSERR
245)
- 246 COMMUNICATIONS INTERFACE ERROR. REMOTE DISCONNECTED.
(DSERR 246)
- 247 COMMUNICATIONS INTERFACE ERROR. LOCAL TIME OUT. (DSERR 247)
- 248 COMMUNICATIONS INTERFACE ERROR. CONNECT TIME OUT. (DSERR
248)
- 249 COMMUNICATIONS INTERFACE ERROR. REMOTE REJECTED
CONNECTION. (DSERR 249)
- 250 COMMUNICATIONS INTERFACE ERROR. CARRIER LOST. (DSERR 250)
- 251 COMMUNICATIONS INTERFACE ERROR. LOCAL DATA SET FOR THE
DS LINE WENT NOT READY. (DSERR 251)
- 252 COMMUNICATIONS INTERFACE ERROR. HARDWARE FAILURE. (DSERR
252)
- 253 COMMUNICATIONS INTERFACE ERROR. NEGATIVE RESPONSE TO THE
DIAL REQUEST BY THE OPERATOR. (DSERR 253)
- 254 COMMUNICATIONS INTERFACE ERROR. INVALID I/O CONFIGURATION.
(DSERR 254)
- 255 COMMUNICATIONS INTERFACE ERROR. UNANTICIPATED CONDITION.
(DSERR 255)

LINE OPENING FAILURES

:DSCONTROL Informatory Messages

Error Code Message

These messages convey status information.

300 DS DEVICE !: MASTER AND SLAVE ACCESS SHUT.
301 DS DEVICE !: SLAVE ACCESS OPENED; MASTER ACCESS SHUT.
302 DS DEVICE !: MASTER ACCESS OPENED; SLAVE ACCESS SHUT.
303 DS DEVICE !: MASTER AND SLAVE ACCESS OPENED.
304 DS DEVICE !: TRACE ACTIVATED USING TRACE FILE !.
305 DS DEVICE !: TRACE DEACTIVATED.
306 DS DEVICE !: MONITORING ACTIVATED.
307 DS DEVICE !: MONITORING DEACTIVATED.
308 DS DEVICE !: DEBUG MODE ACTIVATED.
309 DS DEVICE !: DEBUG MODE DEACTIVATED.
310 DS DEVICE !: SPECIAL DEBUG MODE ACTIVATED.
311 DS DEVICE !: DEFAULT MODE IS NO COMPRESSION.
312 DS DEVICE !: DEFAULT MODE IS COMPRESSION.

:DSCONTROL Error Messages

These messages point out an error in syntax or warn of the consequences of a request.

4100 NUMBER OF PARAMETERS EXCEEDS MAXIMUM OF !. (CIERR 4100)
4101 EXPECTED AT LEAST TWO PARAMETERS: A DS DEVICE
CLASS/NUMBER AND A FUNCTION KEYWORD. (CIERR 4101)
4102 EXPECTED A DEVICE CLASS NAME OR LOGICAL DEVICE NUMBER FOR
ONE OR MORE DS DEVICES. (CIERR 4102)
4103 USER IS NOT ASSOCIATED WITH DS DEVICE !. NO CONTROL
FUNCTIONS EXECUTED FOR THIS DEVICE. (CIWARN 4103)
4104 USER IS NOT ALLOWED TO USE :DSCONTROL AND IS NOT ASSOCIATED
WITH THE DS DEVICE(S). (CIERR 4104)
4105 EXPECTED ONE OR MORE OF THE CONTROL FUNCTIONS: OPEN, SHUT,
MON, MOFF, COMP, NOCOMP, TRACE, OR DEBUG. (CIERR 4105)
4106 INVALID CONTROL FUNCTION. EXPECTED ONE OF: OPEN, SHUT,
MON, MOFF, COMP, NOCOMP, TRACE, OR DEBUG. (CIERR 4106)
4107 MASTER OVERRIDES PREVIOUS MASTER/SLAVE OPTION. (CIWARN
4107)
4108 SLAVE OVERRIDES PREVIOUS MASTER/SLAVE OPTION. (CIWARN
4108)
4109 SPEED OPTION OVERRIDES PREVIOUS SPEED OPTION. (CIWARN
4109)
4110 OPEN OVERRIDES PREVIOUS OPEN/SHUT FUNCTION. (CIWARN 4110)
4111 SHUT OVERRIDES PREVIOUS OPEN/SHUT FUNCTION. (CIWARN 4111)
4112 TRACE OVERRIDES PREVIOUS TRACE FUNCTION(S). (CIWARN 4112)

Error Codes and Messages: Line Opening Failures

4113 DEBUG OVERRIDES PREVIOUS DEBUG FUNCTION(S). (CIWARN 4113)
4114 MON OVERRIDES PREVIOUS MON/MOFF FUNCTION. (CIWARN 4114)
4115 MOFF OVERRIDES PREVIOUS MON/MOFF FUNCTION. (CIWARN 4115)
4116 COMP OVERRIDES PREVIOUS COMP/NOCOMP FUNCTION. (CIWARN 4116)
4117 NOCOMP OVERRIDES PREVIOUS COMP/NOCOMP FUNCTION. (CIWARN
4117)
4118 EXPECTED A ";", ",", ":", OR RETURN AS DELIMITER. (CIERR 4118)
4119 EXPECTED EITHER A ";" OR RETURN AS DELIMITER. (CIERR 4119)
4120 EXPECTED A "=" AS DELIMITER FOR SPEED OPTION. (CIERR 4120)
4121 EXPECTED A ",", AS DELIMITER BETWEEN OPTIONS. (CIERR 4121)
4122 ILLEGAL OPEN/SHUT OPTION. EXPECTED ONE OF: MASTER, SLAVE,
SPEED, OR LINESPEED VALUE. (CIERR 4122)
4123 EXPECTED A POSITIVE DOUBLE VALUE FOR LINESPEED. (CIERR
4123)
4124 CS CAPABILITY REQUIRED TO USE :DSCONTROL. (CIERR 4124)
4125 PM CAPABILITY REQUIRED TO USE DEBUG FUNCTION. (CIERR 4125)
4126 DEBUG FUNCTION MAY ONLY BE USED BY SYSTEM CONSOLE. (CIERR
4126)
4127 EXPECTED NO OPTION FOR DEBUG OR ONE OF THE FOLLOWING:
ON, OFF, OR POSITIVE INTEGER VALUE. (CIERR 4127)
4128 EXPECTED NO OPTION FOR MON/MOFF OR ONE OF THE FOLLOWING:
CS OR DS. (CIERR 4128)
4129 COMP/NOCOMP FUNCTIONS HAVE NO OPTIONS. (CIERR 4129)
4130 SPEED OPTION IGNORED FOR SHUT FUNCTION. (CIWARN 4130)
4131 EXTRANEOUS ";" IGNORED. POSSIBLE MISSING FUNCTION? (CIWARN
4131)
4132 EXTRANEOUS ",", IGNORED. POSSIBLE MISSING OPTION? (CIWARN
4132)
4133 CREATION OF DS MONITOR PROCESS FAILED. (CIERR 4133)
4134 PROGRAM FILE "DSMON.PUB.SYS" MISSING. (CIERR 4134)
4135 DS MONITOR UNABLE TO RUN AS A SYSTEM PROCESS. (CIERR 4135)
4136 CS DEVICE ! IS UNAVAILABLE FOR USE. (CIERR 4136)
4137 DS DEVICE MUST BE OPEN PRIOR TO USE. (CIERR 4137)
4138 USER SPECIFIED TRACE FILE NOT ALLOWED WHEN MORE THAN ONE
DEVICE IN DEVICE CLASS. (CIERR 4138)
4139 DS DEVICE ! CURRENTLY CONTROLLED ELSEWHERE. (CIWARN 4139)
4140 DS DEVICE !: OPEN/SHUT NOT EXECUTED DUE TO ABOVE. (CIWARN
4140)
4141 DS DEVICE !: TRACE NOT EXECUTED DUE TO ABOVE. (CIWARN 4141)
4142 DS DEVICE !: MON/MOFF NOT EXECUTED DUE TO ABOVE. (CIWARN
4142)
4143 DS DEVICE !: COMP/NOCOMP NOT EXECUTED DUE TO ABOVE. (CIWARN
4143)
4144 DS DEVICE !: DEBUG NOT EXECUTED DUE TO ABOVE. (CIWARN 4144)
4145 NO DS DEVICES REMAINING TO BE CONTROLLED. (CIWARN 4145)

Remote FMP Access Errors

These error messages are returned in the error parameter.

Error Code	Meaning	Corrective Action
000	No error.	None
-001	The disc is down.	Try again and then report it to the system manager of the facility.
-002	A file already exists with specified name.	Repeat with new name or purge existing file.
-003	Attempt was made to back-space a device, or type 0 file that cannot be back-spaced.	Check device type.
-004	Attempt was made to create a type 2 file with too many records, or record size too large.	Check size parameter.
-005	Attempt to read or position or to a record not written; or on update to write an illegal record length.	Check position or size parameters.
-006	Attempt to access a cartridge or file that cannot be found, or that has no more room.	Check the file name or cartridge number; if no more room on cartridge try another, or decrease file size.
-007	Attempt to access a file with no security or the wrong code.	Find out the correct code and use it, or do not access the file.
-008	Attempt to open file already open exclusively, or to open eight programs, or cartridge containing file is locked.	Use CL or DL to locate lock; if file is being packed, check if spool shut down.
-009	Attempt to position type 0 file with DAPOSN or force type 0 file to type 1.	Check file type.
-010	Required parameters omitted from call.	Enter the parameters.

REMOTE FMP ACCESS

-011	Attempt to access an unopened DCB.	Use CREATE, or OPEN to open DCB; check for errors.
-012	Attempt to read or write or position beyond file boundaries.	Check record position parameters; result depends on file type and call.
-013	Cartridge is locked.	Initialize cartridge if not initialized; otherwise keep trying.
-014	No more room in file directory.	Purge files and pack directory if possible, or try another cartridge.
-015	File name does not conform to syntax rules.	Correct name.
-016	Wrong type code supplied; attempt to create or purge type 0 file or create a 0-length file.	Check size and type parameters.
-017	Attempt to read, write, or position type 0 file that does not support the operation.	Check file parameters, from FMGR check name.
-018	Destination node does not have FMP.	
-025	Bad FCODE. Internal RFAM error.	
-026	Incorrect entry number in RFAM. DCB destroyed.	
-028	There is no internal table space in RFAM.	
-029	Internal RFAM tables are invalid.	
-053	Illegal record size.	
-056	Illegal request.	
-057	System table error.	
-059	Illegal or missing parameters.	

Error Codes and Messages: Remote FMP Access

If condition code = CCL:

- 3 Not enough parameters.
- 5 Parameter address
 violation at HP 3000.
- 72 Invalid DS line.

DEXEC

DEXEC Errors

If condition code is CCG these four-character ASCII codes are returned in abreg, a double word.

Code: Meaning:

DS03 Illegal record size.

DS06 Illegal request.

DS07 System table error.

DS09 Illegal or missing parameters.

I001 Illegal or missing parameters.

I002 Illegal logical unit.

I003 Logical unit not assigned.

I004 Illegal user buffer.

I007 Call rejected by driver.

I012 LU not available to user's session.

SC01 Missing parameter.

SC02 Illegal parameter.

SC03 Program cannot be scheduled.

SC05 Program cannot be defined.

SC06 There is no resolution code in DEXEC time call.

If condition code = CCL, then a double word abreg can be these binary values:

Code: Meaning:

3 Not enough parameters.

5 Parameter address violation at HP 3000.

72 Invalid DS line.

If condition code = CCE the DEXEC call completed satisfactorily.

DSCONTROL CONSOLE COMMAND

APPENDIX

B

Before establishing a DS/3000 communications link, the console operator's :DSCONTROL command must be used to OPEN a line, so that it is available to DS users. The :DSCONTROL command allows you to enable or disable the DS subsystem on a specific communication link.

For easy reference, this command is shown in the following format:

- SYNTAX Shows the format of the command.
- PARAMETERS Describes the variables in the command.
- OPERATION Describes the command in detail.
- EXAMPLES Shows the command in use.

Syntax

`:DSCONTROL dsdevice;function [;function [;... function]]`

where the command syntax has the following meaning:

function	--	OPEN [,open option [,...open option]] SHUT TRACE,ON [trace options] TRACE,OFF COMP NOCOMP MON [,mon option] MOFF
open option	--	MASTER SLAVE [SPEED=] linespeed
trace options	--	[,[ALL] [, [mask] [, [numentries] [, [WRAP] [,filename]]]]]
mon option	--	DS CS

Parameters

dsdevice	The logical device number or the device class name of the DS/3000 communications device. On your system's I/O configuration listing, the device is back referenced by a pound sign (#) to a previously defined HSI, INP, or SSLC. (Required parameter.)
OPEN	Establishes a communication link with another HP 3000. Makes the line available for remote communication via the DS/3000 Subsystem. (Required parameter.)

SHUT Initiates an orderly line shutdown. Refer to OPERATION for details about the line closing procedure.

MASTER Limits DS/3000 line activity to outgoing requests only. No incoming slave sessions are allowed.

SLAVE Limits DS/3000 line activity to incoming slave requests only; no outgoing master activity is allowed.

NOTE

OPEN, MASTER = SHUT, SLAVE
 OPEN, SLAVE = SHUT, MASTER
 SHUT, MASTER = OPEN, SLAVE
 SHUT, SLAVE = OPEN, MASTER

Default: Both MASTER and SLAVE processing are allowed.

linespeed Transmission rate in characters per second. This parameter is effective only if your system generation for the line controller (HSI, INP, or SSLC) selected SPEED CHANGEABLE. Specify linespeed if yours is a European installation with modems running at half speed, or if the length of cables used for data communications at your site have been changed since the system was configured.

HSI speed: 250,000 (cable lengths less than 1000 ft.)

125,000 (cable lengths greater than 1000 ft.)

INP or SSLC speed: 250, 300, 600, or 1200

Default: System configuration values.

TRACE, ON Activates the TRACE facility to provide a record of communications activities. Trace parameters are positional. The line must already be open, or the OPEN keyword must also be included (to open the line).

ALL Generates trace records for all line activity. Default: Records are written only for transmission errors.

DSCONTROL Console Command

mask An octal number preceded by a percent sign (%nn). Used to select type of trace entries generated. Refer to Appendix D for an explanation of the mask bits. Default: %37 (all except STN).

numentries Decimal integer for the maximum number of entries in a trace record, not greater than 248. Default: 24. (See OPERATION.)

WRAP Trace entries that overflow the trace record overlay the prior trace record entries. Default: Overflow entries are discarded.

filename A name for the trace file. Default: DSTRCxxx.PUB.SYS (where xxx is the ldn of the dsdevice).

TRACE,OFF Deactivates the TRACE facility, so that no records are kept of DS/3000 actions, states, and events. Also closes the trace file.

COMP Activates the data compression facility for all line users. The line need not be open to use COMP.

NOCOMP Deactivates the data compression facility for all line users. The line need not be open to use NOCOMP.

MON | ,DS | Activates internal communication monitoring activity to give additional information on a subsequent cold dump of the system. The line must be open for the use of MON.
 | ,CS |
 -- --

MON Requests monitoring of all levels of activity.

MON,DS Requests monitoring at the DS/3000 level of internal software operation.

MON,CS Requests monitoring at the Communication System level of internal software operation.

Used only for system troubleshooting.

MOFF Deactivates internal DS/3000 monitor records. Line must be open for the use of MOFF.

Operation

Unless :DSCONTROL is issued from the master console, this command requires the user to have CS capability. In addition, all users except the console operator are granted access to :DSCONTROL only if they are ALLOWED to use the command or are ASSOCIATED with the specified DS device.

Only one DS/3000 communications device can be active (OPEN) on a controller at any given time. Once opened (with the :DSCONTROL command), a communications link can be shared by multiple DS/3000 users. It cannot, however, be shared by users of other communications subsystems supported by your system (for example, MRJE/3000). Thus, you must SHUT a DSLINE before it can be opened for use by another subsystem.

Before issuing a :DSCONTROL command, use the :SHOWDEV command to check whether a communications link is already established. The ldn for the INP, SSLC, or HSI port will be UNAVAILABLE if the communication link is in use by any subsystem; the ldn for a DS communication pseudo device (driver IODS0) will be AVAILABLE if it is currently OPEN for use by DS/3000 users.

If a DS device class includes more than one DS device, the functions specified in the :DSCONTROL command apply to all devices in that class. If you have not been ALLOWED to use this command, you can only control those devices in the device class with which you have been ASSOCIATED (if any).

If you include more than one function in a :DSCONTROL command, each function (with its subparameter list) must be separated by a semicolon. A function that duplicates or conflicts with a previous function overrides that function. Functions can appear in any order but are executed in the following order:

1. OPEN/SHUT
2. TRACE
3. MON/MOFF
4. COMP/NOCOMP

The SPEED= keyword in the open option may be omitted from a :DSCONTROL command. For example, the following two commands have exactly the same effect:

```
:DSCONTROL 60;OPEN,MASTER,SPEED=25000
```

```
:DSCONTROL 60;OPEN,MASTER,25000
```


DSCONTROL Console Command

If no trace file exists when you turn on the trace facility and you do not specify numentries, the system creates a file to hold 24 entries in each record. The file is:

```
DSTRCxxx.PUB.SYS
```

where xxx is the logical device number of the dsdevice

If you want to change numentries, purge the current trace file and use the :DSCONTROL command to specify a new value. A new file will be created using the specified value.

The SHUT parameter initiates an orderly line closing procedure. If no sessions or applications are using the line when you shut it, line disconnection occurs immediately. If any user (including applications) has the line open, the line remains connected until all sessions and applications CLOSE the line, or until those accessing the line terminate or are aborted. Once closed by the console operator, no new users may access the line until the operator reopens it.

Examples

To open DS line number 55, thereby making it available for use by the DS/3000 Subsystem, enter:

```
:DSCONTROL 55;OPEN
```

To permit the local HP 3000 to process only master (outgoing) requests on DS line number 55, enter:

```
:DSCONTROL 55;OPEN,MASTER
```

To activate the CS Trace facility for DS line 55 (the line is already open), enter:

```
:DSCONTROL 55;TRACE,ON
```

To open DS line 55 and activate CS Trace with a maximum of 250 entries in a trace record, enter:

```
:DSCONTROL 55;OPEN;TRACE,ON,,,250
```

To open the line named REMSYS and provide compression and internal monitoring, enter:

```
:DSCONTROL REMSYS;OPEN;COMP;MON
```

SYSTEM VERIFICATION TEST

APPENDIX

C

Both the system software and the physical link connecting the computers can be tested with a diagnostic program named DSTEST. DSTEST conducts a simple effective test of the system using Remote File Access (RFA). With DSTEST you can select the number of passes, the word pattern to be transmitted, the mode of transmission, and the block size.

NOTE

To Perform DSTEST you must be connected to a remote HP 1000 system.

To execute the diagnostic program, enter the following command to initiate the line test:

:RUN DSTEST.PUB.SYS,DIAG

Then respond to the questions as follows:

- | | |
|--------------------|--|
| .RFA or PTOP? | Enter RFA for Remote File Access when testing the HP 1000 link. PTOP capability is supported to the HP 1000 if the slave(DSTES) is loaded there. |
| .REMOTE COMPUTER? | Enter 1000. |
| .DSLIN? | Enter the device class or logical device number used in the :DSLIN command. |
| .1000 FILE NAME? | Enter a file name of six or fewer characters beginning with an alphabetic character. The file is purged upon completion of DSTEST. |
| .NUMBER OF PASSES? | Enter the maximum number of passes desired, up to a maximum of 32767. A zero or a carriage return causes a default value of one pass to be used. |
| .PATTERN? | Enter an octal word to be transferred. The percent sign indicating octal (%) must be entered. An illegal input causes the message |

INPUT ERROR

System Verification Test

to be printed. Enter a correct value, or a carriage return to indicate a default value, which is %177777.

- .BLOCKSIZE? Enter the desired blocksize of the transfer, less than 512. If a value greater than 512 is entered, an error message is printed.
- .CONTINUE (Y/N)? Enter a Y to select another set of options, or N to terminate the test.

Example

```
:HELLO TEST.DATACOMM
SESSION NUMBER = #S13
MON, AUG 29, 1977, 10:50 AM
HP32002B.01.02

WELCOME TO THE DATA COMM TEST SYSTEM
:DSLLINE HDS2

DSL
```

```
LINE NUMBER = #L3
:RUN DSTEST.PUB.SYS,DIAG

HEWLETT PACKARD 32195A.00.0 DSTEST/3000 MON, AUG 29,
1977 10:50 AM
.RFA OR P
```

```
TOP? RFA
.REMOTE COMPUTER ? 1000
.DSL
```

```
LINE? HDS2
.1000 FILE NAME? TEMP01
.NUMBER OF PASSES? 22
.PATTERN? %11
.BLOCKSIZE? 256
256 WORD REMOTE RECS WRITTEN/READ: 22 ,SECS: 42.012
,AVE: 1.909
.CONTINUE (Y/N)? N

END OF PROGRAM
:BYE

CPU (SEC) = 5
CONNECT (MIN) = 4
MON, AUG 29, 1977, 10:55 AM
END OF SESSION
```

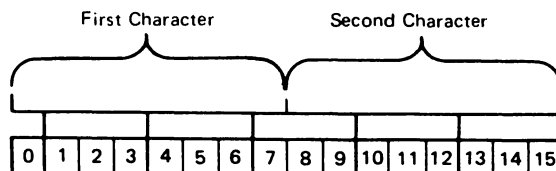
HP CHARACTER SET

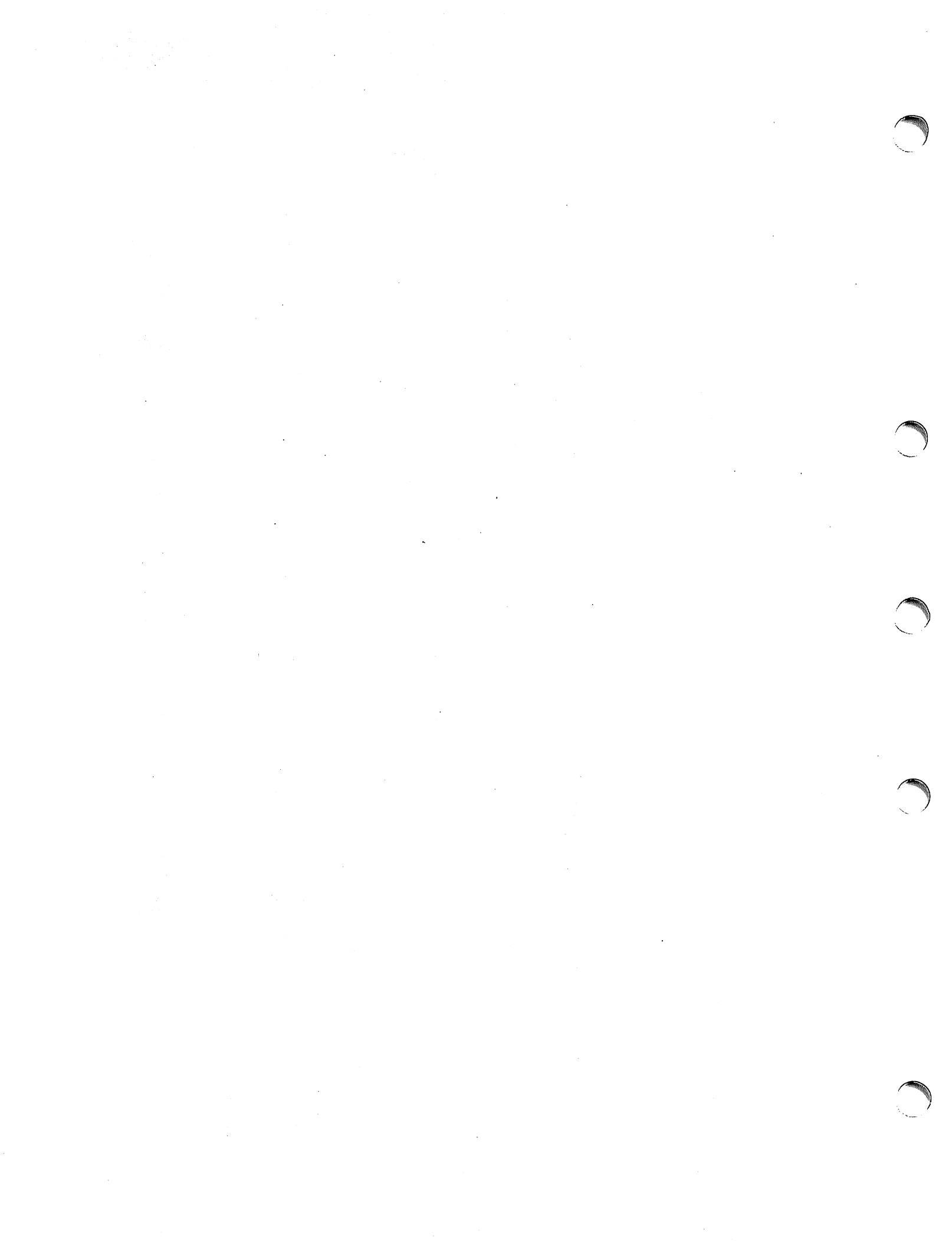
APPENDIX

D

ASCII Character	First Character Octal Equivalent	Second Character Octal Equivalent
A	040400	000101
B	041000	000102
C	041400	000103
D	042000	000104
E	042400	000105
F	043000	000106
G	043400	000107
H	044000	000110
I	044400	000111
J	045000	000112
K	045400	000113
L	046000	000114
M	046400	000115
N	047000	000116
O	047400	000117
P	050000	000120
Q	050400	000121
R	051000	000122
S	051400	000123
T	052000	000124
U	052400	000125
V	053000	000126
W	053400	000127
X	054000	000130
Y	054400	000131
Z	055000	000132
a	060400	000141
b	061000	000142
c	061400	000143
d	062000	000144
e	062400	000145
f	063000	000146
g	063400	000147
h	064000	000150
i	064400	000151
j	065000	000152
k	065400	000153
l	066000	000154
m	066400	000155
n	067000	000156
o	067400	000157
p	070000	000160
q	070400	000161
r	071000	000162
s	071400	000163
t	072000	000164
u	072400	000165
v	073000	000166
w	073400	000167
x	074000	000170
y	074400	000171
z	075000	000172
0	030000	000060
1	030400	000061
2	031000	000062
3	031400	000063
4	032000	000064
5	032400	000065
6	033000	000066
7	033400	000067
8	034000	000070
9	034400	000071
NUL	000000	000000
SOH	000400	000001
STX	001000	000002
ETX	001400	000003
EOT	002000	000004
ENQ	002400	000005

ASCII Character	First Character Octal Equivalent	Second Character Octal Equivalent
ACK	003000	000006
BEL	003400	000007
BS	004000	000010
HT	004400	000011
LF	005000	000012
VT	005400	000013
FF	006000	000014
CR	006400	000015
SO	007000	000016
SI	007400	000017
DLE	010000	000020
DC1	010400	000021
DC2	011000	000022
DC3	011400	000023
DC4	012000	000024
NAK	012400	000025
SYN	013000	000026
ETB	013400	000027
CAN	014000	000030
EM	014400	000031
SUB	015000	000032
ESC	015400	000033
FS	016000	000034
GS	016400	000035
RS	017000	000036
US	017400	000037
SPACE	020000	000040
!	020400	000041
"	021000	000042
#	021400	000043
\$	022000	000044
%	022400	000045
&	023000	000046
'	023400	000047
(024000	000050
)	024400	000051
*	025000	000052
+	025400	000053
,	026000	000054
-	026400	000055
.	027000	000056
/	027400	000057
:	035000	000072
;	035400	000073
<	036000	000074
=	036400	000075
>	037000	000076
?	037400	000077
@	040000	000100
[055400	000133
\	056000	000134
]	056400	000135
^	057000	000136
_	057400	000137
{	060000	000140
	075400	000173
}	076000	000174
~	076400	000175
DEL	077000	000176
	077400	000177





This section contains information regarding applications on the DS 3000 to 1000 communications link. It provides various DS 3000 to 1000 specifications.

Line Buffer Configuration Limits

SOFTWARE

HP 3000 304-4096 (words)

HP 1000 The software is loaded using one of the 3 available Libraries.

304 (words) *
1072 (words) **
4096 (words) *

* Continuation records will be used when the actual transmitted data exceeds the line buffer size.

** No continuation records will be used. This limits the data to approximately 1024 words.

The entire transfer includes users data, plus the DS fixed part of the message format, plus an optional appendage.

HARDWARE

HP 3000 INP 304-1024 (words)

 HSI 304-4096 (words)

 SSLC 304-4096 (words)

HP 1000 PSI 2048 (words)

Applications

Opening a Line from the 3000

To open the DS line from the 3000 side, a `:DSCONTROLxx;OPEN` is done from a system console. Then, the command `:DSLINExx` is done to open a line to the 1000. If the 1000 is not ready to accept a DSLINE, this will fail, and the message 'REMOTE NOT RESPONDING' will appear. At this point the 1000 operator must be instructed to open DS line from the 1000 side. To close a DS line, the command `:DSLINExx;CLOSE` is done. If an attempt at opening a DS line from the 3000 side fails, a `:DSLINExx;CLOSE` must be done before another attempt at opening the line can be accomplished.

Opening a Line from the 1000

To open a line from the 1000 side, type the command `:RU,DINIT`. This will schedule QUEX, a program which exchanges line open information with the 3000. After a period of time (if the line fails), QUEX goes into a secondary state, and will accept an incoming connection with a 3000. The 3000 side operator must type in a `:DSCONTROLxx;OPEN`, before a line from the 1000 can be opened. To reinstate the 1000 side as a primary station (allowing you to make the connection from the 1000 side), type the command `DSMOD/L`.

Session

HP 1000's have the ability to offer users some isolation from other users through a capability called SESSION. SESSION provides a unique environment for each user, where capability levels, private disc space, device access, etc. can be specified for each logon account. For instance, if a user wishes to access a certain file which resides in a private disc area, the user must logon to an account which has access to that area of the disc.

When accessing an HP 1000 (with Session Monitor installed) from an HP 3000, it is necessary to make the access under a session. This is done transparently to the HP 3000 user by providing a default logon for requests received from the HP 3000. The capability level, disc space, device access, etc. is set up for that default session by the HP 1000 Network Manager. This default logon is generally given low capabilities as a security measure. For most HP 3000 users, access will be made by interactively (HELLO), or programmatically (COMMAND HELLO) logging on to a specific account on the HP 1000. The HP 3000 user then has the capabilities of that specific logon. For example, if a HP 3000 user wishes to access data stored in a private file on the HP 1000, the user will have to issue a logon to a specific account that has access to that file before attempting to access the file with remote file access calls.

It is important to note that the session established on the HP 1000 is a non-interactive session. There is no FMGR associated with the session; so the only commands that can be processed are Rte system level commands, and DS level DEXEC, RFA, PTOP, and Utility calls.

Running FMGR on the HP 1000

The only method for issuing remote FMGR commands on the HP 1000 is to place a FMGR transfer file on the HP 1000, and then from REMOTE issue the command 'RU,FMGR,trfile'. FMGR will run on the HP 1000, open the transfer file and execute the commands that it contains.

NOTE

The request to 'RU,FMGR' causes the actual FMGR program to run (not a copy of it). This will tie up FMGR until the transfer file has completed, and keep other users on the HP 1000 from logging on (they will not be able to get a copy of FMGR for their session). You must also be very careful to 'EX,SP' from FMGR at the end of the transfer file or FMGR will be tied up to your session until you terminate it (BYE), or it times out. Exercise caution when running FMGR from the HP 3000.

Applications

The 'MO've Command

DS/1000-IV provides the ability to move files between HP 1000's and HP 3000's with a simple user command from RMOTE (the user interface program to the HP 3000). This capability requires master-slave software to be installed in both computers. On the HP 1000 side:

1. Choose %RMOT1 for the RMOTE program.
2. Initialize DS to the HP 3000
3. Have access to the file !COPY3
4. Load the program %MVCP3
5. Run RMOTE, Switch to the HP 3000, and logon to the System Manager's account
6. Switch back to the HP 1000 and from RMOTE

MVCP3 installs !COPY3 (absolute code) on the HP 3000 as COPY3K.PUB.SYS. Now whenever the Move command is issued from RMOTE, COPY3K.PUB.SYS acts a slave and assists in the file transfer. The Move command converts HP 3000 file format to HP 1000 file and vice versa (ASCII ONLY). Numerical data must still be processed in order to be exchanged between HP 1000's and HP 3000's .

PTOP Testing Using DSTEST

DS/1000-IV provides a slave program, DSTES (%DSTES), which can be loaded on the HP 1000 for verification of the DS PTOp software. Load DSTES, and be sure that it is 'RP'ed (given an ID segment). From the HP 3000, open the DSLINE to the HP 1000 and run DSTEST for verification of the PTOp software.

Abbreviations, device, 4-12
 ACCEPT, 6-21
 ASCII character set, D-1

Basic operating principles, HP 3000, v

Closing a line, 2-4
 Coaxial cables, interfacing, 1-3
 Communication link errors, A-1
 Communication overview, HP 3000 to HP 1000, 1-4
 Communication types, HP 3000 to HP 1000, 1-1
 Condition codes
 ACCEPT, 6-16
 GET, 6-14
 PCHECK, 6-19
 PCLOSE, 6-13
 PCONTROL, 6-11
 POPEN, 6-5
 PREAD, 6-7
 PWRITE, 6-9
 REJECT, 6-18
 Connections, hardware, 2-1
 Cross reference, RFA and FMP, 5-3
 Customer engineer, 2-1

DAPOS, 5-4
 DCLOS, 5-6
 DCONT, 5-8
 DCRET, 5-11
 Device abbreviations, 4-12
 DEXEC call elements, 4-3
 DEXEC call summary, 4-2
 Diagnostic program, C-1
 DLOCF, 5-13
 DNAME, 5-15
 DOPEN, 5-16
 DPURG, 5-20
 DREAD, 5-21
 DSCONTROL operator command, B-1
 DS line number, 1-3
 DSLINE command
 closing a line, 2-4, E-2
 device class name, 2-2
 enable subsystem on link, 2-1
 error numbers, 2-3
 format, 2-2
 opening a line, 2-3, E-2
 parameters, 2-2
 DSTEST, C-1
 DWIND, 5-23
 DWRITE, 5-24
 DSTAT, 5-22

Index

EDITR subsystem, 3-5

Errors

- codes and messages, A-1
- console operator, B-1
- DEXEC, A-11
- line opening failures, A-6
- remote FMP access, A-6

Establishing the communication link with HP 1000 systems, 2-1

Examples

- DSCONTROL, B-3
- DS line buffer, 1-8
- master and slave programs, 6-22, 6-25
- multiple user, 1-6
- multiple use, multiple line, 1-7
- opening DS lines, 3-3
- REMOTE command, series, 3-4
- REMOTE command, single, 3-3
- remote file access, 5-23

Executable RTE operator commands, 3-4

Executing remote EXEC calls (DEXEC calls), 4-1

Execution time, DEXEC call

- initial offset, 4-17

Executing remote operator commands, 3-1

File management package (FMP), 3-5

FMGR, 3-6

Function, DEXEC calls, 4-2

Functional return

- DEXEC, 4-2
- GET, 6-14
- I/O control, 4-11
- PCEHCK, 6-19
- POPEN, 6-5
- PREAD, 6-7
- program schedule, 4-14
- READ, 4-5
- WRITE, 4-8

GET, 6-18

Hardware connections, 2-1

Hardwired communications line, 1-3

Hardwired serial interface (HSI), 1-3

Instruction code, DEXEC calls, 4-2

Interface with remote systems programmatically, 4-1

Intrinsics

- ACCEPT, 6-16
- cross reference, RFA and FMP, 5-3
- DAPOS, 5-4
- DCLOS, 5-6

DCONT, 5-8
DCRET, 5-11
DLOCF, 5-13
DNAME, 5-15
DOPEN, 5-16
DPOSN, 5-19
DPURG, 5-20
DREAD, 5-21
DWIND, 5-23
DWIT, 5-24
DSTAT, 5-22
GET, 6-14
master program-to-program, 6-2
PCHECK, 6-19
PCONTROL, 6-11
PCLOSE, 6-13
POPEN, 6-5
PREAD, 6-7
PWRITE, 6-9
REJECT, 6-18
Introduction, 1-1
I/O control, DEXEC call, 4-11
I/O status, DEXEC call, 4-19

Limitations, remote operator commands, 3-5
Line number, DS, 1-3
Log on, 2-1
Line opening failures, 2-3
Line opening failures, error numbers and messages, A-6
LSTEN, 2-1

Master program on HP 3000, 6-27
Master program-to-program intrinsics, 6-2

Network manager, 2-1
Notes
DCONT, 5-8
DCRET, 5-11
device abbreviations, 4-12, 5-8
DOPEN, 5-16
DSLIN, B-3
modifying RTE clock, 3-5
not using IOWAIT (0) calls, 6-14
POPEN reject, 6-5
terminating slave program unconditionally, 6-11
truncating file size, 5-5

Opening a line, 2-1

Index

Parameters

- ACCEPT, 6-16
- DAPOS, 5-4
- DCLOS, 5-6
- DCONT, 5-8
- DCRET, 5-11
- DLOCF, 5-13
- DNAME, 5-15
- DOPEN, 5-16
- DPOSN, 5-19
- DPURG, 5-20
- DREAD, 5-21
- DSLIN, B-1
- DSTAT, 5-22
- DWIND, 5-23
- DWRIT, 5-24
- execution time
 - initial offset, 4-17
- GET, 6-14
- I/O control, 4-11
- I/O status, 4-19
- PCHECK, 6-19
- PCLOSE, 6-13
- PCONTROL, 6-11
- POPEN, 6-5
- PREAD, 6-7
- program schedule, 4-14
- PWRITE, 6-9
- READ, 4-5
- REJECT, 6-18
- time request, 4-16
- WRITE, 4-8

PCHECK, 6-19

PCONTROL, 6-11

PCONTROL activity, 6-11

PCLOSE, 6-13

POPEN, 6-5

POPEN activity, 6-6

PREAD, 6-7

PREAD activity, 6-8

program schedule, DEXEC call, 4-14

program-to-program communications, 6-1

programmatic access, RFA, 5-1

PWRITE, 6-9

PWRITE activity, 6-10

REJECT, 6-18

REMOTE command, 3-2

Remote file access, 5-1

Remote FMP access errors, A-7

Remote processing, 2-3

RFA and FMP cross referencing, 5-3

RTE operator commands, remote process, 3-4

READ, 4-4

Slave program on HP 3000, 6-25
Slave program-to-program intrinsics, 6-3
Summaries
 communications link capability, 1-1
 DEXEC calls, 4-2
 HP 3000 computer system manuals, v
 master program-to-program intrinsics, 6-2
 remote processing capability, 4-1
 RFA and FMP cross reference, 5-3
 slave program-to-program intrinsics, 6-3
Superscripts
 DEXEC call, 4-3
 PTOP, 6-4
System operator, 2-1
System supervisor, 2-1
System verification test, C-1

Time request, 4-16
Test, system verification, C-1

Users
 manuals, v
 several, 1-3

WRITE, 4-8



Part No. 32190-90005
Printed in U.S.A. 1/82
E0182

