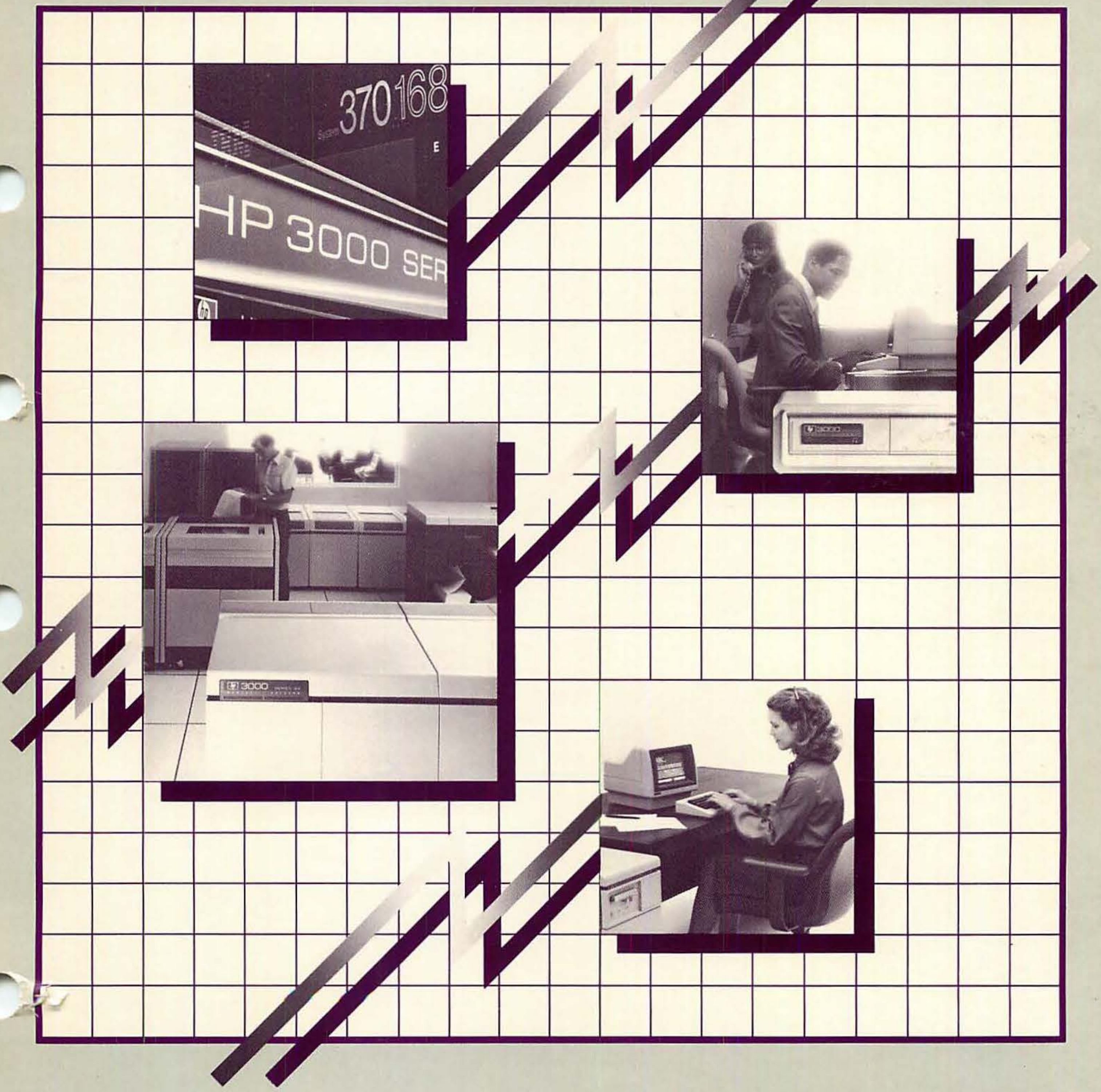
Distributed Systems Network

HEWLETT PACKARD

DSN/DS HP 3000 to HP 3000 Reference Manual



HP Distributed Systems Network

DSN/DS*

HP 3000 to HP 3000

Reference Manual

*(formerly DS/3000)

HEWLETT
PACKARD

INFORMATION NETWORKS DIVISION
19420 Homestead Road, Cupertino, California 95014

Copies of this manual may be ordered through a local Hewlett-Packard sales office. Refer to the pages at the back of the manual for addresses.

Part No. 32190-90001
Product No. 32190A
E0982

Printed in U.S.A. 9/82

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.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates remain.

Second Edition........................September 1982

PRINTING HISTORY

New editions are complete revisions of the manual. 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 of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

The software product part number printed alongside the date indicates the version and update level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

First Edition	Mar 1977	32190A.00.00
Update No. 1	May 1977	32190A.00.00
Update No. 2	May 1978	32190A.02.01
Updates 1 and 2 Incorporated	Feb 1979	32190A.02.02
Update No. 3	Nov 1979	32190A.02.05
Update No. 4	Apr 1980	32190A.02.06
Updates 3 and 4 Incorporated	Aug 1980	32190A.02.06
Update No. 5	May 1981	32190A.03.00
Second Edition	Sep 1982	32190A.04.01

PREFACE

The Hewlett-Packard Distributed Systems Network (HP-DSN) is a set of hardware and software data communications products. One of these data communications products is DSN/Distributed Systems (DSN/DS) which is an integrated software package that provides the capability of communication between HP computer systems.

This manual documents DSN/DS as it applies to an HP 3000 network. (This HP 3000-to-HP 3000 application of the DSN/DS software subsystem was identified in the previous edition of this manual as DS/3000.) The manual explains how an HP 3000 user can communicate with another (or several other) HP 3000 computer systems by establishing a DSN/DS communications link. (Other manuals in the DSN/DS series document the other network combinations of computer types.)

This dual-purpose manual serves as both a reference manual for experienced users of HP DSN/DS and a tutorial text for new HP DSN/DS users. A new user should be familiar with the basic operating principles of the HP 3000 Computer System using the MPE Operating System and should also be knowledgeable in the subjects of the following manuals:

- HP 3000 Computer Systems, MPE Commands Reference Manual (30000-90009).
- HP 3000 Computer Systems, MPE Intrinsics Reference Manual (30000-90010).
- HP 3000 Computer Systems, System Manager/System Supervisor Reference Manual (30000-90014).
- HP 3000 Computer Systems, Console Operator's Guide (32002-90004).
- HP 3000 Computer Systems, Communications Handbook (30000-90105).

For those users who also become involved in the selection and/or connection of the various network components, reference should be made to the appropriate component manuals, including the following:

- HP 30010A Intelligent Network Processor (INP) Installation and Service Manual (30010-90001).
- HP 30020A Intelligent Network Processor (INP) Installation and Service Manual (30020-90001).
- HP 30020B Intelligent Network Processor (INP) Installation and Service Manual (30020-90005).

- HP 30010A/30020A/B Intelligent Network Processor (INP) Diagnostic Procedures Manual (30010-90002).
- HP 30055A Synchronous Single-Line Controller (SSLC) Installation and Service Manual (30055-90001).
- Hardwired Serial Interface (HSI) Installation and Service Manual (30360-90001).

NOTE

Within the text of this manual, cross-references are made to other manuals by title. To obtain the part number of the referenced manual, refer to these lists of manuals in the Preface.

This second edition of the DSN/DS Reference Manual not only incorporates all of the various updates that were issued since the first edition was published, but it also includes some new material. The Network File Transfer (NFT) documentation (that was appended to Section III in Update No. 5) has now been reformatted as Section VI. Another new section -- Section VII, DS Applications -- has been added to expand upon the coverage of the advanced uses and networking possibilities available with the enhanced DSN/DS. The new Appendix H documents the X.25 Network Configurator; and the new Appendix J is a commentary on using Public Data Networks (PDNs) with the new DS/X.25 capability, as well as the X.29 Packet Assembler/Disassembler (PAD) that expands the system-to-system X.25 capability to include a terminal-to-system capability. Appendix A, Configuration Dialogue, has also been updated to include the new driver names required for the utilization of DS/X.25. Additionally, all of the previously existing sections and appendices have been edited and brought up to date.

CONTENTS

CONTENTS (continued)

Section 5 - PROGRAM-TO-PROGRAM COMMUNICATIONS

Section 6 - NETWORK FILE TRANSFER

Section 7 - DS APPLICATIONS

Appendix B - ERROR CODES AND MESSAGES

Appendix C - DSCONTROL CONSOLE COMMAND

Appendix D - DSDUMP

CONTENTS (continued)

Appendix E - SYSTEM VERIFICATION

Appendix F - DSN/DS COBOL INTERFACE

Appendix G - DSN/DS BASIC INTERFACE

CONTENTS (continued)

Appendix H - X.25/X.21 NETWORK CONFIGURATOR

Appendix J - USING X.25/X.29 CAPABILITIES

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

TABLES

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

{ } 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.

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

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

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

Example: NEW NAME? <u>ALPHA1</u>

superscript C Control characters are indicated by a superscript C

Example: Y^C

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.

INTRODUCING DSN/DS

SECTION I

The Hewlett-Packard Distributed Systems Network (HP-DSN) is a combination of hardware and software products that make it possible for Hewlett-Packard computer systems to communicate with one another, and with IBM mainframes as well. The connections can be made over hardwired lines, and/or over the public telephone facility, and/or across Public Data Networks (PDNs), in any mixture. This capability, coupled with our proven remote entry capability to IBM computer systems, provides a total solution to large-company electronic data processing (EDP) needs.

Within the realm of HP-DSN is the software subsystem that accomplishes computer-to-computer communication over these connecting lines. This software subsystem is called DSN/Distributed Systems (DSN/DS). Among other features, DSN/DS includes such capabilities as:

- **Remote File Access.** A user is allowed access to files in remote HP computer systems. An important aspect of this feature is the capability of using Interprocess Communications (IPC) between systems.

- **Remote Data Base Access.** A user can directly access data bases on any remote HP computer under the same security protection used by local data bases.

- **Program-to-Program Communication.** Permits programs residing in different HP computer systems to interactively exchange information with one another in a coordinated manner.

- **Virtual Terminal.** Gives the user remote interactive capabilities, even though the user's terminal is physically connected to the local HP system.

- **Remote Command Execution.** Allows the user to issue commands to a remote HP system as if the local terminal were connected directly to the remote system.

- **Network File Transfer.** A facility that efficiently transfers disc files between HP computer systems.

- **X.25/X.21.** Gives communication capability across X.25 packet switching and X.21 circuit switching networks to the DSN/DS user.

But exactly what does this overall capability mean? It means that a large multidivisional corporation can have a truly coordinated world-wide network of computer systems. They are coordinated in the sense of tying together the various commercial and industrial functions within each division and factory, and they are also coordinated in the larger sense of tying together the various divisions and factories at the corporate level.

For example, imagine a large corporation which has factories in the United States, Canada, France, and West Germany. Within each factory there are HP 3000 computer systems performing such functions as inventory control, factory data collection, and operations management. With a Hewlett-Packard Distributed Systems Network these manufacturing information systems can be tied into an HP 3000 system which handles the factory's administrative functions (such as finance and accounting). The administrative systems of each factory can, in turn, be connected not only to one another but also (via remote job entry) to a large computer facility at corporate headquarters. This overall networking capability makes it possible to perform financial analysis and control at a group and corporate level as well as at the individual factories.

This manual describes how an HP 3000 user can communicate with several HP 3000 computers by establishing a DSN/DS communications link. DSN/DS is that part of the HP Distributed Systems Network in which several HP 3000 computer systems are connected to one another. DSN/DS can also be used for intercomputer communications with other families of computers (such as HP 3000/HP 1000, HP 3000/HP 250, and HP 3000/HP 98xx desktop computers), but these other combinations are described in separate reference manuals.

As a simplified example of a computer network, imagine that you are in the same room with an HP 3000 (labeled "System A" in figure 1-1) and that another HP 3000 (labeled "System B") resides in another part of the building. These two computers are connected to one another by an interconnecting cable and a pair of communications interfaces. By virtue of DSN/DS you can use the processing capability of both of these HP 3000 machines and pass data back and forth between them by entering commands through a single terminal.

To see how DSN/DS works in this simple example, follow through the step-by-step procedure.

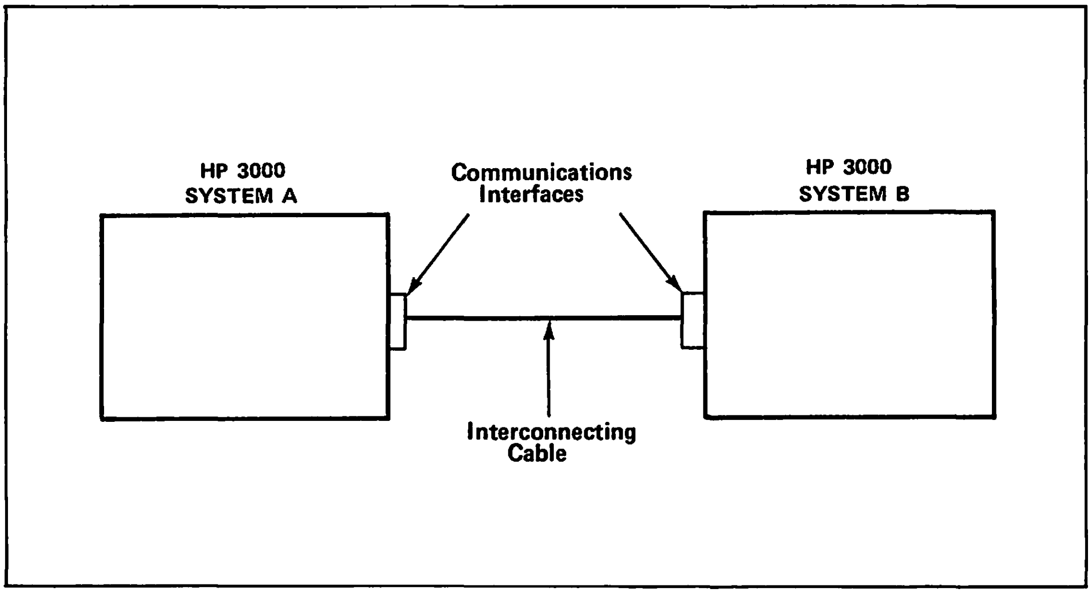


Figure 1-1. HP 3000 to HP 3000 Example

Step 1. Sit down at a terminal connected to System A and initiate a session.

```
carriage return
:HELLO USER.ACCOUNT

 HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:05 AM


WELCOME TO SYSTEM A.
:
```

Within the context of DSN/DS, such a session is referred to as a "local" session because it is active within the HP 3000 to which your terminal is directly connected. This terminology becomes more meaningful later, since all you have actually done, so far, is initiate a standard MPE session. At this point, you have reached the situation illustrated in figure 1-2.

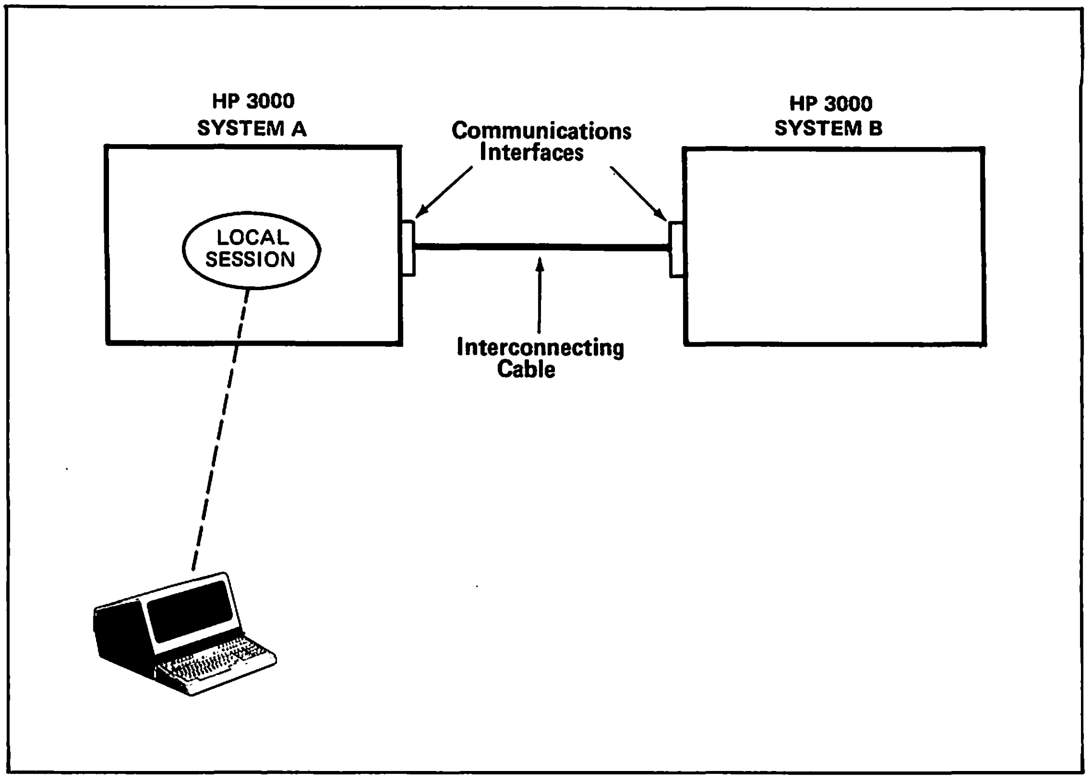


Figure 1-2. Initiating the Local Session

Step 2. Now, open a communications line between System A and System B. Do this by entering a DSLINE command.

```
    :DSLINE HDS2

    DS LINE NUMBER = #L3

    :
```

In this example, HDS2 is the device class name established during system configuration (in System A) for the particular line you wish to use. DSN/DS opens the line and then assigns you a line number (3 in this example). This 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. Within your local session, it uniquely identifies the particular line that you have opened. This becomes significant only if you must open more than one communications line during a session.

Step 3. Now that you have acquired access to a communications line between System A and System B, initiate a session in System B (from your local log-on terminal). Do this by entering a REMOTE command which includes an MPE HELLO command for the remote system.

```
     :REMOTE HELLO RUSER.RACCOUNT

      HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:08 AM

     WELCOME TO SYSTEM B.

     :
```

Within the context of DSN/DS, this type of session is referred to as a "remote" session because it is active within the remotely located HP 3000 that is connected indirectly to your log-on terminal by way of a communications line and your local HP 3000. You now have two distinct sessions in progress concurrently: one in System A (under the user and account names USER.ACCOUNT) and the other in System B (under the user and account names RUSER.RACCOUNT). It is important to keep in mind that within System A your local session is operating under the capabilities and security restrictions defined (by the accounting structure of System A) for USER.ACCOUNT, while within System B your remote session is operating under the capabilities and security restrictions defined (by the accounting structure of System B) for RUSER.RACCOUNT. At this point, the situation is as illustrated in figure 1-3. As will be seen in the next few steps, you can alternate freely between the two sessions.

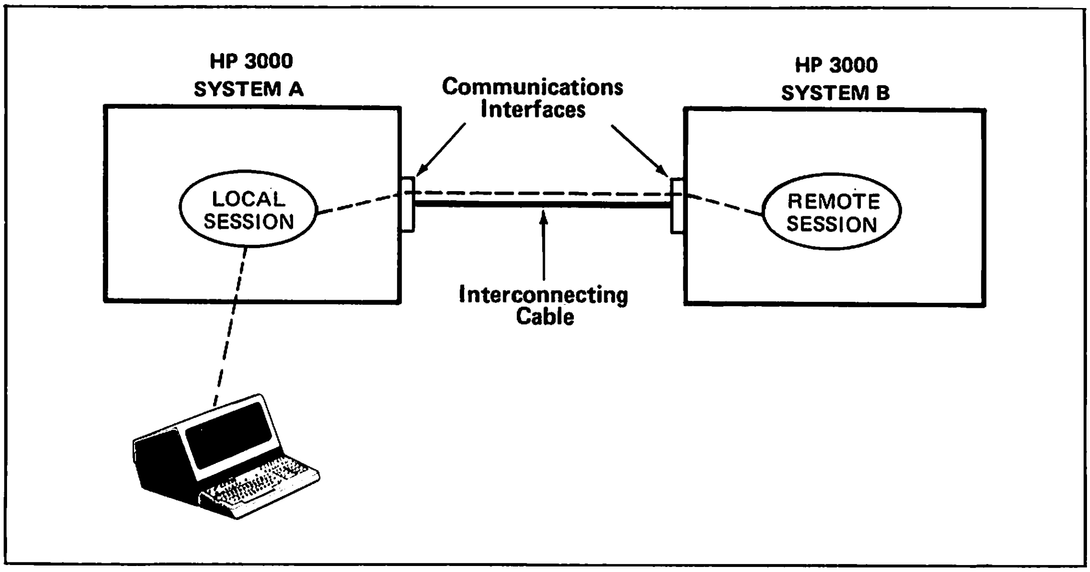


Figure 1-3. Initiating the Remote Session

Step 4. Now, see what files reside in the home group of the ACCOUNT account in System A.

```
:LISTF

FILENAME

DATA1    DATA3    FILE1    SOURCE2    SOURCE5
:
```

You can do the same for the home group of the RACCOUNT account in System B by entering the following command through the same terminal:

```
:REMOTE LISTF

FILENAME

DATA1    DATA5    DATA6    FILE3    SOURCE1
:
```

Notice that in both cases the same command was entered, but in the latter case the prefix REMOTE was used. The presence or absence of that prefix is what determines whether a command is to be executed in the local session or in the remote session.

Step 5. As a result of the LISTF and REMOTE LISTF displays, you can see that a source file, named SOURCE1, exists in System B but not in System A. Suppose you wish to modify one of the statements in that program. To do that, use the text editor in System B. This time, instead of prefixing your remote commands with REMOTE, try a different technique. Enter the following:

```
:REMOTE
#
```

This construct gets into the remote session in such a way that all commands can be entered in their normal form (without the prefix REMOTE). The # is the prompt character issued by DSN/DS (in place of the usual MPE colon prompt). In all other respects it will seem as though you are executing a normal MPE interactive session.

Step 6. Now invoke the text editor, copy the content of SOURCE1 (which is a file in System B) into the editor's work file, display the content of the work file, modify the desired statement, and store the altered source code back in SOURCE1.

```
#EDITOR
HP32201A.7.05 EDIT/3000  WED, MAR  5, 1980,  3:47 PM
(C) HEWLETT-PACKARD CO. 1979
/SET FORMAT=COBOL
/T SOURCE1
/LIST ALL
    1         $CONTROL USLINIT,SOURCE
    1.1        IDENTIFICATION DIVISION.
    1.2        PROGRAM-ID. COBOL-TEST1.
    1.3        ENVIRONMENT DIVISION.
    1.4        DATA DIVISION.
    1.5        WORKING-STORAGE SECTION.
    1.6        77  EDIT-FIELD     PIC $$$$$9.99.
    1.7        77  TOTAL-COST     PIC 999V99.
    1.8        77  COST-OF-SALE   PIC 99V99.
    1.9        77  TAX            PIC 99V99.
    2          77  Y-N            PIC X.
    2.1        PROCEDURE DIVISION.
    2.2        ENTER-ROUTINE.
    2.3            MOVE ZEROS TO TOTAL-COST.
    2.4            DISPLAY SPACE.
    2.5            DISPLAY "ENTER COST OF SALE".
    2.6            ACCEPT COST-OF-SALE.
    2.7            COMPUTE TAX = COST-OF-SALE * .06.
    2.8            ADD COST-OF-SALE, TAX TO TOTAL-COST.
    2.9            MOVE TOTAL-COST TO EDIT-FIELD.
    3              DISPLAY "TOTAL COST= " EDIT-FIELD.
    3.1            DISPLAY "ARE YOU FINISHED? (Y OR N)".
    3.2            ACCEPT Y-N.
    3.3            IF Y-N = "N" OR "n" GO TO ENTER-ROUTINE.
    3.4            STOP RUN.
/MODIFY 2.5
MODIFY     2.5
     DISPLAY "ENTER COST OF SALE".
                                I (NO DECIMAL POINT)
     DISPLAY "ENTER COST OF SALE (NO DECIMAL POINT)".

/KEEP SOURCE1
SOURCE1 ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD? YES
/EXIT

END OF SUBSYSTEM
#
```

Step 7. The work in System B is now completed; so terminate the remote session and return control to your local session.

```
#BYE

CPU=4. CONNECT=7. WED, MAR  3, 1982,  9:15 AM
#:
:
```

Note that you are now back in the local session in System A (signified by the colon prompt). The remote session no longer exists, but the communications line is still open. You could, if you wanted, initiate another remote session over the line by issuing another REMOTE HELLO command. To close the communications line, enter the following variation of the DSLINE command:

```
:DSLINE HDS2 ;CLOSE
1 DS LINE WAS CLOSED.
:
```

Finally, terminate the local session.

```
:BYE

CPU=1. CONNECT=11. WED, MAR  3, 1982,  9:16 AM
```

THE COMMUNICATIONS LINK

SECTION II

WHAT IS A COMMUNICATIONS LINK?

Within the context of DSN/DS, a "communications link" consists of the following elements:

- A normal interactive session in progress in an HP 3000 computer.
- A physical communications line between that HP 3000 computer and another HP 3000 computer at a remote location.
- An interactive session in progress in the remote HP 3000 computer (initiated over the physical communications line from your local session).

Note that your local terminal is the log-on terminal for both the local session and the remote session. (Refer to figure 2-1.)

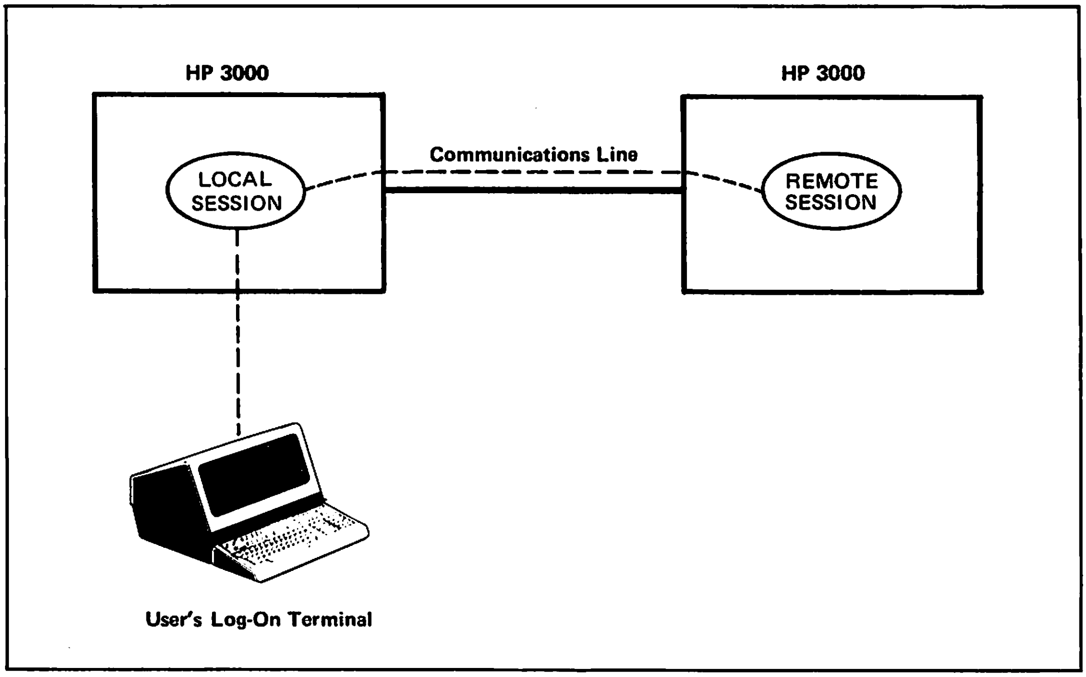


Figure 2-1. DSN/DS Communications Link (HP 3000 to HP 3000)

OPENING A LINE

A communications link can be established over a hardwired communications line, over the public telephone network, or over an X.21 or X.25 Public Data Network (PDN). The procedures for opening hardwired lines and for opening telephone lines differ only slightly. Therefore, the basic differences will be presented first, followed by the procedures that are essentially the same. Generally, once the connection to the remote computer is established, you will perceive no difference in the way DSN/DS performs.

Opening a Hardwired Line

What is a hardwired line? In the general field of data communications there are two types of lines commonly referred to as "hardwired". The first type is a dedicated path on the public telephone network that is leased from the telephone company for the private use of a computer-to-computer configuration. Such a line serves as a permanent connection between the two computers. The other type of hardwired line is a cable that is connected directly to the communications I/O interfaces of the two computers. Within the context of DSN/DS, "hardwired" always refers to the latter. However, the technique used for opening a line is the same for either a direct-connect line or a leased (nonswitched) telephone line.

The hardwired interconnecting cable connects to each HP 3000 by way of a communications interface. The communications interfaces that can be used for a hardwired connection include the HP 30010A, HP 30020A, and HP 30020B Intelligent Network Processor (INP), the HP 30055A Synchronous Single-Line Controller (SSLC), and the HP 30360A Hardwired Serial Interface (HSI). (Although the INPs and the SSLC are the interfaces most commonly used for telephone line connections with modems, they can also be used in hardwired applications without modems.) The HP 30010A INP is used with the HP 3000 Series II/III; the HP 30020A INP is used with the HP 3000 Series 30/33/40/44; the HP 30020B INP is used with the HP 3000 Series 30/33/40/44/64; the HP 30055A SSLC is used with the HP 3000 Series II/III; and the HP 30360A HSI is used with the HP 3000 Series II/III.

It is relatively straightforward to obtain access to a hardwired communications line. All you are required to do is identify the particular communications interface you wish to use. You do this by specifying the device class name or logical device number associated during system configuration with the desired interface. In the example in Section I, the DSLINE command was used for this purpose, as follows:

```
     :DSLINE HDS2
```

In the DSLINE command you may also wish to specify the size of the DSN/DS line buffer to be used in conjunction with the line. The size of this buffer determines the maximum amount of data that can be sent or received in a single physical transmission over the line. Note that a transmission as you normally think of it (sending or receiving all or part of a file) may actually consist of many physical transmissions. In essence, this buffer size defines a blocking factor for the line. (Refer to figure 2-2.) A default buffer size is established during system configuration, and in most cases (as in the example in Section I), you will find it satisfactory to let this default value prevail.

Assume that the DSN/DS line buffer size is 512 words and that the user has initiated the transmission of a block of data 1200 words in length from his HP 3000 to a remote HP 3000. The block of data would actually be sent in three separate physical transmissions, as follows. (DSN/DS appends an average of 20 words of overhead on each transmission.)

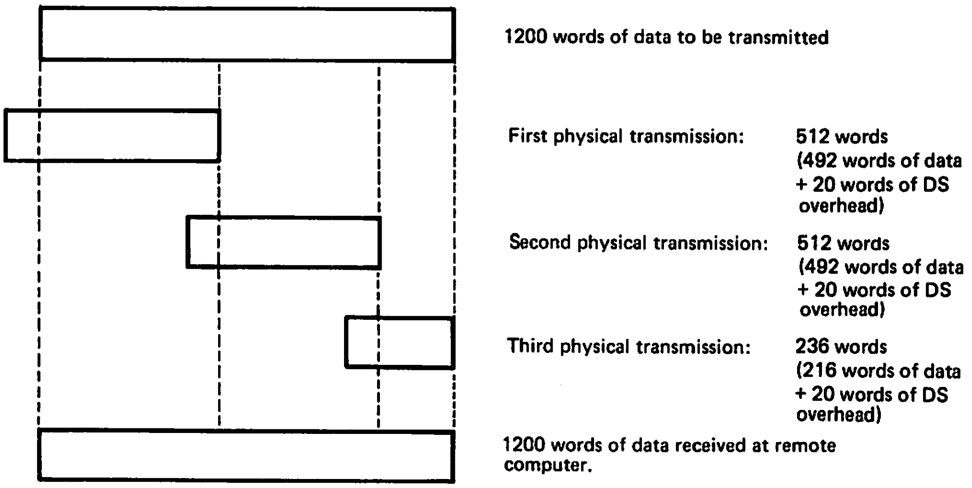

1200 words of data to be transmitted
First physical transmission:
512 words (492 words of data + 20 words of DS overhead)
Second physical transmission:
512 words (492 words of data + 20 words of DS overhead)
Third physical transmission:
236 words (216 words of data + 20 words of DS overhead)
1200 words of data received at remote computer.

Figure 2-2. DSN/DS Line Buffer Example

When you execute a DSLINE command, DSN/DS attempts to give you access to the specified communications line and, if successful, informs you of the assigned DS line number by displaying the following message at your terminal:

```
DS LINE NUMBER = #Lx
```

where x is the assigned DS line number. In the example in Section I, the DS line number "3" was assigned. The DS line number is significant only if you open and use more than one communications line concurrently within a single local session (see "Opening Multiple Lines" later in this section).

At this point you have acquired a physical communications line but the communications link does not yet exist. The actual communications link between the two computers is established by initiating a remote session over the line. You do this by executing a REMOTE HELLO command. In the example in Section I, the REMOTE HELLO command contained the minimum parameters required (a username and an accountname), as follows:

```
:REMOTE HELLO RUSER.RACCOUNT
```

The communications link between the two HP 3000 computers now exists.

Opening a Telephone Line

A DSN/DS communications link can also be established over the public (dial-up) telephone network. In such a case, the information passed back and forth between the two computers travels over the same lines that are used for normal voice traffic. Each computer is interfaced to the telephone lines by way of a modem. (The term "modem" is a contraction of MOdulator-DEModulator.) A modem is a device that translates digital signals (electrical impulses) generated by a computer into analog signals (tones) that can be transmitted over telephone lines, and vice versa.

The modem is connected to the HP 3000 Computer System by way of a communications interface. The communications interfaces used with modems include the HP 30010A, HP 30020A, and HP 30020B Intelligent Network Processor (INP), and the HP 30055A Synchronous Single-Line Controller (SSLC). (The HP 30010A INP is used with the HP 3000 Series II/III; the HP 30020A INP is used with the HP 3000 Series 30/33/40/44; the HP 30020B INP is used with the HP 3000 Series 30/33/40/44/64; and the HP 30055A SSLC is used with the HP 3000 Series II/III.) Each INP or SSLC controls one modem (such as an HP 37210T, 37220T, or 37230A modem, or a Bell System Type 201, 208, or 209 modem) and is capable of both initiating and accepting a telephone connection with a remote computer over the public telephone network or a leased telephone line.

It is a little more complex to obtain access to a telephone line than to a hardwired line. First, you must identify the particular communications interface (INP or SSLC) you wish to use. You do this by specifying the device class name or logical device number of the communication line that was associated during system configuration with the desired interface. You can use the DSLINE command for this purpose, as follows:

```
   :DSLINE SDS1
```

In the DSLINE command, you may also wish to specify the size of the DSN/DS line buffer to be used in conjunction with the line. The size of this buffer determines the maximum sized block that can be sent or received in a single physical transmission over the line. Note that a transmission as you normally think of it (sending or receiving all or part of a file) may actually consist of many physical transmissions. In essence, this buffer size defines a blocking factor for the line. (Refer to figure 2-2.) A default buffer size is established during system configuration, and in most cases (as in the example in Section I), you will find it satisfactory to let this default value prevail.

Next, you may wish to supply a set of identification (ID) sequences to be used in verifying that the desired pair of computers are connected to one another. This is discussed under "ID Sequences" later in this section. Briefly, however, you may supply an ID sequence that identifies your HP 3000 and one or more ID sequences that identify those remote computers with which your HP 3000 may validly be connected. When a telephone connection is established between your HP 3000 and a remote HP 3000, the two computers exchange ID sequences and their validity determines whether or not the connection is to remain in effect. You use the DSLINE command to supply ID sequences, as follows:

```
   :DSLINE SDS1 ;LOCID="SYSTEM A" &
                ;REMID="SYSTEM X"
```

where SYSTEM A is the ID sequence identifying your HP 3000 and SYSTEM X is the ID sequence identifying the remote computer with which you want to establish a telephone connection.

Again, there are default values that can be established during system configuration. In most cases, however, you will at least want to explicitly identify the desired remote HP 3000 to be certain that the proper connection is being established.

Now you must establish the physical connection between the two computers by dialing (at the modem) the telephone number of the remote computer and responding (at the system console) to the dial request. If you wish to have the console operator of your HP 3000 dial the number for you, you may supply the desired number in the DSLINE command and it will be displayed as part of a dial request message at the operator's console. In such a case, you would supply the telephone number as follows:

```
   :DSLINE SDS1 ;LOCID="SYSTEM A" &
                ;REMID="SYSTEM X" &
                ;PHNUM=555-1234
```

If autodial equipment is installed on the SDS1 line, the telephone number supplied in the DSLINE command is used instead of the number configured for the line.

The various possibilities involved in establishing a telephone connection with a remote computer are discussed under "Dialing the Remote Computer" later in this section.

When you execute the DSLINE command, DSN/DS attempts to give you access to the specified communications interface (INP or SSLC) and, if the telephone connection is successfully established, informs you of the assigned DS line number by displaying the following message at your terminal:

```
   DS LINE NUMBER = #Lx
```

where x is the assigned DS line number. In the example in Section I, the DS line number "3" was assigned. The DS line number is significant only if you open and use more than one communications line concurrently within a single local session (see "Opening Multiple Lines" later in this section).

At this point, you have acquired a physical communications line, but the communications link does not yet exist. The actual communications link between the two computers is established by initiating a remote session over the line. You do this by executing a REMOTE HELLO command. In the example in Section I, a REMOTE HELLO command was used that contained the minimum parameters required (a username and an accountname), as follows:

```
   :REMOTE HELLO RUSER.RACCOUNT
```

The communications link between the two HP 3000 computers now exists.

Specifying a Line

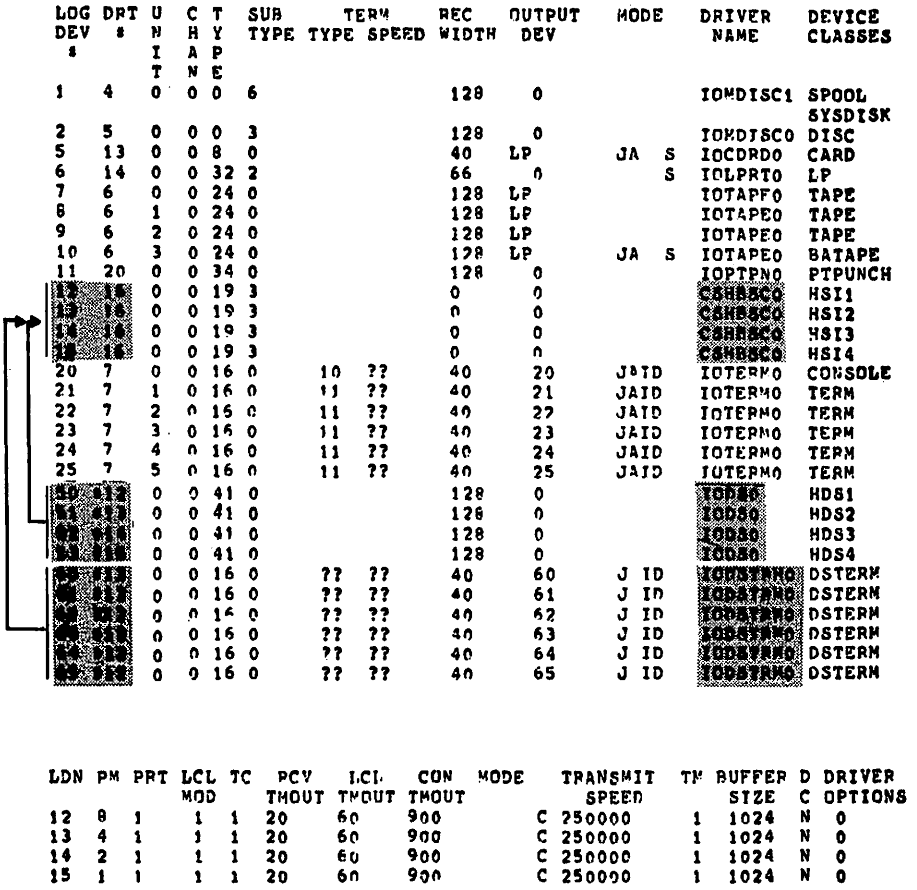
As you have seen, in order to open either a hardwired communications line or a dial-up telephone line, you must specify a device class name or logical device number identifying the particular communication line that is associated with a specific INP, HSI, or SSLC that you wish to use. But how do you figure out which name or number to specify? The remainder of this topic may seem, particularly at first reading, a little complex and tedious. In actual practice, however, once the hardware and software configuration is installed and usable, most DSN/DS sites will post a notice defining all of the available communications lines and the proper device class names and logical device numbers for each. In that case, all of the detective work described in the following paragraphs is already done for you.

For each communications interface, there is a pair of associated drivers. First, there is the actual INP, HSI, or SSLC driver that directly controls the operation of the interface board. In addition, there is a DSN/DS communications driver that controls the operation of the INP, HSI, or SSLC driver. The names of these drivers are as follows:

IOINP0	(INP driver)
CSHBSC0	(HSI driver)
CSSBSC0	(SSLC driver)
IODS0	(DSN/DS communications driver)
IODSX	(DSN/DS communications driver, while utilizing the X.25 capability)

Now look at the appropriate sample I/O device table produced during system configuration (figure 2-3 for a hardwired line or figure 2-4 for a telephone line).

In figure 2-3, the shaded items in the column labeled "DRIVER NAME" shows four HSI lines (CSHBSCO) configured into the system as logical devices 12 through 15. For each one of these lines, there is a DSN/DS communications driver (in this case, IODSO) also configured into the system. Each IODSO (or IODSX) entry is related to the proper HSI (or INP) entry by the number specified in the column labeled "DRT" (the # prefix indicates a back reference to a previously defined logical device number). Logical devices 50 through 53 are paired with logical devices 12 through 15, respectively. It is the device class name or logical device number of the appropriate IODSO entry (or logical node name for the IODSX entry) that you use to specify the desired line. (Refer to Appendix A).

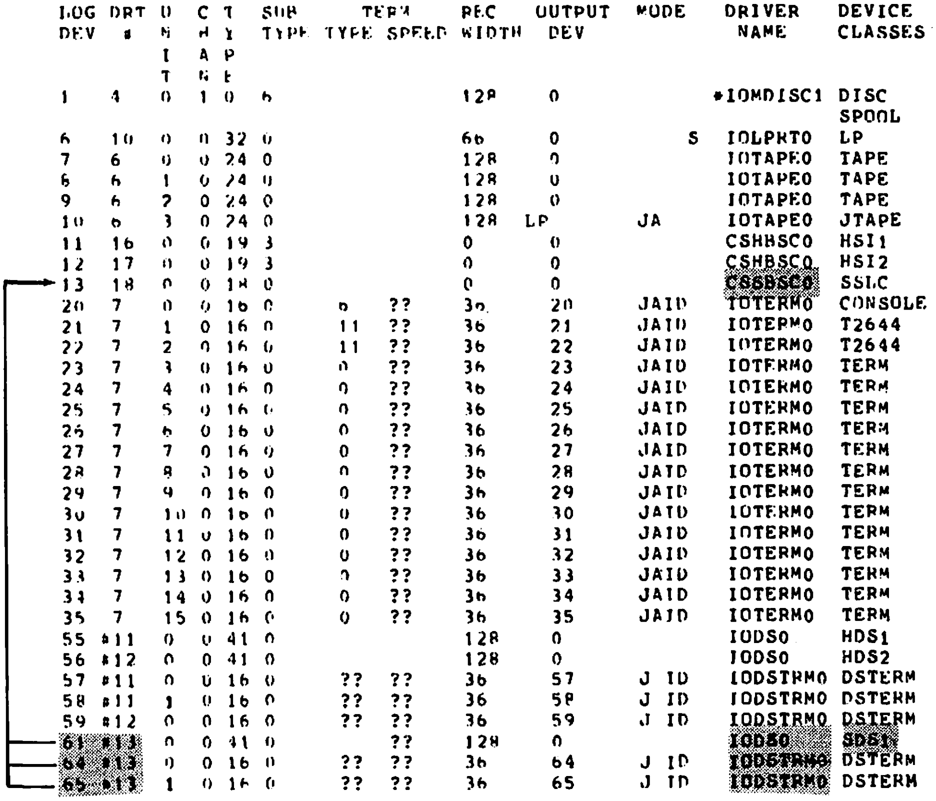


```
LOG DRT U  C T  SUB    TERM     REC   OUTPUT    MODE    DRIVER   DEVICE
DEV  #  N  H Y  TYPE TYPE SPEED WIDTH  DEV               NAME    CLASSES
 #      I  A P
        T  N E
1    4  0  0 0  6                128    0               IOMDISC1 SPOOL
                                                                 SYSDISK
2    5  0  0 0  3                128    0               IOMDISCO DISC
5   13  0  0 8  0                40   LP        JA   S  IOCDRDO  CARD
6   14  0  0 32 2                66     0            S  IOLPRTO  LP
7    6  0  0 24 0                128  LP                IOTAPFO  TAPE
8    6  1  0 24 0                128  LP                IOTAPEO  TAPE
9    6  2  0 24 0                128  LP                IOTAPEO  TAPE
10   6  3  0 24 0                128  LP        JA   S  IOTAPEO  BATAPE
11  20  0  0 34 0                128    0               IOPTPNO  PTPUNCH
12  16  0  0 19 3                0      0               CSHBSCO  HSI1
13  16  0  0 19 3                0      0               CSHBSCO  HSI2
14  16  0  0 19 3                0      0               CSHBSCO  HSI3
15  16  0  0 19 3                0      0               CSHBSCO  HSI4
20   7  0  0 16 0     10   ??    40      20     JAID    IOTERMO  CONSOLE
21   7  1  0 16 0     11   ??    40      21     JAID    IOTERMO  TERM
22   7  2  0 16 0     11   ??    40      22     JAID    IOTERMO  TERM
23   7  3  0 16 0     11   ??    40      23     JAID    IOTERMO  TERM
24   7  4  0 16 0     11   ??    40      24     JAID    IOTERMO  TERM
25   7  5  0 16 0     11   ??    40      25     JAID    IOTERMO  TERM
50 #12  0  0 41 0                128    0               IODSO    HDS1
51 #13  0  0 41 0                128    0               IODSO    HDS2
52 #14  0  0 41 0                128    0               IODSO    HDS3
53 #15  0  0 41 0                128    0               IODSO    HDS4
60 [illegible] 0  0 16 0     ??   ??    40      60     J ID    IODSTRMO DSTERM
61 [illegible] 0  0 16 0     ??   ??    40      61     J ID    IODSTRMO DSTERM
62 [illegible] 0  0 16 0     ??   ??    40      62     J ID    IODSTRMO DSTERM
63 [illegible] 0  0 16 0     ??   ??    40      63     J ID    IODSTRMO DSTERM
64 [illegible] 0  0 16 0     ??   ??    40      64     J ID    IODSTRMO DSTERM
65 [illegible] 0  0 16 0     ??   ??    40      65     J ID    IODSTRMO DSTERM


LDN PM PRT LCL TC  RCV   LCL   CON   MODE  TRANSMIT  TM BUFFER D DRIVER
           MOD     TMOUT TMOUT TMOUT         SPEED       SIZE  C OPTIONS
12  8   1   1  1   20    60    900         C 250000   1  1024  N  0
13  4   1   1  1   20    60    900         C 250000   1  1024  N  0
14  2   1   1  1   20    60    900         C 250000   1  1024  N  0
15  1   1   1  1   20    60    900         C 250000   1  1024  N  0
```

Figure 2-3. Sample I/O Device Table (Hardwired Line)

In figure 2-4, the shaded items in the column labeled "DRIVER NAME" shows one SSLC (CSSBSCO) configured into the system as logical device 13. Notice that there is a DSN/DS communications driver (in this case, IODSO) also configured into the system. The IODSO (or IODSX) entry is related to the proper SSLC (or INP) entry by the number specified in the column labeled "DRT" (the # prefix indicates a back reference to a previously defined logical device number). Logical device 61 is paired with logical device 13. It is the device class name or logical device number of the appropriate IODSO entry (or logical node name for the IODSX entry) that you use to specify the desired line.



```
LOG  DRT U  C T  SUB      TERM      REC   OUTPUT  MODE   DRIVER    DEVICE
DEV   #  N  H Y  TYPE  TYPE SPEED  WIDTH   DEV            NAME     CLASSES
         I  A P
         T  N E
1    4   0  1 0  6                 128     0            *IOMDISC1  DISC
                                                                   SPOOL
6    10  0  0 32 0                 66      0          S  IOLPRTO   LP
7    6   0  0 24 0                 128     0             IOTAPEO   TAPE
8    6   1  0 24 0                 128     0             IOTAPEO   TAPE
9    6   2  0 24 0                 128     0             IOTAPEO   TAPE
10   6   3  0 24 0                 128  LP        JA     IOTAPEO   JTAPE
11   16  0  0 19 3                 0       0             CSHBSCO   HSI1
12   17  0  0 19 3                 0       0             CSHBSCO   HSI2
13   18  0  0 18 0                 0       0             CSSBSCO   SSLC
20   7   0  0 16 0     0    ??     36      20     JAID   IOTERMO   CONSOLE
21   7   1  0 16 0     11   ??     36      21     JAID   IOTERMO   T2644
22   7   2  0 16 0     11   ??     36      22     JAID   IOTERMO   T2644
23   7   3  0 16 0     0    ??     36      23     JAID   IOTERMO   TERM
24   7   4  0 16 0     0    ??     36      24     JAID   IOTERMO   TERM
25   7   5  0 16 0     0    ??     36      25     JAID   IOTERMO   TERM
26   7   6  0 16 0     0    ??     36      26     JAID   IOTERMO   TERM
27   7   7  0 16 0     0    ??     36      27     JAID   IOTERMO   TERM
28   7   8  0 16 0     0    ??     36      28     JAID   IOTERMO   TERM
29   7   9  0 16 0     0    ??     36      29     JAID   IOTERMO   TERM
30   7   10 0 16 0     0    ??     36      30     JAID   IOTERMO   TERM
31   7   11 0 16 0     0    ??     36      31     JAID   IOTERMO   TERM
32   7   12 0 16 0     0    ??     36      32     JAID   IOTERMO   TERM
33   7   13 0 16 0     0    ??     36      33     JAID   IOTERMO   TERM
34   7   14 0 16 0     0    ??     36      34     JAID   IOTERMO   TERM
35   7   15 0 16 0     0    ??     36      35     JAID   IOTERMO   TERM
55   #11 0  0 41 0                 128     0             IODSO     HDS1
56   #12 0  0 41 0                 128     0             IODSO     HDS2
57   #11 0  0 16 0     ??   ??     36      57     J ID   IODSTRMO  DSTERM
58   #11 1  0 16 0     ??   ??     36      58     J ID   IODSTRMO  DSTERM
59   #12 0  0 16 0     ??   ??     36      59     J ID   IODSTRMO  DSTERM
61   #13 0  0 41 0          ??     128     0             IODSO     SDS1
64   #13 0  0 16 0     ??   ??     36      64     J ID   IODSTRMO  DSTERM
65   #13 1  0 16 0     ??   ??     36      65     J ID   IODSTRMO  DSTERM
```

Figure 2-4. Sample I/O Device Table (Telephone Line)

If you have only one communications interface (INP, HSI, or SSLC) configured into your system, there is no question about which name or number to specify in a DSLINE command. If there is more than one communications interface, however, you must know (or ask someone who knows) which CSHBSC0, CSSBSC0, or IOINP0 pertains to the physical line you want to use.

One or more virtual terminal drivers (IODSTRM0 or IODSTRMX) should also be configured into the system. The IODSTRM0 or IODSTRMX entries allow users on another system to be logged on to this system and regulate the number of remote Session Main Processes (SMP) that can be assigned to a given line. Each IODSTRM0 or IODSTRMX entry is related to the proper communications interface entry by the number specified in the column labeled "DRT". Figure 2-3 (the hardwired example) shows logical devices 60 through 65 are paired with logical devices 12 through 15; figure 2-4 (the telephone line example) shows logical devices 64 and 65 are paired with logical device 13.

In figure 2-3, notice that the HSI board entries (logical devices 12 through 15) look the same except for the PORTMASK. The PORTMASK specifies which port on the board is to be used. There are also virtual terminals (logical devices 60 through 65) referencing back to logical device 12.

Since only one port on the HSI board can be opened at a time, only one block of virtual terminal entries is needed for that board. As each port is opened individually by specifying the corresponding DS entry in the :DSCONTROL command (see Appendix C), the system automatically reallocates the block of virtual terminal entries to the proper HSI board entry. This reallocation will not, however, show up in the I/O configuration table.

The DSLINE Command

The format of the DSLINE command, as used to open a line, is presented in figure 2-5. In addition to opening a hardwired line or a telephone line, this command can also be used for closing one or more communications lines (discussed later in this section).

```
:DSLINE    dsdevice    [;LINEBUF=buffer-size]
                       [;LOCID=local-id-sequence]
                       [;REMID=remote-id-sequence1[,remote-id-sequence2] . . .]
                       [;PHNUM=telephone-number]
                       [;EXCLUSIVE]
                       [;COMP]
                       [;NOCOMP]
                       [;QUIET]
```

Figure 2-5. Opening a Line with the DSLINE Command

The parameters that pertain to opening either a hardwired communications line or a telephone line are as follows:

dsdevice

This is the device class name or logical device number assigned to the DSN/DS communications driver (IODSO) during system configuration, or a logical node name. This parameter specifies what physical hardwired line or what communications interface (and modem) you wish to use.

(Required parameter.)

NOTE

DS/X.25 users should always use a node name rather than a line identifier.

buffer-size

A decimal integer specifying the size (in words) of the DSN/DS line buffer to be used in conjunction with the communications line. The integer must be within the range 304 < buffer-size < 1024 when used with the INP or within the range 304 < buffer-size < 4095 when used with the SSLC or HSI. The default value is the buffer size entered in response to the PREFERRED BUFFER SIZE prompt during system configuration. This parameter overrides the MPE configured value when specified by the first user to open the given line.

(Optional parameter.)

EXCLUSIVE

This parameter, if present, specifies that you want exclusive use of the communications line. If the requested line or specified communications interface is already open and you have specified the EXCLUSIVE option, DSN/DS will deny you access to the line (you cannot open it). (See "Line Opening Failures" later in this section.) Opening an EXCLUSIVE line requires the user to have CS and ND capability.

(Optional parameter.)

COMP

By using this parameter, you can override the current system default, which was set at configuration time (see Appendix A) or set by the system operator (see Appendix C), and activate data compression. In this way, the mode of operation is set for your subsequent DS activity. This parameter does not affect other users sharing the line.

(Optional parameter.)

NOCOMP

This parameter deactivates the data compression mode.

(Optional parameter.)

QUIET

When you issue the DSLINE command with this parameter added, the message identifying the DS line number is suppressed. The messages associated with the subsequent REMOTE HELLO and REMOTE BYE commands will also be suppressed.

(Optional parameter.)

The additional parameters (shown in figure 2-5) that pertain only to opening a telephone line are as follows:

telephone-number

A telephone number consisting of digits and dashes. The maximum length permitted (including both digits and dashes) is 20 characters. Provided that YES was entered in response to the DIAL FACILITY prompt during system configuration, this telephone number will be displayed at the operator's console of your HP 3000 and the operator will then establish the telephone connection by dialing that number at the modem. (When the autodial feature is present in your system, the number provided here is dialed automatically.) The default telephone number is the one entered in response to the PHONE NUMBER prompt during system configuration.

(Optional parameter.)

local-id-sequence

A string of ASCII characters contained within quotation marks or a string of octal numbers separated by commas and contained within parentheses. If you wish to use a quotation mark within an ASCII string, use two successive quotation marks. In the case of an octal sequence, each octal number represents one byte and must be within the range 0-377. The maximum number of ASCII characters or octal numbers allowed in the string is 16.

The supplied string of ASCII characters or octal numbers defines the ID sequence that will be sent from your HP 3000 to the remote HP 3000 when you attempt to establish the telephone connection. If the remote HP 3000 does not recognize the supplied ID sequence as a valid one, the telephone connection is terminated. The default value is the ASCII or octal string entered in response to the LOCAL ID SEQUENCE prompt during system configuration.

(Optional parameter.)

remote-id-sequence

Same format as local-id-sequence.

The supplied strings of ASCII characters or octal numbers define those remote HP 3000 ID sequences that will be considered valid when you attempt to establish the telephone connection. If the remote HP 3000 does not send a valid ID sequence, the telephone connection is terminated. The default set of remote ID sequences consists of the ASCII and octal strings entered in response to the REMOTE ID SEQUENCE prompt during system configuration.

(Optional parameter.)

NOTE

The logical node name (mentioned in the dsdevice parameter description) appears in the configuration file for a Public Data Network (PDN). (See Appendix H.) A Remote Node (RN) table relates the logical node name (specified in this command) to the logical device number of the appropriate IODSX driver (the X.25 driver), and to the PDN address of the destination node.

Dialing the Remote Computer

When you are opening a telephone line, you may supply a telephone number (as an optional parameter in the DSLINE command, see figure 2-5) to be dialed at the modem connected to the specified INP or SSLC. If you supply a telephone number, DSN/DS displays a message on the system console telling the operator to dial that number. The operator, after dialing the specified number, enters YES or NO through the system console =REPLY command to let DSN/DS know whether or not the telephone connection was successfully made. If the operator enters YES, DSN/DS proceeds with the exchanging of ID sequences. If the operator enters NO, your DSLINE request is denied (you cannot open the line). In either case, your terminal's keyboard is disabled until the console operator responds.

If you do not supply a telephone number, the sequence of events is as described in the above paragraph, except that DSN/DS uses (by default) the first telephone number in the PHONELIST established during system configuration.

If you do not supply a telephone number and no PHONELIST was established during system configuration, an I/O request message is displayed at the system console, but it does not include the number to be dialed. This method might be used when you will dial the remote HP 3000 yourself. Remember, however, that the console operator must still know whether you dialed successfully, since he must respond to the console message before you are granted access to the line. Because your terminal's keyboard is disabled until the console operator responds with YES or NO, it is recommended that you always supply a telephone number in the DSLINE command.

ID Sequences

Once a telephone connection to a remote HP 3000 exists, the two computers exchange ID sequences with one another. Within the context of DSN/DS, an ID sequence is a string of up to 16 ASCII characters or octal numbers that identifies a particular computer.

During system configuration, each HP 3000 can be assigned a local ID sequence and a list of remote ID sequences. The local ID sequence identifies the particular HP 3000 in which it is established; the remote ID sequences identify those remote computers with which a communications link can be established over the public telephone network.

In the DSLINE command, you can supply (as optional parameters) a local ID sequence and one or more remote ID sequences to be used instead of those established during system configuration. (Refer to figure 2-5.)

When a telephone connection is established between your HP 3000 and a remote HP 3000, the local ID sequence supplied in your DSLINE command is transmitted to the remote system. The remote system compares that ID against its list of remote ID sequences. If that ID sequence is found to be valid, the remote system transmits its local ID sequence over the telephone line to your HP 3000. The received ID sequence is then compared against the remote ID sequence(s) supplied in your DSLINE command. If that ID sequence is found to be valid, the telephone connection is considered successful and DSN/DS grants you access to the line. If the ID sequence received at either end of the line is not considered valid, your DSLINE request is denied (you cannot open the line).

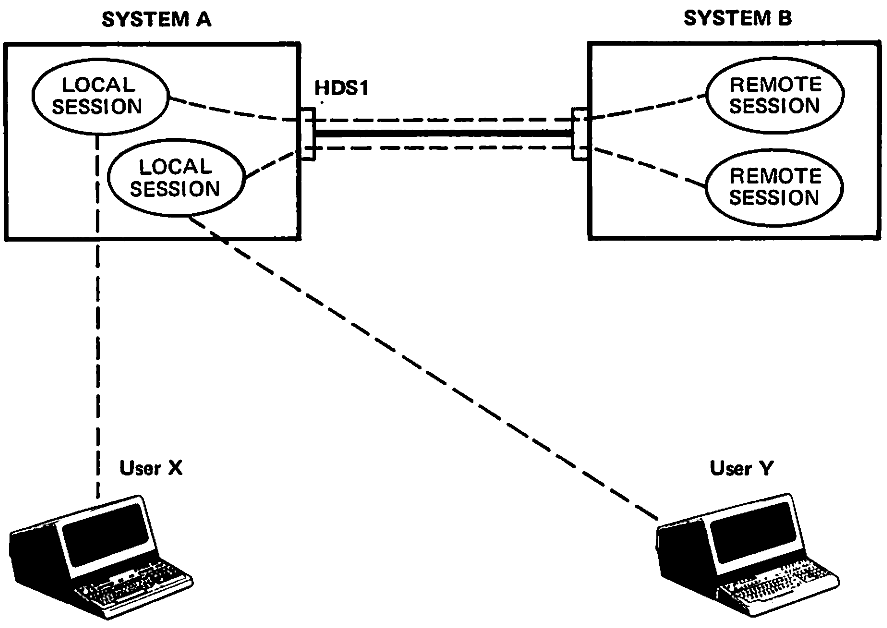
If you do not supply any ID sequences, DSN/DS uses those established during system configuration. If no ID sequences were established during system configuration and you do not supply any, no local ID sequence is transmitted from your HP 3000 to the remote system and any remote ID sequence received is considered valid.

Multiple Users

Within a DSN/DS environment, it is possible for several users at either end of the line to share access to the same physical communications line or for a single user at one end of the line to obtain exclusive access to the line.

As previously mentioned in the presentation of the DSLINE command, the EXCLUSIVE parameter can be used to obtain exclusive access to the specified physical communications line. If you specify this parameter (and if access to the line is granted), no other user in either computer will be permitted to open that line until you close it. If you ask for exclusive access to a particular line and that line is already in use, DSN/DS denies your request (you cannot open the line). (See "Line Opening Failures" later in this section.)

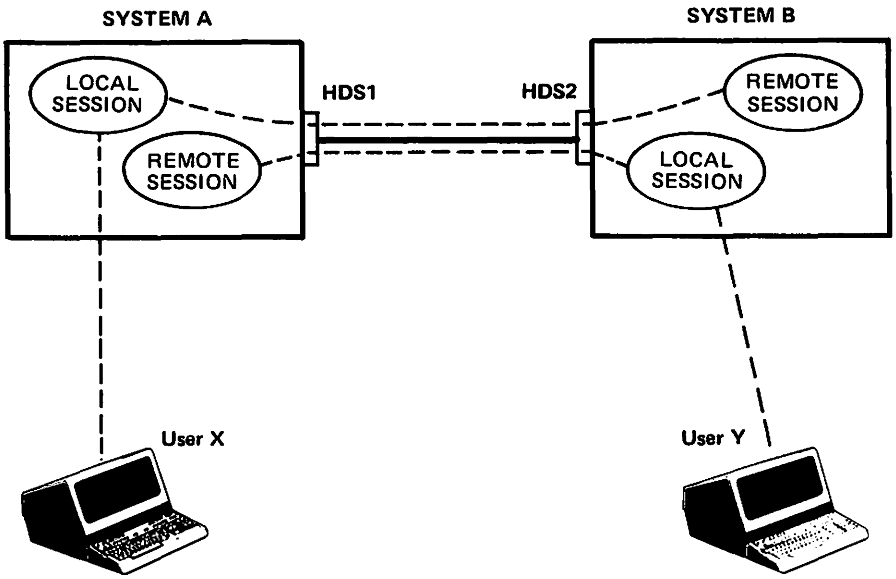
For hardwired lines and for dial-up lines, multiple users at either end of the line can specify the same physical line in DSLINE commands and obtain access to that line as long as none of them requests exclusive access. In such a case, the users' data is multiplexed, so that each user's access to the line appears to be completely independent of all others. The exception for a telephone line is that all users, other than the one who originally opened the line, specify (explicitly or by default) the currently active remote ID sequence. Figures 2-6 through 2-21 present annotated examples, illustrating successful and unsuccessful attempts by different users to obtain access to the same line.



```
:HELLO USER.X                     :HELLO USER.Y
:DSLINE HDS1                      :DSLINE HDS1
:REMOTE HELLO USER.X              :REMOTE HELLO USER.Y
```

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 DSN/DS because neither user asked for exclusive access to the line.

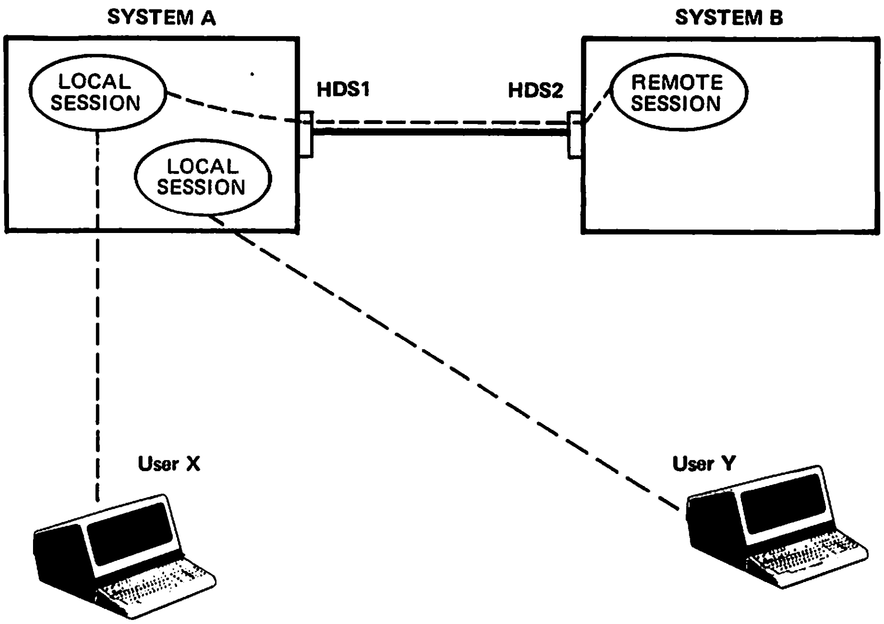
Figure 2-6. Multiple User Example 1



```
:HELLO USER.X                    :HELLO USER.Y
:DSLINE HDS1                     :DSLINE HDS2
:REMOTE HELLO USER.X             :REMOTE HELLO USER.Y
```

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 B, obtains access to the same communications line, and initiates a remote session in System A. The request by User Y for the particular communications line is granted by DSN/DS because neither user asked for exclusive access to the line.

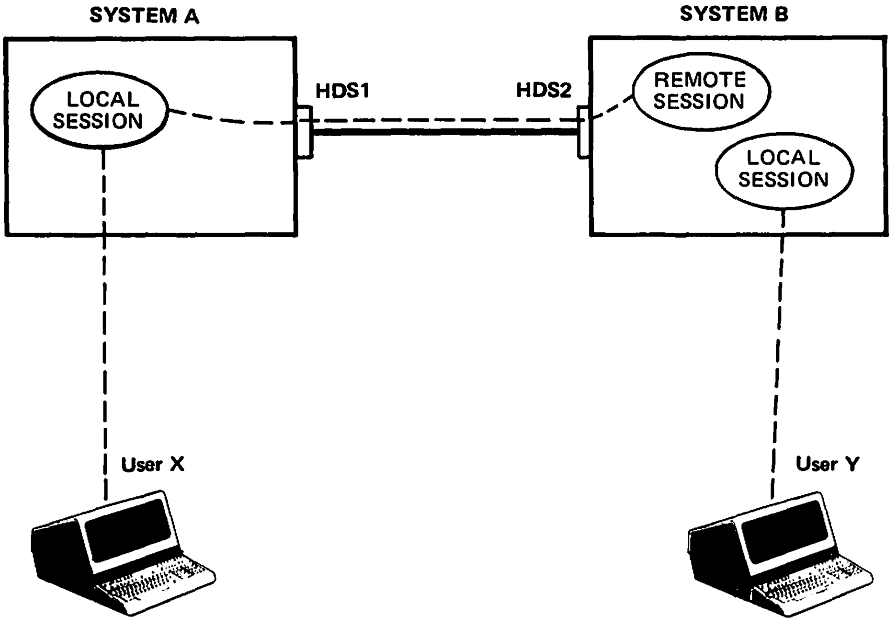
Figure 2-7. Multiple User Example 2



```
:HELLO USER.X                          :HELLO USER.Y
:DSLINE HDS1 ;EXCLUSIVE                :DSLINE HDS1
:REMOTE HELLO USER.X
```

In this example, User X initiates a local session in System A, obtains exclusive 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 and requests access to the same communications line. The request is denied by DSN/DS because User X already has exclusive access to the specified line.

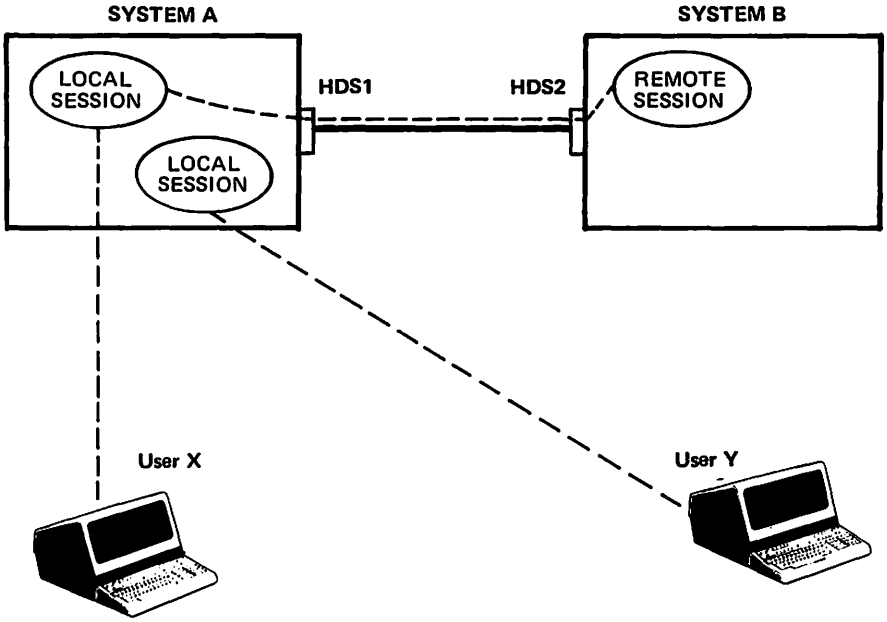
Figure 2-8. Exclusive Option Example 1



```
:HELLO USER.X                     :HELLO USER.Y
:DSLINE HDS1 ;EXCLUSIVE           :DSLINE HDS2
:REMOTE HELLO USER.X
```

In this example, User X initiates a local session in System A, obtains exclusive 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 B and requests access to the same communications line. The request is denied by DSN/DS because User X already has exclusive access to the specified line.

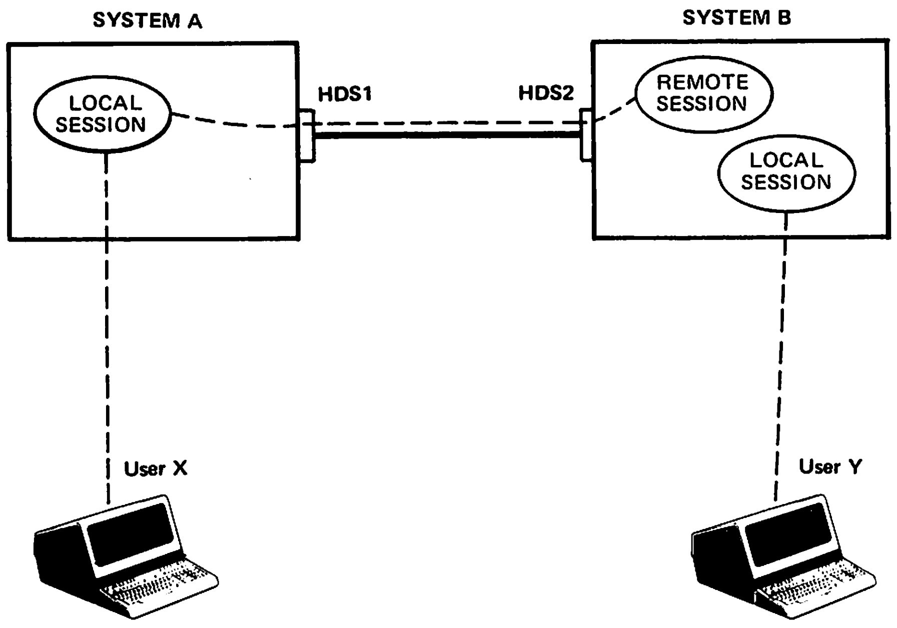
Figure 2-9. Exclusive Option Example 2



```
:HELLO USER.X                        :HELLO USER.Y
:DSLINE HDS1                         :DSLINE HDS1 ;EXCLUSIVE
:REMOTE HELLO USER.X
```

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 and requests exclusive access to the same communications line. The request is denied by DSN/DS because the specified line is already in use and therefore cannot be assigned exclusively to User Y.

Figure 2-10. Exclusive Option Example 3



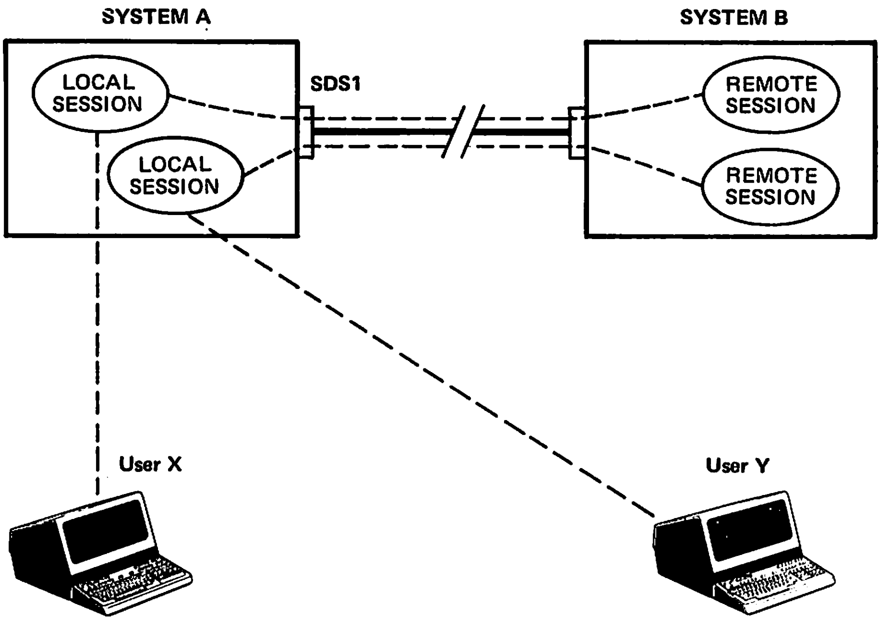
```
:HELLO USER.X                  :HELLO USER.Y
:DSLINE HDS1                   :DSLINE HDS2 ;EXCLUSIVE
:REMOTE HELLO USER.X
```

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 B and requests exclusive access to the same·communications line. The request is denied by DSN/DS because the specified line is already in use and therefore cannot be assigned exclusively to User Y.

Figure 2-11. Exclusive Option Example 4

Configured Local ID: A
Configured Remote IDs: B,C

Configured Local ID: B
Configured Remote IDs: A,C



```
:HELLO USER.X
:DSLINE SDS1 &
        ;PHNUM=555-1234 &
        ;REMID="B"
:REMOTE HELLO USER.X
```

```
:HELLO USER.Y
:DSLINE SDS1 &
        ;PHNUM=555-1234 &
        ;REMID="B"
:REMOTE HELLO USER.Y
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone

(continued)

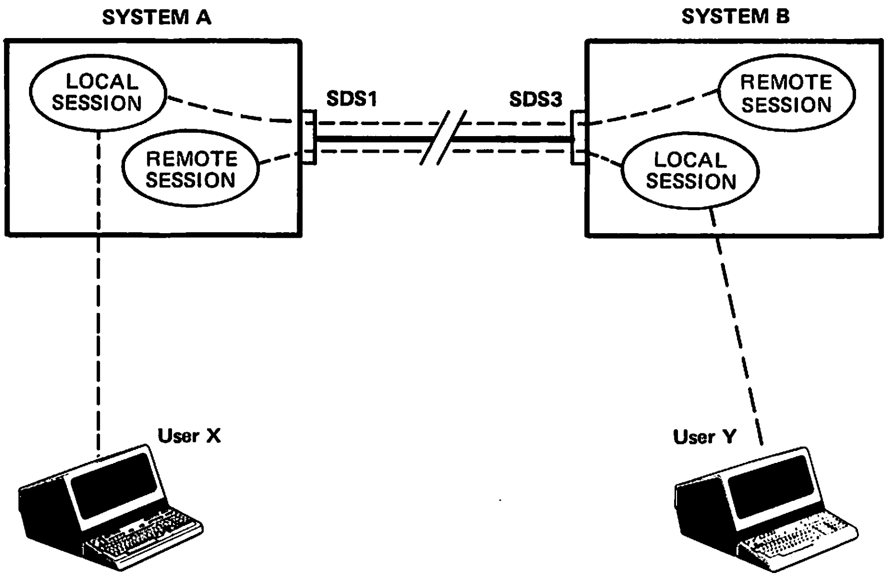
Figure 2-12. Dial-up Line Multiple User Example 1

connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System A and requests access to the same line (SDS1). Since that line is already open, DSN/DS ignores the supplied telephone number (no message is displayed at the system console). Access to the currently opened line is granted to User Y because neither user requested exclusive access and User Y specified the currently active remote ID sequence (REMID="B") in his DSLINE command.

Figure 2-12. Dial-up Line Multiple User Example 1 (Continued)

```
Configured Local ID:  A           Configured Local ID:  B
Configured Remote IDs:  B,C       Configured Remote IDs:  A,C
```



```
:HELLO USER.X                    :HELLO USER.Y
:DSLINE SDS1 &                   :DSLINE SDS3 &
        ;PHNUM=555-1234 &                ;PHNUM=777-4321 &
        ;REMID="B"                       ;REMID="A"
:REMOTE HELLO USER.X             :REMOTE HELLO USER.Y
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone

(continued)

Figure 2-13. Dial-up Line Multiple User Example 2

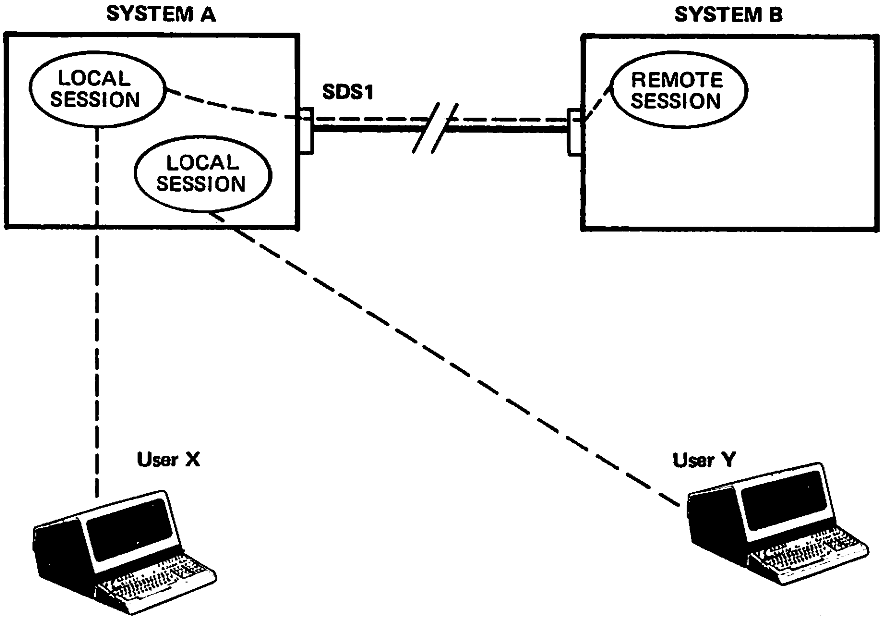
connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System B and requests access to the line identified by the device class name SDS3. Since that line is already open, DSN/DS ignores the supplied telephone number (no message is displayed at the system console). Access to the currently opened line is granted to User Y because neither user requested exclusive access and User Y specified the currently active remote ID sequence (REMID="A") in his DSLINE command.

Figure 2-13. Dial-up Line Multiple User Example 2 (Continued)

Configured Local ID: A
Configured Remote IDs: B,C

Configured Local ID: B
Configured Remote IDs: A,C



```
:HELLO USER.X
:DSLINE SDS1 &
        ;PHNUM=555-1234 &
        ;REMID="B" &
        ;EXCLUSIVE
:REMOTE HELLO USER.X
```

```
:HELLO USER.Y
:DSLINE SDS1 &
        ;PHNUM=555-1234 &
        ;REMID="B"
```

In this example User X initiates a local session in System A and obtains exclusive access to the line identified by the device class name SDS1 (he is granted access to it because at the time no one else was using that line). The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone

(continued)

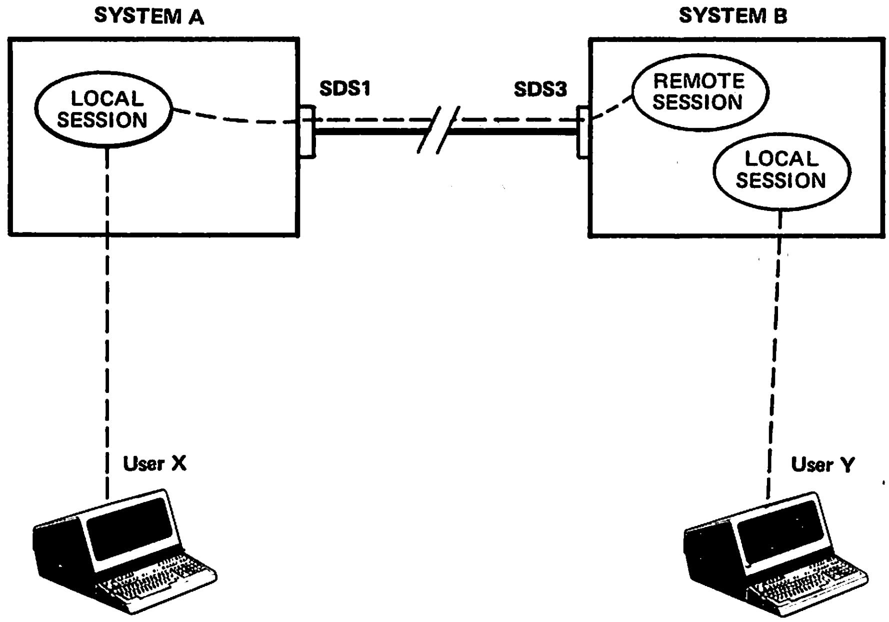
Figure 2-14. Dial-up Line Multiple User Example 3

connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System A and requests access to the same line (SDS1). The request is denied by DSN/DS because the specified line is already open and User X has exclusive access to it.

Figure 2-14. Dial-up Line Multiple User Example 3
(Continued)

```
Configured Local ID:  A          Configured Local ID:  B
Configured Remote IDs:  B,C      Configured Remote IDs:  A,C
```



```
:HELLO USER.X                    :HELLO USER.Y
:DSLINE SDS1 &                   :DSLINE SDS3 &
        ;PHNUM=555-1234 &                ;PHNUM=777-4321 &
        ;REMID="B" &                     ;REMID="A"
        ;EXCLUSIVE
:REMOTE HELLO USER.X
```

In this example User X initiates a local session in System A and obtains exclusive access to the line identified by the device class name SDS1 (he is granted access to it because at the time no one else was using that line). The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone

(continued)

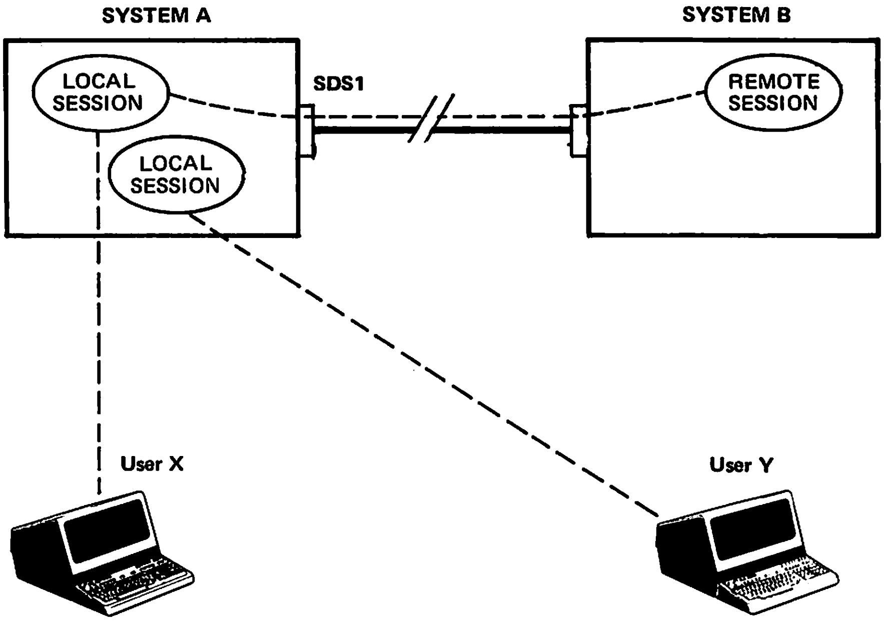
Figure 2-15. Dial-up Line Multiple User Example 4

connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System B and requests access to the line identified by the device class name SDS3. The request is denied by DSN/DS because the specified line is already open and User X has exclusive access to it.

Figure 2-15. Dial-up Line Multiple User Example 4 (Continued)

```
Configured Local ID:  A          Configured Local ID:  B
Configured Remote IDs:  B,C      Configured Remote IDs:  A,C
```



```
:HELLO USER.X                     :HELLO USER.Y
:DSLINE SDS1 &                    :DSLINE SDS1 &
        ;PHNUM=555-1234 &                 ;PHNUM=555-1234 &
        ;REMID="B"                        ;REMID="B" &
:REMOTE HELLO USER.X                      ;EXCLUSIVE
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone

(continued)

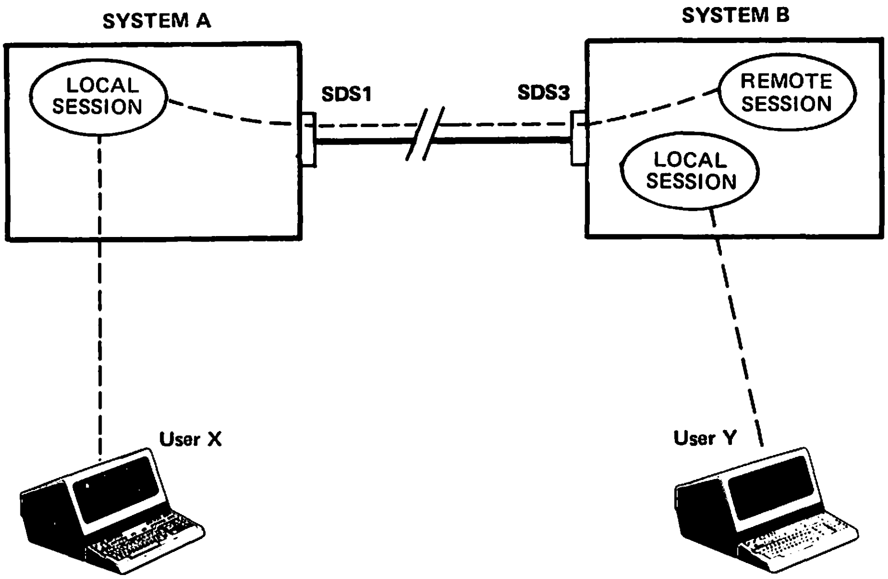
Figure 2-16. Dial-up Line Multiple User Example 5

connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System A and requests exclusive access to the same line (SDS1). The request is denied by DSN/DS because the specified line is already open and therefore cannot be assigned exclusively to User Y.

Figure 2-16. Dial-up Line Multiple User Example 5 (Continued)

```
Configured Local ID:  A           Configured Local ID:  B
Configured Remote IDs:  B,C       Configured Remote IDs:  A,C
```



```
:HELLO USER.X                   :HELLO USER.Y
:DSLINE SDS1 &                  :DSLINE SDS3 &
        ;PHNUM=555-1234 &               ;PHNUM=777-4321 &
        ;REMID="B"                      ;REMID="A" &
:REMOTE HELLO USER.X                    ;EXCLUSIVE
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone

(continued)

Figure 2-17. Dial-up Line Multiple User Example 6

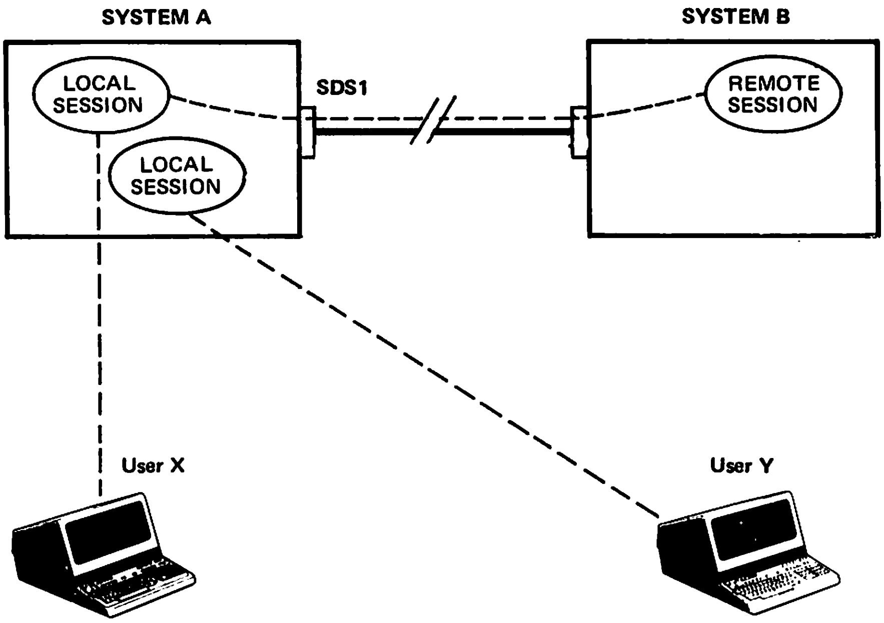
connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System B and requests exclusive access to the line identified by the device class name SDS3. The request is denied by DSN/DS because the specified line is already open and therefore cannot be assigned exclusively to User Y.

Figure 2-17. Dial-up Line Multiple User Example 6 (Continued)

Configured Local ID: A
Configured Remote IDs: B,C

Configured Local ID: B
Configured Remote IDs: A,C



```
:HELLO USER.X
:DSLINE SDS1 &
        ;PHNUM=555-1234 &
        ;REMID="B"
:REMOTE HELLO USER.X
```

```
:HELLO USER.Y
:DSLINE SDS1 &
        ;PHNUM=555-2001 &
        ;REMID="C"
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone

(continued)

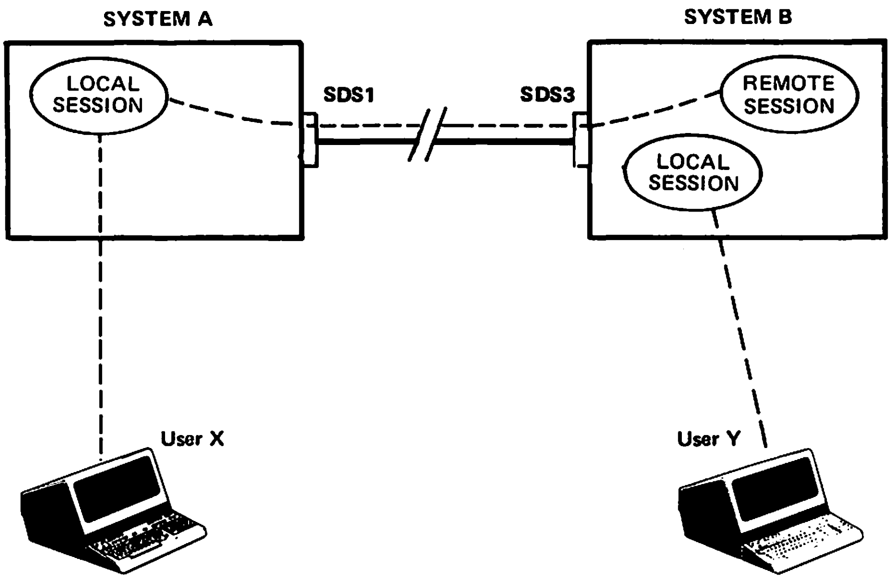
Figure 2-18. Dial-up Line Multiple User Example 7

connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System A and requests access to the same line (SDS1). The request is denied by DSN/DS because the specified line is already open and User Y did not specify the currently active remote ID sequence (B) in his DSLINE command.

Figure 2-18. Dial-up Line Multiple User Example 7 (Continued)

```
Configured Local ID:  A         Configured Local ID:  B
Configured Remote IDs:  B,C     Configured Remote IDs:  A,C
```



```
:HELLO USER.X                   :HELLO USER.Y
:DSLINE SDS1 &                  :DSLINE SDS3 &
        ;PHNUM=555-1234 &               ;PHNUM=555-2001 &
        ;REMID="B"                      ;REMID="C"
:REMOTE HELLO USER.X
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone

(continued)

Figure 2-19. Dial-up Line Multiple User Example 8

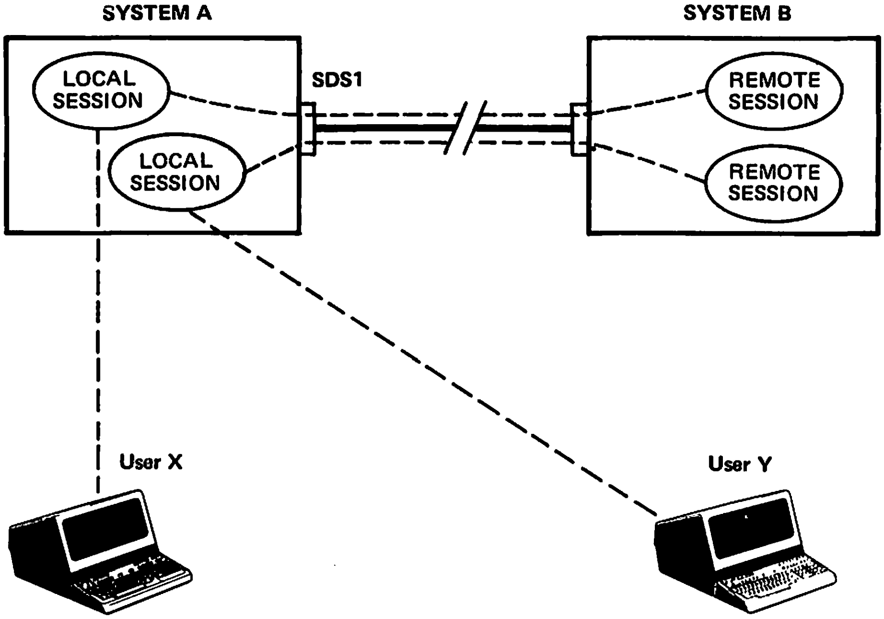
connection was successfully made. The two computers exchange their configured local ID sequences. System A compares the received ID sequence (B) against the remote ID sequence specified by User X (REMID="B") and System B compares the received ID sequence (A) against its list of configured remote ID sequences (A,C). Since the received ID sequences are found to be valid at both ends of the line, the telephone connection is allowed to remain in effect. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System B and requests access to the line identified by the device class name SDS3. The request is denied by DSN/DS because the specified line is already open and User Y did not specify the currently active remote ID sequence (A) in his DSLINE command.

Figure 2-19. Dial-up Line Multiple User Example 8 (Continued)

Configured Local ID: (none)
Configured Remote IDs: (none)

Configured Local ID: (none)
Configured Remote IDs: (none)



```
:HELLO USER.X
:DSLINE SDS1 ;PHNUM=555-1234
:REMOTE HELLO USER.X
```

```
:HELLO USER.Y
:DSLINE SDS1 ;PHNUM=555-1234
:REMOTE HELLO USER.Y
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone connection was successfully made. No ID sequences are exchanged because none were established (in either HP 3000)

(continued)

Figure 2-20. Dial-up Line Multiple User Example 9

during system configuration and User X didn't specify any in his DSLINE command. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

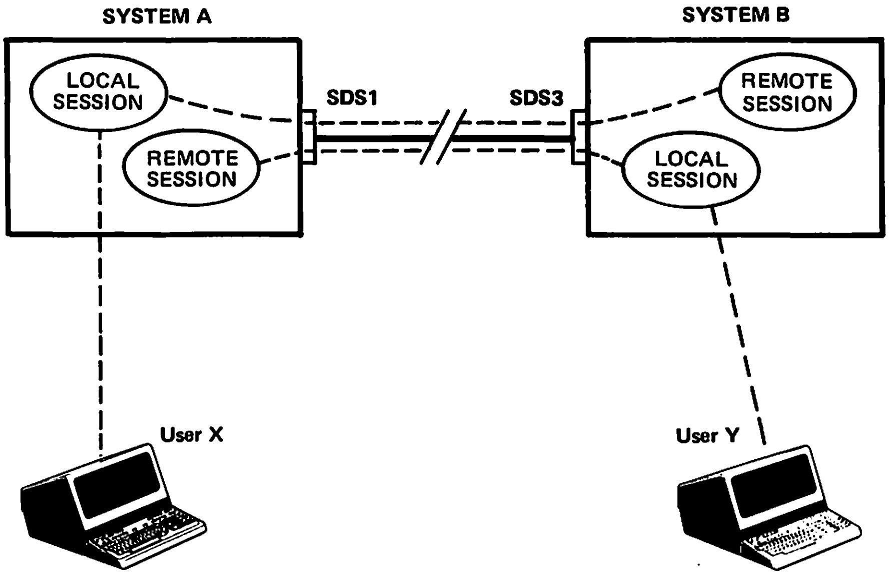
User Y subsequently initiates a local session in System A and requests access to the same line (SDS1). Since that line is already open, DSN/DS ignores the supplied telephone number (no message is displayed at the system console). Access to the currently opened line is granted to User Y because neither user requested exclusive access and User Y specified the currently active remote ID sequence (in this case none) in his DSLINE command.

Note that when no ID sequences are configured and the users don't supply any in their DSLINE commands, both are taking it on faith that they are connected to the proper remote computer. The total absence of configured or supplied ID sequences is safe only under very controlled circumstances. It is strongly recommended that all computers in a DSN/DS network that are capable of communicating over telephone lines have default local and remote ID sequences established during system configuration and that all line users specify the ID sequence of the desired remote computer (REMID=x) in their DSLINE commands.

Figure 2-20. Dial-up Line Multiple User Example 9 (Continued)

Configured Local ID: (none)
Configured Remote IDs: (none)

Configured Local ID: (none)
Configured Remote IDs: (none)



```
:HELLO USER.X
:DSLINE SDS1 ;PHNUM=555-1234
:REMOTE HELLO USER.X
```

```
:HELLO USER.Y
:DSLINE SDS3 ;PHNUM=777-4321
:REMOTE HELLO USER.Y
```

In this example User X initiates a local session in System A and obtains access to the line identified by the device class name SDS1. The supplied telephone number is displayed at the system console of System A. The console operator establishes the telephone connection by dialing the number at the modem connected to the particular line and then enters "YES" through the system console to let DSN/DS know that the telephone connection was successfully made. No ID sequences are exchanged because none were established (in either HP 3000)

(continued)

Figure 2-21. Dial-up Line Multiple User Example 10

during system configuration and User X didn't specify any in his DSLINE command. User X then initiates a remote session in System B over the telephone line from his local log-on terminal.

User Y subsequently initiates a local session in System B and requests access to the line identified by the device class name SDS3. Since that line is already open, DSN/DS ignores the supplied telephone number (no message is displayed at the system console). Access to the currently opened line is granted to User Y because neither user requested exclusive access and User Y specified the currently active remote ID sequence (in this case none) in his DSLINE command.

Note that when no ID sequences are configured and the users don't supply any in their DSLINE commands, both are taking it on faith that they are connected to the proper remote computer. The total absence of configured or supplied ID sequences is safe only under very controlled circumstances. It is strongly recommended that all computers in a DSN/DS network that are capable of communicating over telephone lines have default local and remote ID sequences established during system configuration and that all line users specify the ID sequence of the desired remote computer (REMID=x) in their DSLINE commands.

Figure 2-21. Dial-up Line Multiple User Example 10
(Continued)

:REMOTE HELLO

The REMOTE HELLO Command

Once you have obtained access to a physical communications line using the DSLINE command, you use the REMOTE HELLO command to actually establish the communications link. The REMOTE HELLO command initiates a remote session on your behalf in the HP 3000 connected to the other end of the communications line.

The format of the REMOTE HELLO command is presented in figure 2-22. Notice that, except for the three shaded items, it has exactly the same format as the standard MPE HELLO command.

Because the REMOTE HELLO command is initiating a session for you in a remote HP 3000, the parameters in that command specify information which pertains to the operating environment of the remote HP 3000 (not your local one). More specifically you must keep the following in mind:

- sessionname (if present) identifies the remote session and has no relationship to your local session.

- username, accountname, groupname, and their passwords (if any) must all be valid as defined by the accounting structure of the remote HP 3000.

- cpusecs (if present) refers to central-processor time in the remote system.

- BS, CS, DS, ES, inputpriority, and HIPRI (if present) all specify priorities for the remote session within the remote system.

- termtype (if present) has no meaning and is ignored because output from the remote session is directed to the communications line instead of to a terminal. The termtype parameter for your local session implicitly defines your log-on terminal type for any remote sessions that you initiate.

:REMOTE HELLO

```
:REMOTE HELLO   [sessionname,] username [/userpass] .acctname [/acctpass]
                [,groupname [/grouppass] ]

                [;TERM=termtype]

                [;TIME = cpusecs]

                         ( BS )
                         ( CS )
                [;PRI =  (    ) ]
                         ( DS )
                         ( ES )

                [;INPRI = inputpriority ]
                [;HIPRI                 ]

                [;DSLINE=dsdevice]
```

Figure 2-22. The REMOTE HELLO Command Syntax

The parameters for the REMOTE HELLO command are as follows:

sessionname — Arbitrary name used in conjunction with username and acctname parameters to form a session identity. Contains from 1 to 8 alphanumeric characters, beginning with a letter. Default is that no session name is assigned.

username — A user name, established by the account manager, that allows you to log on under this account. Contains from 1 to 8 alphanumeric characters, beginning with a letter.

(Required parameter.)

userpass

User password, optionally assigned by the account manager. Contains from 1 to 8 alphanumeric characters, beginning with a letter.

acctname

Name of account, as established by the account manager. Contains from 1 to 8 alphanumeric characters, beginning with a letter. The acctname parameter must be preceded by a period.

(Required parameter.)

acctpass

Account password, optionally assigned by the system manager. Contains from 1 to 8 alphanumeric characters, beginning with a letter.

groupname

Name of the group to be used for local file domain and CPU time charges. Established by the account manager. Contains from 1 to 8 alphanumeric characters, beginning with a letter. Default is your home group, if you are assigned one by the account manager.

(Optional if you have a home group; required if a home group has not been assigned.)

grouppass

Group password, optionally assigned by the account manager. Contains from 1 to 8 alphanumeric characters, beginning with a letter.

(Required if assigned and you are logging on under other than your home group; optional if you are logging on under your home group.)

termtype

Ignored. The TERM=termtype parameter of the HELLO command that initiated the local session also implicitly defines the log-on terminal type for any remote sessions initiated from the local session.

cpusecs

Maximum CPU time that the session can use, entered in seconds. When the limit is reached, the session is aborted. Must be a value from 1 to 32767. To specify no limit, enter a question mark (?) or UNLIM, or omit the parameter. Default is no limit.

BS, CS, DS, ES

Execution priority class. BS is highest priority; ES is lowest. If you specify a priority that exceeds the highest permitted for your account or user name by the system, MPE assigns the highest priority possible below BS. Default is CS.

NOTE

DS and ES are used primarily for batch jobs. Their use for sessions is discouraged.

inputpriorty

Relative input priority used in checking against access restrictions imposed by the jobfence, if one exists. Takes effect at log-on time. Must be a value from 1 (lowest priority) to 13 (highest priority). If a value is specified that is less than or equal to current jobfence set by the console operator, the session is denied access. Default is 8 or 13, depending upon the System Logging options in effect.

HIPRI

Request for maximum session-selection input priority, causing the session to be scheduled regardless of current jobfence or execution limit for sessions. This parameter can be specified only by users with System Manager or System Supervisor capability. (If not, the system tries to log you on with INPRI= 13.) Default is the current jobfence and execution limit.

dsdevice

The device class name or logical device number assigned to the DSN/DS communications driver (IODS0 or IODSX) during system configuration. This parameter, if present, specifies which line you wish to use.

(Optional parameter if a line is already open; otherwise it is required.)

Opening a Line

So far, we have been talking entirely about the DSLINE and REMOTE HELLO commands being used in conjunction with one another: the DSLINE command obtaining access to a physical line and the REMOTE HELLO command actually establishing the communications link by initiating a remote session over the acquired line. As you may have guessed from the above parameter definitions, the DSLINE parameter of the REMOTE HELLO command gives you a new, and simpler, way to obtain a line and establish a communications link. If you are satisfied to use the default DSN/DS line buffer size and you do not need exclusive use of the line, you can acquire a line and initiate a remote session over that line by using a single command: a REMOTE HELLO command with the DSLINE parameter. If you open a line in this way, however, it remains open only for the duration of the particular remote session (when the remote session is terminated the line is automatically closed). If, on the other hand, you use the DSLINE command to open a line, the line remains open for the duration of the local session (or until you explicitly close the line).

To illustrate this, look again at the example in Section I. In that example, the DSLINE command was used to obtain access to the hardwired line HDS2 and the REMOTE HELLO command was used to initiate a remote session over the line:

```
:DSLINE HDS2

REMOTE HELLO RUSER.RACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:08 AM

WELCOME TO SYSTEM B.
:
```

NOTE

In this case the acquired line remains open when the remote session is terminated.

By including the DSLINE parameter in the REMOTE HELLO command, essentially the same operations could be performed while using a single command, as follows:

```
:REMOTE HELLO RUSER.RACCOUNT;DSLINE=HDS2

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:08 AM

WELCOME TO SYSTEM B.
:
```

NOTE

In this case the acquired line is closed when the remote session is terminated.

Another example, this time using the DSLINE command to obtain access to a telephone line (by way of an SSLC whose associated communications driver's device class name is SDS1) and the REMOTE HELLO command to initiate a remote session over the line, is as follows:

```
:DSLINE SDS1

DS LINE NUMBER = #L3

:REMOTE HELLO RUSER.RACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:08 AM

WELCOME TO SYSTEM B.
:
```

NOTE

In this case the acquired line remains open when the remote session is terminated.

By including the DSLINE parameter in the REMOTE HELLO command, you can perform essentially the same operations using a single command, as follows:

```
:REMOTE HELLO RUSER.RACCOUT;DSLINE=SDS1

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:08 AM

WELCOME TO SYSTEM B.
:
```

NOTE

In this case the acquired line is closed when the remote session is terminated.

This telephone line example will work properly for you only under very limited circumstances:

- You must be satisfied to use the default DSN/DS line buffer size established during system configuration.
- The default ID sequences established in both computers during system configuration must properly identify both your local HP 3000 and the desired remote HP 3000 (or no ID sequences were established during system configuration in either computer).
- You must dial the remote computer yourself at the proper modem, or the line must be connected to (and configured for) autodialing. Note that if you cannot successfully make the telephone connection you cannot abort the REMOTE HELLO command; the command will be rejected by DSN/DS if no connection is established within 15 minutes.

The likelihood of all of the above conditions existing for a particular use of DSN/DS is rather slim. In most DSN/DS environments you will want to explicitly define the ID sequence of the desired remote computer to guarantee that the proper connection is established, and you will want to provide a telephone number so that you can let DSN/DS know immediately if a telephone connection cannot be made. (It is not acceptable to tie up a communications interface and your log-on terminal for 15 minutes waiting for an unsuccessful DSLINE or REMOTE HELLO request to be rejected.)

Opening Multiple Lines

Within your local session, you can open more than one physical communications line and you can have remote sessions active concurrently over all of the opened lines. However, when operating without DS/X.25 capability, you are limited to one remote session per physical line at any given time.

If access to the specified line is obtained, DSN/DS responds to each DSLINE command by displaying a DS line number at your log-on terminal. This line number is roughly analagous to the file number returned by the MPE FOPEN intrinsic, in that it is an arbitrary number that uniquely identifies (within your local session) your current access to a particular communications line. It has no relationship to the logical device number or any other configuration parameter associated with the 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 numbers to specify which line you wish to use for a given remote command (or sequence of remote commands) or to close a particular line without closing the others.

Figure 2-23 illustrates a situation where a user has established two hardwired communications links concurrently from within a single local session. Take a closer look at that situation and examine the sequence of commands that was used to create it.

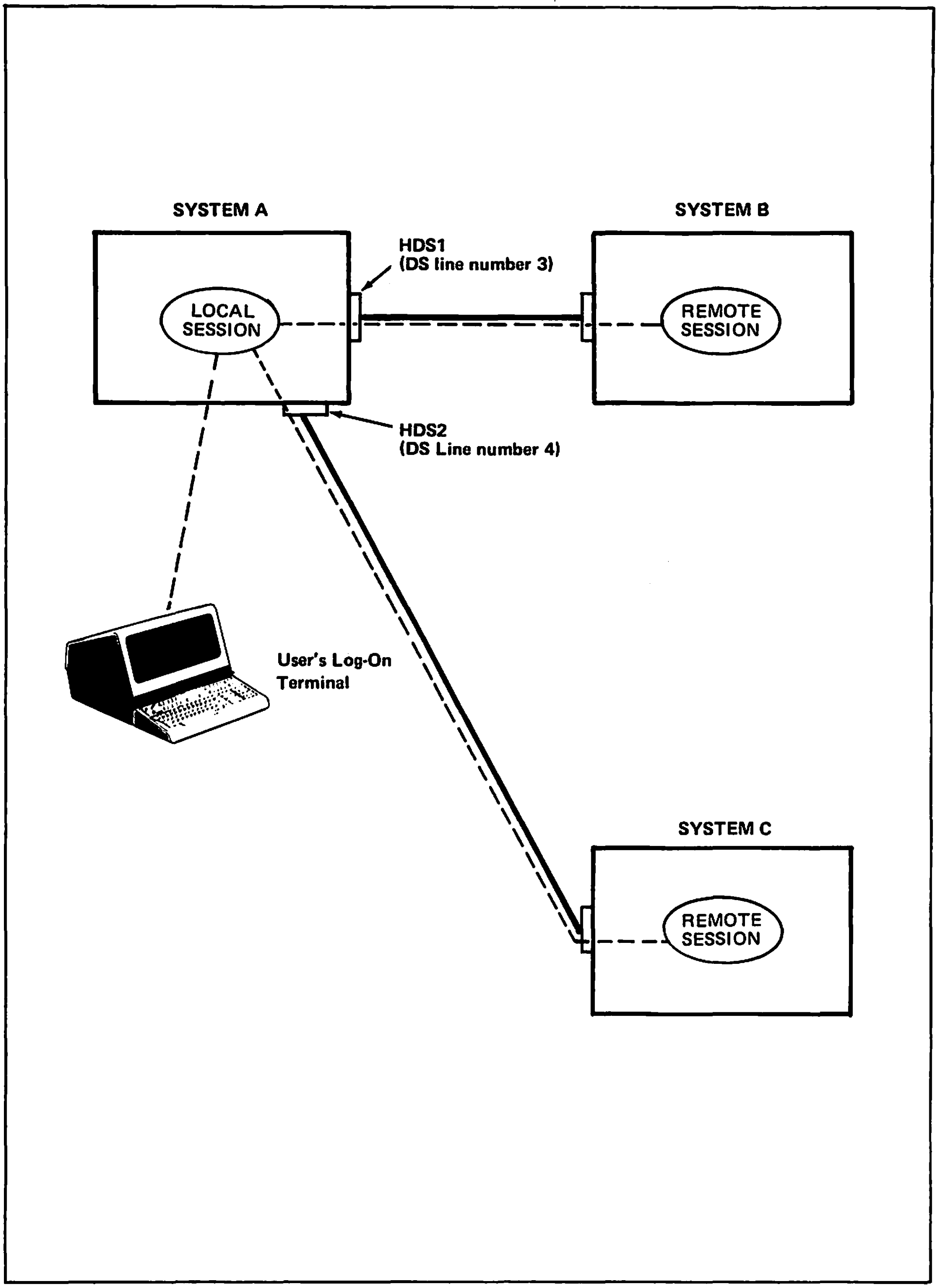


Figure 2-23. Multiple Line Example (Hardwired Lines)

First the user sat down at a terminal connected to System A and initiated a local session:

```
:HELLO USER.ACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  1:37 PM

WELCOME TO SYSTEM A.
:
```

USER and ACCOUNT are valid user and account names, respectively, as defined by the accounting structure of System A.

Now, we have the situation illustrated in figure 2-24. Notice that, at this point, no communications link exists between any of the three systems.

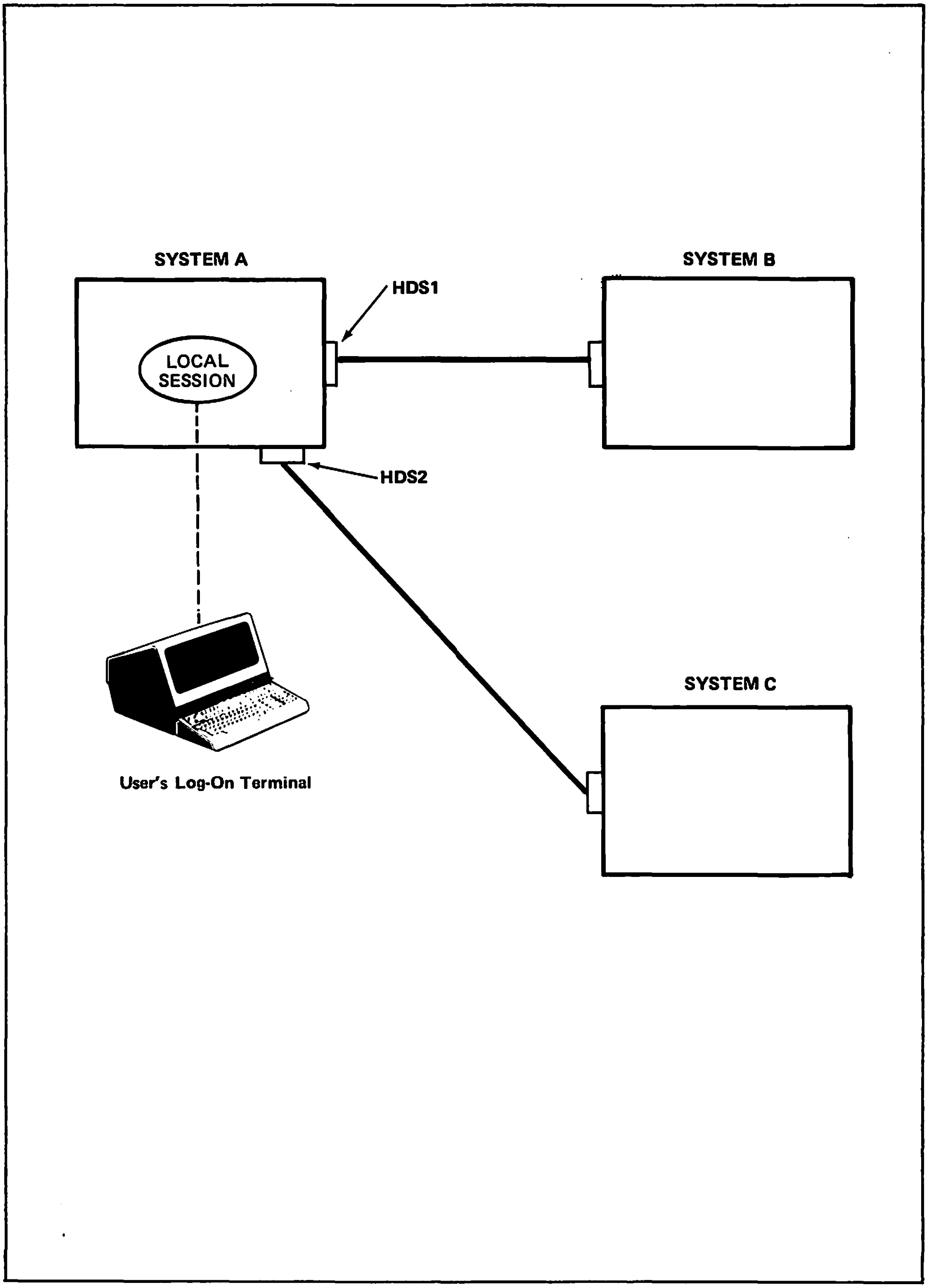


Figure 2-24. Initiating the Local Session (Hardwired Example)

Next, the user acquired access to a line between Systems A and B and initiated a remote session in System B:

```
:DSLINE HDS1

DSLINE NUMBER = #L3

:REMOTE HELLO RUSER.RACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  1:38 PM

WELCOME TO SYSTEM B.
:
```

HDS1 is the device class name (as defined within System A) associated with the particular line. RUSER and RACCOUNT are valid user and account names, respectively, as defined by the accounting structure of System B.

Now we have the situation illustrated in figure 2-25.

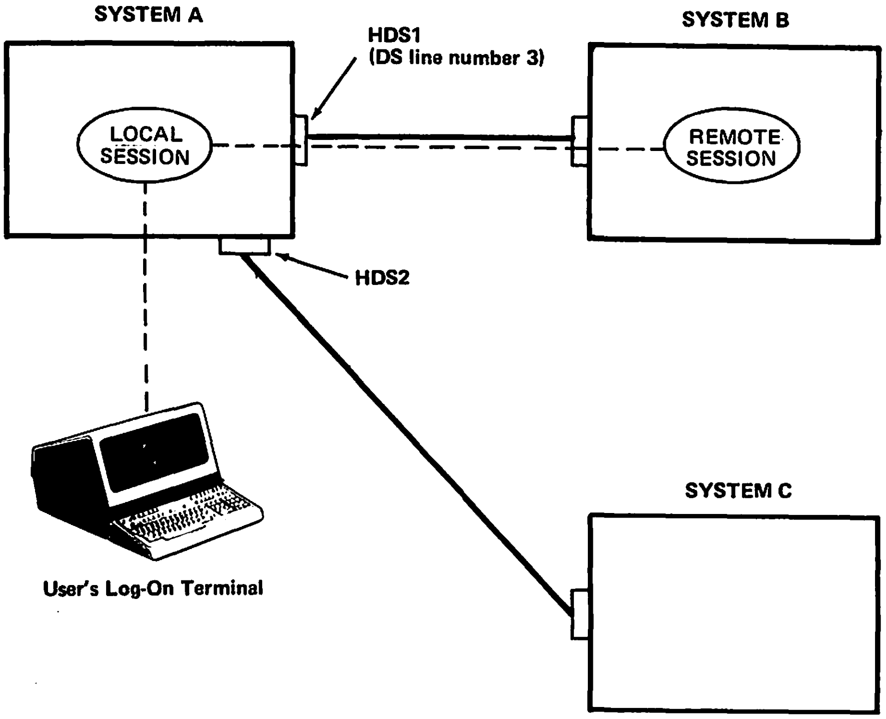


Figure 2-25. Establishing the Link With System B (Hardwired Example)

Finally, the user acquired access to a line between Systems A and C and initiated a remote session in System C:

```
:DSLINE HDS2

DS LINE NUMBER = #L4

:REMOTE HELLO RUSER.RACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  1:39 PM

WELCOME TO SYSTEM C
:
```

HDS2 is the device class name (as defined within System A) associated with the particular line. RUSER and RACCOUNT are valid user and account names, respectively, as defined by the accounting structure of System C.

We end up with the situation illustrated in figure 2-26, which is identical to figure 2-23 that started this example.

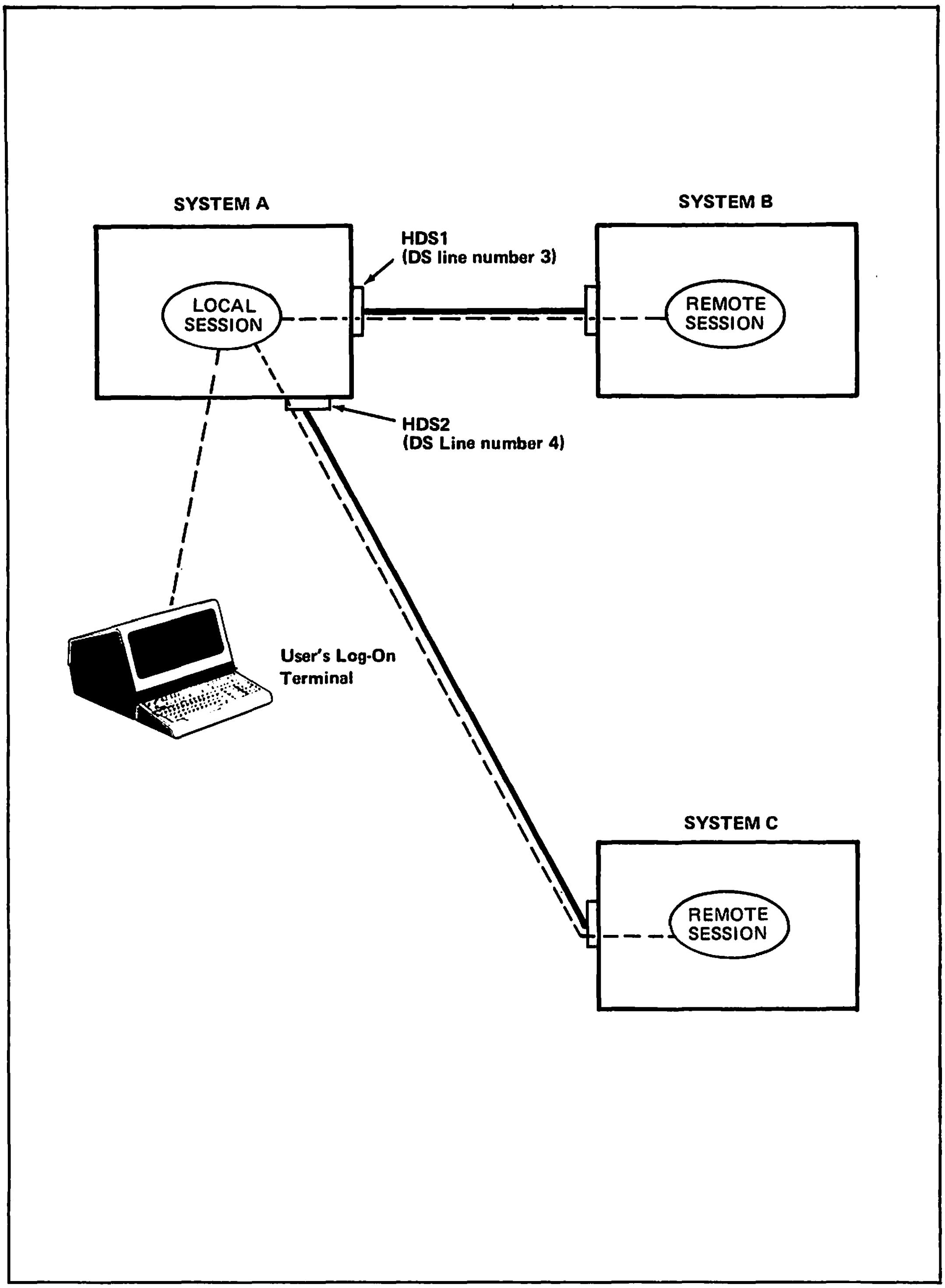


Figure 2-26. Establishing the Link With System C
(Hardwired Example)

Figure 2-27 illustrates a situation where a user has established two telephone communications links concurrently from within a single local session. Take a closer look at that situation and examine the sequence of commands that was used to create it.

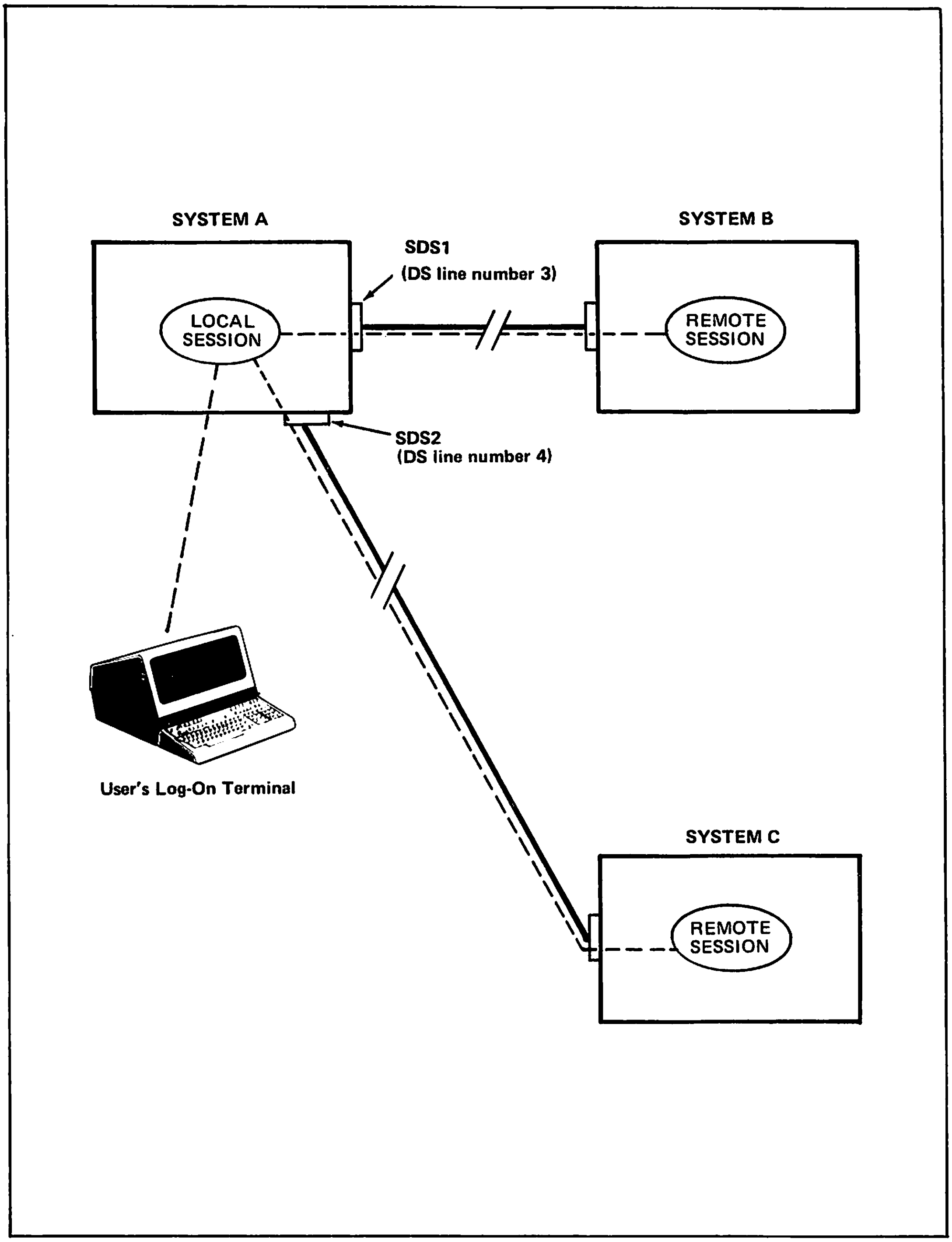


Figure 2-27. Multiple Line Example (Telephone Lines)

Opening a Line

First the user sat down at a terminal connected to System A and initiated a local session:

```
:HELLO USER.ACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  1:37 PM

WELCOME TO SYSTEM A.
:
```

USER and ACCOUNT are valid user and account names, respectively, as defined by the accounting structure of System A.

At this point, we have the situation illustrated in figure 2-28. Notice that, so far, no communications link exists between any of the three systems.

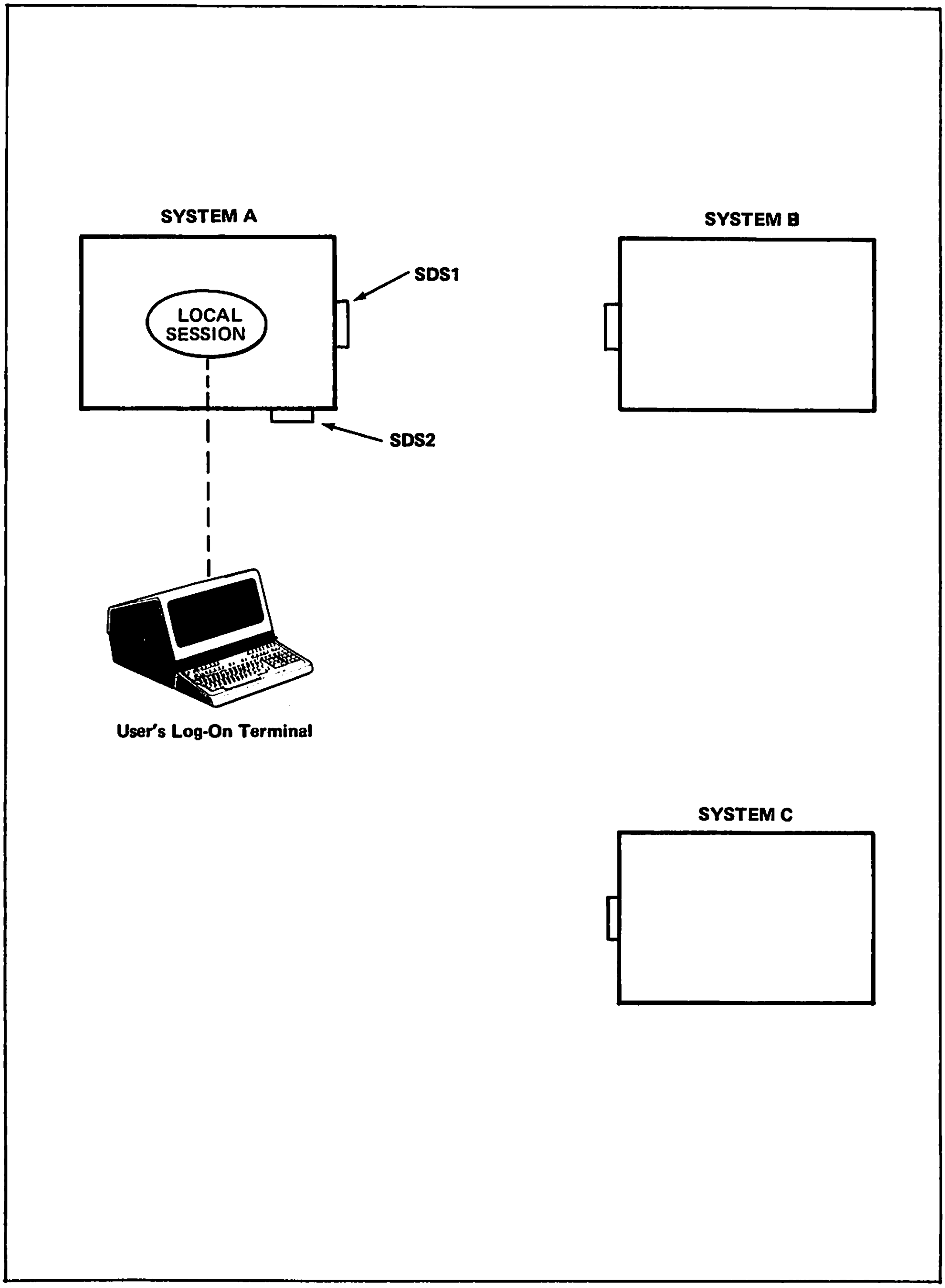


Figure 2-28. Initiating the Local Session (Dial-up Example)

Opening a Line

Next, the user acquired access to a telephone connection between Systems A and B and initiated a remote session in System B:

```
:DSLINE SDS1 ;LOCID="A" ;REMID="B" ;PHNUM=257-8001

DS LINE NUMBER = #L3

:REMOTE HELLO RUSER.RACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  1:38 PM

WELCOME TO SYSTEM B.
:
```

SDS1 is the device class name (as defined within System A) associated with the particular line, A and B are the ID sequences identifying Systems A and B, respectively, and 257-8001 is the telephone number of the modem connected to the communications interface at System B. RUSER and RACCOUNT are valid user and account names, respectively, as defined by the accounting structure of System B.

Now we have the situation illustrated in figure 2-29.

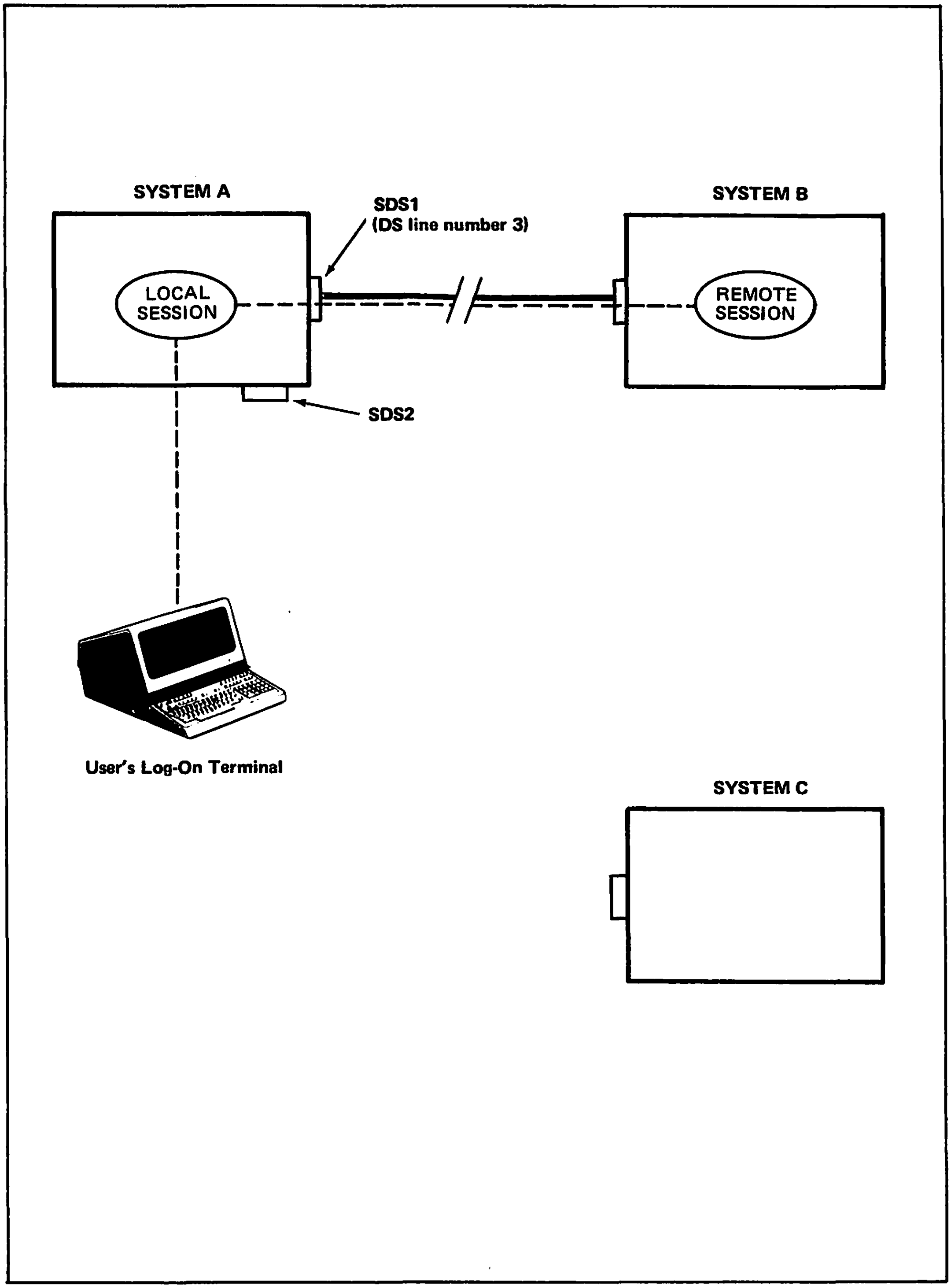


Figure 2-29. Establishing the Link With System B
(Dial-up Example)

Finally the user acquired access to a line between Systems A and C and initiated a remote session in System C:

```
:DSLINE SDS2 ;LOCID="A" ;REMID="C" ;PHNUM=377-2000

DS LINE NUMBER = #L4

:REMOTE HELLO RUSER.RACCOUNT

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  1:39 PM

WELCOME TO SYSTEM C.

:
```

SDS2 is the device class name (as defined within System A) associated with the particular line, A and C are the ID sequences identifying Systems A and C, respectively, and 377-2000 is the telephone number of the modem connected to the communications interface at System C. RUSER and RACCOUNT are valid user and account names, respectively, as defined by the accounting structure of System C.

We end up with the situation illustrated in figure 2-30, which is identical to figure 2-27 that started this example.

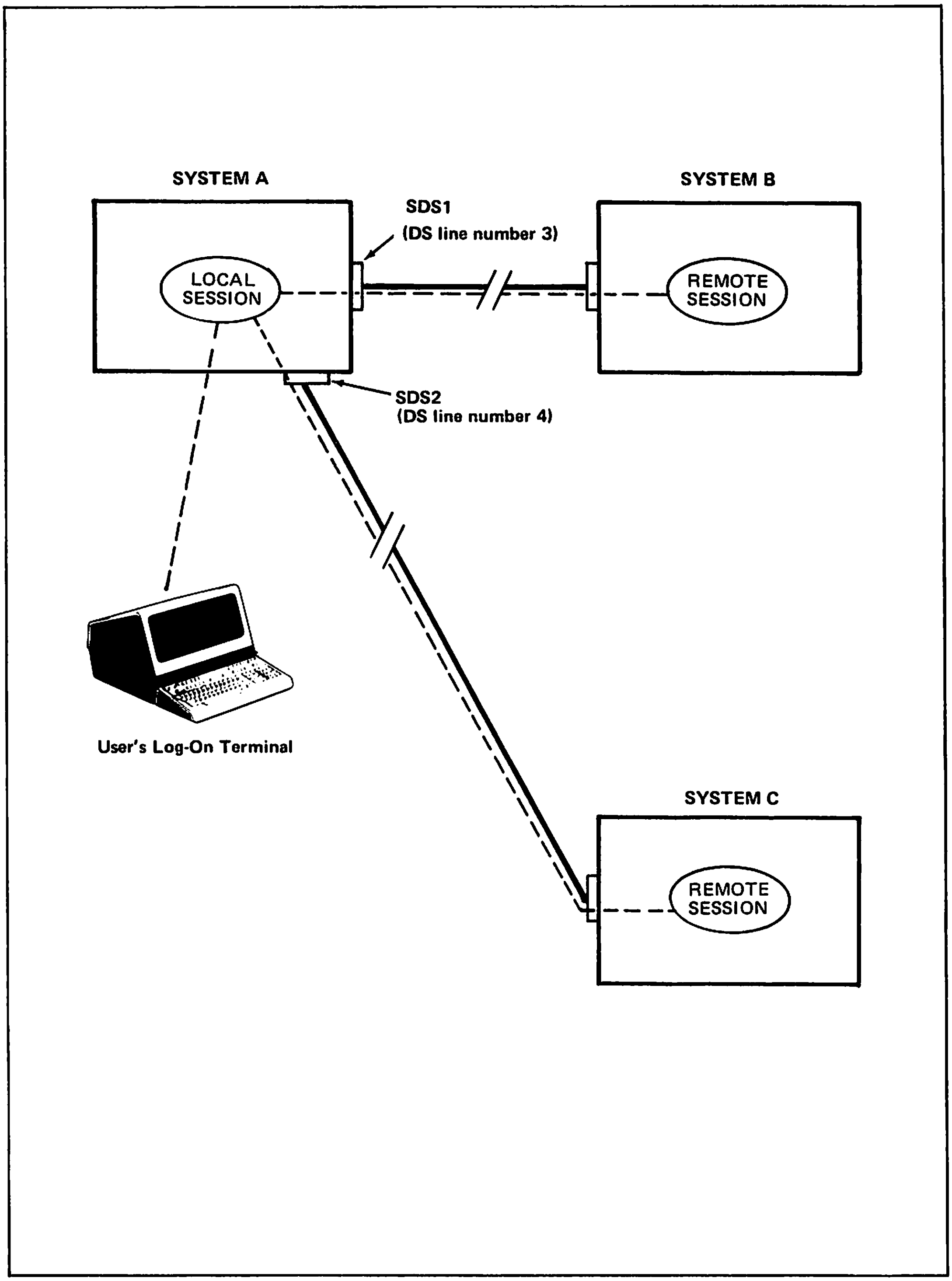


Figure 2-30. Establishing the Link With System C (Dial-up Example)

Line Opening Failures

There are several reasons why a DSLINE command for opening a communications line might be rejected by DSN/DS, some of which have already been illustrated earlier in this section.

The following list summarizes the likely causes of a line opening failure that are common to hardwired lines, leased lines, and dial-up telephone lines:

- You made a syntax error in the DSLINE command.
- You gave an erroneous line specification (dsdevice) in the DSLINE command. (There is no IODSO or IODSX entry in the system configuration with the specified device class name or logical device number.)
- The line was not opened by the local console operator.
- The line was not opened by the remote console operator.
- Someone already has exclusive access to the specified line.
- You asked for EXCLUSIVE access to a line which was already in use.
- DSN/DS detected a hardware problem (the communications interface board is not responding correctly).

The following list summarizes the additional causes of a line opening failure on a dial-up telephone line:

- The operator was not able to make the requested telephone connection and entered NO through the system console in response to the dial request message.
- The remote computer rejected your local ID sequence.
- The remote computer did not send a valid ID sequence (the received ID sequence did not match any of the remote ID sequences that you specified or, if you didn't specify any, did not match any of the configured remote ID sequences).
- The specified line is already in use and the remote ID sequence you supplied did not match the one used by the currently connected remote HP 3000.

The various error numbers and messages that might appear as a result of line opening failures are included in the summary of error codes and messages in Appendix B.

CLOSING A LINE

Once you have opened one or more communications lines, you can close any or all of them by using a variation of the DSLINE command. The line closing format of the DSLINE command is presented in figure 2-31.

```
            ⎧  dsdevice          ⎫
  :DSLINE   ⎨  ds-line-number    ⎬       ; CLOSE
            ⎩  @                 ⎭
```

Figure 2-31. Closing a Line with the DSLINE Command

The parameters that pertain to closing one or more communications lines are as follows:

dsdevice — The device class name, logical device number, or logical node name specified in the DSLINE command that opened a particular line.

(Optional parameter.)

ds-line-number — The DS line number assigned to you by DSN/DS when the particular line was opened. When this parameter is used, it must appear in the format #Ln , where n is the line number (see "Examples" on the following page).

(Optional parameter.)

@ — This parameter specifies that you wish to close all of the lines that you currently have open.

(Optional parameter.)

Closing a Line

;CLOSE — This parameter specifies that you wish to close the specified line(s).

(Required parameter.)

If no line identifier (dsdevice, ds-line-number, or @) is specified, DSN/DS closes the line that you most recently opened.

Examples

The following examples illustrate the variations of the DSLINE command that can be used for closing one or more communications lines.

```
   :DSLINE  HDS1 ;CLOSE
```

This form closes the line that is identified by the device class name HDS1.

```
   :DSLINE  55 ;CLOSE
```

This form closes the line that is identified by the logical device number 55.

```
   :DSLINE  @ ;CLOSE
```

This form closes all the lines that you currently have open.

```
:DSLINE #L3 ;CLOSE
```

This form closes the line that is identified by #L3.

```
:DSLINE  ;CLOSE
```

This form closes the line that you most recently opened.

If you are sharing one or more physical communications lines with other users, the above forms of the DSLINE command close the line(s) for your application only (the other user's applications are not affected).

REMOTE SESSIONS

SECTION III

A communications link exists after you have initiated a session in the remote HP 3000 under the username, accountname, and groupname specified in the REMOTE HELLO command. You now have two distinct sessions in existence simultaneously from the same log-on terminal: a local session (in the HP 3000 to which you first logged on) and a remote session (in the HP 3000 at the other end of the communications line). Now pause for a moment to see what this implies.

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, of course). 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 DSN/DS were not present. This local session is running under the username, accountname, and groupname specified in the HELLO command that you used to first log on. All user capabilities and file access available to you within the local session are determined by those log-on parameters.

Within the remote session, you have access to all I/O devices and disc files in the remote HP 3000 (again, subject to the usual MPE file security). With the few minor exceptions described in the following pages, this is also a normal MPE interactive session. All MPE commands and subsystems are, however, executed in the remote HP 3000. The output resulting from the executed commands and subsystems appears at your local log-on terminal. The remote session is running under the username, accountname, and groupname specified in the REMOTE HELLO command that you used in establishing the communications link. All user capabilities and file access available to you within the remote session are determined by those log-on parameters.

For the sake of clarity and as a learning aid, the remainder of this section will treat local and remote sessions as separate (and essentially unrelated) entities that use only those resources available in the particular HP 3000 in which they are running. Actually, it is possible to access the I/O devices and disc files of the remote HP 3000 computer from your local session, and it is also possible to access the I/O devices and disc files of the local HP 3000 from your remote session. This more advanced activity will be covered in Section IV, "Remote File Access".

ISSUING REMOTE COMMANDS

Remember that, in the previous sections, the following sequence of commands was used to establish the communications link:

```
   carriage return
   :HELLO USER.ACCOUNT

   HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:05 AM

   WELCOME TO SYSTEM A.
```

HELLO command and log-on display for local session.

```
   :DSLINE HDS2
   DS LINE NUMBER = #L3

   :REMOTE HELLO RUSER.RACCOUNT

   HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982,  9:06 AM

   WELCOME TO SYSTEM B.
   :
```

HELLO command and log-on display for remote session.

At this point, the remote session has been initiated, but you are currently in the local session (as signified by the colon prompt character). To execute a command in the remote session, use the following construct:

```
     :REMOTE  [xxx]  command
```

where xxx is the DS line number returned by DSN/DS when the communications line was opened, and command is the desired MPE command in its normal format. (The DS line number is necessary only if you have more than one communications line open simultaneously; if it is omitted, then the line which you most recently opened is referenced by default). In the example in Section I, this construct was used to execute a LISTF command, as follows:

```
     :REMOTE LISTF

     FILENAME

     DATA1    DATA5    DATA6    FILE3    SOURCE1
     :
```

Because the prefix REMOTE was included, the LISTF command is executed in the remote session (the implied account and group names are those established by the REMOTE HELLO command that initiated the remote session). Although the LISTF command is executed in the remote HP 3000, the output generated by the command is displayed at your local log-on terminal.

Notice, in the above example, that the DS line number associated with the particular communications line was not specified (3 in this example). This is because, if no line number is specified, DSN/DS uses (by default) the line most recently opened. Only one communications line is open from your local session; so DSN/DS uses that line by default. If you had opened a second line, you would need to tell DSN/DS in which remote computer the remote command is to be executed. To tell DSN/DS, include the appropriate DS line number in the remote command, as follows:

```
:REMOTE 3 LISTF

FILENAME

DATA1     DATA5     DATA6     FILE3     SOURCE1
:
```

The above construct only allows you to execute a single remote command. After the remote command has been executed, control returns to your local session (as signified by the colon prompt character).

But suppose that you want to execute a whole series of remote commands. It would obviously be a nuisance to have to prefix each command with the word REMOTE. DSN/DS provides a convenient solution to this situation. To execute a series of commands in the remote session, use the following construct:

```
:REMOTE  [xxx]
```

where xxx is again the DS line number of the desired communications line (specifying in which remote HP 3000 we want to execute commands). DSN/DS then prompts you for each command by displaying a # in column 1 of your terminal (in place of the standard MPE colon prompt). In the example in Section I, this construct was used for entering two remote MPE commands, EDITOR and BYE.

After reviewing the example in Section I, try another example that uses more than those two remote commands:

```
     :REMOTE
     #LISTF

     FILENAME

     DATA1    DATA5    DATA6    FILE3    SOURCE1
     #PURGE DATA5
     #PURGE DATA6
     #LISTF

     FILENAME

     DATA1    FILE3    SOURCE1
     #RUN FCOPY.PUB.SYS

     HP32212A.0.03 FILE COPIER

     >FROM=DATA1 ;TO=DATA2 ;NEW
     EOF FOUND IN FROMFILE AFTER RECORD 679

     680 RECORDS PROCESSED *** 0 ERRORS

     >EXIT

     END OF PROGRAM
     #LISTF

     FILENAME

     DATA1    DATA2    FILE3    SOURCE1
     #BYE


   CPU=4. CONNECT=7. WED, MAR  3, 1982,  9:13 AM
   #:
   :
```

Notice that except for the # prompt (in place of the standard colon prompt) this looks exactly like a normal MPE interactive session. All of the commands shown in the previous example are entered through the local log-on terminal, but the MPE and FCOPY commands are executed in the remote session within the remote HP 3000. After each remote MPE command was executed, however, control remained in the remote session (as signified by the # prompt character). When the remote session was terminated and the user typed a colon (:) in response to the # prompt following the log-off message, control was then returned to the local session (as signified by the colon prompt).

Using The Remote Subsystem From a Batch Job

While in a batch job, you can establish a remote session by using the DSLINE or REMOTE HELLO command.

The job to be streamed may be similar to the following:

```
:JOB USER.ACCOUNT
:DSLINE HDS2
:REMOTE HELLO RUSER.RACCOUNT
:REMOTE
#FILE OUT;DEV=LP
#BUILD WORK;DISC=50
#RUN USERPROG
#PURGE WORK
#:
:REMOTE BYE
:DSLINE;CLOSE
:EOJ
```

NOTE

The remote # prompt is optional.

An important point to remember is that, once established, the remote session is interacting with the job in the same way as it would interact with a terminal. If the remote session detects an error, the error is printed to $STDLIST and the next record in the job file is read (in the same manner as waiting for a character or carriage return on a terminal). The record is then lost to the job.

The BREAK Key

Within a remote session, you can use the BREAK key to temporarily interrupt remote processing. When doing so, either you may return control to the MPE Command Interpreter of your local HP 3000, or you may temporarily suspend the remote subsystem that you are executing without returning control to the local HP 3000. This is determined by how you execute commands in the remote session. There are two ways to execute commands in a remote session:

- By prefixing each command with the word REMOTE.
- By entering the word REMOTE, which prompts you for each command.

Prefixing Each Command With REMOTE When you are conducting a remote session by prefixing each command with the word REMOTE, pressing the BREAK key returns control to the local Command Interpreter and you receive the colon (:) prompt. To continue remote processing at the point where it was interrupted, you merely enter REMOTE RESUME in response to the local MPE colon prompt.

As an example, assume that you are in the midst of using the text editor in a remote session when you suddenly decide to start a job stream executing concurrently in your local HP 3000. The sequence of commands would be similar to the following:

```
     :REMOTE EDITOR

     HP32201A.7.05 EDIT/3000  FRI, MAY 9, 1980, 9:11 AM
     (C) HEWLETT-PACKARD CO. 1979
     /ADD
         1   DOE, JOHN                29  M  CHI
         2   BLACK, PATRICIA          23  F  SF
         3   SIMON, NEIL              43  M  NY
         4   MACK, SHIRLEY            38  F  DET
         5   <-┐
Local          └── BREAK key pressed here.
session
prompt.
   \
    \
     :STREAM  COBTEST1        }      Control is now
       #J19                   }      in the local
     :REMOTE RESUME           }      session.

     READ PENDING                          }
     MICHAELS, WILLIAM      32  M  CHI     }  Control is
         6   O'LEARY, TIMOTHY         49  M  DET }  now back in
         7   MARTIN, MARY             34  F  LA  }  the remote
         8   MURIN, JOICE             42  F  CHI }  session.

                 .
                 .
                 .
```

Notice that when the BREAK key was pressed, the text editor in the remote HP 3000 was waiting for you to enter the text for line 5. The BREAK key interrupted the remote session and passed control to the MPE Command Interpreter of the local HP 3000 (as signified by the colon prompt). The STREAM command was issued within the local session, which caused the file COBTEST1 to be executed in the local HP 3000. Then, when the RESUME command was issued, control was passed back to the remote session at the point where it was interrupted (that is, the text editor in the remote HP 3000 is now waiting for you to enter the text for line 5). When the text for line 5 is entered, the remote session proceeds as though nothing had happened.

Note that by the end of the example, the local job stream, the local session, and the remote session are all operational simultaneously.

Entering REMOTE When you are interacting with the remote Command Interpreter by having entered the word REMOTE and you are receiving the remote prompt (#), pressing the BREAK key will temporarily suspend the subsystem you are executing and will return control to the remote Command Interpreter. To continue remote processing at the point where it was interrupted, you merely enter RESUME in response to the remote prompt.

As an example, assume that you are in the midst of using the text editor in a remote session when you suddenly decide to start a job stream executing in your remote HP 3000. The sequence of commands would be similar to the example shown previously, but with a few minor differences, as follows:

```
:REMOTE
#EDITOR

HP 32201A.7.05 EDIT/3000  FRI, FEB 13, 1981, 9:20 AM
(C) HEWLETT-PACKARD CO. 1979
/ADD
    1   LEWIS, LEO              51  M  SV
    2   LAGERGREN, FRED         25  M  SJ
    3   DICKINSON, MARY         21  F  SC
    4   LAGREGREN, LINDA        24  F  SJ
    5  <-
        BREAK key pressed here.

#STREAM APLTEST1          }      Control is still
  #J20                    }      in the remote
#RESUME                   }      session.

READ PENDING
MELLO, HENRY          44  M  SJ
    6   SOARES, JOE             59  M  LA
    7   LAWRENCE, ALICE         44  F  SJ
    8   LEWIS, BOB              29  M  WASH
```

Notice that when the BREAK key was pressed, the text editor in the remote HP 3000 was waiting for you to enter the text for line 5. The BREAK key interrupted the remote session, but control remained in the remote HP 3000 (as signified by the remote # prompt). The STREAM command executed the file APLTEST1 within the remote HP 3000. Then, when the RESUME command was issued, control was passed back to the point where the text editor was interrupted (that is, the text editor is waiting for you to enter the text for line 5). When the text for line 5 is entered, the remote session proceeds as though nothing had happened.

The Control Keys

Within a remote session Control-H, Control-X, and Control-Y perform exactly the same functions as they do in a normal MPE interactive session.

For example, if you are using FCOPY or the text editor in a remote session, you can use Control-Y to prematurely terminate an FCOPY or text editor operation. When the operation terminates, control is still in the particular subsystem within the remote session.

Similarly, you can use Control-H to delete the last character entered or Control-X to delete the line of text currently being entered. In both of these cases, after the deletion occurs, control remains in the remote session.

ISSUING LOCAL COMMANDS

Whenever the standard MPE colon prompt is displayed at your terminal, you are in the local session. Within the local session, you enter MPE commands in their normal format in response to the colon prompt. If you are in the midst of a remote session (that is, you used the command :REMOTE, and DSN/DS is issuing the # prompt character), you can return control to your local session by entering a colon, as follows:

```
    #:
```

In response to the remote 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 the remote session; you can resume processing in the remote session by again using either of the constructs described under "Issuing Remote Commands".

TERMINATING A REMOTE SESSION

You can terminate a remote session either from within the local session or from within the remote session itself.

From The Local Session

Whenever the standard MPE colon prompt is displayed at your terminal, you are in the local session. To terminate a remote session from within your local session, use the following command:

```
     :REMOTE  [xxx]  BYE
```

where xxx is the DS line number associated with the communications line connecting the particular remote session to your local session. (The DS line number is necessary only if you have more than one communications line open simultaneously; if it is omitted then the line that you most recently opened is referenced by default.)

For instance, in the example in Section I, either of the following sequences could have been used to terminate the remote session:

```
   #:
   :REMOTE BYE

   CPU=4. CONNECT=7. WED, MAR  3, 1982,  9:13 AM
   :

                 *** OR ***

   #:
   :REMOTE 3 BYE

   CPU=4. CONNECT=7. WED, MAR  3, 1982,  9:13 AM
   :
```

In both cases, the remote colon was used to return control from the remote session to the local session. In either case, the remote session is terminated.

If the communications line was opened using the DSLINE= parameter of the REMOTE HELLO command, the line is automatically closed when the remote session terminates. To initiate another remote session over the same communications line, you must once again open the line (using either the DSLINE command or the DSLINE= parameter of the REMOTE HELLO command) and then issue another REMOTE HELLO command.

If the communications line was opened using the DSLINE command, it is still open. To initiate another remote session over the same communications line, merely issue another REMOTE HELLO command (you do not need to issue another DSLINE command because the communications line is still open). To close the communications line, use the constructs presented in Section II.

From The Remote Session

Whenever the # prompt is displayed at your terminal, you are in the remote session. To terminate a remote session from within the remote session itself, use the following command:

```
#BYE
```

Note that you do not need to supply a DS line number in this case, because DSN/DS knows implicitly which remote session you wish to terminate (that is, the one in which the #BYE command is executed).

Remember that this command was used to terminate the remote session in the example at the end of Section I, as follows:

```
#BYE

CPU=4. CONNECT=7. WED, MAR  3, 1982,  9:15 AM
#
```

Notice that although the remote session is terminated, DSN/DS is still issuing the # prompt character. To return control to the local session, issue a colon (described earlier under "Issuing Local Commands").

If the communications line was opened using the DSLINE command, it is still open. To initiate another remote session over the same communications line, merely issue an appropriate remote MPE HELLO command. (You do not need to use the prefix REMOTE because

DSN/DS is still waiting for you to enter a remote command; nor do you need to issue another DSLINE command because the communications line is still open.) To close the communications line, use the constructs presented in Section II.

If the communications line was opened using the DSLINE= parameter of the REMOTE HELLO command, the line is automatically closed when the remote session terminates. To initiate another remote session over the same line, you must once again open the line (using the DSLINE command or the DSLINE= parameter of a REMOTE HELLO command) and then issue another REMOTE HELLO command.

REMOTE FILE ACCESS

SECTION

IV

In the preceding sections, you have seen how you can establish a communications link between two HP 3000s and thereby use the computing power of the remote HP 3000. But that is only part of the story! Through the use of the DSN/DS Remote File Access (RFA) capability, programs running in your local session can:

- Use any of the devices connected to the remote HP 3000 as though they were connected directly to your local HP 3000

- Access any of the disc files of the remote HP 3000 (subject to the normal MPE file security, of course) as though they resided at your local HP 3000 site.

The RFA capability, in conjunction with the remote session capability, suddenly puts all of the computing power and all of the hardware and software resources of a remote HP 3000 at your fingertips.

Section IV is divided into two main parts. The first part, "Command Access", describes how you can issue local MPE FILE commands that define devices and/or files residing at the remote HP 3000 site. The second part, "Programmatic Access", describes how you can use the standard set of MPE File System intrinsics within your local programs to access devices and/or files residing at the remote HP 3000 site.

COMMAND ACCESS

After a DSN/DS communications link has been established, you can issue local MPE FILE commands that define devices and/or files residing at the remote HP 3000 site. To make this possible, the DEV= parameter of the MPE FILE command has been expanded to include a DS line specification in addition to the usual device specification. To specify a file that resides across a DS line, the format of the DEV= parameter is as follows:

```
;DEV=[dsdevice]#[device]
```

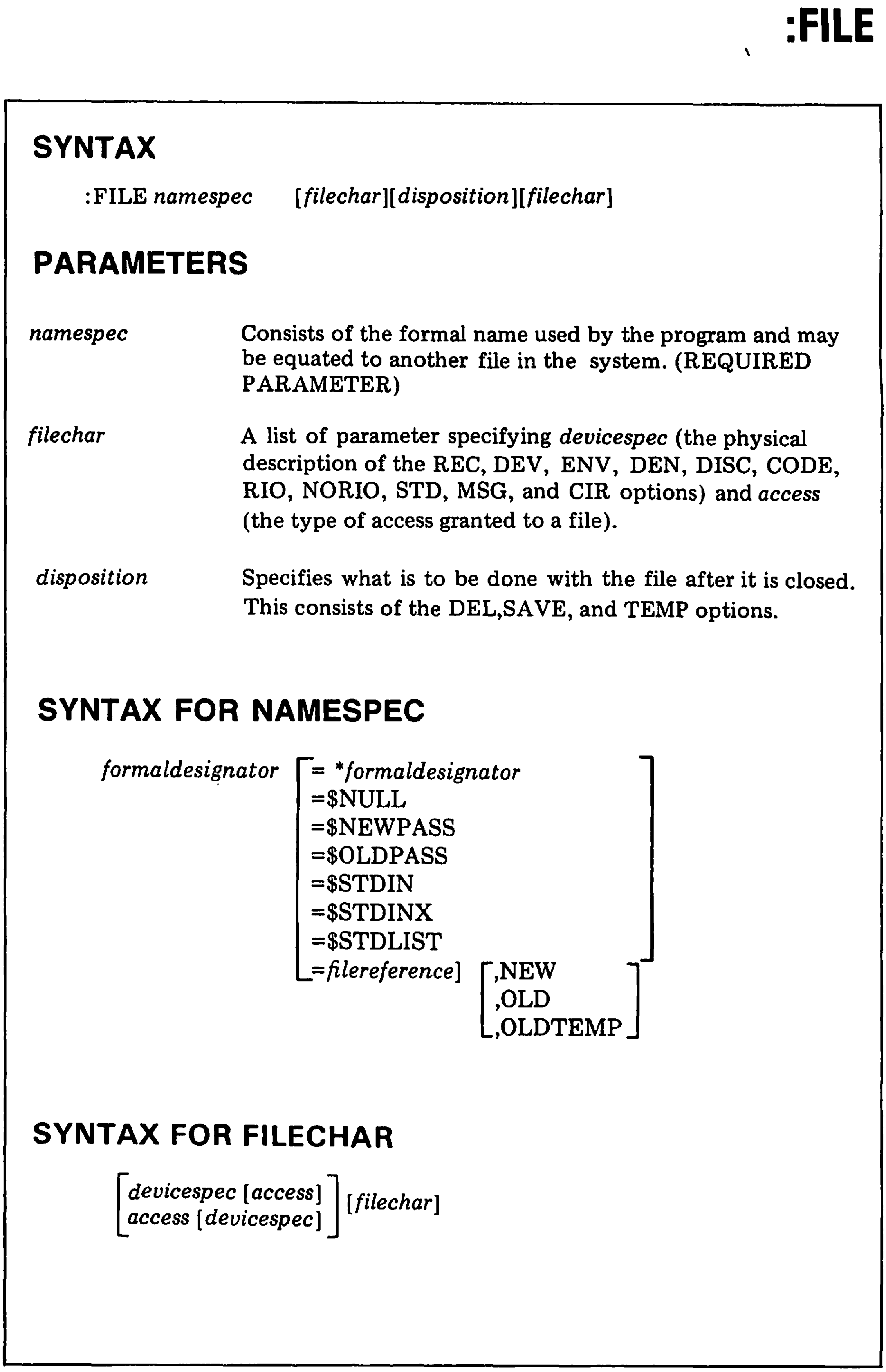
where dsdevice is the device class name, logical device number, or node name that you used when establishing the particular communications link (this specifies the physical line connecting the two computers); and device is the device class name or logical device number of the desired remote device as established within the remote HP 3000.

NOTE

> When the FILE command is entered on a remote system to point back across to a file on the local system, dsdevice is omitted.

Figure 4-1 is an excerpt from the MPE Commands Reference Manual, showing only the parts of the syntax and parameter specifications pertinent to and including the dsdevice# parameter. (Refer to the MPE Commands Reference Manual for the complete syntax and all parameters.) The dsdevice# parameter (within the DEV= parameter) is the only parameter unique to DSN/DS. This one small item of syntax is enormously powerful. It means that from within your local session you can access any of the devices and/or disc files of a remote HP 3000 as though they resided at your local HP 3000 site. Access to remote disc files is, of course, subject to the usual MPE file security. The user, account, and group names that you specified in the REMOTE HELLO command when establishing the communications link are the ones used by MPE in the remote HP 3000 for determining your file access capabilities.

Following figure 4-1 are five annotated examples illustrating remote device and file access from a local session.



:FILE

SYNTAX

:FILE *namespec* [*filechar*][*disposition*][*filechar*]

PARAMETERS

namespec Consists of the formal name used by the program and may be equated to another file in the system. (REQUIRED PARAMETER)

filechar A list of parameter specifying *devicespec* (the physical description of the REC, DEV, ENV, DEN, DISC, CODE, RIO, NORIO, STD, MSG, and CIR options) and *access* (the type of access granted to a file).

disposition Specifies what is to be done with the file after it is closed. This consists of the DEL,SAVE, and TEMP options.

SYNTAX FOR NAMESPEC

```
formaldesignator [= *formaldesignator                 ]
                 [=$NULL                              ]
                 [=$NEWPASS                           ]
                 [=$OLDPASS                           ]
                 [=$STDIN                             ]
                 [=$STDINX                            ]
                 [=$STDLIST                           ]
                 [=filereference]  [,NEW     ]        ]
                                   [,OLD     ]
                                   [,OLDTEMP ]
```

SYNTAX FOR FILECHAR

```
[devicespec [access]]
[access [devicespec]] [filechar]
```

Figure 4-1. MPE FILE Command (Pertinent Excerpts)

SYNTAX FOR DEVICESPEC

```
[;REC= [recsize ] [, [blockfactor ] [, [F ] [,BINARY ]]]]
                                       [U ] [,ASCII  ]
                                       [V ]
[;DEV = [[dsdevice] # ] [device] [,[outpriority] [,numcopies] ] ]
[;ENV = [environment] ]
[;DEN = [density] ]
[;DISC = [numrec] [,numextents] [,initalloc] ] ]
[;CODE = [filecode] ]
[;RIO   ]
[;NORIO ]
[;STD   ]
[;MSG   ]
[;CIR   ]
```

SYNTAX FOR ACCESS

```
[;NOCCTL]                                        [;NOMULTI]
[;CCTL  ]                                        [;MULTI  ]
                                                 [;GMULTI ]

             ( IN       )                        [;NOMR   ]
             ( OUT      )                        [;MR     ]
[;ACC =      { UPDATE   }   ]
             ( OUTKEEP  )                        [;WAIT   ]
             ( APPEND   )                        [;NOWAIT ]
             ( INOUT    )

[;BUF [ = [numbuffers]]]                         [;NOLOCK ]
[;NOBUF                ]                         [;LOCK   ]

[;EXC  ]                                         [;NOCOPY ]
[;SHR  ]                                         [;COPY   ]
[;EAR  ]
[;SEMI ]

[;NOLABEL                                              ] [;FORMS=formsmsg]
[;LABEL [ = [volid] [ , [type] [ , [expdate][, seg]]]] ]
```

SYNTAX FOR DISPOSITION

```
[;DEL ]
[;TEMP]
[;SAVE]
```

Figure 4-1. MPE FILE Command (Pertinent Excerpts) (continued)

Example #1

Assume that you are maintaining an ASCII file containing both uppercase and lowercase characters using the Text Editor on your HP 3000 but that you don't have an upper/lowercase line printer. Assume further that elsewhere in the same building there is another HP 3000 with an upper/lowercase line printer, that both HP 3000s have DS capability, and that they are connected to one another by an interconnecting cable and communications interfaces. You can access the remote line printer as follows.

First, the console operators of both computer systems OPEN the line. Then, you log on to your HP 3000 and establish a communications link with the remote HP 3000.

```
:HELLO USER.ACCOUNT
:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2
```

where USER and ACCOUNT are valid user and account names (respectively) within the accounting structure of your local HP 3000, RUSER and RACCOUNT are valid user and account names (respectively) within the accounting structure of the remote HP 3000, and LINE2 is the device class name of the IODS0 entry (or the node name of the IODSX entry) for the local communications interface to which the interconnecting cable is connected.

Next, issue a local MPE FILE command that defines the desired line printer as a remote device.

```
:FILE LIST;DEV=LINE2#SLOWLP
```

where LIST is the formaldesignator by which you will subsequently reference the line printer, LINE2 is the device class name you used when establishing the particular communications link, the # symbol tells the local file system that the next parameter references a device on the remote system, and SLOWLP is the device class name (as established within the remote HP 3000) of the upper/lowercase line printer.

Then, invoke the Text Editor of your local HP 3000, specifying the remote line printer as the off-line listing device:

```
:EDITOR *LIST
```

Thereafter, direct the Text Editor offline output to the remote upper/lowercase line printer as though it were connected directly to your local HP 3000. For example, you could print the content of the file TEXTFILE on the upper/lowercase line printer as follows:

```
/TEXT TEXTFILE
/LIST ALL,OFFLINE
```

The entire command sequence is as follows (refer to Figure 4-2):

```
:HELLO USER.ACCOUNT
HP3000 / MPE IV C.00.04.  TUE, AUG  3, 1982, 12:51 PM

WELCOME TO SYSTEM A.

:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  TUE, AUG  3, 1982, 12:52 PM

WELCOME TO SYSTEM B.

:FILE LIST;DEV=LINE2#SLOWLP
:EDITOR *LIST

HP32201A.7.10 EDIT/3000  TUE, AUG  3, 1982, 12:53 PM
(C) HEWLETT-PACKARD CO. 1981
/TEXT TEXTFILE
/LIST ALL,OFFLINE
*** OFF LINE LISTING BEGUN. ***

    .
    .
    .
```

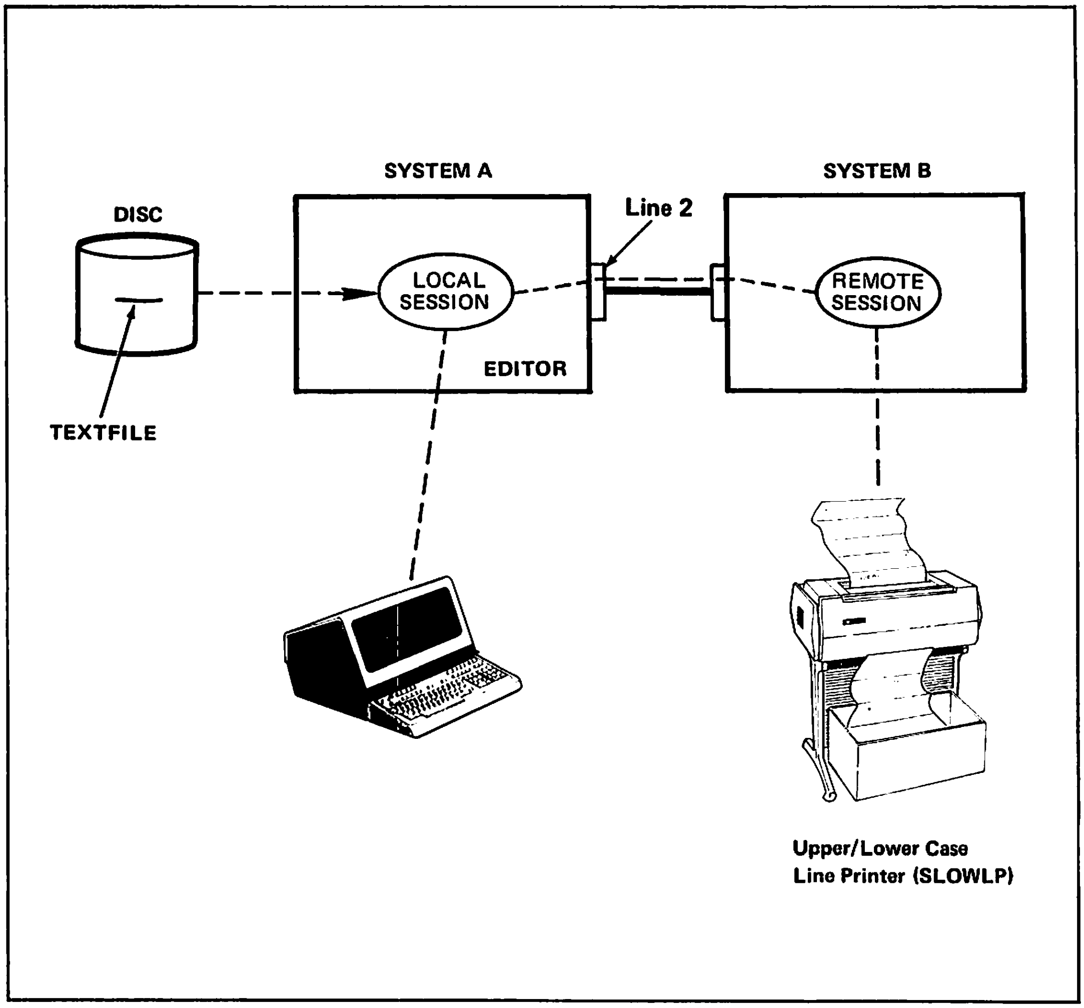


Figure 4-2. Remote Off-Line Listing Example

Example #2

Assume that there is a file named SOURCE residing on a disc connected to a remote HP 3000 and that SOURCE contains a list of clients sorted alphabetically by the clients' names. Assume further that the remote HP 3000 and your local HP 3000 both have DSN/DS configured and that they are interconnected by a hardwired connection. You wish to access the remote file SOURCE from your local HP 3000, sort its content alphabetically by the names of the states in which the clients reside, and store the sorted version in a newly created disc file named SORTED on your local HP 3000. You can do that (without disturbing the original content of SOURCE) as follows.

First, the console operators of both systems open the line to make it available. Then, log on to your local HP 3000 and establish a communications link with the remote HP 3000.

```
:HELLO USER.ACCOUNT
:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2
```

where USER and ACCOUNT are valid user and account names (respectively) within the accounting structure of your local HP 3000, RUSER and RACCOUNT are valid user and account names (respectively) within the accounting structure of the remote HP 3000, and LINE2 is the device class name of the local DSN/DS Communications Driver (IODSO or IODSX) that is associated with the line you want to use.

Next, issue a local MPE BUILD command to create the local disc file SORTED that will receive the sorted output.

```
:BUILD SORTED;DISC=250,1,1;REC=-80,16,F,ASCII
```

Then, issue two local MPE FILE commands: one that defines the remote disc file SOURCE as the sort input file and one that defines the local disc file SORTED as the sort output file.

```
:FILE INPUT=SOURCE;DEV=LINE2#DISC
:FILE OUTPUT=SORTED
```

Then, invoke the Sort program, specify the sort key, and initiate the actual sort.

```
:RUN SORT.PUB.SYS
>KEY 50,9
>END
```

Note that the sort is performed in your local HP 3000, using the remote disc file SOURCE as the sort input file; the output of the sort is stored in the local disc file SORTED; and the original content of SOURCE is not altered.

The entire command sequence is as follows (refer to figure 4-3):

```
:HELLO USER.ACCOUNT
HP3000 / MPE IV C.00.04.  TUE, AUG  3, 1982, 12:51 PM

WELCOME TO SYSTEM A.

:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  TUE, AUG  3, 1982, 12:52 PM

WELCOME TO SYSTEM B.

:BUILD SORTED;DISC=250,1,1;REC=-80,16,F,ASCII
:FILE INPUT=SOURCE;DEV=LINE2#DISC
:FILE OUTPUT=SORTED
:RUN SORT.PUB.SYS

HP32214C.02.06  SORT/3000  TUE, AUG  3, 1982,  12:53 PM
(C) HEWLETT-PACKARD CO. 1980

>KEY 50,9
>END

                   STATISTICS

NUMBER OF RECORDS =                             221
RECORD SIZE (IN BYTES) =                         80
NUMBER OF INTERMEDIATE PASSES =                   0
SPACE AVAILABLE (IN WORDS) =                 13,346
NUMBER OF COMPARES =                             45
```

```
     NUMBER OF SCRATCHFILE IO'S =                         10
     CPU TIME (MINUTES) =                                .01
     ELAPSED TIME (MINUTES) =                            .14

      END OF PROGRAM
     :
          .
          .
          .
```

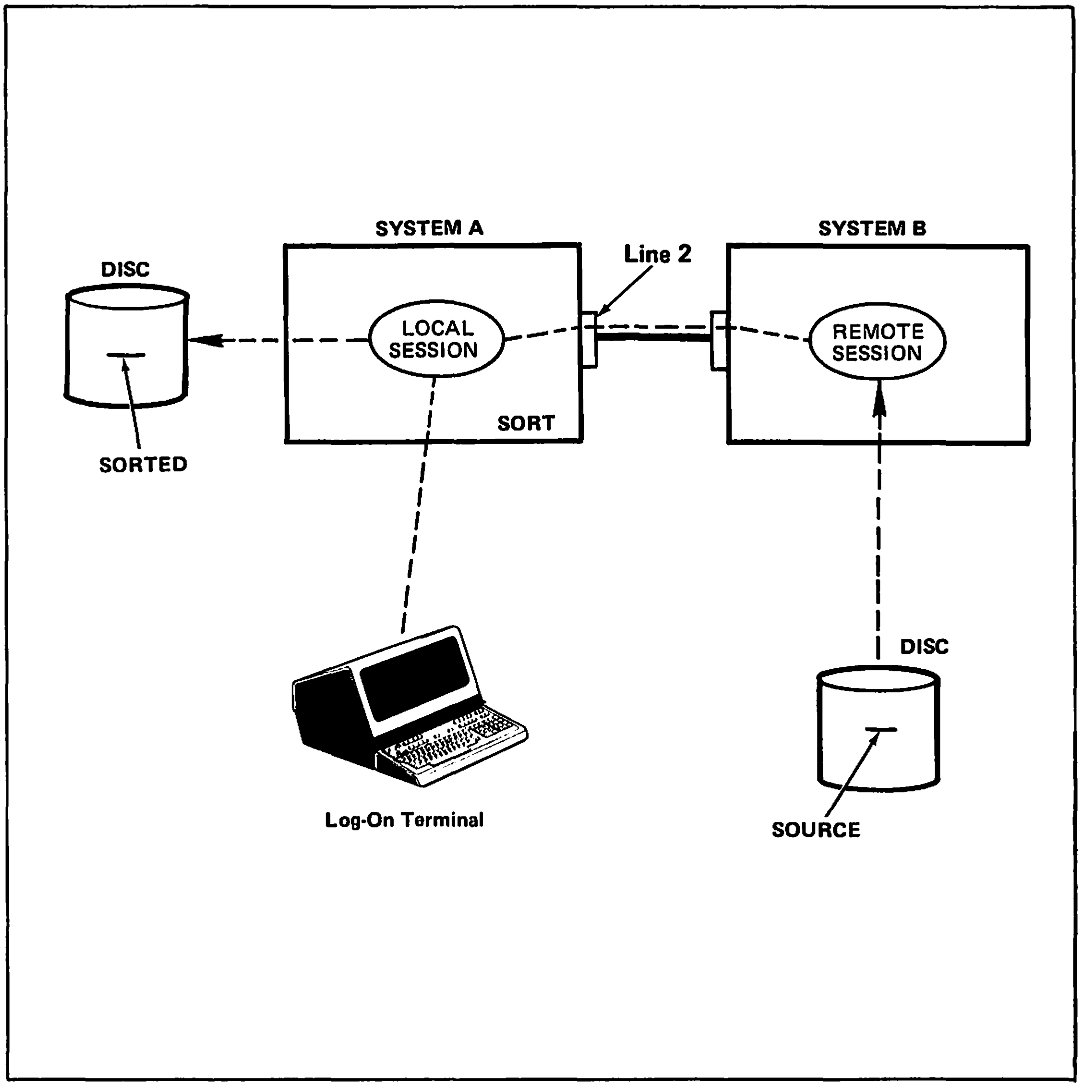


Figure 4-3. SORT Remote File Access Example

Example #3

Suppose that you want to copy a disc file from your local HP 3000 to a remote HP 3000. Assume a hardwired connection and DSN/DS is configured. You can perform the file copy operation as follows.

First, both ends of the line must be opened by the console operators. Then, you log on to your local HP 3000 and establish a communications link with the remote HP 3000.

```
:HELLO USER.ACCOUNT
:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2
```

where USER and ACCOUNT are valid user and account names (respectively) within the accounting structure of your local HP 3000, RUSER and RACCOUNT are valid user and account names (respectively) within the accounting structure of the remote HP 3000, and LINE2 is the node name or the device class name of the local IODS0 that is associated with the line that you want to use.

Next, issue a local MPE FILE command defining the destination file (REMFILE) as being a remote disc file.

```
:FILE REMFILE;DEV=LINE2#DISC
```

Then, invoke the File Copier and specify the file copy parameters.

```
:RUN FCOPY.PUB.SYS
>FROM=LOCFILE;TO=*REMFILE;NEW
```

A new disc file named REMFILE is created in the home group of the RACCOUNT account in the remote HP 3000 and the content of the local disc file LOCFILE is then copied over the communications line into REMFILE.

The entire command sequence is as follows (refer to figure 4-4):

```
:HELLO USER.ACCOUNT
HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982, 12:51 PM

WELCOME TO SYSTEM A.

:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982, 12:52 PM

WELCOME TO SYSTEM B.

:FILE REMFILE;DEV=LINE2#DISC
:RUN FCOPY.PUB.SYS

HP32212A.3.14 FILE COPIER (C) HEWLETT-PACKARD CO. 1981

>FROM=LOCFILE;TO=*REMFILE;NEW
EOF FOUND IN FROMFILE AFTER RECORD 2017

2018 RECORDS PROCESSED *** 0 ERRORS

>EXIT

END OF PROGRAM

    .
    .
    .
```

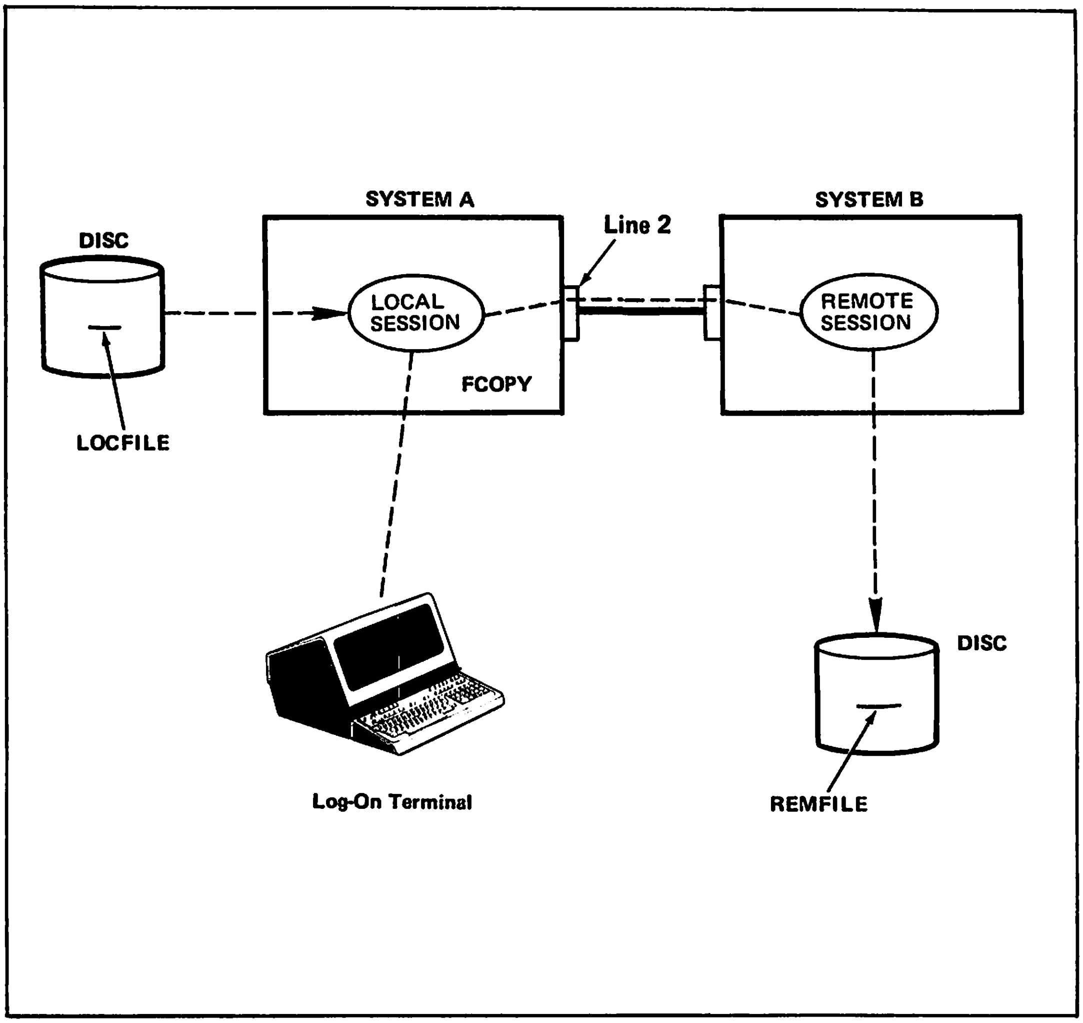


Figure 4-4. FCOPY Remote File Access Example

Example #4

Assume that there is a COBOL source file named SOURCE1 residing on a disc connected to a remote HP 3000 and that you want to compile, prepare, and execute that program on your local HP 3000. Assume further that the remote HP 3000 and your local HP 3000 both have DSN/DS configured and a hardwired interconnection. You can locally compile, prepare, and execute the remote source file as follows.

First, the console operators must open both ends of the DS line. Then, log on to your HP 3000 and establish a communications link with the remote HP 3000.

```
:HELLO USER.ACCOUNT
:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2
```

where USER and ACCOUNT are valid user and account names (respectively) within the accounting structure of your local HP 3000, RUSER and RACCOUNT are valid user and account names (respectively) within the accounting structure of the remote HP 3000, and LINE2 is the device class name of the local IODS0 entry associated with the line that you want to use.

Next, issue a local MPE FILE command defining the file SOURCE1 as being a remote disc file.

```
:FILE SOURCE1;DEV=LINE2#DISC
```

where LINE2 is the node name or the device class name that you used when establishing the communications link and DISC is the device class name (as established within the remote HP 3000) of the disc on which SOURCE1 resides.

Then, invoke the COBOL compiler and the Segmenter of your local HP 3000, specifying the remote disc file SOURCE1 as the inputfile.

```
:COBOLGO *SOURCE1
```

The content of the remote disc file SOURCE1 is compiled, prepared and executed in your local HP 3000.

The entire command sequence is as follows (refer to figure 4-5):

```
:HELLO USER.ACCOUNT
HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982, 12:51 PM

:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982, 12:52 PM

:FILE SOURCE1;DEV=LINE2#DISC
:COBOLGO *SOURCE1

PAGE 0001   HP322130.01.00  (C) HEWLETT-PACKARD CO. 1976

          (SOURCE1 is now being compiled.)

    DATA AREA IS %000341 WORDS.
    CPU TIME = 0:00:01.  WALL TIME  == 0:00:07.
END COBOL/3000 COMPILATION.  NO ERRORS.  NO WARNINGS.

 END OF COMPILE

  (The compiled version of SOURCE1 is now being
   prepared by the MPE Segmenter.)

 END OF PREPARE

  (The compiled and prepared version of SOURCE1 is
   now being executed.)

 END OF PROGRAM
:

    .
    .
    .
```

NOTE

Due to the amount of time and system resources required for COBOL activity, this example (and also Example #5) does not make efficient use of a DS line. The general rule is to do the COBOL compile, preparation, and run on the same system where the data resides. Sometimes this means copying the data files to another system before (or after) COBOL activity, rather than copying across the line during the COBOL activity.

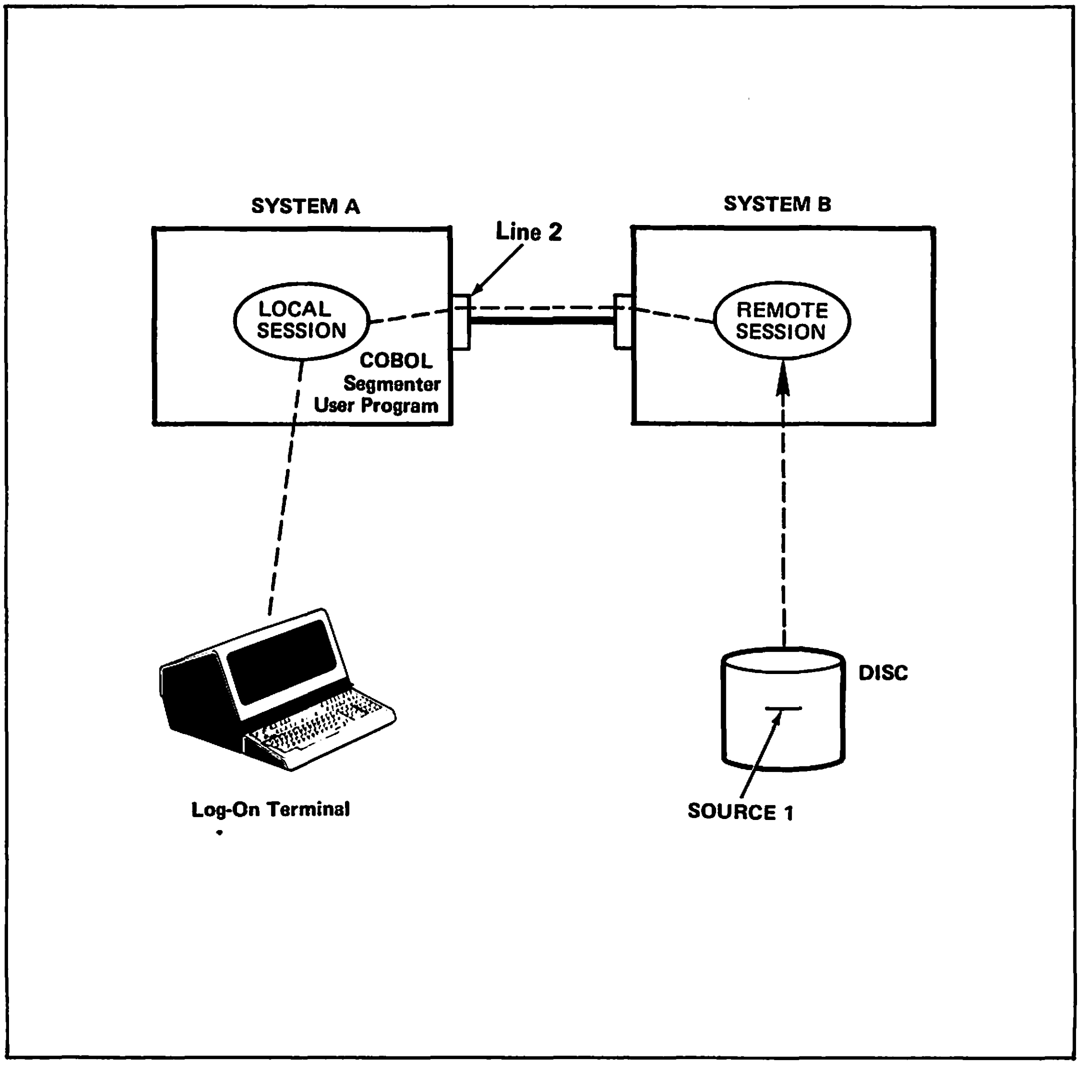


Figure 4-5. COBOLGO Remote File Access Example

Example #5

Assume that there is a COBOL source program named SOURCE1 residing on a disc connected to a remote HP 3000 and that you want to incorporate changes into the content of that file from a local file named CHANGES, compile the updated source code on your local HP 3000, and store a copy of the updated source code in a new file named SOURCE1A on the disc connected to the remote HP 3000 Assume further that the remote HP 3000 and your local HP 3000 both have DS capability and a hardwired interconnection. You can perform the update and compilation as follows:

First, log on to your HP 3000 and establish a communications link with the remote HP 3000.

```
     :HELLO USER.ACCOUNT
     :REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2
```

where USER and ACCOUNT are valid user and account names (respectively) within the accounting structure of your local HP 3000, RUSER and RACCOUNT are valid user and account names (respectively) within the accounting structure of the remote HP 3000, and LINE2 is the device class name of the local IODS0 entry associated with the line that you want to use.

Next, issue two local MPE FILE commands: one that defines the source file SOURCE1 as being a remote disc file and one that defines the file SOURCE1A as a new remote disc file.

```
     :FILE SOURCE1;DEV=LINE2#DISC
     :FILE SOURCE1A,NEW;SAVE;DEV=LINE2#DISC
```

where LINE2 is the node name or the device class name you used when establishing the particular communications link, DISC is the device class name (as established within the remote HP 3000) of the disc on which SOURCE1 resides and SOURCE1A will reside, and NEW;SAVE specifies that SOURCE1A is to be a new permanent file.

Then, invoke the local COBOL compiler, specifying the local disc file CHANGES as the update input file (textfile), the remote disc file SOURCE1 as the source input file (masterfile), and the remote disc file SOURCE1A as the updated source file (newfile).

```
:COBOL CHANGES,,,*SOURCE1,*SOURCE1A
```

The source code in the remote disc file SOURCE1 is updated by the content of the local disc file CHANGES, a new permanent disc file named SOURCE1A is created in the remote HP 3000, and the resultant source code is stored in the remote disc file SOURCE1A. Note that the updating operation is performed by the COBOL compiler in your local HP 3000.

The entire command sequence is as follows (refer to figure 4-6):

```
:HELLO USER.ACCOUNT
HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982, 12:51 PM

:REMOTE HELLO RUSER.RACCOUNT;DSLINE=LINE2

DS LINE NUMBER = #L3

HP3000 / MPE IV C.00.04.  WED, MAR  3, 1982, 12:52 PM

:FILE SOURCE1;DEV=LINE2#DISC
:FILE SOURCE1A,NEW;SAVE;DEV=LINE2#DISC
:COBOL CHANGES,,,*SOURCE1,*SOURCE1A

PAGE 0001   HP322130.01.00  (C) HEWLETT-PACKARD CO. 1976

     (SOURCE1 is now being updated and compiled.)

    DATA AREA IS %000341 WORDS.
    CPU TIME = 0:00:01.  WALL TIME = 0:00:17.
END COBOL/3000 COMPILATION.  NO ERRORS.  NO WARNINGS.

 END OF COMPILE
:
    .
    .
    .
```

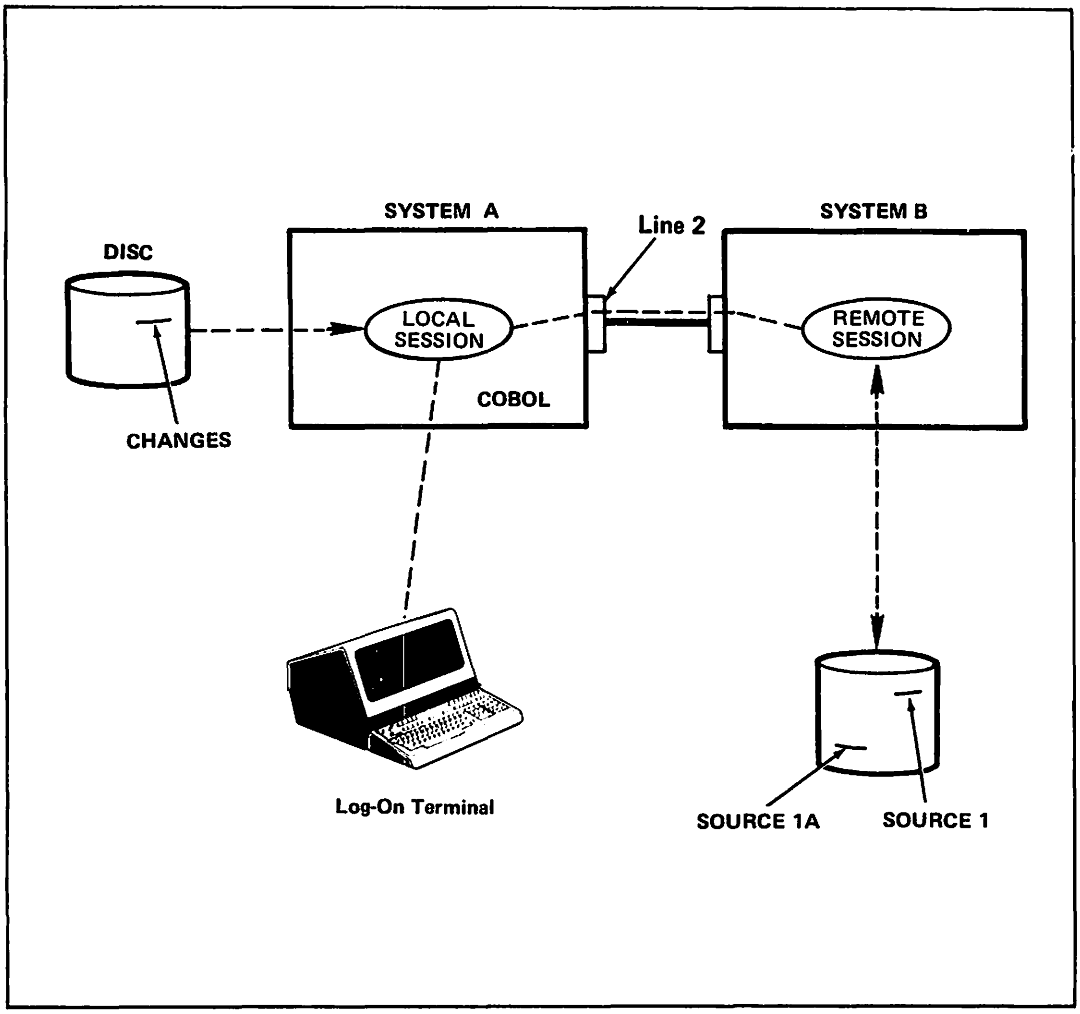


Figure 4-6. COBOL Remote File Access Example

NOTE

Many aspects of system resources (for example, memory size, CPU load, type of CPU load, time quantums, etc.) affect how COBOL activities (and also remote data base access activities) are conducted in a DSN/DS environment. In general, it is more efficient to transfer data, USL, and SL files before or after (but not during) COBOL activity.

PROGRAMMATIC ACCESS

Once a DSN/DS communications link has been established between your HP 3000 and a remote HP 3000, you can use the standard set of MPE File System intrinsics within your local programs to access devices and/or files residing at the remote HP 3000 site. To make this possible, the format of the byte array referenced by the device parameter of the MPE FOPEN intrinsic has been expanded to include a DS line specification in addition to the usual device specification. The format of the device byte array is as follows:

```
dsdevice#[device]
```

where dsdevice is the device class name, logical device number, or node name that you used when establishing the particular communications link (this specifies the physical line connecting the two computers) and device is the device class name or logical device number of the desired remote device as established within the remote HP 3000.

The full syntax for the MPE FOPEN intrinsic is presented in figure 4-7. However, for conciseness, only the "device" parameter specifications are shown in figure 4-8. (For a complete presentation of all FOPEN intrinsic parameters, refer to the MPE Intrinsics Reference Manual.) The addition of "dsdevice#" to the format of the byte array referenced by the device parameter has enormously powerful implications. It means that programs executing in your local HP 3000 can easily access any of the devices and/or disc files of a remote HP 3000 as though they resided at your local HP 3000 site. Access to remote files is, of course, subject to the usual MPE file security. The user, account, and group names that you specified in the REMOTE HELLO command when establishing the communications link are the ones used by MPE in the remote HP 3000 for determining your file access capabilities.

On the pages following figures 4-7 and 4-8, an annotated example illustrates remote device and file access from a local program running within a local session.

The Condition Codes for the various MPE File System intrinsics retain their normal meanings. Any communications line errors will return a CCL. In the event of an error, you can call the MPE FCHECK intrinsic to determine what happened. When using the MPE File System intrinsics for remote file access, the Message Block B (File System) error codes apply to the remote file. You may also use the MPE PRINTFILEINFO intrinsic to display the status of a remote file.

```
      I                      BA       LV       LV      IV

filenum:=FOPEN(formaldesignator,foptions,aoptions,recsize,


                   BA      BA         IV          IV

               device,formmsg,userlabels,blockfactor,


                       IV       DV         IV        IV

               numbuffers,filesize,numextents,initalloc,


                     IV    O-V

               filecode);
```

Figure 4-7. MPE FOPEN Intrinsic Syntax

device

byte array (optional)
Contains a string of ASCII characters terminating with any non-alphanumeric character except a slash or period, designating a local or remote device on which the file is to reside. For a local device the string may represent a device class name up to eight alphanumeric characters beginning with a letter or a logical device number consisting of a three-byte numeric string. For a remote device the string may represent a DS line identifier (the device class name, logical device number, or node name that you used when establishing the particular communications link) followed by a # followed by the device class name or logical device number of the desired remote device.

The format of the array referenced by device is as follows:

```
dsdevice#device
```

where dsdevice is the device class name, logical device number, or node name that you used when establishing the particular communications link (this specifies the physical line connecting the two computers). The dsdevice class names and logical device numbers are defined and assigned to devices and communications interfaces during system configuration. Node names are defined in a Network Configuration data base, where each name is logically associated with a communications interface and a remote node address.

... and where device is the device class name or logical device number of the desired remote device as established within the remote HP 3000. If the file is a newly-created disc file and the device specification is a device class, then all extents of the file are restricted to members of the class. Similarly, if the device specification is a logical device number, then all extents are restricted to the specified logical device.

Default: Local DISC and remote DISC.

Figure 4-8. Pertinent Parameter for MPE FOPEN Intrinsic

Example

The following program illustrates how remote files can be accessed by using file system intrinsics.

```
$CONTROL USLINIT,ADR,MAP,CODE
 BEGIN
 INTEGER
    A,
    I:=-1,
    RDISCNUM,
    RLPNUM;

 BYTE ARRAY RMTLP'FILNAM(0:3):="RLP ";
 BYTE ARRAY RLPDEV(0:11);
 BYTE ARRAY RMTDISC'FILNAM(0:5):="RDISC ";
 BYTE ARRAY MSG(0:71);
 BYTE ARRAY RDISCDEV(0:11);

 LOGICAL ARRAY LMSG(*)=MSG;

 INTRINSIC PRINT,READ,FOPEN,FWRITEDIR,FREADDIR,FWRITE,FCLOSE;

           <<BEGIN OUTER BLOCK>>

 MOVE MSG:="INPUT REMOTE DISC DEVICE CLASS NAME ";
 PRINT(MSG,-35,0);
 MOVE MSG:="IN THE FORM .. DSDEVICE#DISCDEV ";
 PRINT(MSG,-31,0);
 A:=READ(LMSG,-12);
 MOVE RDISCDEV:=MSG,(A);

 MOVE MSG:="INPUT REMOTE LP DEVICE CLASS NAME ";
 PRINT(MSG,-33,0);
 MOVE MSG:="IN THE FORM .. DSDEVICE#LPDEV ";
 PRINT(MSG,-29,0);
 A:=READ(LMSG,-12);
 MOVE RLPDEV:=MSG,(A);

 MOVE MSG:="OPENING REMOTE DISC FILE ";
 PRINT(MSG,-24,0);
 RDISCNUM:=FOPEN(RMTDISC'FILNAM,4,%104,-80,RDISCDEV);
 <<NEW,ASCII>>
 IF <> THEN
    BEGIN
       MOVE MSG:="COULD NOT OPEN REMOTE DISC FILE ";
       PRINT(MSG,-31,0);
       GO TO OUT;
    END;
```

```
MOVE MSG:="WRITING TO REMOTE DISC FILE ";  <<INITIALIZE DISC
FILE>>
PRINT(MSG,-27,0);
MOVE MSG:=" ";
MOVE MSG(1):=MSG(0),(71);

WHILE (I:=I+1) <10 DO
   BEGIN
      MOVE MSG:="REMOTE FILE ACCESS TEST ";
      FWRITEDIR(RDISCNUM,LMSG,36,DOUBLE(I));  <<RECORD TO BE
                                              WRITTEN>>
      IF <> THEN
         BEGIN
            MOVE MSG:="ERROR WHEN WRITING TO REMOTE DISC ";
            PRINT (MSG,-33,0);
            GO TO OUT;
         END;
      END;

MOVE MSG:="OPENING REMOTE LP FILE ";
PRINT(MSG,-22,0);
RLPNUM:=FOPEN(RMTLP'FILNAM,4,1,,RLPDEV);
IF <> THEN
   BEGIN
      MOVE MSG:="COULD NOT OPEN REMOTE LP FILE ";
      PRINT (MSG,-29,0);
   END;

I:=-1;                   <<READING REMOTE DISC>>
WHILE (I:= + 1) < 10 DO
   BEGIN
      FREADDIR(RDISCNUM,LMSG,36,DOUBLE(I));
      IF <> THEN
         BEGIN
            MOVE MSG:="COULD NOT READ REMOTE DISC FILE ";
            PRINT(MSG,-31,0);
         END;
      FWRITE(RLPNUM,LMSG,36,0);
      IF <> THEN
         BEGIN
            MOVE MSG:="COULD NOT PRINT TO REMOTE LP FILE ";
            PRINT(MSG,-34,0);
         END;
      END;

OUT;
END.
```

PROGRAM-TO-PROGRAM COMMUNICATIONS

SECTION V

In the preceding sections, you have seen how you can establish communications links between several HP 3000 computers to form a telecommunications network and how you can execute programs in any of the HP 3000s from a single log-on terminal. Furthermore, you have seen that programs running within any HP 3000 in the network can, under the proper circumstances, obtain access to any of the hardware or software resources available throughout the network. At this point, you already have a powerful telecommunications network capability at your disposal. But if you stopped here, there would be some very important features missing -- features that make DSN/DS a complete teleprocessing tool.

For most teleprocessing applications, it is essential that separate user programs be able to be run simultaneously in separate computers within the network and that they be able to communicate efficiently with one another.

Two capabilities answer that need: DSN/DS Program-to-Program (PTOP) Communications (described in this section) and Interprocess Communications (IPC) (described in Section VII).

You might ask, "Why can't the normal process-handling capabilities of MPE be used for this purpose?" As you probably recall, the process-handling capabilities of MPE permit a user process (referred to as the "father" process), to create and activate one or more "son" processes that then run concurrently with the father process. Father and son processes can communicate efficiently with one another through the use of the SENDMAIL intrinsics, shared extra data segments, or a shared user file. Unfortunately, however, the process-handling capabilities of MPE were designed for use within a single processor. They cannot handle the intervention of a communications line between father and son processes.

Suppose you were to log on to an HP 3000, gain access to a DS line, and initiate a remote session. Within the local session, you use a STREAM command to initiate the execution of a program named PROGA; and within the remote session, you use a STREAM command to initiate execution of a program named PROGB. You now have two programs executing simultaneously: PROGA in your local HP 3000 and PROGB in the remote HP 3000.

At this point, the two programs are entirely independent of one another: neither knows the other exists. If you add a sharedaccess disc file to the situation, PROGA and PROGB can now read from and write to that file, and thereby communicate indirectly with one another. This arrangement works well as long as the data being deposited in the shared file does not have to be retrieved, processed, and responded to within a finite period of time.

There are teleprocessing applications where this type of arrangement is not only adequate but makes a great deal of sense. For example, consider the case where a branch office is accumulating information that must be merged once a day into a data base residing at the main office. In this case, the two programs can make very effective use of the message file approach.

As soon as an application tries to be truly interactive, however, this arrangement falters because the two programs cannot communicate directly. Each must know whether or not the other program is trying to transmit data. The more dependent each program is upon receiving data from the other, the more likely it is that PTOP should be used for the application.

With the remote file access method of program-to-program communication, the two programs had no way of knowing if the other program was actually executing. With the POPEN intrinsic, the master program knows that the slave program is executing, because it created and activated the slave program's process. Likewise, the slave program knows that the master program is executing, because without an active corresponding master program, the slave itself would not be executing.

The DSN/DS program-to-program communications facility provides nine intrinsics that make it possible for two or more user programs residing in separate HP 3000s to exchange data and control information directly (and efficiently) over DSN/DS communications links.

The nature of any 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 a "slave" program) always responds to requests received from the master. Those intrinsics used within a master program are summarized in table 5-1, and those used within a slave program are summarized in table 5-2.

Table 5-1. Master Program-to-Program Intrinsics

Intrinsic Name	Function
POPEN	Initiates and activates a slave process in a remote HP 3000 and initiates program-to-program communication with the slave program.
PREAD	Sends a read 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 program's process.
PCHECK	Returns an integer code specifying the completion status of the most recently executed master program-to-program intrinsic.

Table 5-2. Slave Program-to-Program Intrinsics

Intrinsic Name	Function
GET	Receives the next request from the remote master 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.

Conceptually, the DSN/DS 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 5-3 compares the intrinsics used for process-to-process communication within a single-system environment to those used for program-to-program communication within a distributed systems environment.

Table 5-3. Single System / Distributed Systems Comparison

Function	Single System (Process Handling)	Distributed Systems (Program-to-Program)
Initiate another process.	CREATE ACTIVATE	POPEN
Communicate with the other process.	Mail Intrinsics: SENDMAIL RECEIVEMAIL . . . User Managed Extra Data Segment: GETDSEG DMOVEIN DMOVEOUT . . . Shared User File: FOPEN FREAD FWRITE FCONTROL FCLOSE FCHECK . . .	Master (father) Requests: PREAD PWRITE PCONTROL PCHECK Slave (son) Responses: GET ACCEPT REJECT PCHECK
Terminate the other process.	Father: KILL (a son) TERMINATE (self and all sons)	Master (father): PCLOSE (a slave) TERMINATE (self and all slaves)

When a DSN/DS communications link exists between two HP 3000s, a user program (the master program) can create and activate a son process (a slave program) in the remote HP 3000. The POPEN intrinsic performs this function, in place of the standard MPE CREATE and ACTIVATE intrinsics.

After the master and slave programs are both executing, the master program can:

- Send data (PWRITE) or control information (PCONTROL) directly to the slave program

- Send a read request (PREAD) or control request (PCONTROL) to the slave program asking that the slave send data or control information back to the master

- Check status (PCHECK) and terminate (PCLOSE) a slave program.

Notice the striking similarity between this method of communication and the use of the MPE File System intrinsics FREAD and FWRITE. It is as though the master program is reading from or writing to a file -- a very intelligent file that is capable of making decisions, controlling input/output devices, and performing productive processing.

PTOP INTRINSICS

The following pages contain detailed descriptions of the PTOP intrinsics that were summarized in tables 5-1 and 5-2. For convenience in locating specific items of information in this reference section, these detailed descriptions are presented in a format consistant with that used in the MPE Intrinsics Reference Manual. Also, since this part of the section will be used for repeated reference, the intrinsics are arranged in alphabetical sequence, rather than in the order of normal usage as they were presented in the summary tables.

To call a DSN/DS PTOP intrinsic from an SPL program, use the following procedure:

1. Refer to the intrinsic description to determine the parameter types and their positions in the parameter list.

2. Declare the variables or array names to be passed as parameters by type at the beginning of the program.

3. Include the name of the PTOP intrinsic in an INTRINSIC declaration statement.

4. Issue the intrinsic call at the appropriate place in your program.

ACCEPT

(Slave callable)

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);
```

PARAMETERS

itag

integer array (optional)

A twenty-word array used for transmitting a tag field. The format of the tag field is defined by the user's master and slave programs.

target

integer array (optional)

An array used for transmitting or receiving blocks of data.

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

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

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

tcount

integer by value (optional)

An integer specifying the number of words (if positive) or bytes (if negative) to be transmitted or received.

For PREAD requests, this parameter specifies how much data is to be transmitted from target to the master program.

For PWRITE requests, this parameter specifies how much data is to be moved from the DSN/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.

OPERATION

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 or bytes (tcount) from target to the master program and transmits an optional tag field (itag) to the master program.

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

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

GET

(Slave callable)

Receives the next request from the remote master program.

```
   I        IA  I        I   O-V

ifun:=GET(itag,il,ionumber);
```

FUNCTIONAL RETURN

When the GET intrinsic executes, it returns to the slave program a number (ifun) specifying what type of request was received from the remote master program, as follows:

ifun

0 An error occurred. This value is returned only when the condition code CCL is also returned. Issue a PCHECK intrinsic call (with a dsnum 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 also returned. It indicates that a pending MPE File System I/O without wait request was completed (instead of the expected remote DSN/DS I/O request). ionumber contains the file number associated with the completed I/O request.

PARAMETERS

itag

integer array (optional)

A twenty-word array used for receiving a tag field. The format of the tag field is defined by the master and slave programs.

il

integer (optional)

A word that has meaning only when a PREAD or PWRITE request is received from the master program.

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

For a PWRITE request, il contains an integer specifying the number of words or bytes transmitted from the master program to the DSN/DS buffer on the remote system.

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 file number associated with the completed I/O without wait request.

Default: 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 DSN/DS remote I/O request. 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.

OPERATION

The GET intrinsic receives the next request from the remote master program and accepts an optional tag field (available in itag). The GET intrinsic call implicitly issues an IOWAIT(0) intrinsic call. An ifun of 0 indicates that an IOWAIT error occurred. An ifun of 5 will occur 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 DSN/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 DSN/DS remote I/O request).

NOTE

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

PCHECK

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

(Slave and Master callable)

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

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 DSN/DS program-to-program intrinsic. The various values of icode are shown in Appendix B under the heading "DSN/DS Functional Errors".

PARAMETERS

dsnum integer by value (required)

MASTER PROGRAM: The link identifier returned by the particular POPEN intrinsic that initiated communication with the remote slave program.

SLAVE PROGRAM: 0 (zero); no link identifier is returned to a slave program.

PCHECK Intrinsic

CONDITION CODES

CCE	PCHECK request successfully completed.
CCG	(Not returned.)
CCL	PCHECK request denied because dsnum was invalid.

OPERATION

The PCHECK intrinsic returns an integer value that specifies the completion status of the most recently executed DSN/DS program-to-program intrinsic.

PCLOSE

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

(Master callable)

```
                    IV

             PCLOSE(dsnum);
```

PARAMETERS

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.

OPERATION

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

PCONTROL

(Master callable)

Exchanges tag fields with the remote slave program.

```
            IV   IA   O-V

PCONTROL(dsnum,itag);
```

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 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 and may serve any purpose you desire.

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.

OPERATION

The PCONTROL intrinsic 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.

Although this intrinsic was designed specifically for the exchanging of tag fields, you will notice that itag is an optional parameter (it is also optional for the ACCEPT and REJECT slave program-to-program calls). If the master program did not transmit a tag field, the returned tag field (if any) is not accessible.

The PCONTROL activity is illustrated in figure 5-1.

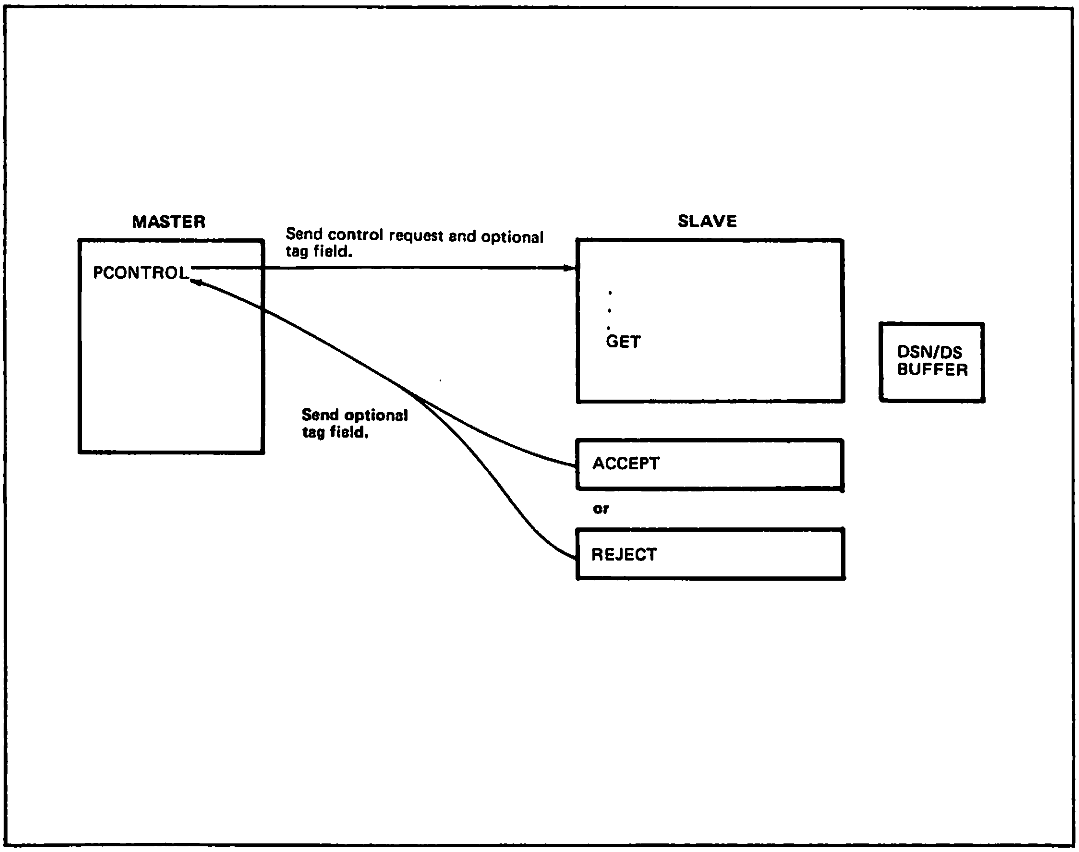


Figure 5-1. PCONTROL Activity

POPEN

(Master callable)

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

```
    I              BA       BA   IA        BA    IV

dsnum:=POPEN(dsdevice,progname,itag,entryname,param,

                LV        IV     IV      IV      IV    O-V

             flags,stacksize,dlsize,maxdata,bufsize);
```

FUNCTIONAL RETURN

When the POPEN intrinsic executes, it returns to the master program a number (dsnum) by which DSN/DS uniquely identifies the particular communications link. This number is analagous to the file number returned by the MPE FOPEN intrinsic in that it is used in all subsequent master program-to-program intrinsic calls to reference the remote slave program.

PARAMETERS

dsdevice byte array (required)

Contains a string of ASCII characters terminated by a space. This string must be the device class name, logical device number, or node name 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 ASCII characters terminated by a space. This string is the name (with optional group and account names) of an MPE program file (residing on a disc connected to the remote HP 3000) containing the remote slave program.

itag

integer array (optional)

A twenty-word array that is used for transmitting and receiving tag fields. The format of the tag field is defined as part of the user's application.

Default: A tag field of all zeros is sent; the returned tag field (if any) is not available to the master program.

entryname

byte array (optional)

Contains a string of ASCII characters terminated by a space. This string is the name of the entry point (label) at which execution of the remote slave program is to begin.

Default: Primary entry point.

param

integer by value (optional)

A word used to transfer control information to the new (remote) process. Any instruction in the outer block of code in the new process can access this information in location Q-4.

Default: Word is filled with zeros.

flags

logical by value (optional)

A word whose bits, if on, specify the loading options for the slave program:

NOTE

> Bit groups are denoted using the standard SPL notation. Thus bit (15:1) indicates bit 15, bits (10:3) indicates bits 10,11, and 12.

Bit(15:1) - (Always set on.)

Bit(14:1) - LOADMAP bit. If on, a listing of the allocated (loaded) program is produced on the job/session list device. This map shows the Code Segment Table (CST) entries used by the new process. If off, no map is produced.

Default: Off.

Bit(13:1) - DEBUG bit. Bit must be off (0) -- no breakpoint can be set.

Default: Off.

Bit(12:1) - If on, the slave program is loaded in non-privileged mode. If this bit is off, the program is loaded in the mode specified when the program file was prepared.

Default: Off.

Bits(10:2) - LIBSEARCH bits. These bits denote the order in which remote libraries are to be searched for the slave program:

00 - System Library.
01 - Account Public Library, followed by System Library.
10 - Group Library, followed by Account Public Library and System Library.

Default: 00.

Bit(9:1) - NOCB bit. If on, file system control blocks are established in an extra segment. If off, control blocks may be established in the Process Control Block Extension (PCBX) area.

Default: Off.

NOTE

This bit should be set on if the slave program uses a large stack.

Bits(7:2) - Reserved for MPE. Should be set to zero.

Bits(5:2) - STACKDUMP bits. Bits must be off (00).

Default: 00

Bit(4:1) - Reserved for MPE. Should be set to zero.

NOTE

The following bits (0:4) are ignored, because the bit pair (5:2) must be 00.

Bit(3:1) - DL to QI bit. If on, the portion of the stack from DL to QI is dumped. If off, this portion of the stack is not dumped.

Default: Off.

Bit(2:1) - QI to S bit. If on, the portion of the stack from QI to S is dumped. If off, this portion of the stack is not dumped.

Default: Off.

Bit(1:1) - Q-63 to S bit. If on, the portion of the stack from Q-63 to S is dumped. If off, this portion of the stack is not dumped.

Default: Off

stacksize integer by value (optional)

An integer (Z - Q) denoting the number of words assigned to the local stack area bounded by the initial Q and Z registers.

Default: The same as that specified in the program file.

dlsize integer by value (optional)

An integer (DB - DL) denoting the number of words in the user-managed stack area bounded by the DL and DB registers.

Default: The same as that specified in the program file.

maxdata integer by value (optional)

The maximum size allowed for the process' stack (Z-DL) area in words. When specified, this value overrides the one established at program-preparation time.

Default: If not specified, and not specified in program file either, MPE assumes that the stack will remain the same size.

bufsize integer by value (optional)

The size in words of the communications buffer (DSN/DS buffer) that is to be established by the remote DSN/DS software. Note that this parameter defines the maximum number of words of data that can be transmitted by a PWRITE or PREAD intrinsic call.

Default: Same size as the line buffer defined by the DSLINE command (LINEBUF=) for the first DSLINE issued to the dsdevice. Will never be smaller than 304 words.

If no LINEBUF= is specified by the first DSLINE command, then the default configuration length is used.

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.

OPERATION

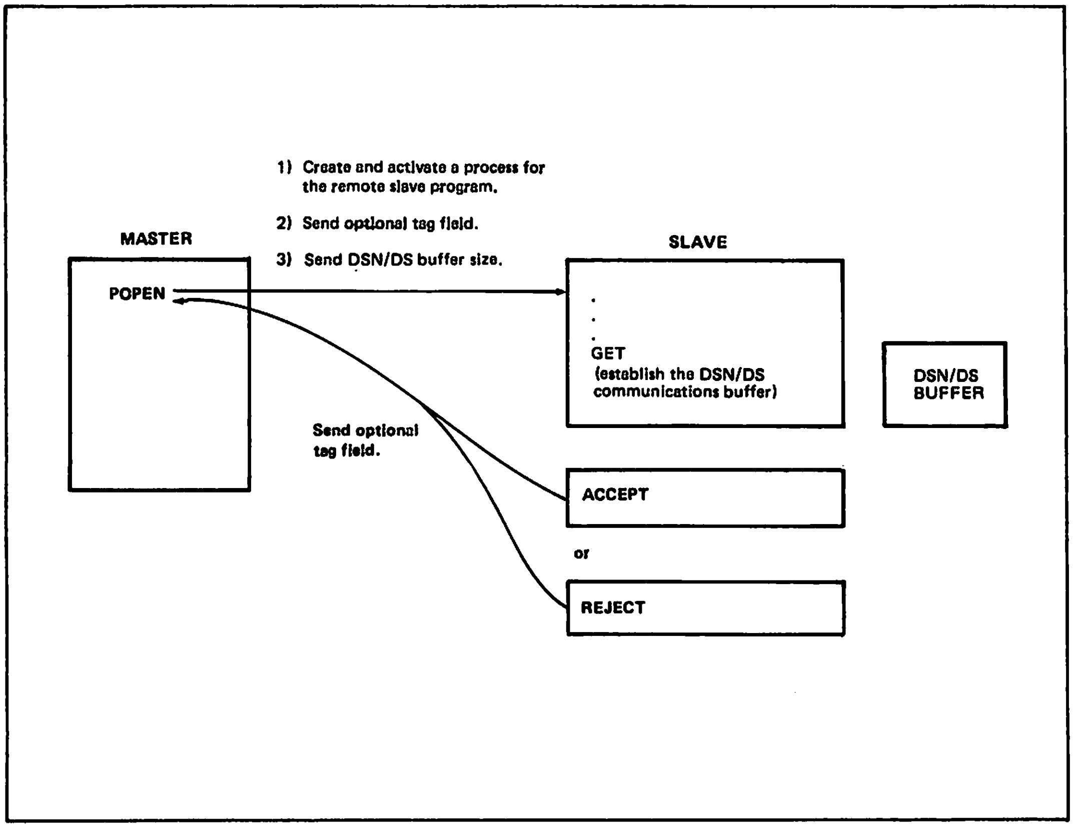
The POPEN intrinsic creates and activates a process in the remote HP 3000 for the specified remote slave program (progname) and optionally transmits a tag field (itag) to that remote slave program. The remote slave program must issue a GET intrinsic call followed by either an ACCEPT or REJECT call to 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 bufsize parameter specifies the length in words of an area to be established by the remote DSN/DS software as a communications buffer. This buffer is established implicitly as part of the GET call that receives the POPEN request.

NOTE

The master program is limited to one slave program on each line. Thus, only one POPEN (to a given node) is permitted. After a POPEN intrinsic call, the remote slave program remains activated, and both the communications link and the DSN/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 design of the user's application.

The POPEN activity is illustrated in figure 5-2.



1) Create and activate a process for the remote slave program.
2) Send optional tag field.
3) Send DSN/DS buffer size.
MASTER
POPEN
SLAVE
GET
(establish the DSN/DS communications buffer)
DSN/DS BUFFER
Send optional tag field.
ACCEPT
or
REJECT

Figure 5-2. POPEN Activity

PREAD

(Master callable)

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

```
   I           IV     IA     IV   IA    O-V

lgth:=PREAD(dsnum,target,tcount,itag);
```

FUNCTIONAL RETURN

The PREAD intrinsic returns a positive integer value showing the length of the information transferred. If the tcount parameter in the PREAD call was positive, the positive value returned represents a word count; if the tcount parameter was negative, the positive value returned represents a byte count.

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 requested number of words (if positive) or bytes (if negative) of data.

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 and may serve any purpose the user desires.

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.

OPERATION

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 GET call moves the tag field from the master program into the itag field provided in the remote slave program. The ACCEPT call moves the requested block of data from the remote program's data buffer into the target in the master program, and it also sends the optional itag back to the master program. The REJECT call transmits no data; it only returns the optional tag field. If the master program did not transmit a tag field, the returned field (if any) is not accessible.

The PREAD activity is illustrated in figure 5-3.

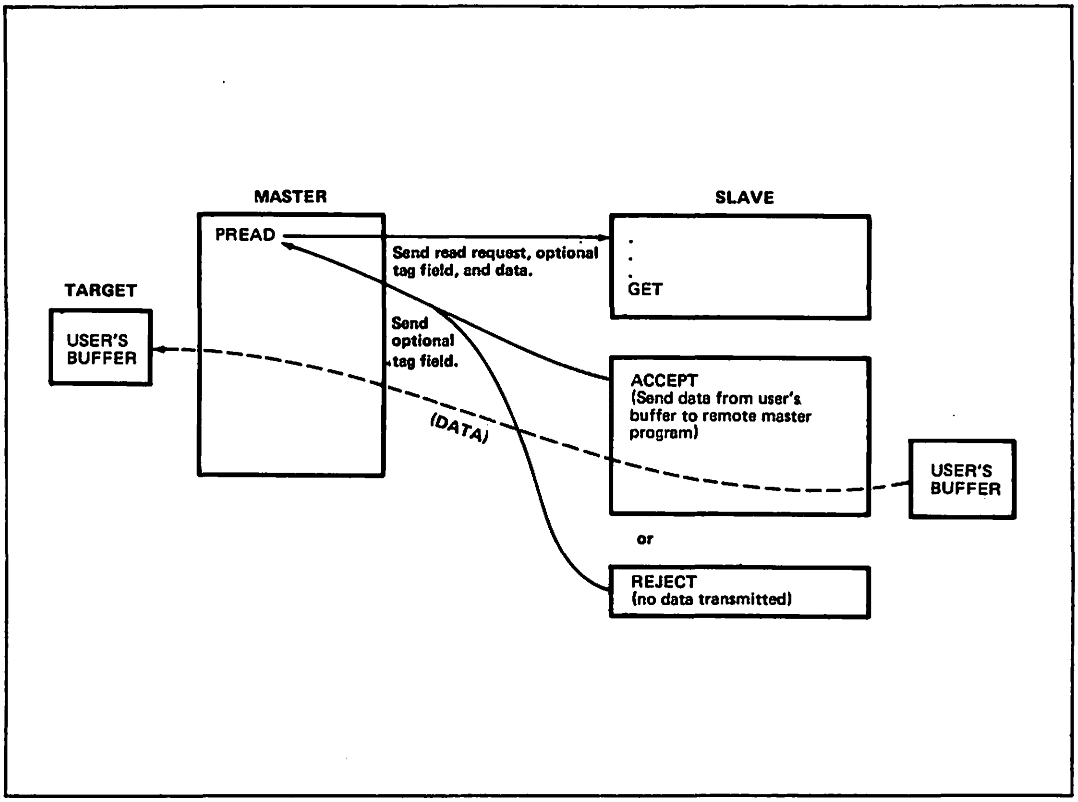

MASTER
PREAD
SLAVE
Send read request, optional tag field, and data.
GET
TARGET
USER'S BUFFER
Send optional tag field.
ACCEPT (Send data from user's buffer to remote master program)
(DATA)
USER'S BUFFER
or
REJECT (no data transmitted)

Figure 5-3. PREAD Activity

PWRITE

Sends a block of data to the remote slave program.

(Master callable)

```
          IV     IA     IV    IA    O-V

PWRITE(dsnum,target,tcount,itag);
```

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 from which data will be transmitted to a remote slave program.

tcount

integer by value (required)

The requested number of words (if positive) or bytes (if negative) of data.

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 and may serve any purpose the user desires.

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.

OPERATION

The PWRITE intrinsic transmits a block of data (number of words = tcount) from target to the DSN/DS buffer in the remote HP 3000, transmits a write 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 PWRITE operation. The GET call moves the tag field from the master program into the itag field provided in the remote slave program, and it also moves the data across the line into the DSN/DS data buffer. The ACCEPT call moves the data from the remote DSN/DS buffer into the remote slave program's buffer, and it also sends the optional itag back to the master program. The REJECT call refuses the write request (the data in the DSN/DS buffer is no longer accessible to the slave program) and returns the optional tag field to the master program.

The PWRITE activity is illustrated in figure 5-4.

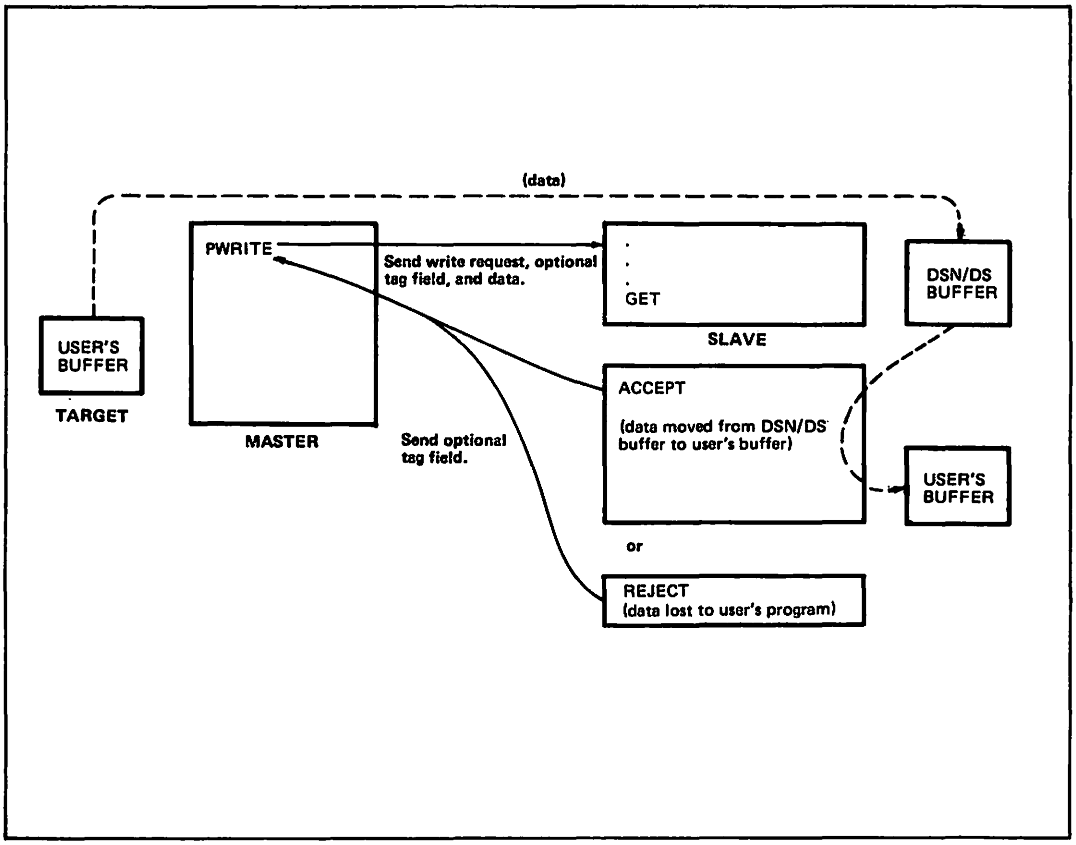

(data)
PWRITE
Send write request, optional tag field, and data.
GET
SLAVE
DSN/DS BUFFER
USER'S BUFFER
TARGET
MASTER
Send optional tag field.
ACCEPT
(data moved from DSN/DS buffer to user's buffer)
USER'S BUFFER
or
REJECT
(data lost to user's program)

Figure 5-4. PWRITE Activity

REJECT

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

(Slave callable)

```
                    IA    O-V

              REJECT(itag);
```

PARAMETERS

itag — integer array (optional)

A twenty-word array used for transmitting a tag field. The format of the tag field is defined by the user's master and slave programs.

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.

OPERATION

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.

INTERFACING WITH COBOL AND BASIC

Access to the program-to-program communications capability is available to ANS COBOL (COBOL/I) and BASIC users only through interface routines. DSN/DS COBOL Interface is covered in Appendix F, and DSN/DS BASIC Interface is covered in Appendix G.

It is not necessary to use the DSN/DS COBOL Interface with COBOL II/3000, however.

PTOP EXAMPLE

This example shows how two programs can communicate with one another by using the master and slave program-to-program intrinsics. The comments included within each program tell what is happening.

Master Program

```
$CONTROL USLINIT,ADR,MAP,CODE
BEGIN

COMMENT
   NAME OF PROGRAM IS MASTERP(S).
   THE SOURCE IS MASTERS.
   THIS PROGRAM IS TO BE RUN ON THE MASTER CPU.  IT
   WILL START THE "SLAVE" PROGRAM ON THE SLAVE CPU.
   THE PROGRAM WILL THEN RECEIVE A KNOWN TEST PATTERN
   FROM THE USER TERMINAL, WRITE IT TO THE REMOTE DISK
   FILE, READ IT BACK, AND PRINT IT ON THE LOCAL LP.
   THE TRANSFER OF DATA IS DONE THRU PTOPC.;


INTEGER
  ERROR,
  LINE'NUM,
  I,
  J,
  LPDEV'NUM;

BYTE ARRAY DS'DEVICE(0:6):="       ";
BYTE ARRAY LPDEV(0:2):="LP ";
BYTE ARRAY LPFILE(0:6):="LPFILE ";
BYTE ARRAY MSG(0:79);
BYTE ARRAY PROG'NAME(0:19):="SLAVEP.PUB.SUPPORT ";


LOGICAL ARRAY IOBUF(0:39);
LOGICAL ARRAY ITAG(0:19):=20(%020040);
LOGICAL ARRAY MSGW(*)=MSG;
LOGICAL ARRAY DS'DEVW(*)=DS'DEVICE;


INTRINSIC DEBUG,FCLOSE,FOPEN,FWRITE,PCONTROL;
INTRINSIC PCLOSE,POPEN,PREAD,PRINT,PWRITE,READ;

MOVE MSG:="      INPUT NAME OF DSDEVICE";
PRINT(MSGW,-28,0);
READ(DS'DEVW,-7);

MOVE MSG:="      POPEN ISSUED";
PRINT(MSGW,-18,0);

LINE'NUM:=POPEN(DS'DEVICE,PROG'NAME,ITAG);
  IF <> THEN

    BEGIN
      PRINT(ITAG,20,0);
      ERROR := 1;
      GO TO ERR'PROC;
    END
    ELSE
      PRINT(ITAG,20,0);


MOVE MSG:="      POPEN COMPLETED SUCCESSFULLY";
PRINT(MSGW,-33,0);

LPDEV'NUM:=FOPEN(LPFILE,4,1,40,LPDEV);
  IF <> THEN BEGIN ERROR:=2;GO TO ERR'PROC; END;

MOVE MSG:="IN PUT TEST RECORD MAX. 80 CHAR";
PRINT(MSGW,-30,0);

MOVE IOBUF:="  ";             <<CLEAR OUT BUFFER AREA>>
MOVE IOBUF(1):=IOBUF,(39);

READ(IOBUF,-80);               <<GET RECORD TO WRITE>>

PWRITE(LINE'NUM,IOBUF,40);     <<SEND RECORD TO REMOTE>>
IF <> THEN BEGIN ERROR:=3;GO TO ERR'PROC; END;

MOVE MSG:="      DISK FILES BEING XFERRED FROM REMOTE";
PRINT(MSGW,-41,0);
J:=-1;                   <<START READING FROM REMOTE>>
WHILE (J:=J+1)<5 DO
  BEGIN
    MOVE MSG:="       PREAD ISSUED";
    PRINT(MSGW,-19,0);

    MOVE IOBUF:="  ";
    MOVE IOBUF(1):=IOBUF,(39);

    I:=PREAD(LINE'NUM,IOBUF,40,ITAG);
    IF = THEN
      BEGIN
        IF J=4 THEN
          BEGIN
            MOVE MSG:="      ALL DISK RECORDS XFERRED";
            PRINT(MSGW,-29,0);
          END;
      END
        ELSE
      BEGIN ERROR:=4;GO TO ERR'PROC; END;
    FWRITE(LPDEV'NUM,IOBUF,I,0);
      IF <> THEN BEGIN ERROR:=4;GO TO ERR'PROC;END;
  END;

FCLOSE(LPDEV'NUM,0,0);
PCLOSE(LINE'NUM);
IF <> THEN BEGIN ERROR:=5;GO TO ERR'PROC;END;
MOVE MSG:="END OF MASTER PROGRAM";
PRINT(MSGW,-21,0);  GO TO END'IT;

ERR'PROC:      <<HANDLE ERROR CONDITIONS>>
  DEBUG;
  FCLOSE(LPDEV'NUM,0,0);
  PCLOSE(LINE'NUM);
  MOVE MSG:="ERROR, END MASTER PROGRAM";
  PRINT(MSGW,-25,0);

END'IT:  END.
```

Slave Program

```
$CONTROL USLINIT,ADR,MAP,CODE

BEGIN
COMMENT
   THE NAME OF THIS PROGRAM IS SLAVEP(S).
   THE NAME OF THE SOURCE IS SLAVES.
   THIS PROGRAM IS TO BE COMPILED AND PREP'ED ON THE
   SLAVE HP3000 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
  ERROR,
  DISK'FILENUM,
  I,
  IL,
  IONUMBER,
  J;

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

LOGICAL ARRAY DISK'BUF(0:39);
LOGICAL ARRAY ITAG(0:19):=7(020040);
LOGICAL ARRAY MSGW(*)=MSG;

INTRINSIC FOPEN,DEBUG,FWRITEDIR,FREADDIR,FCLOSE;
INTRINSIC GET,ACCEPT,PRINT,READ,REJECT;


IL:=40;
MOVE MSG:="ISSUING A GET (REMOTE)";
PRINT(MSGW,-22,0);
I:=GET(ITAG);                  <<GET FOR POPEN>>
IF < THEN
  BEGIN
    MOVE ITAG:="ERROR ON GET;POPEN";
    GO TO ERR'OPEN;
  END;

IF I=1 THEN
  BEGIN
    MOVE MSG:="POPEN RCVD...ISSUING AN ACCEPT (REMOTE)";
    PRINT(MSGW,-39,0);
  END;

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

DISK'FILENUM:=FOPEN(TEST,4,%104,-80,,,,1,1,10D);
  IF <> THEN BEGIN ERROR:=1;GO TO ERR'PROC;  END;

I:=GET;                       <<TEST REC FROM MASTER>>
IF <> THEN BEGIN ERROR:=2; GO TO ERR'PROC; END;
IF I=3 THEN                   <<PWRITE RECEIVED>>
  BEGIN
    ACCEPT( ,DISK'BUF);
    IF <> THEN BEGIN ERROR:=3; GO TO ERR'PROC; END;
  END;


I:=-1;                 <<START WRITING TEST FILE>>
WHILE(I:=I+1) < 5 DO
  BEGIN                          <<WRITE REC TO DISK>>
    FWRITEDIR(DISK'FILENUM,DISK'BUF,40,DOUBLE(I));
      IF <> THEN BEGIN ERROR:=4; GO TO ERR'PROC; END;

  END;                      <<END WRITING TEST FILE>>

J:=-1;                      <<SEND DISK FILE TO MASTER>>
WHILE(J:=J+1)<5 DO
```

```
  BEGIN
    MOVE MSG:="ISSUING A GET (REMOTE)";
    PRINT(MSGW,-22,0);
    I:=GET(ITAG,IL,IONUMBER);
    IF < THEN BEGIN ERROR:=5; GO TO ERR'PROC; END;
    IF I=2 THEN
      BEGIN
        MOVE MSG:="PREAD RCVD...ISSUING AN ACCEPT
        (REMOTE)";
        PRINT(MSGW,-39,0);
      END
      ELSE
        BEGIN ERROR:=6;GO TO ERR'PROC; END;
    MOVE DISK'BUF:=%020040;
    MOVE DISK'BUF(1):=DISK'BUF(0),(39);
    FREADDIR(DISK'BUF,40,DOUBLE(J));
    IF <> THEN BEGIN ERROR:=7;GO TO ERR'PROC; END;
    ACCEPT(ITAG,DISK'BUF,40);
    IF <> THEN BEGIN ERROR:=8;GO TO ERR'PROC; END;
  END;

FCLOSE(DISK'FILENUM,0,0); GO TO END'IT;

ERR'PROC:          <<HANDLE ERROR CONDITIONS>>
  DEBUG;   <<WILL PROMPT OT MASTER SIDE TERMINAL>>
  REJECT;
  I:=GET;  <<ALLOW PCLOSE>>
GO TO ERR'PROC;

END'IT: END;
```

NETWORK FILE TRANSFER

SECTION
VI

The Network File Transfer (NFT) program runs on an HP 3000 Computer System to provide the ability to efficiently copy disc files. When initiated over a DSN/DS communication link, the NFT program can copy a file to or from any other adjacent HP 3000 computer which also provides this service.

FEATURES OF NFT

- You can initiate copy operations from sessions, jobs, or programs.
- DSCOPY can be used to copy users' files and MPE system files, as well as data management files, such as KSAM/3000 files.
- There is only one NFT command to learn -- :DSCOPY.
- There are two intrinsics: DSCOPY and DSCOPYMSG. The intrinsics are callable from programs written in SPL, COBOL, FORTRAN, and BASIC.
- NFT can be used in Interactive Mode to submit a series of copy requests. When a DSCOPY command or intrinsic initiates Interactive Mode, users' requests are placed in a transaction file whose formal designator is DSCOPYI. The default for this file is $STDINX.
- * NFT can record a history of all copy operations performed by DSCOPY requests. The history report can be printed to $STDLIST, as well as to a secondary file.
- You can initiate a copy operation from a system other than the system(s) where the source and target files are located.
- NFT can efficiently copy disc files within your local HP 3000.
- The files referenced by a DSCOPY command (or intrinsic) may reside on system or private volumes.

File transfers can involve one or more computers. In all transfers, there are three distinct roles a system can play:

1. The initiator is always the system where the :DSCOPY command originates. The initiator functions only in an outgoing sense. It is similar to PTOP operation, where the PTOP master program always issues a POPEN out across a DS line to cause a slave to be created and activated on a remote system.

2. The producer is the source computer where the file that is to be copied resides.

3. The consumer is the target computer where the new file will reside.

You should remember that one system may be performing two or all three of these roles.

When a DSCOPY request names a remote source, the DS line to that computer must be open and a remote session must exist. The same is also true when a remote target is specified.

When DSCOPY is used to transfer files over two or more systems, the following restrictions apply:

1. DSCOPY must be initiated only from the master side of the DS line. The slave (remote) side cannot be the initiator of a DSCOPY command.

2. DSCOPY must not be initiated programmatically from either a master or a slave PTOP program in any direction.

SYNTAX

```
                                ,              TO                 tdsdev      ,
:DSCOPY [sfile [,[sdsdev] [     [sdev]]] [        [tfile [,   [         ] [     [tdev]]]]]]
                                #           ;                     *           #
```

To submit a series of transfer requests, omit all of the source and target parameters to initiate Interactive Mode. NFT prompts you for input and, after the transfer completes, prompts you again.

Terminate Interactive Mode by typing // or Control-Y.

PARAMETERS

sfile (Required Parameter) Identifies the file to be copied. The name can be written in the following format:

sfile[/lockword][.groupname][.accountname]

If the source file is in a group.account different from the requestor's log-on group.account, the requestor must have read and lock access to the source file.

sdsdev (Optional Parameter) The device classname, logical device number, or node name that was used to open the communications link to the remote computer where the source file resides.

Default: The local system (that is, the system where the transfer request is submitted).

sdev (Optional Parameter) The classname or logical device number of the disc where the source file resides.

Default: DISC.

tfile (Optional Parameter) Specifies the file to receive the data. The name can be written in the following format:

tfile[/lockword][.groupname][.accountname]

Default: The new file has the same filename as the source file. The default groupname and accountname are the log-on groupname and accountname. Security is on for the new file, even though the source file may have been released.

:DSCOPY

tdsdev (Optional Parameter) The device clasname or logical device number that was used to open the communications link to the remote computer where the target file will reside.

Default: DSCOPY copies the sourcefile to the local computer and assigns the same filename as the sourcefile name. If the source computer is the local system, this default causes a file system error (because the file already exists).

* Means the target dsdevice (the target computer) is the same as the source dsdevice (the source computer).

tdev (Optional Parameter) The device classname or logical device number of the disc where the new file should reside.

Default: DISC

USE

Available	in Session?	YES
	in Job?	YES
	in Break?	NO
	Programmatically?	NO*
Breakable?		YES

* Call the DSCOPY intrinsic rather than use the COMMAND intrinsic.

OPERATION

NEVER BREAK AND ABORT DSCOPY DURING A COPY OPERATION.

Control-Y can be used to show how much of a file has been transferred and to cancel a currently executing copy request. If you enter Control-Y during a copy operation, DSCOPY prints the percentage of the transfer that is complete and prompts whether to cancel or continue the operation.

Source and Target Files

In a DSCOPY command, source and target files are referenced as defined by the systems upon which they reside.

There is no default for a sourcefile.

A default for a targetfile is derived from the sourcefile. The default consists of the first sequence of characters in the sourcefile name which constitutes a legal HP 3000 file name. For example:

```
     :DSCOPY SFILE.SGROUP,SNODE
```

Here the source file is SFILE (in group SGROUP on a remote system). The targetfile is generated in the users' log-on group (on the local system) and is assigned the default name SFILE. The characteristics of the new file are the same as those of the source file.

If a source file has a negative file code, the user requesting the transfer must have Account Manager (AM), System Manager (SM), or Privilege Mode (PM) capability to be allowed to copy the file. The log-on user on the target node must also have AM, SM, or PM capability. Remember that the person requesting the transfer (the initiator) is not necessarily the consumer (the log-on user at the target). After a successful copy operation, the new file has the same negative file code as the source file.

When copying KSAM files, both the data file and its key file are copied. The DSCOPY user can specifically name a data file/key file pair by enclosing the file names in quotes and separating them by a comma. For example:

```
     :DSCOPY SFILE TO "DATAFILE,KEYFILE"
```

When a user specifies a source KSAM data file and the NFT subsystem must generate a default key file, it uses the data file name and appends a K. For example:

```
     :DSCOPY SFILE, LINE1 TO TFILE, LINE2
```

In the case where SFILE is a KSAM data file, the new data file on the computer connected to LINE2 will be named TFILE and the associated key file will be named TFILEK by default.

Interactive Mode

To execute a series of transactions, enter the :DSCOPY command without parameters. Now the system prompts you for input with the word DSCOPY and accepts your response from the file DSCOPYI (whose default is $STDINX).

The syntax required for your response follows the format already described for source and target parameters.

Note the following about Interactive Mode:

- To continue your response on the next line, enter an ampersand (&) as the last non-blank character on the current line and press RETURN. A continuation prompt is printed so that you can continue your response.

- To cancel a response while entering a line, use Control-Y.

  You can issue MPE comands while in Interactive Mode by entering a colon (:) before the command. The MPE commands allowed in Interactive Mode are those allowed by the COMMAND intrinsic.

- To terminate Intractive Mode, enter // or Control-Y in response to a DSCOPY prompt.

Event Recording

DSCOPY produces printed output to document user input and copy results. This output may be sent to a primary file and/or a secondary file, either of which may be disabled. The primary file is $STDLIST and the secondary file has the formal designator DSCOPYL. All user requests and DSCOPY prompts are printed on $STDLIST and echoed on the secondary file (and on the primary, if not duplicative). Primary output is enabled by a DSCOPY command, or by the DSCOPY intrinsic with the OPT parameter set to 4, 5, or 6 (refer to the parameters of the DSCOPY intrinsic). Output for the secondary file, DSCOPYL, defaults to $NULL so that secondary output is disabled by default. It can be enabled by using a :FILE command to equate DSCOPYL to a file or a line printer, or to $STDLIST.

EXAMPLES

Local Copy

To make a local copy of SFILE and name the new file TFILE, use either of the following:

```
     :DSCOPY SFILE TO TFILE   or    :DSCOPY SFILE; TFILE
```

The following example copies a file named SFILE from another group on the local system into a file in the log-on group. The new file is also named SFILE.

```
     :DSCOPY SFILE.SGROUP
```

Remote-to-Local Copy

To copy a file from the computer connected to DS line SYSA into your log-on group (on the local system), enter:

```
     :DSCOPY SFILE,SYSA;TFILE
```

Local-to-Remote Copy

To copy a file named SFILE (on the local system) to the computer attached to DS line SYSB and name the new file SFILE, enter:

```
     :DSCOPY SFILE TO ,SYSB
```

Remote Copy

An asterisk (*) means the target system is also the source system. The following example copies a file named SFILE to a new file named TFILE. Both files reside on the remote computer connected to the dsline named SYSA.

```
    :DSCOPY SFILE,SYSA TO TFILE,*
```

Remote-to-Remote Copy

The next example illustrates a command that copies a file from one remote system to another. In this case, the communications lines to both remote computers must be open and a remote session must exist on each system.

```
     :DSCOPY SFILE,SYSA TO TFILE,SYSB
```

NFT INTRINSICS

Programs can use the DSCOPY intrinsic to copy disc files.

Programs can also print a message which corresponds to the result code returned by a DSCOPY intrinsic call. The DSCOPYMSG intrinsic is used for this purpose.

The rules for using the intrinsics are consistent with those for using other MPE intrinsics. Specifically, the following rules apply.

- Both intrinsics can be called from programs written in the SPL/3000, COBOL, FORTRAN, and BASIC languages.
- Calling sequences for all of the languages are basically the same.
- All parameters are passed by reference.
- The intrinsics are not option variable.
- Neither of the intrinsics are typed (returns a parameter as its value).
- Neither returns a condition code (they both return a result).
- Split stack calls are not allowed.
- For COBOL, data types should be defined as follows:

Data Type	**Data Description**
Numeric	PICTURE S9(4) COMPUTATIONAL
Alphanumeric	PICTURE X(n) or picture A(n)
Numeric Array	PICTURE S9(4) COMPUTATIONAL SYNCHRONIZED OCCURS n TIMES

THE DSCOPY INTRINSIC

SPL Procedure Declaration

```
PROCEDURE DSCOPY (OPT, SPEC, RESULT);
     VALUE SPEC, RESULT;
     LOGICAL OPT;
     LOGICAL POINTER SPEC, RESULT;
```

OPT OPT controls the primary output (i.e. output to $STDLIST) and specifies the type of copy operation.

Bits 0 through 12 are reserved for future use and should be set to zero. The remaining bits can be set to indicate the following:

Value	Meaning
0	Single transaction; primary output disabled.
1	Multiple transactions; return after first unsuccessful transaction; primary output disabled.
2	Multiple transactions; return after all transactions have been attempted or after an internal error occurs; primary output disabled.
4	Single transaction; primary output enabled.
5	Multiple transactions; return after first unsuccessful transaction; primary output enabled.
6	Multiple transactions; return after all transactions have been attempted or after an internal error occurs; primary output enabled.

SPEC The logical array should contain ASCII text terminated by an 8-bit binary zero. In the single transaction case, the syntax required is the same as for the DSCOPY command parameters.

In the multiple transaction case, the array should contain only a zero. Zero causes NFT to read the copy request from the DSCOPYI file (whose default is $STDIN).

RESULT   A two-word array returned to the caller which indicates the outcome of the intrinsic call.

RESULT(0)   Result=0 indicates the copy operation was successful. Any other value represents an error as defined in "DSCOPY Error Messages" listed in Appendix B.

RESULT(1)   Shows the number of files that were successfully copied.

COBOL Calling Sequence

```
CALL "DSCOPY" USING OPT, SPEC, RESULT.
```

OPT   Numeric data item.

SPEC   Alphanumeric data item.

RESULT   Numeric array of two or more data items.

FORTRAN Calling Sequence

```
CALL DSCOPY (OPT, SPEC, RESULT)
```

OPT   INTEGER*2 variable

SPEC   CHARACTER array

RESULT   An array of two or more INTEGER*2 variables

BASIC Calling Sequence

```
CALL BDSCOPY (0, S$, R)
```

0   Numeric variable

S$ A string variable

R   An array of two or more numeric variables

Programmatic DSCOPY Operation

Simultaneous DSCOPY requests cannot be issued from two processes in the same session.

The only valid values for the OPT parameter are: 0, 1, 2, 4, 5, or 6.

The ASCII text passed by the SPEC parameter must be terminated by a binary zero.

The values passed in the parameters are verified as being in bounds and valid.

The system creates the NFT process and passes the contents of OPT and SPEC to it.

The specified files are copied by the NFT process.

The intrinsic returns the result to the user.

After processing, OPT and SPEC remain unchanged except that any ASCII lower case characters in SPEC may have been shifted to upper case.

After processing, the first word of the RESULT contains a number which indicates the outcome of the DSCOPY request. A zero value indicates a successful transfer operation; the meaning of any other value is given under "DSCOPY Error Messages" in Appendix B. The second word contains the number of files successfully copied.

THE DSCOPYMSG INTRINSIC

SPL Procedure Declaration

```
PROCEDURE DSCOPYMSG (RESULT, FNUM, R);

     VALUE FNUM;
     LOGICAL ARRAY RESULT;
     INTEGER FNUM, R;
```

RESULT The two-word result returned by the DSCOPY intrinsic.
0 = DSCOPY was successful.
n = An error occurred. Refer to the Error Messages in Appendix B for the meaning.

FNUM When FNUM=0, the message associated with RESULT is printed on $STDLIST.

When FNUM contains a file number returned by an FOPEN call, the message associated with RESULT is written to the file.

R Result returned by this DSCOPYMSG call.
0 = Successful call
n = Unsuccessful call. Refer to the Error Messages in Appendix B.

COBOL Calling Sequence

```
CALL "DSCOPYMSG" USING RESULT, FNUM, R.
```

RESULT An array of two or more data items.

FNUM A numeric data item.

R A numeric data item.

FORTRAN Calling Sequence

```
CALL DSCOPYMSG (RESULT, FNUM, R)
```

RESULT   An array of two or more INTEGER*2 variables.

FNUM   INTEGER*2 variable

R   INTEGER*2 variable

BASIC Calling Sequence

```
Call BDSCOPYMSG (R, F, R0)
```

R   An array of two or more numeric variables

F   An integer variable

R0   An integer variable

EXAMPLES

A very simple example of a programmatic DSCOPY request is shown coded in the COBOL, FORTRAN, and BASIC languages.

The example copies a file (NFTTESTS) to a new file (TEMP1). The source file resides on the local machine, and the new file will be created on a remote machine connected to line "HDS".

DSCOPY COBOL Example

```
$CONTROL CODE
$TITLE " DSCOPY INTRINSIC TEST"
 IDENTIFICATION DIVISION.
 PROGRAM-ID.       DSCOPY00.
 AUTHOR.           JIM BRANDT.
 DATE-WRITTEN.     APRIL 1980.
 DATE-COMPILED.
 REMARKS.
   THIS PROGRAM DOES A SIMPLE INTRINSIC CALL TO DSCOPY.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. HP3000
 OBJECT-COMPUTER. HP3000
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  OPT         PIC S9(4) COMP VALUE 0.
 01  STRING1.
  02 ASCIIPART   PIC X(24) VALUE "NFTESTS TO TEMP1,HDS".
  02 TERMINATOR    PIC S9(4) COMP VALUE 0.
 01  RESULT1.
  02 RESULT2       PIC S9(4) COMP OCCURS 2 TIMES.
 PROCEDURE DIVISION.
 BEGINLABEL.
  CALL "DSCOPY" USING OPT, STRING1, RESULT1.
  STOP RUN.
```

DSCOPY FORTRAN Example

```
$CONTROL MAP,LIST,CODE,CROSSREF,LOCATION,STAT
     PROGRAM DSCOPY
     CHARACTER*40 STRING1
     INTEGER*2 FNUM
     INTEGER*2 OPT
     INTEGER ARRAY IRESULT(4)
C
     DATA STRING1/" NFTTEST TO TEMP1,HDS "/
C
C    THIS PROGRAM DOES A SIMPLE DSCOPY INTRINSIC REQUEST
C
     OPT=0
     FNUM=0
     CALL DSCOPY(OPT, STRING1, IRESULT)
     IF (IRESULT .GT. 0) CALL DSCOPYMSG(IRESULT, FNUM)
     STOP
     END
```

DSCOPY BASIC Example

```
 10     REM THIS WILL DO A SIMPLE DSCOPY REQUEST
 20     DIM A$[30],R[4]
 30     0=R2=Z=0
 40     MAT R=ZER
 50     A$=" NFTTESTS TO TEMP1,HDS "
 60     PRINT A$
 70     CALL BDSCOPY(0,A$,R[*])
 80     IF R[1] <>0 THEN PRINT " ERROR IN DSCOPY. ERROR= ",R[1]
 90     IF R[1] <>0 THEN CALL BDSCOPYMSG(R[*],Z,R2)
100     STOP
110     END
```

DS APPLICATIONS

SECTION VII

DSN/DS is particularly useful in applications that involve transaction processing and that are geographically or functionally dispersed. Any local-system command can be executed remotely through a simple extension to that command. Many operating system intrinsics are also extended in a similar fashion. No knowledge of the communication protocol or physical link being used is required of the terminal user or application programmer. Every application-level capability operates transparently across each connection-level alternative.

DSN/DS on the HP 3000 provides facilities for point-to-point connection between processors. These connections can be made on a variety of types of communication lines, including switched (dial-up), leased, or hardwired, and they can also be mixed throughout the network. Applications can easily obtain access to systems more than one "hop" away, through multiple :REMOTE HELLO log-ons. In addition, HP 3000 computers can connect to X.25 packet-switched and X.21 circuit-switched networks and communicate across those networks with HP 1000 or other HP 3000 computers. In fact, DSN/DS can maintain concurrent connections to multiple remote systems, and/or multiple connections to the same remote system, over a single physical link to the X.25 network.

DSN/DS requires users to pass all of the security checks imposed by MPE (such as passwords) when logging on to a remote system. DSN/DS also provides additional security features applicable only to a network environment. For example, the operator can restrict incoming or outgoing access to the communications link. And incoming calls from an X.25 network are accepted only if the remote host is already in the local system's network data base.

DS Applications

DSN/DS offers the tools to facilitate the sharing of resources within a network. Examples of such resources are programs, data structures, or physical hardware elements of the network. You can access these resources in any of several modes:

- Remote command execution allows you to direct commands to any CPU in the network.

- Remote File Access (RFA) permits the application of processing power to files and devices remote from the CPU. RFA also provides the means for extending Interprocess Communications (IPC) across a DS link.

- Program-to-program (PTOP) communication permits direct communications between master and slave programs, each resident in its own CPU within the network.

- Remote Data Base Access (RDBA) gives the capability for direct and indirect access of data bases on any HP 3000 computer in the network. Combining the distributed processing capability of DSN/DS with the use of data management subsystems such as V/3000, KSAM/3000, and IMAGE/3000 makes possible the sharing of data.

- Network File Transfer (NFT) is a more efficient mechanism than FCOPY for transferring disc files across a communications link.

The chief advantages of Program-to-Program (PTOP) communication are coprocessing capabilities and control of data transmission blocking. Coprocessing master-slave programs execute in multiple systems. Program-to-Program communication allows decisionmaking to be distributed within the master-slave relationship. The exchange of data and control information between the executing programs can be used to alter program flow to adjust to current conditions in the network. Remote command execution and remote file/device access allow one program executing in one CPU to utilize data and/or devices anywhere in the network. All decisionmaking is embodied in this one program. Coprocessing capabilities assume importance in networks where synchronization of modifications to related data structures is important.

Blocking control can be utilized in such a manner as to decrease the number of transmissions to move a specific amount of data. Since transmission time on a high-speed link is a negligible factor in communications performance, the required number of transmissions is the key to performance. Reducing the number of transmissions correspondingly reduces the number of line turnarounds. This may become a significant performance factor in half-duplex networks or satellite communication links where propagation delay affects response time.

TRANSMISSIONS BETWEEN SYSTEMS

Underlying all modes of utilizing DSN/DS is the transmission of data from one system to another. Now compare the building of these transmission units for remote file access and for PTOP.

DSN/DS is supported on three controllers: the Intelligent Network Processor (INP), the Hardwired Serial Interface (HSI), and the Synchronous Single-Line Controller (SSLC). To configure any of these devices into the system, you must specify a buffer length. The buffer length value that you specify represents the maximum number of words to be transmitted between systems in one transmission and it is the system's default buffer size. When you activate DSN/DS with a :DSLINE command, a LINEBUF parameter may be specified to override the configured buffer size. Only the first user to activate the line may use LINEBUF to alter the data communication buffer size. This buffer size may not be respecified until all concurrent DSN/DS users have closed their links. In this way, the pertinent buffer limiting factors for inter-CPU transmissions are set.

How is LINEBUF utilized in accessing remote files and remote peripheral devices? The basic unit for file system operations is the record (or block of records). In remote file access and remote command execution, file system blocking limits the transmission unit to a single record or sequential multiple records. File system's FREAD is satisfied by moving a logical record from a file to the user's buffer. An FREAD on a file open with multi-record access is satisfied by a byte count which can be specified to be blocksize. Thus, an FREAD on a remote file will pack LINEBUF with a record or a specified byte count of sequential records. Contrast this record-orientation to the array-orientation of PTOP communications. PTOP's PREAD is satisfied by the transmission between programs of the contents of LINEBUF. The PTOP programmer must construct the buffer by packing it with array(s), record(s), or fields of records. The records in one transmission need not even come from the same file.

In addition to transmitting specified data, DSN/DS attaches a header of varying lengths. The header always contains eight words transmitted in a fixed format and can contain additional words in an appendage area. For remote command execution and remote file and peripheral device access, the data field is usually preceded by a header of 14 words. Some intrinsics, such as FREAD (multirecord), require a longer header to convey all parameter information. The header for PTOP communications includes the 20-word tag field in the appendage; thus the typical PTOP header is 34 words long. The ideal LINEBUF size will allow the user's data field plus DSN/DS header information to fit into LINEBUF.

To illustrate: Assume that you want to read six 80-byte records from a remote file. Specify a LINEBUF of 300 words.

a. If the remote file is thus defined: REC=-80,1,F, then Remote File Access must retrieve a block of one record from the disc, FREAD one record, and transmit one record. The complete data transfer requires six disc accesses, six FREADs, and six data transmissions.

b. If the remote file is thus defined: REC=-80,6,F, then Remote File Access must access the disc to retrieve a block of six records, satisfy an FREAD with one record, and transmit one record. The complete transfer requires one disc access, six FREADs, and six data transmissions.

c. If the remote file is thus defined: REC=-80,6,F, and opened with the NOBUF and multirecord aoption, then Remote File Access must access the disc to retrieve a block of six records, satisfy an FREAD of 480 bytes with six records, and transmit the six records. The complete data transfer requires one disc access, one FREAD, and one data transmission.

d. In PTOP, the master program can issue a PREAD. The slave program can pack the buffer with all six records, utilizing any of the above three methods. Note that a LINEBUF of 300 words is ample to permit transmission of 480 bytes (240 words) of data plus 34 words of DSN/DS header information in one transmission. A LINEBUF of 256 words requires two transmissions.

COORDINATING MASTER AND SLAVE PROGRAMS

PTOP communication programming requires synchronizing two separate programs at specific points in time. For this reason, it is often helpful to block diagram the transmissions and their contents on a simulated time line.

Where the PTOP programmer wants to loop on certain PTOP operations, the loop's terminating condition must, of course, be defined. The master program has direct control over the interprogram communications and can terminate a loop under conditions defined locally. More difficult are the situations when the slave must communicate to the master that the terminating condition has been met. To do this, the slave might send a REJECT response. A REJECT does not allow transmission of data, and so requires a terminating exchange of transmissions after all data has been transmitted.

Another method is to utilize the 20-word itag field (the ITAG parameter) of the PTOP intrinsics. This field is not accessible by the slave unless designated as a parameter in the corresponding master's PTOP operation. For example:

	Master Program	Slave Program
Example A.	PREAD(dsnum,target,tcount);	GET(itag);
Example B.	PREAD(dsnum,target,tcount,itag);	GET(itag);

In example A, PREAD doesn't utilize the itag field. The slave program can't access itag on this transaction. The second PREAD (Example B) might not even initialize the itag array, but the array has been specified as a parameter. The slave program can now return control information to the master via this field. The master program logic can inspect itag and take corresponding action.

A PCONTROL from the master will also cause an exchange of itag fields and may be used for passing control information. This intrinsic will not pass a data field, however.

The control information passed between programs may terminate a loop, may branch to another part of the program, may transmit an index to be used in a CASE statement, or may serve any other purpose the programmer desires.

It is important to bear in mind the accessibility of transmitted data. When the master program PWRITEs, the slave program cannot process the received data until the ACCEPT intrinsic has moved the data into the slave process stack. The slave program can, however, examine the itag array before doing the ACCEPT or REJECT. After examining the itag, the slave can then alter the itag array. The ACCEPT or REJECT will transmit the slave's itag to the master. Slave local processing can then proceed.

DEBUGGING

Where the amount of local processing in a PTOP application is significant, it may be helpful to debug the master and slave programs as local programs. MOVEs on dummy arrays or FREADs on dummy files can be substituted for communication operations to simplify debugging of the local processing.

When the time arrives to run the programs in master-slave fashion, a :RUN PROG;DEBUG is sufficient to invoke the Debug Utility for the master. This will not, however, allow the programmer to break-point in the slave program or to examine the slave process stack. To facilitate debugging slave programming, the first executable statement of the slave program should be the DEBUG intrinsic.

LINE BUFFERS/CONTINUATION BUFFERS

DSN/DS is designed to send across the line, in a single transfer operation, the amount of data configured as the PREFERRED BUFFER SIZE for the line controller (INP, SSLC, or HSI). The first person to use a DS line can override the configured line buffer size by specifying a different value with the LINBUF parameter of a :DSLINE command.

When a user specifies LINBUF=xxxx, the xxxx value tells the Communication Software (CS/3000) the maximum amount of data DSN/DS will ever send across the line in a single request. For example, if you say LINBUF=1074, you are saying the largest buffer DSN/DS can pass to the Communication Software is 1074 words.

The 1074 words will always consist of both user data and DSN/DS fixed header and variable-length appendage characters. These additional characters (approximately 20 to 50 words) give to and from information, intrinsic names, etc., and vary for RFA and PTOP operations.

For RFA, the DSN/DS header and appendage usually adds about 20 words to the data; for PTOP, the header and appendage also includes the 20-word tag field, for a total of approximately 40 words. The ideal LINBUF size should be able to accommodate the user's data plus these DSN/DS overhead characters.

When a DSN/DS user requests the transfer of more data in a single operation than the line buffer can accomodate, the Communication Software automatically fills the line buffer, makes the transfer, refills, and transfers again -- until all of the user's data has been sent. When a user's single request causes CS/3000 to make several transfer operations, the additional buffers of data are known as "continuation buffers". As stated before, the ideal line buffer should be large enough to eliminate the need for continuation buffers.

COMPRESSION

Compression of data on the communications link may be specified in order to achieve higher throughput.

The COMPRESSION capability can be specified at generation time by use of SUBTYPE=1 while configuring IODS0 or IODSX (refer to Appendix A). This configured subtype sets the default for the line.

A compression parameter may be specified while executing the :DSCONTROL console command. A console operator uses the parameter to override a line default or to reset to the configured state.

A compression parameter may also be specified while executing a :DSLINE command in a session or job. Use of the DSLINE parameter allows individual users to control whether or not their data will be compressed.

The compression technique compresses any occurance of three or more consecutive characters. The compression takes place in the data only, not in the fixed part or the appendage of the request or reply header.

Compression generally increases throughput by reducing redundancy in the data, which results in a reduction in the number of characters being transmitted over the communications link.

In some cases, however, compression could actually result in an increase in the number of characters to be transmitted. For this reason, DSN/DS examines each case when compression is specified. If a situation is found where compression would be detrimental to performance, DSN/DS sends the data uncompressed.

Compression is most helpful in applications using line speeds up to 56 K bps. However, compression is generally not helpful nor desirable in applications that use the HSI at high data rates.

Doing compression and decompression increases the system overhead at both ends of the link. The decision on whether to use compression depends on the communications link data rate, system load, and the amount of redundancy in the data being transmitted. Often, a test of relative throughput with normal system load and "typical" data will provide an indication of the benefits of using compression.

The amount of redundancy in data or files may vary significantly. Source or listing files may compress by as much as 75 percent. But a more typical random assortment of HP 3000 files may reduce by an amount closer to 25 percent. Obviously, the actual reduction will vary from application to application. Comparative tests with and without compression will indicate the benefits.

The DSN/DS initialization procedures allow compression only if both systems are capable of performing compression. Compression is handled on an individual basis, so that on a non-exclusive line, some users may compress while others use NOCOMP.

Formats for Inserted Compression Characters

Octal Value	Meaning
xx nnn nnn	xx = compression type 00 = uncompressed character string 10 = repeated blanks 11 = repeated non-blank character (next byte is the character) nnn nnn = octal character count 1 to 77.

Examples :

036	36 (octal) non-compressed characters
217	17 (octal) blank characters
323.052	23 (octal) compressed * characters

PERFORMANCE

The performance achieved while using the DSN/DS link may vary widely, and it depends on many factors.

Computer System Dependent

The activity mix on the respective HP 3000 will affect performance. It depends upon the character of the simultaneous activity: such as the number of jobs, number of CPU-bound jobs and their relative priority, contention for disc, memory size and amount of swapping, quantum size, etc.

Communication Links

The choice of the communications link will provide an upper limit to the performance. Generally, a full-duplex line will outperform a half-duplex line by reducing line turnaround delays. A half-duplex line with a smaller request-to-send/clear-to-send delay will be faster (such as a 208B at 50 milliseconds versus a 208B at 150 milliseconds).

Line quality can result in wide variations in performance at times when line errors are high. A leased line is generally better and more predictable than a dial-up line. Some telephone offices provide cleaner lines depending on the age and nature of their switching gear.

Applications

For a given amount of data, the buffer size selected will affect performance. The smaller the number of requests required to transmit a given quantity of data, the higher the throughput. This also includes continuation requests. The data may be packed into larger buffers while using PTOP applications. The data may also be blocked into larger records for RFA applications. (RFA and FCOPY handle one record at a time, even though the file may use blocking).

As described earlier, use an appropriate line buffer size. Use a line buffer large enough to contain the full record or buffer, plus the DSN/DS fixed blocks and appendage header words. (The "rule-of-thumb" is 50 words larger than the data size.)

For applications to be run on dial-up lines, the line errors normally suggest a reasonable maximum of 1024 words. Analysis of :SHOWCOM xx ;ERRORS output and trace listings for error rates will allow modification of this recommendation for "typical" conditions. (This suggested maximum value of 1024 could be either increased or decreased when an SSLC is being used; but the value could only be decreased when the communications interface is an INP, since the maximum buffer size for the INP is 1024 words.)

PTOP applications allow both the master and slave programs to do a larger share of data searching, qualification, and manipulation at each local computer, thus reducing the quantity of data which must be sent across the line.

Remote Listing

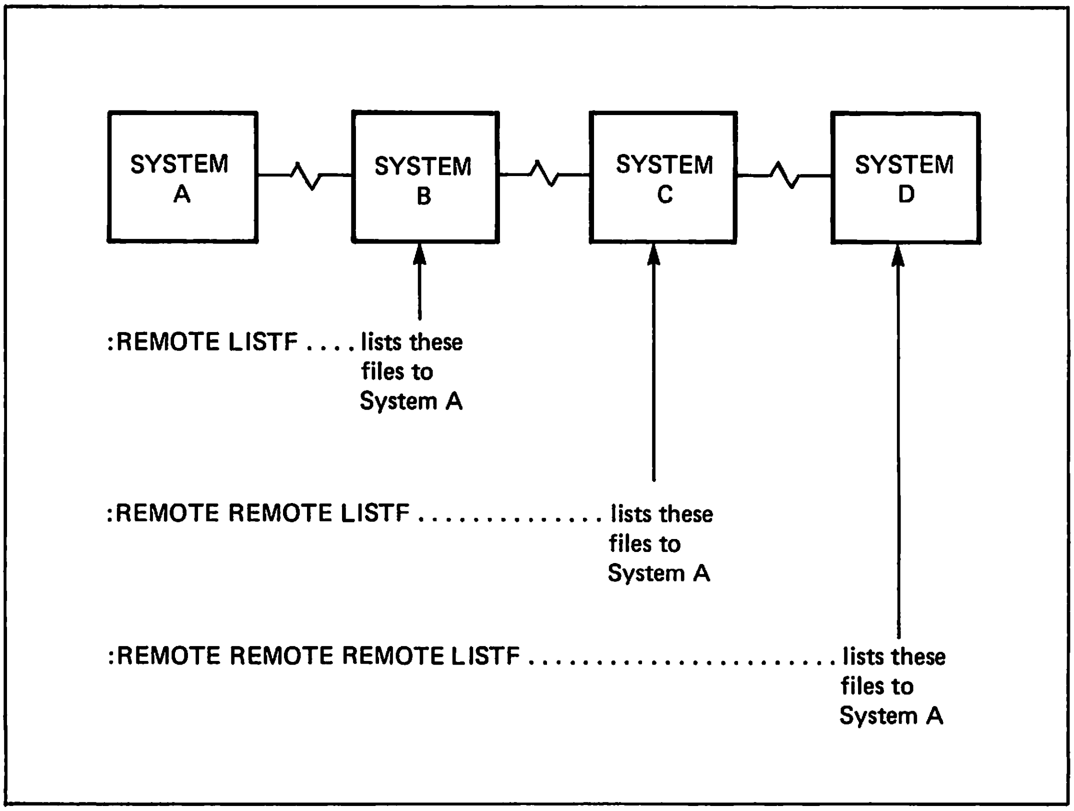
Where data must be sent to a remote device (such as a line printer or a magnetic tape) it may be possible to send the program which generates the data to the remote computer for execution. For example, since a compiler listing can be quite large, it might be more efficient to transmit the source across the line and do a remote compilation and remote list, rather than doing a remote list for a local compilation.

MULTIPLE REMOTE ACCESS

While presenting the basic concepts of DSN/DS in the tutorial sections of this manual, the examples were intentionally limited to simple networks. From those somewhat simplistic illustrations, it might appear as though the only way your local computer can talk to more than one other computer is through additional parallel communication lines from your local system to the additional remote systems. Actually, it is possible to communicate with other remote computers in the network that have no direct connection with your local computer. This communication is made possible by going through one remote computer (to which you do have a direct communication line) to reach another remote computer to which the first remote computer is connected. To reach a second remote system through a first remote system, a multiple REMOTE command is used. The syntax is as follows:

```
    :REMOTE [xxx] [REMOTE [xxx]] ... [command]
```

In this way, the local user can initiate a session sequentially on each remote system. Refer to figure 7-1.



```
:REMOTE LISTF .... lists these
                   files to
                   System A

:REMOTE REMOTE LISTF .............. lists these
                                    files to
                                    System A

:REMOTE REMOTE REMOTE LISTF ....................... lists these
                                                    files to
                                                    System A
```

Figure 7-1. Multiple Remote Accessing Example

Figure 7-1 shows how your local system (System A) can obtain a list of the files in the first remote system (System B) by issuing the command:

```
:REMOTE LISTF
```

To obtain a similar list of files from System C in this kind of network (where the communications link is through an intermediate remote computer), use the command:

```
:REMOTE REMOTE LISTF
```

Likewise, you can route your request through to System D by expanding the command to:

```
:REMOTE REMOTE REMOTE LISTF
```

Using this compound command accomplishes the same result as if you had issued the following series of separate commands:

```
:REMOTE
#REMOTE
#REMOTE
#LISTF
```

There is an important difference in the way of returning to your local system, however. When you reach System D (figure 7-1) by entering the compound command

```
:REMOTE REMOTE REMOTE
#
```

the # prompt is coming from the Command Interpreter (CI) on System D. If you now type a colon (:)

```
#:
:
```

you are being switched back to your local CI (System A). But if you were to use the alternative method of reaching System D with a series of separate commands

```
:REMOTE
#REMOTE
#REMOTE
#
```

and then you typed a colon as before, you would be switched to System C. To get back to your local system (System A), you must return a step at a time (just as you went out to System D a step at a time) as follows:

```
:REMOTE
#REMOTE
#REMOTE
#:
#:
#:
:
```

INTERPROCESS COMMUNICATIONS

Interprocess Communications (IPC) is a capability of the MPE operating system that is very beneficial in the DSN/DS environment. For some applications, IPC may be easier to implement than Program-to-Program Communications (PTOP) and may provide other advantages as well. A basic description of the use of IPC and the changes made to the file system is included in the MPE Intrinsics Reference Manual.

A simple example of the use of IPC for communication between two remote sessions is presented in figure 7-2. User Bill establishes a local session on Node A and a remote session on Node B. His application, called BILLPROG, opens a local MSGFILE as a reader and a remote MSGFILE as a writer. Then, user Jack establishes a local session on Node B and a remote session on Node A. Jack's application, called JACKPROG, opens a local MSGFILE as a reader and a remote MSGFILE as a writer. Now these two unrelated processes can communicate with each other through the IPC capability.

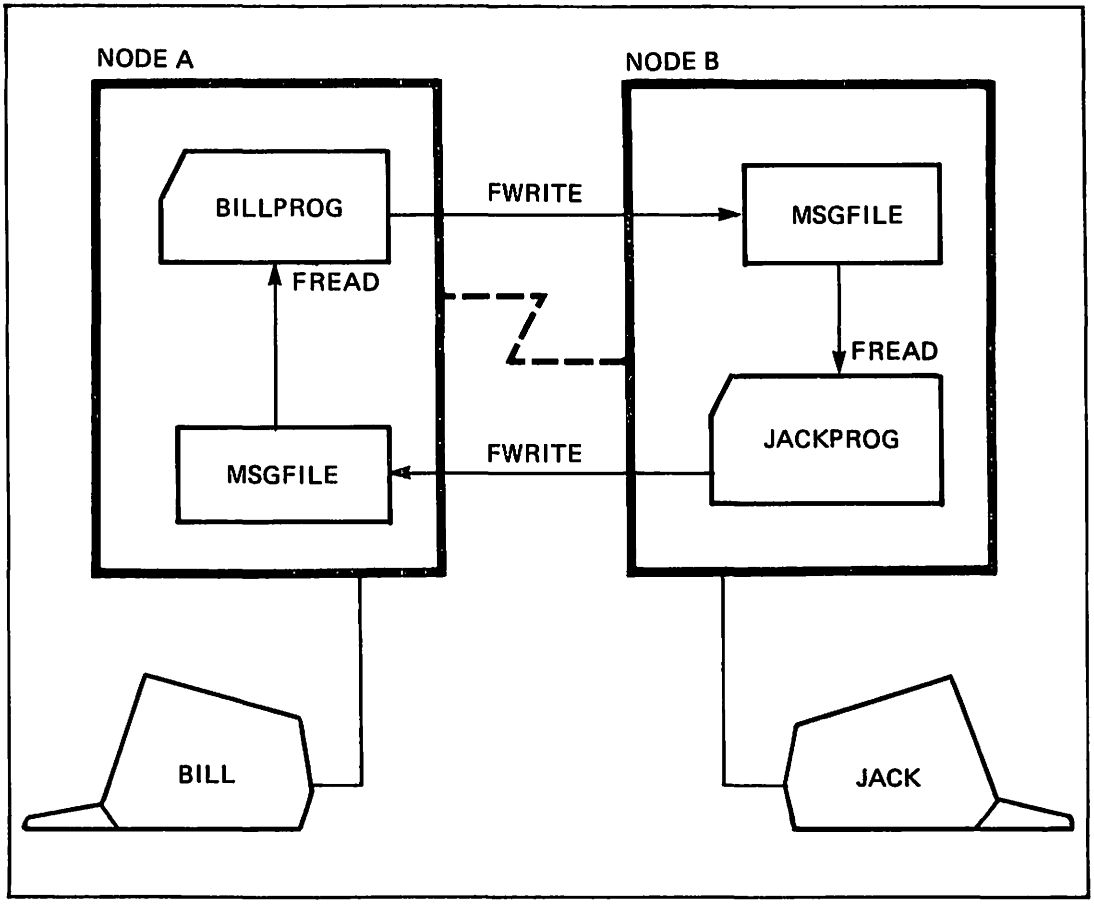


Figure 7-2. Two-node IPC Communication

If PTOP had been used in the example in figure 7-2, a PTOP master program would need to be executing in one node and a slave program would have been initiated by the master in the other node. The master-slave programs would also require coordination because of their relationship.

The advantage of IPC becomes more dramatic when two or more processes desire to communicate with each other, or when the network is more complex than two nodes. Figure 7-3 shows a network consisting of three nodes and a solution that seems very useful in the general DSN/DS applications environment.

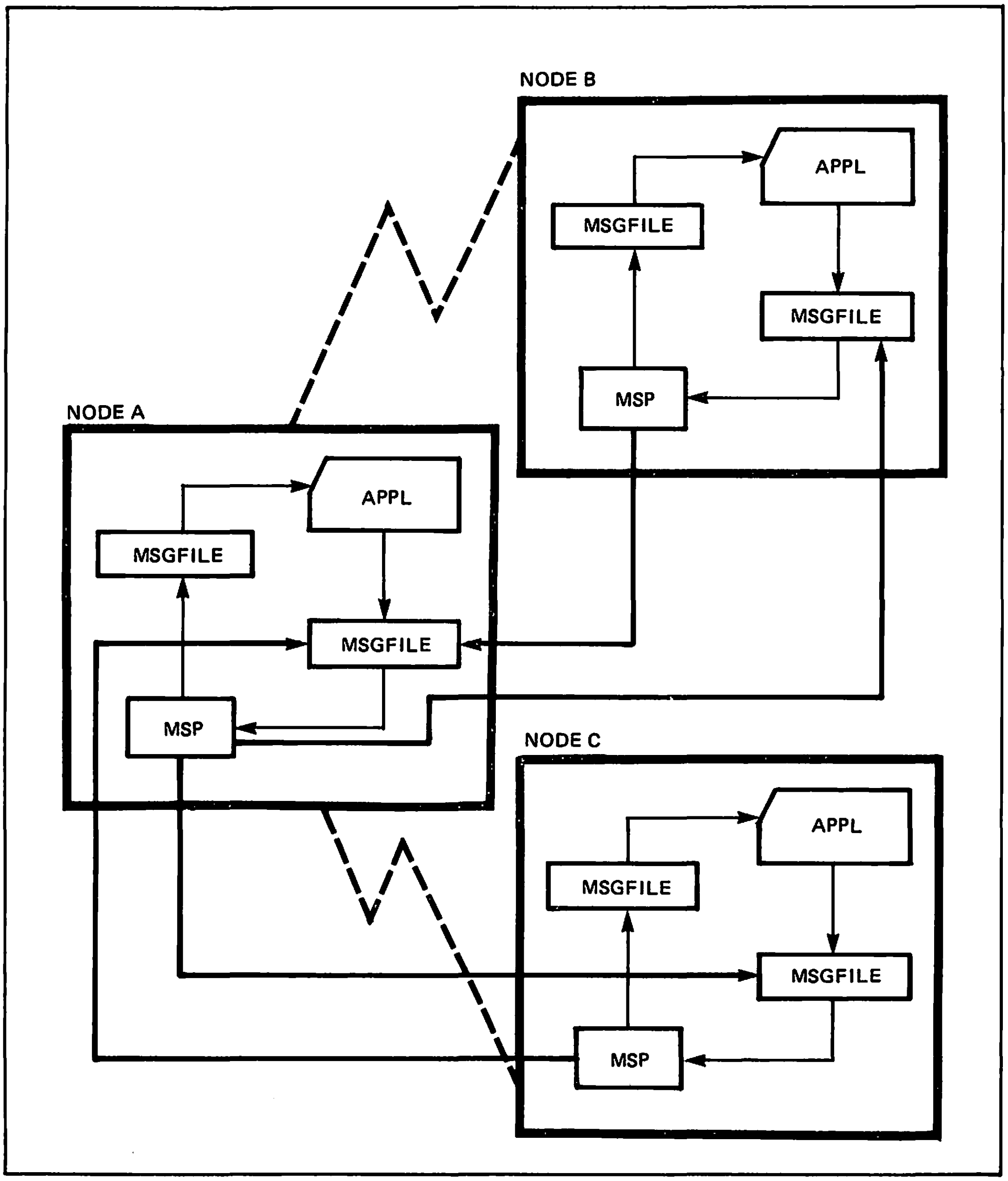


Figure 7-3. Three-node IPC Communication

In figure 7-3, a general application program called Message Switching Procedure (MSP) is written and executed on each node. The MSP performs the following functions:

- Opens a local message file as a reader
- Opens any local applications message files as a writer
- Opens all DS lines to adjacent nodes
- Establishes a remote session on each of these nodes
- Opens a message file on each adjacent node as a writer to be used for communication with each MSP.

The MSP handles all outgoing requests by forwarding them to the MSP programs on adjacent nodes. MSP also handles all incoming requests by routing them to a local application program or by passing them on to the next node in the network.

If the network is complex and it is desirable to shift the responsibility for routing from the user to the MSP, a solution might include addressing within the user's data buffer and the use of a directory file in conjunction with the MSP. The MSP would then use the directory file to determine to what node it should forward the message. A more advanced directory file could provide alternate routes in case of downed lines. If alternate routes were not available, the unserviceable requests could be stored in a disc file and then be rewritten to the MSP's MSGFILE when the downed lines are restored.

In a simple network, it may not be desirable to design an MSP; but it is still possible that using IPC may be more advantageous than using PTOP. In this case, each user application could set up one or more remote sessions on the appropriate node(s) and communicate with other processes using the normal file intrinsics (FOPEN, FREAD, FWRITE, and FCLOSE) and message files. Also, by using the :FILE command, it can be transparent to the user or to the application program that the MSGFILE is located on a remote node.

The advantage of using an MSP is that several users on a system can communicate with a number of remote processes, but only one remote session is required per node. Since fewer remote sessions are necessary, the amount of memory required is decreased.

The major advantage of IPC versus PTOP is that there is no limitation to the number of local or remote processes with which a single process can communicate. The processes are fully bilateral with IPC making it easier to implement and expand the application for more complex networks. Also, activities such as development, testing, and debugging can all be done on one node, and then the resulting application can be distributed.

CONFIGURATION DIALOGUE

APPENDIX A

DSN/DS operation requires the installation and configuration of one communications interface for each line to a remote computer; or, in the case of DS/X.25, one communications interface is required for each physical link to a Public Data Network (PDN).

This appendix explains how to configure the following:

- Intelligent Network Processor (INP)
- Synchronous Single-Line Controller (SSLC)
- Hardwired Serial Interface (HSI)
- DS Line Monitor (communications driver IODS0 or IODSX)
- DS Virtual Terminals (IODSTRM0, IODSTRMX, or IOPAD0) -- one for each session that will be allowed on your system from a remote system or from a Packet Assembler/Disassembler (PAD).

The same communications interface (INP, SSLC, or HSI) can be used by another HP 3000 data communications subsystem (such as DSN/MRJE) when it is not being used by DSN/DS. In such a case, the communications interface is configured once for each subsystem (each time with a unique logical device number, but always with the same DRT number). Keep in mind that the following dialogue applies only when the interface is used for DSN/DS activity, and that a response that is optional for DSN/DS may not be optional for one of the other subsystems. Configuration guidelines pertaining to the other subsystems are given in the reference manual for each subsystem. Configuration summary tables for each of the communications interface types are included in the Communications Handbook.

For any data communications subsystem to function, CS/3000 modules must be present on the system. It is presumed in this configuration description that the Account Systems Engineer (SE) has already installed CS/3000.

If you are making any other changes to the MPE I/O system, refer to the System Manager / System Supervisor Reference Manual.

Configuration Dialogue

Configuration is accomplished through an interactive dialogue between you and the computer system. As the questions or prompts appear on your console, enter the appropriate replies through the console keyboard for your desired system configuration.

NOTE

> In all responses, Y or N can be used for YES and NO. A carriage return is equivalent to NO.

Prior to entering the dialogue, log onto the system and input at least a file reference to a magnetic tape, as follows:

```
:FILE name;DEV=TAPE
:SYSDUMP*name
```

The dialogue commences as follows:

Step No.	Dialogue
1	ANY CHANGES? YES
2	SYSTEM ID = HP 32002 v.uu.ff? return
3	MEMORY SIZE= xxx? return
3.1	I/O CONFIGURATION CHANGES? YES
3.2	LIST I/O DEVICES? YES

All I/O devices currently configured on the system are listed with the following column headings:

LOG DEV   Logical device number.

DRT #   Hardware device address (Device Reference Table number) configured on the interface board.

UNIT #   Hardware unit number of device on its controller.

CHAN   Channel number of device on its controller.

Step No.	Dialogue

TYPE	Device type.
SUBTYPE	Device subtype.
TERM TYPE	Terminal type.
TERM SPEED	Terminal speed.
REC WIDTH	Record width in decimal words.
OUTPUT DEV	Device class name or device ldn.

MODE	J = Accept jobs
	A = Accept data
	I = Interactive device
	D = Duplicative device
	S = Spooled device

DRIVER NAME	Driver name.
DEVICE CLASSES	Class name assigned to the interface.

NOTE

The prompt in Step 3.3, below, appears only if a communications subsystem (CS) device was previously configured into the system.

3.3 LIST CS DEVICES? <u>YES</u>

A list of all CS devices currently assigned to the system is printed with the following column headings:

LDN	Logical device number.
PM	Port Mask. (Not used by INP and SSLC.)
PRT	Protocol.
LCL MOD	Local mode.
TC	Transmission code.

Step No. **Dialogue**

RCV TMOUT	Receive timeout (in seconds).
CON TMOUT	Connect timeout (in seconds).

MODE	O = Dial out.
	I = Manual answer.
	A = Automatic answer.
	D = Dual speed.
	H = Half speed.
	C = Speed changeable.

TRANSMIT SPEED	Transmission speed (characters per second).
TM	Transmission mode.
BUFFER SIZE	Default buffer capacity, in words.
DC	Driver changeable or not changeable.
DRIVER OPTION	Driver options.

If you have a switched device, such as those that are connected through a dial-up telephone line, then you receive the following additional information:

LDN	INP or SSLC logical device number.
CTRL LEN	Not currently implemented.
PHONE NUMBER LIST	A single telephone number -- the default for the data communications line.
LOCAL ID SEQUENCE	The default identification of the local computer.
REMOTE ID SEQUENCE	The default identification of the remote computer.

3.4 HIGHEST DRT=xx?

In the output, xx is a value denoting the present highest DRT entry number that can be assigned to a device.

To change xx, enter the new value desired. If the highest-numbered device in the configuration is a device that uses more than one DRT entry (such as a

Step No. **Dialogue**

terminal controller with one or two data set controllers), be sure to enter the highest of the DRT numbers.

To maintain the current xx, enter a carriage return.

3.5 LOGICAL DEVICE #?

To specify a device to be added or removed, enter the logical device number of that device. An HSI has four ports and thus can be configured up to four times with a unique logical device number for each port. In addition, a communications driver (IODS0) with a unique logical device number must be configured for each HSI port configured.

This prompt is repeated later in the configuration dialogue, so that you can return to this point to configure more than one device.

3.6 DRT #?

To add a device, enter its DRT entry number. For a communications driver and a virtual terminal, you must assign the logical device number of the associated communications interface (INP, SSLC, or the HSI port), preceded by a number sign (#).

Virtual terminals need to be configured for only one port of any HSI (back referenced to only one logical device number for the HSI). The terminals will be dynamically allocated to the proper port when a user opens it.

To remove a device and return to Step 3.3, enter zero.

3.7 UNIT #? 0

Enter zero for an INP, an SSLC, an HSI, the DSN/DS Communications Driver (IODS0 or IODSX), or Virtual Terminals (IODSTRM0, IODSTRMX, or IOPAD0).

3.8 SOFTWARE CHANNEL #? 0

Step No. **Dialogue**

3.9 TYPE?

Enter the device type, where
16 = Virtual Terminal or PAD Terminal
17 = Intelligent Network Processor (INP)
18 = Synchronous Single-Line Controller (SSLC)
19 = Hardwired Serial Interface (HSI)
41 = DSN/DS Communications Driver

NOTE

When configuring Device Type 16, consider the maximum number of terminals supported by your system. Each virtual terminal configured is added to the total number of terminals already on the system.

3.10 SUBTYPE?

Communications Interface:
For an INP, enter 0, 1, or 3
For an SSLC, enter 0 or 1
For an HSI, enter 3, where
0 = switched line with modem
1 = nonswitched line with modem
3 = hardwired line, synchronous transmission

Communications Driver:
For IODS0 or IODSX, enter 0 or 1, where
0 = no data compression
1 = data compression

Virtual Terminal or PAD Terminal:
For IODSTRM0, IODSTRMX, or IOPAD0, always enter 0.

NOTE

If you are configuring a terminal (Type 16), the dialogue continues to Step 3.11. If you are configuring an HSI (Type 19), the dialogue skips to Step 3.13. If you are configuring an SSLC (Type 18), the dialogue skips to Step 3.14. If you are configuring an INP (Type 17), the dialogue skips to Step 3.17. For all other device types, the dialogue skips to Step 3.40.

Step No. **Dialogue**

3.11 TERM TYPE? 0

This question is asked only if Type is 16. Term Type is always zero for DSN/DS Virtual Terminals or PAD Terminals.

3.12 SPEED IN CHARACTERS PER SECOND? 0

This question is asked only if device Type is 16, then the dialogue skips to Step 3.40.

3.13 PORTMASK?

This question is asked only if device Type is 19 (HSI). The values allowable are shown below and must be entered in decimal. This forms a mask indicating which HSI channel will be used. Only one of the four channels may be designated for each unique logical device number.

Enter 8 for HSI cable connector port 0.

Enter 4 for HSI cable connector port 1.

Enter 2 for HSI cable connector port 2.

Enter 1 for HSI cable connector port 3.

Since only one port on the HSI PCA can be opened at a time, only one block of virtual terminals (entered later in this configuration) are needed for that HSI PCA. This same block is automatically reallocated to each new port opened. One block of virtual terminals serves all ports.

3.14 PROTOCOL? 1

This response defines Binary Synchronous Communications.

3.15 LOCAL MODE?

For an HSI, enter 1.

For an SSLC, enter 1 or 2 where
1 = Local is a primary contention station
2 = Local is a secondary contention station

Step No. **Dialogue**

To resolve the contention problem in point-to-point operations, each station is assigned a priority (primary or secondary). Because the secondary station can gain control of the line for a transmission only when the line is left free by the primary station, the SSLC is usually configured as a primary station.

Local mode determines the amount of time a local station will wait in response to a line bid; primary station timeout is two seconds and secondary is three seconds. If a response from the remote system is not received within the allowed time (two seconds for primary or three seconds for secondary), the line bid is re-transmitted until the number of retries permitted by the communications software is exhausted.

3.16 TRANSMISSION CODE?

For an HSI, enter 1.

For an SSLC, enter 1, 2, or 3 where
1 = Automatic code sensing of ASCII and EBCDIC if initially receiving; ASCII if initially sending; or for Hardwired Serial Interface (HSI).
2 = ASCII transmission.
3 = EBCDIC transmission.

Select the most frequently used method of transmission because your response establishes the configuration default. In DSN/DS, all transmissions are ASCII. Only in certain other data communication subsystems can users optionally transfer EBCDIC characters.

3.17 RECEIVE TIMEOUT?

Enter the positive number of seconds the CS device will wait to receive text before terminating the read mode. Entering a carriage return provides a 20-second timeout.

Step No. **Dialogue**

NOTE

For all timeout responses: Entering 0 disables the timeout; maximum timeout is 32000 seconds; DS displays an error when the communications software (CS) disconnects because of a timeout.

3.18 LOCAL TIMEOUT?

Enter the positive number of seconds a connected local station will wait to transmit or receive before disconnecting. Entering a carriage return provides a 60-second timeout. (Your response is not used for DSN/DS activity.)

3.19 CONNECT TIMEOUT?

Enter the positive number of seconds the local station will wait after one attempt to make a connection to a remote station. Entering a carriage return provides a 900-second timeout. For an HSI, 100 to 500 is recommended. For an INP or SSLC, 300 is recommended.

NOTE

Steps 3.20 through 3.22 apply only to CS devices with switched lines connected through a modem (dial telephones, Subtype 0). For CS devices with non-switched lines connected through a modem (private lines, Subtype 1) the dialogue skips to Step 3.23. If the CS device is either an HSI or a hardwired INP (Subtype 3), the dialogue skips to Step 3.25.

3.20 DIAL FACILITY?

Enter YES if manual dial-up is required. Enter INP LDEV# if the AUTO DIAL feature is used. Enter NO if no dial facility is required.

Step No.	Dialogue

3.21 ANSWER FACILITY?

Enter YES if the local modem can answer calls, either manually or automatically. Enter NO if it cannot. A NO response causes the next step to be skipped.

3.22 AUTOMATIC ANSWER?

Enter YES if the local modem can automatically answer calls. Enter NO if manual answering is required.

3.23 DUAL SPEED?

Enter YES if the local modem is dual speed (European models). Enter NO if it is single speed. A NO response causes the next step to be skipped.

3.24 HALF SPEED?

Enter YES if the local modem is to operate at half speed. Enter NO if it is to operate at full speed. The dialogue skips to Step 3.26.

3.25 SPEED CHANGEABLE?

For an HSI, enter YES.

For an INP or an SSLC, enter YES if the speed of the line is changeable. Enter NO if the line speed is fixed. In general, the speed is changeable when the communications interface provides the clocking, and it is not changeable when a single-speed modem or other external device provides the clocking. You must respond YES if the console operator will be using the speed parameter in the :DSCONTROL command to change the configured transmission speed (see Step 3.26).

3.26 TRANSMISSION SPEED?

For INP (Type 17) or SSLC (Type 18) devices, enter the transmission speed of the line in characters per second (Bit Rate/8). For HSI (Type 19) devices, enter 250 000 for cable lengths up to 1000 feet, or

Step No. **Dialogue**

enter 125 000 for cable lengths greater than 1000 feet.

The transmission speed you specify is ignored for modems that provide internal clocking signals. This allows modems of different speeds to be used without reconfiguring the Operating System. The speed specified is used if the modems are eliminated and the controllers are hardwired together.

The speed you specify becomes the default. The console operator can override the default by including the speed parameter in the :DSCONTROL command.

3.27 TRANSMISSION MODE?

Enter the appropriate number for the transmission mode in use. The mode numbers are:

0 = Full duplex
1 = Half duplex

INP and SSLC:
Configure the communications interface (INP or SSLC) to operate in Full Duplex (0) if your facility uses one of the following:

- A leased line with four-wire, point-to-point installation.
- A dial network with two lines (four-wire equivalent).
- A dial network with Wide Band Service.
- Any Direct Connect cable between two INPs.

Configure the communications interface to operate in Half Duplex (1) if your facility uses one of the following:

- A dial network with a single-line (two-wire) installation.
- An INP-to-SSLC Direct Connect cable (a "modem eliminator" cable between an INP and an SSLC).

Your response must agree with the remote system's

Step No. **Dialogue**

configuration and with the characteristics of the communications line.

HSI:
Always configure an HSI as Full Duplex (0).

3.28 PREFERRED BUFFER SIZE?

Enter the desired buffer size in words, up to a maximum of 4095 words for an HSI or SSLC, or up to a maximum of 1024 words for an INP. For a dial-up line, 1024 is generally recommended; for a leased line with an SSLC, the size may be larger than 1024 if the line quality is good. Note that although large buffer sizes increase transmission efficiency, they also use up memory space. Match buffer sizes for sender and receiver whenever possible, since the effective buffer size that can be utilized is the smaller of the two.

3.29 DRIVER CHANGEABLE? NO

3.30 DRIVER OPTIONS? 0

NOTE

The dialogue skips to Step 3.50.

3.40 RECORD WIDTH?

For IODSO, enter 128.

For all Virtual Terminals, enter 40.

3.41 OUTPUT DEVICE?

For the communications driver, enter zero.

For a virtual terminal or a Packet Assembler/Disassembler (PAD) terminal, enter the class name or logical device number to be used for the corresponding job/session listing device.

Step No. **Dialogue**

3.42 ACCEPT JOBS/SESSIONS?

For the communications driver, enter NO.

For virtual terminals or PAD terminals, enter YES.

3.43 ACCEPT DATA? NO

3.44 INTERACTIVE?

For the communications drivers, enter NO.

For virtual terminals or PAD terminals, enter YES.

3.45 DUPLICATIVE?

For the communications drivers, enter NO.

For virtual terminals or PAD terminals, enter YES.

3.46 INITIALLY SPOOLED? NO

3.50 DRIVER NAME?

Enter the name of the driver for this device as follows:

IOINP0 = INP

CSSBSC0 = SSLC

CSHBSC0 = HSI

IODS0 = Communications driver

IODSX = Communications driver, while utilizing the X.25 capability

IODSTRM0 = Virtual terminals

IODSTRMX = Virtual terminals, while utilizing the X.25 capability

IOPAD0 = Packet Assembler/Disassembler (PAD) terminals, while utilizing the X.29/X.25 capability

Step No. **Dialogue**

NOTE

Steps 3.52 through 3.55 apply to CS devices with switched (dial-up) lines (Types 17 and 18, Subtype 0). The dialogue for all other devices skips to Step 3.70.

3.52 PHONELIST?

Enter YES or NO.

You can supply one number (usually a frequently dialed number) which will be the system default.

3.53 PHONE NUMBER?

Enter a string of numbers and hyphens, but not more than 30 characters. This number will be included in the I/O request on the system console when a user OPENs a dial-up (manual call) line. This number will be dialed automatically on an autodial line unless the user overrides with a phone number in the :DSLINE command.

3.54 LOCAL ID SEQUENCE?

The default local ID sequence can be specified in terms of code or number system. Enter a carriage return for a null local ID sequence. Enter one of the letters below, followed by the ID sequence in quotes, if code, or parentheses, if number system.

A = ASCII	Example: A "JOE"
E = EBCDIC	Example: E "STRING"
O = Octal	Example: O (7, 35, 5)
H = Hexadecimal	Example: H (A1, 1F, BB)

NOTE

Do not enter more than 16 characters for the local or remote ID sequence.

Step No. **Dialogue**

3.55 REMOTE ID SEQUENCE?

Enter the default remote ID sequence in the same format as the local ID sequence (above). This can be repeated until a carriage return is entered.

3.70 DEVICE CLASSES?

Enter a list containing a device class name (up to eight alphanumeric characters, beginning with a letter). Class names are separated from each other by commas. These names are left to the discretion of the System Supervisor. They will be used in certain commands and intrinsics when any member of a group of devices (such as any disc drive) can be referenced. No name need be entered.

NOTE

For IODSX entries, the destination logical node name cannot be specified as a device class name. (Refer to Appendix H.)

The dialogue now prints the LOGICAL DEVICE #? prompt described in Step 3.5. If all I/O configuration is complete, press RETURN and the dialogue continues at Step 3.80. Otherwise, enter a logical device number and repeat the configuration procedure from Step 3.5.

3.80 MAX # OF OPENED SPOOLFILES= xxx? return

3.81 LIST OF I/O DEVICES? YES

To print a listing of the new input/output device configuration, enter YES. This list appears in the format described in Step 3.2.

3.82 LIST CS DEVICES? YES

Enter YES to list the characteristics of the new CS device configuration.

Step No. **Dialogue**

3.83 CLASS CHANGES? return

3.93 LIST I/O DEVICES? return

NOTE

The prompt in Step 3.94 appears only if a CS device is configured or if additional drivers exist (for the CS driver-changeable option in Step 3.29). If neither case exists, the dialogue skips to Step 4.

3.94 ADDITIONAL DRIVER CHANGES? NO

4 SYSTEM TABLE CHANGES? NO

5 MISC CONFIGURATION CHANGES? NO

6 LOGGING CHANGES? NO

7 DISC ALLOCATION CHANGES? NO

8 SCHEDULING CHANGES? NO

9 SEGMENT LIMIT CHANGES? NO

10 SYSTEM PROGRAM CHANGES? NO

11 SYSTEM SL CHANGES? NO

The NO response assumes CS/3000 modules are already present on the system.

11.1 LIST LIBRARY? NO

11.2 DELETE SEGMENT? NO

Step No. **Dialogue**

11.3 REPLACE SEGMENT? NO

11.4 ADD SEGMENT? YES

12 ENTER DUMP DATE?

return — Copies the modified MPE. When this copy is used to COLDSTART the system, the account structure and all files remain intact.

mm/dd/yy — where mm/dd/yy is some date in the future. Copies the modified MPE and the current accounting, but no files.

mm/dd/yy — where mm/dd/yy is usually the date of the most recent system backup. Copies the modified MPE, the current accounting structure, and any files that were changed on or since the specified date.

0 — Copies the entire system (MPE, the current accounting structure, and all files).

12.01 ENTER DUMP FILE SUBSETS?

Enter a carriage return, or enter a filename or series of filenames. (Example: @.PUB.SYS)

12.1 LIST FILES DUMPED? YES or NO

13 The console operator must now use the =REPLY command to assign the magnetic tape drive on which you have arranged for a fresh tape reel to be mounted.

After the SYSDUMP is complete, the tape produced should be used to COLDSTART the system. During COLDSTART, the old I/O device configuration is replaced with the new one from your SYSDUMP tape.

ERROR CODES AND MESSAGES

APPENDIX B

The following is a summary of the error code numbers and messages that may be encountered. The messages, as listed here, have been grouped into several categories. For example, the first group contains all messages pertaining to :DSLINE syntax problems, while the second group contains the messages that report a DSN/DS functional problem. Each group is identified with an explanatory heading, and the messages are listed in numerical sequence within each category for easy reference.

:DSLINE SYNTAX ERRORS

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 DSLINE CANNOT CONTAIN BOTH OPEN AND CLOSE.  (CIERR 1301)
1302 DSLINE REQUIRES AT LEAST ONE PARAMETER.  (CIERR 1302)
1303 DSNUMBER SPECIFICATION MUST BE A NUMBER FROM 1 THRU 255.
     (CIERR 1303)
1304 DSLINE #1! 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 DSLINE.  (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 DSLINE OPEN REQUIRES A VALID DS DEVICE NAME AS THE FIRST PARAMETER. (CIERR 1323)
1324 FROM ADDRESS MUST BE BETWEEN 1 AND 14 CHARACTERS INCLUSIVE. (CIERR 1324)
1325 TO ADDRESS MUST BE BETWEEN 1 AND 14 CHARACTERS INCLUSIVE. (CIERR 1325)
1326 FROM AND TO ADDRESS MUST BE A DECIMAL NUMBER. (CIERR 1326)
1392 ONLY ! WORDS WERE ALLOCATED FOR THE LINE BUFFER. (CIWARN 1392)
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 :DSLINE COMMAND OR A REQUIRED :REMOTE HELLO HAS NOT BEEN DONE. (DSERR 214)
215 DSLINE 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 SLAVE PROGRAM HAS TERMINATED BEFORE EXECUTING "GET". (DSERR 226)
227 REMOTE HELLO MUST BE DONE TO INITIATE REMOTE SESSION. (DSERR 227)
228 EXCEEDED MAXIMUM NUMBER OF VIRTUAL CHANNELS PER JOB. (DSERR 228)
231 INVALID FACILITY IN CONNECTION REQUEST. (DSERR 231)
232 THE REMOTE COMPUTER IS NOT OBTAINABLE. (DSERR 232)
233 VIRTUAL CIRCUIT IS NOT AVAILABLE. (DSERR 233)
235 DS MESSAGE SEQUENCING ERROR. (DSERR 235)
236 COMMUNICATIONS HARDWARE HAS DETECTED AN ERROR. (DSERR 236)
237 CANNOT CURRENTLY GAIN ACCESS TO THE TRACE FILE. (DSERR 237)
238 COMMUNICATIONS INTERFACE ERROR. INTERNAL FAILURE. (DSERR 238)
239 COMMUNICATIONS INTERFACE ERROR. TRACE MALFUNCTION. (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 REMOTE OR PDN IS NOT RESPONDING. (DSERR 243)
244 COMMUNICATIONS INTERFACE ERROR. LINE RESET OCCURRED. (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)

:DSCONTROL INFORMATORY MESSAGES

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.
313 DS DEVICE !: RETRY COUNT NOW EQUALS !.
314 DS DEVICE !: CALL REQUEST CANCELED.

: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)

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)

4146 RETRY OVERRIDES PREVIOUS RETRY FUNCTION. (CIWARN 4146)

4147 EXPECTED AN "=" AS DELIMITER FOR RETRY FUNCTION. (CIERR 4147)

4148 INVALID RETRY COUNT, MUST SPECIFY "DEFAULT" OR A NUMBER BETWEEN 0 AND 255 INCLUSIVE. (CIERR 4148)

4149 DS DEVICE !: RETRY NOT EXECUTED DUE TO ABOVE. (CIWARN 4149)

4150 DS INTERNAL FIX NUMBERS DIFFER. (CIWARN 4150)

4151 INCOMPATIBLE OR MISSING NONCRITICAL DS MODULE: DSCOPY, DSTEST, DS2026, OR DS2026CN. (CIWARN 4151)

4152 CRITICAL DS MODULES ARE INCOMPATIBLE, NO CONTROL FUNCTIONS EXECUTED. (CIERR 4152)

4153 MISSING CRITICAL DS SOFTWARE, NO CONTROL FUNCTIONS EXECUTED. (CIERR 4153)

4154 REMOTE PROMPT MUST BE 1 TO 8 PRINTABLE CHARACTERS. (CIERR 4154)

4155 PROMPT OVERRIDES PREVIOUS PROMPT FUNCTION(S). (CIERR 4155)

:DSCOPY GENERAL ERROR MESSAGES

0 SUCCEEDED.

1 SUCCESSFULLY INITIATED.

4 UNABLE TO OPEN TRANSACTION FILE. (NFTERR 4)

5 UNABLE TO OPEN LIST FILE (DSCOPYI). (NFTERR 5)

6 IC ERROR ON TRANSACTION FILE. (NFTERR 6)

7 TRANSACTION RECORD > 200 CHARS LONG. (NFTERR 7)

9 TEMPORARY TRANSACTION FILE FULL. (NFTERR 9)

10 PARAMETERS IMPLY CONFLICTING MODES. (NFTERR 10)

11 CAN'T "RUN" COPY PROCESS IN THIS MODE. (NFTERR 11)

13 UNRECOGNIZED PARAMETER. (NFTERR 13)

14 CONFLICTING OPTIONS HAVE BEEN SPECIFIED. (NFTERR 14)

16 UNIMPLEMENTED FEATURE. (NFTERR 16)

17 CANNOT CONTACT REMOTE NODE. (NFTERR 17)

18 FILE SYSTEM ERROR ON SOURCE FILE. (NFTERR 18)

19 FILE SYSTEM ERROR ON TARGET FILE. (NFTERR 19)

21 ILLEGAL DSLINE NAME. (NFTERR 21)

24 UNSUPPORTED STANDARD DEVICE TYPE. (NFTERR 24)

25 CAN'T FIND OR OPEN THE SOURCE FILE. (NFTERR 25)

26 CAN'T CREATE OR OPEN THE TARGET FILE. (NFTERR 26)

27 CANNOT CONTACT REMOTE SYSTEM. (NFTERR 27)

28 SOURCE AND TARGET FILES CANNOT BE ACCESSED THROUGH REMOTE FILE ACCESS. (NFTERR 28)

29 COMMUNICATION IO ERROR. (NFTERR 29)

30 INSUFFICIENT CAPABILITIES. (NFTERR 30)

33 NO SOURCE FILE WAS SPECIFIED. (NFTERR 33)

36 DS/3000 HAS NOT BEEN INSTALLED ON THIS SYSTEM. (NFTERR 36)

37 REMOTE SYSTEM UNABLE TO USE TRANSPARENT MODE. (NFTERR 37)

38 CAN'T FIND THE EXTRA DATA SEGMENT, USE THE DSCOPY INTRINSIC TO INVOKE NFT. (NFTERR 38)

39 INVALID EXTRA DATA SEGMENT CONTENTS, USE THE DSCOPY INTRINSIC TO INVOKE NFT. (NFTERR 39)

40 NEGOTIATIONS FAILED, NO COPY CAN BE PERFORMED. (NFTERR 40)

41 FILE TRANSFER ABORTED. (NFTERR 41)

42 COPY CANCELLED BY USER. (NFTERR 42)

:DSCOPY INTRINSIC ERROR RETURNS

80 BOUNDS VIOLATION. (NFTERR 80)
81 SPLITSTACK MODE CALLS NOT ALLOWED. (NFTERR 81)
82 FIRST PARAMETER VALUE IS OUT OF RANGE (-1:6). (NFTERR 82)
83 SECOND PARAMETER TOO SHORT TO CONTAIN VERSION STRING. (NFTERR 83)
84 NFT PROCESS IS BUSY, CAN'T START NEW TRANSACTION. (NFTERR 84)
85 NFT PROCESS IS NOT RUNNING. (NFTERR 85)
86 ILLEGAL BASIC CALLING SEQUENCE. (NFTERR 86)

:DSCOPY INTERNAL ERRORS

101 INTERNAL ERROR ON REMOTE SYSTEM. (NFTERR 101)
102 REMOTE SYSTEM NFT VERSION IS INCOMPATIBLE. (NFTERR 102)
103 INTERNAL - STRING STORAGE OVERFLOW. (NFTERR 103)
104 UNABLE TO CREATE TEMPORARY TRANSACTION FILE. (NFTERR 104)
105 AN UNEXPECTED MESSAGE WAS RECEIVED. (NFTERR 105)
106 AN ILLEGAL VALUE WAS RECEIVED IN A MESSAGE. (NFTERR 106)
107 A MESSAGE RECEIVED IN INVALID FORMAT. (NFTERR 107)
108 A REQUIRED ELEMENT WAS MISSING FROM A RECEIVED MESSAGE. (NFTERR 108)
109 NFT PROCESS CREATE FAILED. (NFTERR 109)
110 ATTEMPT TO GET EXTRA DATA SEGMENT FAILED. (NFTERR 110)

DSCONTROL CONSOLE COMMAND

APPENDIX C

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

:DSCONTROL

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
                      RETRY=[retry option]

open option     --    MASTER
                      SLAVE
                      [SPEED=] linespeed

trace options   --    [,[ALL][,[mask][,[numentries][,[WRAP]
                               [,filename]]]]]

mon option      --    DS
                      CS

retry option    --    DEFAULT
                      count
```

PARAMETERS

dsdevice — The logical device number or the device class name of the DSN/DS communications device. On your system's I/O configuration listing, the device is back referenced by a pound sign (#) to a previously defined INP, HSI, or SSLC. (Required parameter.)

OPEN — Establishes a communication link with another HP 3000. Makes the line available for remote communication via the DSN/DS Subsystem. (Required parameter.)

SHUT

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

MASTER

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

SLAVE

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

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 selected SPEED CHANGEABLE. Specify linespeed if yours is a European installation with modems running at half speed, or if the line is hardwired and you want to override the configured default. It may be necessary to include this parameter if the length of cables used for HSI communications has 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

Remember, both ends of the line must operate at the same speed.

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.

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 PSTN).

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 DSN/DS 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]
    [,CS]

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.

MON — Requests monitoring of all levels of activity.

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

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

Default: No monitoring.

Used only for system troubleshooting.

MOFF

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

RETRY= [DEFAULT]
       [count   ]

Changes the communications error retry count to the specified value. The retry counter controls the number of times the system attempts to send or receive a message across a DS line.

DEFAULT — Specifies a limit of 15 retries when a line error occurs.

count — Can be any value within the range of 0 to 255.

Default: 15.

:DSCONTROL

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 and are ASSOCIATEd with the specified DS device.

Only one DSN/DS 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 DSN/DS users. It cannot, however, be shared by users of other communications subsystems supported by your system (for example, DSN/MRJE). Thus, you must SHUT the DSN/DS communications device before the controller 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 communications link is in use by any subsystem; the ldn for a DSN/DS communication pseudo device (driver IODS0 or IODSX) will be AVAILable if it is currently OPEN for use by DSN/DS 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
```

The name of the trace file is:

```
DSTRCxxx.PUB.SYS
```

where xxx is the logical device number of the dsdevice.

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 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. When using the X.25 capability of DSN/DS, the SHUT parameter disconnects the line immediately, even if there are current users on the line.

NOTE

Occasionally you may not be able to SHUT a standard (non-X.25) DS line. This could happen, for example, if a DS user forgot to issue a :DSLINE xxx;CLOSE command but still has a local session. It could also happen if a remote session is "hung". In such a situation, you can "KILL" all activity across the line by issuing an :ABORTIO xxx (where xxx is the logical device number of the dsdevice). Following the use of the :ABORTIO xxx command, a second :DSCONTROL xxx;SHUT command will complete successfully.

EXAMPLES

To open DS line number 55, thereby making it available for use by the DSN/DS 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,ALL
```

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
```

DSDUMP

APPENDIX D

The DSDUMP program is a CS Trace Facility analyzer that formats and dumps DSN/DS messages from a CS Trace file. DSDUMP is a support tool to be used in conjunction with CSDUMP (not as a replacement for CSDUMP). The program can be run either interactively or in batch mode. DSDUMP has several commands that allow you to specify certain messages to be formatted. In addition to the formatted DS messages, the OPR id and CMP id, as well as any error code for the formatted DS message, is printed. However, the OPR id is not printed when the ERRORS option is turned on, but the OPR id for any DS message should be obvious. When the DSDUMP program is being run in a job, a file equation is needed for the CSTRACE file. Otherwise, the file equation is optional, since you will be prompted for the CSTRACE file name if the FOPEN to the CSTRACE file fails.

The file equation for the CS Trace file is:

```
:FILE CSTRACE=CS trace file name
```

DSDUMP COMMANDS

The commands used with the DSDUMP program are as follows:

CLEAR	Resets all options to their default values.
DATA=	Places a limit on the number of words in the data section to be printed per DS message.
DISPLAY	Shows the status of all commands and parameters.
ERRORS	To format only those DS messages whose completion entry has an error code not equal to zero.
EXIT	Terminates the program. This command may be used any time the user is prompted.
GO	To get out of the Command Interpreter and start the dump.

HELP — For an explanation of the commands. HELP does not accept any parameters. (Only allowed in a session.)

ONES — To include idle (-1) DS messages in the dump. The default is to exclude these messages.

NEWDEV — To specify a new output device. (Only allowed in a session.)

NEWFILE — To change CSTRACE files. (Only allowed in a session.)

PINS= — To format only those DS messages whose To or From PIN is equal to one of the specified PINs. (A maximum of ten PINs is permitted.)

RANGE — To find the trace times of the first and last entries.

TIMES= — To format only those DS messages whose trace times are within the specified range.

TYPES= — To format only the specified type of DS messages. The parameters for the TYPES= command are:

- COMMANDS — Formats the REMOTE command, Remote HELLO, Remote BYE, Control-Y message, BREAK message, RESUME message, ABORT message, KILLJOB message, and First Slave DSOPEN.
- PTOP — Formats PREAD, PWRITE, PCONTROL, POPEN, PCLOSE, ACCEPT, and REJECT.
- RFA — Formats RFA and IMAGE messages.
- RTE — Formats RTE DS messages.
- QTOQ — Formats QTOQ (NFT) DS messages.
- TERMINAL — Formats PRINT messages, READ and READX messages, and FCONTROLs to the terminal.

Several commands can be combined on one line if they are delimited (separated) with a semicolon (;). All parameters for a command must be on the same line. When the PINS= command or the TYPES= command are entered several times, they do not cancel the previous command, but instead they are added to the previous parameters. The PINS= command checks for duplicate PINs. A new TIMES= or DATA= command, however, does replace the previous command. If a colon (:) is typed in the first column of a line, that line is assumed to contain an MPE command. An End-of-File will initiate the dump. The default designator is $STDINX. The command file may be equated to $NULL if no options are desired. If you are having all messages formatted (including idle messages), then sequence numbers will be included in the output.

The file equation for the Command Interpreter is:

```
:FILE COMFILE=command file name
```

OPERATION

When the DSDUMP program begins, a procedure is called to read the driver name from the first COPEN intrinsic entry. This begins in word 5. The WHO intrinsic is used to determine whether it is in batch mode or interactive mode. If it is interactive, you are queried as to where you want your output sent. After the FOPEN for the output is done, the Command Interpreter is called. Then, after the commands are read, the program processes the data. There is a procedure for HSI drivers and another for INP and SSLC drivers.

The procedure for the HSI driver entries does a search for STX and EDT entries. EDT entries contain receive-text DS messages if word 5 is a %12. EDT's word 6 contains the length of the DS message before truncation. If this length is greater than 26 bytes (the maximum number of bytes for which the data segment has space), the length is set to 13 words. The length of STX DS messages is determined by bits 11:5 of word 0 of the entry. This is the length of the entire entry. The length of the header (five) is subtracted from this value to get the length of the DS message. The DS message is simply extracted "as is" and the FORMAT procedure is called to output it.

The procedure for traces from INP and SSLC drivers looks for STX and RTX entries. When they are found, their data section is searched for protocol start-texts (DLE STX). The data from this point until a corresponding entry's protocol end-text (DLE ETX) is found is then put into a buffer, omitting any control characters. When the ETX is found, the FORMAT procedure is called to format the DS message(s).

Both procedures check for overlaid or truncated records, as well as searching for OPR id and CMP id. It is these procedures which also handle the TIMES= and ERRORS options.

Preparation

The DSDUMP program file is prepared with STACK=3000 and MAXDATA=25000.

Running the Program Interactively

When the DSDUMP program is being run interactively, it is not necessary to specify any file equations. DSDUMP commands can be read from a file, but a file equation for the output file is not permitted. The program will ask you whether you want the output to go to the terminal or to the printer. If you don't specify a device, the program defaults to LP. If no priority is specified, the default is 6. After the FOPEN has been performed, you will be prompted for commands. If the output is going to the printer, all DSDUMP commands are echoed. If the output is going to the terminal, then after all messages have been listed, the CSTRACE file is rewound and control goes to the Command Interpreter. Also, when output is going to the terminal, typing a Control-Y will cause the CSTRACE file to be rewound and contol to go to the Command Interpreter. When the program returns to the Command Interpreter (CI), the input is then read from $STDINX rather than COMFILE.

Streaming the Program

A file equation for the CS trace file is required for batch jobs. However, file equations for the command input file and the output files are optional, since the default designators are $STDINX and $STDLIST. The formal designator for the list file is DSLIST. The HELP, NEWDEV, and NEWFILE commands are ignored in batch mode. Any error in the command file will terminate the program. If command input is supplied on $STDINX, it must be terminated by :EOD if the GO command is not used.

Output

Output on the printer includes the name of the CS trace file and the date of the trace file in the heading. Also included is the CS LDEV and the name of the driver. Messages from sent-text entries appear on the left side and received messages are on the right side.

Operating Tips

When using the TIMES= command, be sure to specify the upper bound a little higher than you actually want it. This is to ensure that you get all the desired entries. This technique is necessary because the clock on the INP is accurate to hundredths of seconds while the mainframe clock is accurate to milliseconds, and DSDUMP tests the millisecond trace times against an upper bound that is accurate to hundredths of seconds. It is also possible that you may not get all the messages you wanted if you use the TIMES= command when the trace has already been running for a while. This is a result of the INP and mainframe clocks getting out of synchronization, which can cause later trace times to be less than previous trace times. There is still a possibility that DS messages can be truncated by DSDUMP (as it was with CSDUMP) if the numentries parameter in the DSCONTROL statement is small. Therefore, take caution to prevent misinterpretation of later messages.

APPENDIX E

SYSTEM VERIFICATION

Both the system software and the physical link connecting the computers can be tested with a diagnostic program called DSTEST. DSTEST conducts a simple, yet effective, test of the system, including Remote File Access (RFA) and Program-to-Program (PTOP) communications.

DSTEST can run in Diagnostic or Normal mode. In the Diagnostic mode of DSTEST, you can select the number of passes, the word pattern to be transmitted, the mode of transmission, and the block size. In the Normal mode, DSTEST automatically assigns typical values for each option. An additional entry point provides a report of the version and fix level of DS and CS software installed on a system.

NOTE

> To perform DSTEST, you must have a remote session. Also, remote command execution can be used independently of DSTEST for checkout of the various system configurations.

SOFTWARE VERSION REPORT

The DSTEST program has an alternate entry point which provides a list of the software module version numbers for the DSN/DS modules installed on your system. This list must be available for all troubleshooting activities, and it must accompany each Service Request (SR) that you submit to your HP Systems Engineer (SE).

In order to obtain the list, you must have READ access to the DSN/DS program files in PUB.SYS. The command syntax is:

```
     :RUN DSTEST.PUB.SYS,VERS
```

Version Report Example

```
:RUN DSTEST.PUB.SYS,VERS
HEWLETT PACKARD 32190A.03.04 DSTEST/3000 THU, OCT 7,1982, 1:18 PM
   MODULE      VERSION
SL MODULE 00  A.04.01, INTERNAL FIX 002
SL MODULE 01  A.04.01, INTERNAL FIX 002
SL MODULE 02  A.04.01, INTERNAL FIX 000
   DSCOPY     A.04.01, INTERNAL FIX 001
   DSMON      A.04.01, INTERNAL FIX 002
   DSTEST     A.04.01, INTERNAL FIX 001
   DS2026     A.04.01, INTERNAL FIX 000
   DS2026CN   A.04.01, INTERNAL FIX 000
   DSDUMP     A.04.01, INTERNAL FIX 000
   IODS0      A.04.01, INTERNAL FIX 000
   IODSTRM0   A.04.01, INTERNAL FIX 001
   DSMONX     A.04.01, INTERNAL FIX 007
   IODSX      A.04.01, INTERNAL FIX 005
   IODSTRMX   A.04.01, INTERNAL FIX 002
   IOPAD0     A.04.01, INTERNAL FIX 005
   NETCONF    A.05.03, INTERNAL FIX 000
     CS       A.05.03, INTERNAL FIX 000

END OF PROGRAM
:
```

DIAGNOSTIC MODE

To run the diagnostic mode, perform the following steps:

1. Enter the following line to initiate the line test:

   ```
   :RUN DSTEST,DIAG
   ```

   If you are testing RFA, a :FILE command is required before initiating the DSTEST to direct the accessed file to the proper DS line. The file equation is:

   ```
   :FILE REMOTE;DEV=dsdevice#DISC
   ```

2. Answer the following questions:

.RFA or PTOP?

Enter RFA for Remote File Access or enter PTOP for Program-to-Program testing.

.REMOTE COMPUTER?

Enter "3000" or "1000".

.DSLINE?

Enter the device class, logical device number, or node name that was assigned to IODS0 or IODSX during system configuration.

.NUMBER OF PASSES?

Enter the number of actual transmissions desired, up to a maximum of 32767 (decimal). Zero (0) or a carriage return causes the test pattern to be transmitted once.

.PATTERN?

Enter an octal word to be transferred (the % sign must be entered).

NOTE

Illegal input causes the message

INPUT ERROR

to be printed. Enter a correct value or enter a carriage return to specify the default value %177777.

.BLOCKSIZE?

Enter the desired blocksize of the transfer (<4096). If a value equal to or greater than 4096 is entered, an error message will be printed.

.CONTINUE(Y/N)?

Enter an affirmative response (Y) to return to the beginning of the option selection phase if you wish to repeat the cycle, or enter a negative response (N) to terminate the test.

NORMAL MODE

To run the normal mode, perform the following steps:

1. Enter the following line to initiate the DSTEST:

```
:RUN DSTEST
```

In the normal mode, you are not required to select options; the default values are automatically used.

2. Answer the following question:

.DSLINE?

Enter the device class, logical device number (decimal), or node name that was assigned to IODS0 or IODSX during system configuration.

NOTE

The normal mode default is a 512-word program-to-program transfer with all words containing %177777.

DSN/DS COBOL INTERFACE

CONVENTIONS

To call an external procedure from ANS COBOL (COBOL/I), the parameters must be passed by word reference. This requirement effectively prevents the COBOL/I user from calling system-level intrinsics in general, and specifically, the DSN/DS Program-to-Program intrinsics. The following interface routines are provided to allow the ANS COBOL user access to the program-to-program communications capability. The user of COBOL II/3000 need not use these interface intrinsics, since the call-by-value capability can access the Program-to-Program intrinsics (as outlined in the COBOL/II Reference Manual).

The parameters in the COBOL calling sequences must be of the following types:

> If the parameter is an integer, it must be a COBOL picture 9 through 9(4) or S9(3) computational, synchronized.
>
> If the parameter is a character string, it must be defined as COBOL picture X(n) or A(n), where n is large enough for the required number of characters.

In the following parameters, those not specifically defined as characters will be assumed to be integers.

It is assumed that the user is already familiar with DSN/DS, in general, and the program-to-program intrinsics, specifically. Information regarding formal usage or content of the interface intrinsic parameters can be found in Section V.

COMMON PARAMETERS

Parameters whose use is the same through all the procedures are:

CCODE - integer (required)
The condition code returned by the PTOP intrinsic.
-1 = CCL
0 = CCE
1 = CCG

DSNUM - integer (required)
The number returned by CPOPEN, and which is required for all subsequent master PTOP calls.
The number is always 0 for slave programs.

ITAG - A 20-word integer field (optional).

TARGET - A character field used for reading or writing data (required).

TCOUNT - The number of words or bytes to be read or written (required).
Words are a positive integer; bytes are negative.

INTERFACE INTRINSICS

CPOPEN

This procedure is the COBOL callable interface to POPEN.

Calling Sequence:

```
CALL "CPOPEN" USING CCODE, DSNUM, DSDEVICE, PROGNAME,
                    ITAG, ENTRYNAME, PARM, FLAGS,
                    STACKSIZE, DLSIZE, MAXDATA,
                    BUFFSIZE
```

Where:

DSDEVICE is a character field containing the node name, device class, or logical device number of the desired DS line.

PROGNAME is a character field containing the name (terminated by a space) of the remote slave program.

ITAG is the 20-word integer field sent to and received from the remote program.

ENTRYNAME is the character field specifying the secondary entry point (or spaces) where the remote program will begin execution. It is ignored if the slave system is an RTE system.

PARM is an integer value to be placed in Q-4 of the slave program. It is ignored if the slave system is an RTE system.

FLAGS, STACKSIZE, DLSIZE, and MAXDATA are all MPE parameters used to specify slave program loading options. See Section V of this manual or the MPE Intrinsics Reference Manual for usage. It is ignored if the slave system is an RTE system.

BUFFSIZE is an integer specifying the maximum number of words which will be transferred by any of the PTOP intrinsics.

CPREAD

This procedure is the COBOL callable interface to PREAD.

Calling Sequence:

```
     CALL "C PREAD" USING CCODE, DSNUM, LENGTH, TARGET,
                          TCOUNT, ITAG
```

Where:

LENGTH is the actual number of words or bytes (depending on the value of TCOUNT) read into TARGET. (Required.)

CPWRITE

This procedure is the COBOL callable interface to PWRITE.

Calling Sequence:

```
     CALL "CPWRITE" USING CCODE, DSNUM, TARGET, TCOUNT, ITAG
```

CPCONTROL

This procedure is the COBOL callable interface to PCONTROL.

Calling Sequence:

```
     CALL "CPCONTROL" USING CCODE, DSNUM, ITAG
```

CPCLOSE

This procedure is the COBOL callable interface to PCLOSE.

Calling Sequence:

```
     CALL "CPCLOSE" USING CCODE, DSNUM
```

CGET

This procedure is the COBOL callable interface to GET.

Calling Sequence:

```
     CALL "CGET" USING CCODE, IFUN, ITAG, IL, IONUMBER
```

Where:

IFUN is the function number of the current pending PTOP operation. (Required.)

0 = An error occurred. This value is returned only when the condition code CCL is also returned. Issue a PCHECK intrinsic call (with a dsnum 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 also returned. It indicates that a pending MPE File System I/O without wait request was completed (instead of the expected remote DSN/DS I/O request). ionumber contains the file number associated with the completed I/O request.

IL is the number of words sent by a PWRITE or the number of words requested by a PREAD.

IONUMBER is the file number of a non-DS file which completed an I/O without wait.

CACCEPT

This procedure is the COBOL callable interface to ACCEPT.

Calling Sequence:

```
     CALL "CACCEPT" USING CCODE, ITAG, TARGET, TCOUNT
```

CREJECT

This procedure is the COBOL callable interface to REJECT.

Calling Sequence:

```
CALL "CREJECT" USING CCODE, ITAG
```

CPCHECK

This procedure is the COBOL callable interface to PCHECK.

Calling Sequence:

```
CALL "CPCHECK" USING CCODE, DSNUM, ICODE
```

Where:

DSNUM is an integer.
For a master program, this number is returned by the CPOPEN call. (Required.)
For a slave program, this number is always 0. (Optional.)

ICODE is an integer identifying the last error encountered. The error code meanings are given in Appendix B, under "DSN/DS Functional Errors".

EXAMPLE

The following example illustrates how two COBOL programs, residing on two HP 3000 computers, pass data back and forth. These two programs demonstrate and test the intrinsics available to the user of the COBOL Program-to-Program facility of DSN/DS. The Slave program must be entered on the remote system, compiled, and PREPed before the test. The PREPed file must then be made a permanent file. In this example, the MPE commands were:

```
:COBOL COBOLSS  (COBOL Slave Source)
:PREP $OLDPASS, COBOLS
:SAVE COBOLS
```

The Master program must then be entered, compiled, PREPed, and run on the local system.

A brief outline of the test is as follows:

1. The Master program opens the Slave program with CPOPEN. The ITAG array is filled with the value of the subscript of each array element, and the CPOPEN intrinsic is called. The Slave displays certain parameters involved in the opening. Then the Master also displays the value of the parameters used for

opening the remote program. After each call is made to a COBOL intrinsic, the status of the call is checked in the STATUS-CK-RTN paragraph.

2. The Master next tests the CPREAD intrinsic by requesting that a message from the Slave be sent back.

3. CPWRITE is tested by sending a message to the Slave. The Slave then displays the message as it was received to demonstrate the validity of the text.

4. The CREJECT-TEST paragraph of the Master is used to test the CPREJECT intrinsic available to the Slave as well as the CPCONTROL intrinsic of the Master. The value 14 is moved into the first element of ITAG and CPCONTROL is called. Within the paragraph that handles a call to CPCONTROL, the Slave tests this value and rejects the request.

5. The master then calls CPCLOSE to close the remote program before terminating itself.

The individual programs are shown on the following pages.

Master PTOP Program

```
001000$CONTROL USLINIT,SOURCE,MAP
001100 IDENTIFICATION DIVISION.
001200 PROGRAM-ID. MASTER-COBOL.
001300 ENVIRONMENT DIVISION.
001400 DATA DIVISION.
001500************************************
001600 WORKING-STORAGE SECTION.
001700 77 CCODE        PIC S99    COMP  VALUE 0.
001800 77 DSNUM        PIC S99    COMP  VALUE 0.
001900 77 PARAM        PIC S99    COMP  VALUE 0.
002000 77 FLAGS        PIC S99    COMP  VALUE 33.
002100 77 STACKSIZE    PIC S9(4)  COMP  VALUE IS -1.
002200 77 DLSIZE       PIC S9(4)  COMP  VALUE IS -1.
002300 77 MAXDATA      PIC S9999  COMP  VALUE IS 8000.
002400 77 BUFSIZE      PIC S999   COMP  VALUE IS 304.
002500 77 LGTH         PIC S99    COMP  VALUE 0.
002600 77 ICODE        PIC S99    COMP  VALUE 0.
002700 77 TCOUNT       PIC S99    COMP  VALUE IS 33.
002800 77 I            PIC S99    COMP  VALUE 0.
002900 77 DATA-BUF     PIC X(66)        VALUE SPACES.
003000**********************************************
003100 77 A-DOLLAR     PIC X(12) VALUE "<< ACCEPT >>".
003200 77 K-DOLLAR     PIC X(12) VALUE "## PCHECK ##".
003300 77 M-DOLLAR     PIC X(26) VALUE "######### MASTER #########".
003400 77 O-DOLLAR     PIC X(11) VALUE "## POPEN ##".
003500 77 R-DOLLAR     PIC X(11) VALUE "## PREAD ##".
003600 77 S-DOLLAR     PIC X(18) VALUE "## STATUS CHECK ##".
003700 77 W-DOLLAR     PIC X(12) VALUE "## PWRITE ##".
```

```
 77 C0-DOLLAR   PIC X(14) VALUE "## PCONTROL ##".
 77 C9-DOLLAR   PIC X(12) VALUE "## PCLOSE ##".
 77 D1-DOLLAR   PIC XXXX  VALUE "INDY".
 77 E0-DOLLAR   PIC XX    VALUE "  ".
 77 P0-DOLLAR   PIC X(7)  VALUE "COBOLS ".
 77 R0-DOLLAR   PIC X(12) VALUE "<< REJECT >>".
**********************************************
 01 ITAG-ARRAY.
     02 ITAG-ARRAY-MEM PIC 99 OCCURS 20 TIMES.
**********************************************
*            PROCEDURE DIVISION              *
**********************************************
 PROCEDURE DIVISION.
 DRIVER-PARA.
     PERFORM ISSUE-AN-OPEN.
     PERFORM ISSUE-A-READ.
     PERFORM ISSUE-A-WRITE.
     GO TO CREJECT-TEST.
     GO TO PCLOSE-CALL.
 ISSUE-AN-OPEN.
     DISPLAY M-DOLLAR.
     DISPLAY O-DOLLAR.
     PERFORM LOOPI VARYING I FROM 1 BY 1 UNTIL
      I IS GREATER THAN 20.
     DISPLAY " TAG TO BE SENT: ".
     DISPLAY ITAG-ARRAY.
     CALL "CPOPEN" USING CCODE, DSNUM, D1-DOLLAR, P0-DOLLAR,
      ITAG-ARRAY, E0-DOLLAR, PARAM, FLAGS, STACKSIZE, DLSIZE,
      MAXDATA, BUFSIZE.
     DISPLAY O-DOLLAR.
     DISPLAY " CCODE=", CCODE, " DSNUM=", DSNUM,
      " PARAM=", PARAM.
     DISPLAY " FLAGS=", FLAGS, " STACKSIZE=", STACKSIZE.
     DISPLAY " DLSIZE=", DLSIZE, " MAXDATA=", MAXDATA.
     DISPLAY " BUFSIZE=", BUFSIZE, " LGTH=", LGTH.
     DISPLAY " PROGNAME=", P0-DOLLAR.
     DISPLAY " ITAG-ARRAY RECEIVED: ".
     DISPLAY ITAG-ARRAY.
     PERFORM STATUS-CK-RTN.
 ISSUE-A-PREAD.
     DISPLAY R-DOLLAR.
     PERFORM LOOPI VARYING I FROM 1 BY 1 UNTIL
      I IS GREATER THAN 20.
     DISPLAY "ITAG TO BE SENT: ".
     DISPLAY ITAG-ARRAY.
     CALL "CPREAD" USING CCODE, DSNUM, LGTH, DATA-BUF,
      TCOUNT, ITAG-ARRAY.
     DISPLAY " CCODE=", CCODE, " DSNUM=", DSNUM,
      " LGTH=", LGTH.
     DISPLAY " DATA RECEIVED FROM SLAVE: ".
     DISPLAY DATA-BUF.
     DISPLAY " ITAG RECEIVED: ".
     DISPLAY ITAG-ARRAY.
     PERFORM STATUS-CK-RTN.
 ISSUE-A-WRITE.
```

```
009300     DISPLAY W-DOLLAR.
009400     PERFORM MULTIPLY-LOOP VARYING I FROM 1 BY 1 UNTIL
009500      I IS GREATER THAN 20.
009600     DISPLAY " ITAG TO BE SENT: ".
009700     DISPLAY ITAG-ARRAY.
009800     MOVE "THIS IS THE DATA FROM PWRITE TEST." TO DATA-BUF.
009900     CALL "CPWRITE" USING CCODE, DSNUM, DATA-BUF, TCOUNT,
010000      ITAG-ARRAY.
010100     DISPLAY " CCODE=", CCODE, " DSNUM=", DSNUM.
010200     DISPLAY " ITAG RECEIVED: ".
010300     DISPLAY ITAG-ARRAY.
010400     PERFORM STATUS-CK-RTN.
010500 CREJECT-TEST.
010600     DISPLAY C0-DOLLAR.
010700     MOVE SPACES TO ITAG-ARRAY.
010800     MOVE 14 TO ITAG-ARRAY-MEM(1).
010900     DISPLAY "ITAG TO BE SENT: ".
011000     DISPLAY ITAG-ARRAY.
011100     CALL "CPCONTROL" USING CCODE, DSNUM, ITAG-ARRAY.
011200     PERFORM STATUS-CK-RTN.
011300     STOP RUN.
011400 LOOPI.
011500     MOVE I TO ITAG-ARRAY-MEM(I).
011600 MULTIPLY-LOOP.
011700     MULTIPLY 2 BY ITAG-ARRAY-MEM(I).
011800 STATUS-CK-RTN.
011900     IF CCODE IS LESS THAN ZERO GO TO
012000      SOMETHING-WENT-WRONG.
012100     IF CCODE IS GREATER THAN ZERO GO TO
012200      REQUEST-REJECTED.
012300     DISPLAY S-DOLLAR, "EVERYTHING OKAY".
012400 REQUEST-REJECTED.
012500     DISPLAY S-DOLLAR, "REQUEST REJECTED BY SLAVE".
012600     GO TO PCLOSE-CALL.
012700 SOMETHING-WENT-WRONG.
012800     DISPLAY S-DOLLAR, "CCL--SOMETHING IS WRONG".
012900     CALL "CPCHECK" USING CCODE, DSNUM, ICODE.
013000     DISPLAY K-DOLLAR, " CCODE=", CCODE,
013100      " ICODE=", ICODE.
013200 PCLOSE-CALL.
013300     DISPLAY C9-DOLLAR.
013400     CALL "CPCLOSE" USING CCODE, DSNUM.
013500     DISPLAY " CCODE=", CCODE, " DSNUM=", DSNUM.
013600     STOP RUN.
```

Slave PTOP Program

```
001000$CONTROL USLINIT,SOURCE
001100 IDENTIFICATION DIVISION.
001200 PROGRAM-ID. SLAVE-COBOL.
001300 ENVIRONMENT DIVISION.
001400 DATA DIVISION.
001500********************************************
001600 WORKING-STORAGE SECTION.
001700 77 T       PIC S99  USAGE COMP.
001800 77 I       PIC S99  COMP VALUE 0.
001900 77 CCODE     PIC S99     COMP VALUE 0.
002000 77 IFUN      PIC S9      COMP VALUE 0.
002100 77 IL        PIC S99     COMP VALUE 0.
002200 77 IONUMBER  PIC S99     COMP VALUE 0.
002300 77 ICODE     PIC S99     COMP VALUE 0.
002400 77 DSNUM     PIC S99     COMP VALUE 0.
002500********************************************
002600 77 C-DOLLAR  PIC X(11)  VALUE "## CHECK ##".
002700 77 G-DOLLAR  PIC X(9)   VALUE "## GET ##".
002800 77 A-DOLLAR  PIC X(12)  VALUE "## ACCEPT ##".
002900 77 R-DOLLAR  PIC X(12)  VALUE "## REJECT ##".
003000 77 S-DOLLAR  PIC X(25)  VALUE "######### SLAVE #########".
003100********************************************
003200 77 OO-DOLLAR  PIC X(11)  VALUE "<< POPEN >>".
003300 77 CO-DOLLAR  PIC X(14)  VALUE "<< PCONTROL >>".
003400 77 SO-DOLLAR  PIC X(18)  VALUE "## STATUS CHECK ##".
003500 77 RO-DOLLAR  PIC X(11)  VALUE "<< PREAD >>".
003600 77 WO-DOLLAR  PIC X(12)  VALUE "<< PWRITE >>".
003700********************************************
003800 01 DATA-ARRAY.
003900     02 DATA-ARRAY-MEM  PIC 99 OCCURS 33 TIMES.
004000 01 ITAG-ARRAY.
004100     02 ITAG-ARRAY-MEM  PIC 99 OCCURS 20 TIMES.
004200***************************************************
004300*            PROCEDURE DIVISION                  *
004400***************************************************
004500 PROCEDURE DIVISION.
004600 START-OF-SLAVE.
004700     DISPLAY S-DOLLAR.
004800     CALL "CGET" USING CCODE, IFUN, ITAG-ARRAY, IL, IONUMBER.
004900     DISPLAY G-DOLLAR.
005000     PERFORM PRINT-GET-PARAMS THROUGH CHECK-RETURN-CC.
005100     IF CCODE IS NOT EQUAL TO ZERO GO TO CREJECT-PAR.
005200     GO TO POPEN, PREAD, PWRITE, PCONTROL DEPENDING ON IFUN.
005300 POPEN.
005400     DISPLAY S-DOLLAR.
005500     DISPLAY OO-DOLLAR.
005600     MOVE ZEROES TO ITAG-ARRAY.
005700     GO TO CACCEPT-PAR.
005800 PREAD.
005900     DISPLAY S-DOLLAR.
006000     DISPLAY RO-DOLLAR.
006100     PERFORM LOOP1 VARYING I FROM 1 BY 1 UNTIL
006200      I IS GREATER THAN 20.
```

```
006300     PERFORM INCREASE-LOOP VARYING T FROM 1 BY 1 UNTIL
006400      T IS GREATER THAN IL.
006500     GO TO CACCEPT-PAR.
006600 LOOP7.
006700     MOVE 7 TO ITAG-ARRAY-MEM(I).
006800 LOOP1.
006900     MOVE 1 TO ITAG-ARRAY-MEM(I).
007000 LOOP2.
007100     MOVE 2 TO ITAG-ARRAY-MEM(I).
007200 INCREASE-LOOP.
007300     MOVE T TO DATA-ARRAY-MEM(T).
007400 PWRITE.
007500     DISPLAY S-DOLLAR.
007600     DISPLAY WO-DOLLAR.
007700     PERFORM LOOP7 VARYING I FROM 1 BY 1 UNTIL
007800      I IS GREATER THAN 20.
007900 CACCEPT-PAR.
008000     CALL "CACCEPT" USING CCODE, ITAG-ARRAY, DATA-ARRAY, IL.
008100     DISPLAY A-DOLLAR.
008200     PERFORM CHECK-RETURN-CC.
008300     IF IFUN = 3 PERFORM PRINT-DATA.
008400     GO TO START-OF-SLAVE.
008500 CREJECT-PAR.
008600     CALL "CREJECT" USING CCODE, ITAG-ARRAY.
008700     DISPLAY R-DOLLAR.
008800     PERFORM CHECK-RETURN-CC.
008900     GO TO START-OF-SLAVE.
009000 PCONTROL.
009100     DISPLAY CO-DOLLAR.
009200     IF ITAG-ARRAY-MEM(1) = 14 GO TO CREJECT-PAR.
009300     PERFORM LOOP2 VARYING I FROM 1 BY 1 UNTIL
009400      I IS GREATER THAN 20.
009500     GO TO CACCEPT-PAR.
009600 PRINT-DATA.
009700     DISPLAY "DATA RECEIVED FROM THE MASTER: ".
009800     DISPLAY DATA-ARRAY.
009900 PRINT-GET-PARAMS.
010000     DISPLAY " CCODE=", CCODE, " IFUN=", IFUN, " IL=", IL,
010100      " IONUMBER=", IONUMBER.
010200     DISPLAY "ITAG RECEIVED: ".
010300     DISPLAY ITAG-ARRAY.
010400 CHECK-RETURN-CC.
010500     IF CCODE IS NOT EQUAL TO ZERO
010600      PERFORM SOMETHING-WENT-WRONG.
010700     DISPLAY SO-DOLLAR, "EVERYTHING OKAY".
010800 SOMETHING-WENT-WRONG.
010900     CALL "CPCHECK" USING CCODE, DSNUM, ICODE.
011000     DISPLAY C-DOLLAR.
011100     DISPLAY " CCODE=", CCODE, " ICODE=", ICODE.
011200 ERROR-EXIT.
011300     DISPLAY "## SLAVE PROGRAM EXITING ##".
011400     STOP RUN.
```

DSN/DS BASIC INTERFACE

APPENDIX G

CONVENTIONS

When parameters are specified in the CALL statement the BASIC/3000 Interpreter (and compiled BASIC) sets up a parameter address table. The parameter address table consists of:

- The number of parameters.

- A code word for each parameter, specifying data type and dimensioning.

- A reference pointer to each parameter.

See Appendix F of the BASIC/3000 Interpreter Manual.

Because the DSN/DS intrinsics are program-to-program, the BASIC/3000 slave must be a compiled and PREPed program. The master program may be either running on the Interpreter or run as a compiled program.

It is assumed that the user is already familiar with DSN/DS in general and the program-to-program intrinsics specifically. Information regarding formal usage or content of the interface intrinsic parameters can be found in Section V.

COMMON PARAMETERS

Parameters whose usage is the same throughout the procedures are:

CCODE - integer (required)
Condition code returned by the DSN/DS Program-to-Program intrinsic.

-3 = not enough user stack for data transfer.

-2 = CCL

0 = CCE

1 = CCG

DSNUM - integer (required)
The DSN/DS communication line number.
(analagous to FOPEN file number)

ITAG - integer (optional)
A 20-word array.

Parameters - The BASIC-DSN/DS interface routines pack and unpack the data specified in the parameter lists of the master and the slave programs. The user must insure that the number of parameters specified on the master and the slave sides are the same and that the data types correspond. If the sending and receiving data types are not the same, the resulting data will be unpredictable.

INTERFACE INTRINSICS

BPOPEN

This procedure is the BASIC callable interface to POPEN.

Calling Sequence:

```
                                                [{,ITAG}
   CALL BPOPEN (CCODE,DSNUM,DSDEVICE,PROGNAME   [{,0   }

               [{,ENTRYNAME}  [{,PARAM}  [{,FLAGS}
               [{,0        }  [{,0    }  [{,0    }

               [{,STACKSIZE}  [{,DLSIZE}  [{,MAXDATA}
               [{,-1       }  [{,-1    }  [{,-1     }

               [,BUFFSIZE]]]]]]]])
```

Where:

DSDEVICE - string (required).
The DS line class, node name, or logical device number (string must have at least one trailing blank).

PROGNAME - string (required).
Name of remote slave program (terminated with a blank).

ENTRYNAME - string (optional).
Secondary entry point into the slave program (terminated with a blank).

PARAM - integer (optional).
Value placed in Q-4 of the slave program stack.

FLAGS
STACKSIZE
DLSIZE
MAXDATA - MPE parameters used to control slave program loading. See Section V of this manual or the MPE Intrinsics Reference Manual for usage.

BUFSIZE - integer (optional)
Maximum number of words per PTOP transfer.

BPREAD

This procedure is the BASIC interface routine to PREAD.

Calling Sequence:

```
                                   ┌┌ ,ITAG ┐                ┐
CALL BPREAD(CCODE,DSNUM,LGTH       ││       │  [,param list] │)
                                   └└ ,0    ┘                ┘
```

Where:

LGTH - integer (required).
Number of words received in transfer.

param list ::= param [, param list]

param Any BASIC supported data type (such as, STRING, INTEGER, REAL ARRAY)

NOTE

The BASIC interface routines: BPREAD, BPWRITE, and BACCEPT differ significantly from the PTOP intrinsics in the TARGET/TCOUNT vs. parameter list data schemes. The parameter list allows the BASIC user to send and receive heterogeneous data items. A contiguous buffer is built on the User's stack for the transfers.

BPWRITE

This procedure is the BASIC callable interface to PWRITE.

Calling sequence:

```
                              ┌┌ ,ITAG ┐                ┐
CALL BPWRITE(CCODE,DSNUM      ││       │  [,param list] │)
                              └└ ,0    ┘                ┘
```

BPCONTROL

This procedure is the BASIC interface routine to PCONTROL.

Calling Sequence:

```
CALL BPCONTROL(CCODE,DSNUM[,ITAG])
```

BPCLOSE

This procedure is the BASIC interface routine to PCLOSE.

Calling Sequence:

```
CALL BPCLOSE(CCODE,DSNUM)
```

BGET

This procedure is the BASIC interface routine to GET.

Calling Sequence:

```
                         ┌,ITAG  ┌,IL   ┌,IONUMBER┐┐┐
CALL BGET( CCODE,IFUN    │       │      │         │││ )
                         └,0     └,0    └,0       ┘┘┘
```

Where:

IFUN - integer (required).
Receives the function code from the request issued by the remote master program. (Refer to Section V for IFUN meanings.)

IL - integer (optional).
The number of words expected or sent on BPREAD or BPWRITE.

IONUMBER - integer (optional)
Valid if both CCODE=1 and IFUN=5
File number of completed non-DS I/O without wait.

BACCEPT

This is the BASIC callable interface routine to ACCEPT.

Calling Sequence:

```
                          ┌,ITAG┐
CALL BACCEPT( CCODE,IFUN  │     │  [,param list] )
                          └,0   ┘
```

BREJECT

This is the BASIC callable interface routine to REJECT.

Calling Sequence:

```
CALL BREJECT(CCODE[,ITAG])
```

BPCHECK

This is the BASIC callable routine to PCHECK.

Calling Sequence:

```
CALL BPCHECK(CCODE,DSNUM,ICODE)
```

Where:

DSNUM - integer.
For a master program, this number is returned by BPOPEN. (Required.)
For a slave program, this number is always 0. (Optional.)

ICODE - integer (required).
The number returned identifies the last error encountered. Refer to Appendix B, under the heading "DSN/DS Functional Errors", for meaning.

EXAMPLES

Master PTOP Program

MASTER

```
 1 REM:**********************************************
 2 REM:
 3 REM:          MASTER PTOP PROGRAM
 4 REM:
 5 REM:**********************************************
 6 REM:
 7 REM: THIS PROGRAM ISSUES A BPOPEN TO THE SALVE PROGRAM
 8 REM: AND USES THE TAG FIELD TO SEND SUBTYPES FOR THE
 9 REM: BREAD/BWRITE OPERATIONS.
10 REM:
11 REM: THE SUBFUNCTIONS ARE:
12 REM:       WRITE:     1  -  CHANGE NAME N1 $ TO N2$
13 REM:                  2  -  CHANGE ADDRESS TO A1$ FOR NAME N
14 REM:                  3  -  ADD PERSON N1$, A1$
15 REM:                  4  -  DELETE PERSON N1$
16 REM:       READ:      1  -  SEND RECORD FOR NAME N1$
```

```
 17 REM:                  2  -  SEND ALL RECORDS
 18 REM: THE TAG RETURN VALUES ARE:
 19 REM:                  1  -  WRITE - OPERATION COMPLETE
 20 REM:                        READ - BUFFER CONTAINS VALID DAT
 21 REM:                  0  -  ON READ(2), MORE RECORDS COMING
 22 REM:                        ( CHANGES TO 1 ON LAST RECORD )
 23 REM:                 -1  -  RECORD NOT FOUND
100 REM:*********************************************
101 REM:         DATA DECLARATIONS
102 REM:*********************************************
105 REM: ***BASIC PTOP INTRINSIC PARAMETERS:::
110 INTEGER C,D,F,IO,L
115 REM: C=CC; D=DSUM; F=FLAGS; IO=ICODE; L=LENGTH
120 INTEGER I1[20]
125 REM: I1[*]=TAG FIELD
130 DIM DO$[4],PO$[6]
135 REM: DOS=DSLINE; PO$=REMOTE PROGRAM (SLAVE)
150 REM: *** LOCAL VARIABLES ***
155 DIM N1$[20], N2$[20], A1$[20], R$[40]
160 REM: N1$, N2$=NAMES; A1$=ADDRESS; R$=FULL RECORD (N1$,A1$)
165 DIM C$[20], N$[22]
170 REM: C$=USER COMMAND/TEXT LINE
175 REM: N$=ENTER NAME MESSAGE
200 REM:*********************************************
201 REM:
202 REM:         START OF PROCESSING
203 REM:
204 REM:*********************************************
205 REM << INITIALIZATION OF VALUES AND BPOPEN >>
210 DO$="PR2 "
215 PO$="SLAVE "
220 F=32
225 CALL BPOPEN (C,D,DO$,PO$,0,0,0,F)
230 IF C=0 THEN GOTO 300
235 PRINT "### ERROR ON BPOPEN ###"
240 GOSUB 1900
245 STOP
300 REM:*********************************************
301 REM:     WELCOME MESSAGE AND MENU
302 REM:*********************************************
305 PRINT "MASTER AND SLAVE PTOP RUNNING"
310 PRINT " "
315 PRINT "*** OPERATIONS MENU ***"
320 PRINT "    N  -  NAME CHANGE"
325 PRINT "    A  -  ADDRESS CHANGE"
330 PRINT "    I  -  INSERT PERSON"
335 PRINT "    D  -  DELETE PERSON"
340 PRINT "   LN  -  LIST NAME AND ADDRESS"
345 PRINT "   LA  -  LIST ALL NAMES AND ADDRESSES"
350 PRINT "   EX  -  EXIT PROGRAM"
400 REM:********************************************
401 REM:       NOW ASK FOR OPERATION
402 REM:********************************************
405 PRINT " "
410 PRINT "ENTER OPERATION"
```

```
 415 INPUT C$
 420 IF C$[1,1]="N" THEN GOTO 1000
 425 IF C$[1,1]="A" THEN GOTO 1100
 435 IF C$[1,1]="I" THEN GOTO 1200
 440 IF C$[1,1]="D" THEN GOTO 1300
 445 IF C$[1,2]="LN" THEN GOTO 1400
 450 IF C$[1,2]="LA" THEN GOTO 1500
 455 IF C$[1,2]="EX" THEN GOTO 2000
 460 PRINT "***UNRECOGNIZED OPERATION ***"
 465 GOTO 310
1000 REM:*********************************************
1001 REM:           NAME CHANGE
1002 REM:*********************************************
1005 PRINT N$
1010 LINPUT N1$
1015 IF I1[1]<0 OR I1[1]>1 THEN GOTO 1900
1020 PRINT "ENTER NEW NAME"
1025 LINPUT N2$
1030 L1[1]=1
1035 CALL BPWRITE (C,D,I1[*],N1$,N2$)
1045 GOSUB 1900
1050 GOTO 1000
1100 REM:*********************************************
1101 REM:            ADDRESS CHANGE
1102 REM:*********************************************
1105 PRINT N$
1110 LINPUT 01$
1115 IF N1$="" THEN GOTO 400
1120 PRINT "ENTER NEW ADDRESS"
1125 LINPUT A1$
1130 I1[1]=2
1135 CALL BPWRITE(C,D,I1[*],N1$,A1$)
1145 GOSUB 1900
1150 GOTO 1100
1200 REM:*********************************************
1201 REM:           INSET NAME
1202 REM:*********************************************
1205 PRINT N$
1210 LINPUT N1$
1215 IF N1$="" THEN GOTO 400
1220 PRINT "ENTER ADDRESS"
1225 LINPUT A1$
1230 I1[1]=3
1235 CALL BPWRITE(C,D,I1[*],N1$,A1$"
1245 GOSUB 1900
1250 GOTO 1200
1300 REM:*********************************************
1301 REM:            DELETE PERSON
1302 REM:*********************************************
1305 PEINR N$
1310 LINPUT N1$
1315 IF N1$="" THEN GOTO 400
1320 I1[1]=4
1325 CALL BPWRITE(C,D,I1[*],N1$)
1335 GOSUB 1900
```

```
1340 GOTO 1300
1400 REM:*********************************************
1401 REM:           LIST NAME AND ADDRESS
1402 REM:*********************************************
1405 PRINT N$
1410 LINPUT N1$
1415 IF N1$="" THEN GOTO 400
1420 I1[1]=1
1425 CALL BPWRITE(C,D,I1[*],N1$,N1$)
1430 IF C=0 THEN CALL BPCONTROL(C,D,I1[*])
1435 IF C<>0 OR I1[1]<>1 THEN GOTO 400
1440 L=-80
1445 CALL BPREAD(C,D,LI1[*],R$)
1450 GOSUB 1900
1455 PRINT R$
1460 GOTO 1400
1500 REM:*********************************************
1501 REM:            LIST WHOLE LIST
1502 REM:*********************************************
1505 L=-80
1510 I1[1]=2
1515 CALL BPREAD(C,D,L,I1[*],R$)
1520 GOSUB 1900
1525 PRINT R$
1530 IF I1[1]=0 THEN GOTO 1500
1535 GOTO 400
1900 REM:*********************************************
1901 REM:       CONDITION CODE AND STATUS CHECK
1902 REM:*********************************************
1905 IF C>0 THEN GOTO 1945
1910 IF C<0 THEN GOTO 1960
1913 CALL BPCONTROL(C,D,I1[*])
1915 IF I1[1]< OR I1[1]>1 THEN GOTO 1930
1920 REM: ***EVERYTHING OKAY ***
1925 RETURN
1930 REM: *** BAD RECORD ***
1935 PRINT "### NON-EXISTENT RECORD ###"
1940 RETURN
1945 REM: *** CCG ***
1950 PRINT."### REQUEST REJECTED BY SLAVE ###"
1955 RETURN
1960 REM: *** CCL ***
1965 CALL BPCHECK(C,D,I0)
1970 PRINT "### PTOP ERROR:";I0;"###"
1975 RETURN
2000 REM:*********************************************
2001 REM:            EXIT
2002 REM:*********************************************
2005 CALL BPCLOSE (C,D)
2010 GOSUB 1900
2015 END
```

Slave PTOP Program

```
SLAVEBF
  1 REM:#############################################
  2 REM:
  3 REM:             SLAVE PTOP PROGRAM
  4 REM:
  5 REM:#############################################
  6 REM:
  7 REM: THIS PROGRAM ACEPTS DATA FROM THE MASTER
  8 REM: AND ACCORDINGLY CHANGES, INSERTS OR DELETES
  9 REM: ENTRIES. IT ALSO TRANSMITS NAME/ADDRESS
 10 REM: RECORDS TO THE MASTER.
100 REM:*********************************************
101 REM:              DATA DECLARATIONS
102 REM:*********************************************
105 REM:*** BASIC PROP INTRINSIC PARAMETERS ***
110 INTEGER C,F,L,IO,I,D
115 REM:    C=CC; F=FUNCTION; L=IL; IO=I/O; I=ICODE; D=DSNUM
120 INTEGER I1[20]
125 REM:    I1[*]=TAG FIELD
130 REM:*** LOCAL VARIABLES ***
135 DIM N1$[20],N2$[20],A1$[20]
140 REM:     N1$,N2$=NAME; A1$=ADDRESS
145 DIM NO$[50,20],AO$[50,20]
150 REM     LIST OF NAMES AND ADDRESSES
155 INTEGER P,S
160 REM:    P=LAST RECORD POINTER; S=STATUS
300 REM:*********************************************
301 REM:              START OF PROGRAM
302 REM:*********************************************
305 CALL BGET(C,F,I1[*])
310 GOSUB 1900
315 GOSUB F OF 500,600,700,1100
320 GOTO 300
500 REM:*********************************************
501 REM:               BPOPEN
502 REM:*********************************************
505 CALL BACCEPT(C,F)
510 GOSUB 1900
515 FOR P=1 TO 12
520   READ NO$[P],AO$[P]
525 NEXT P
530 RETURN
600 REM:*********************************************
601 REM:               READ
602 REM:*********************************************
605 IF I1[1]=2 THEN GOTO 650
606 REM:**** LIST SINGLE RECORD ****
610 I1[1]=S=1
615 CALL BACCEPT(C,F,I1[*],NO$[PO],AO$[PO])
620 GOSUB 1900
625 RETURN
650 REM:**** LIST ALL RCORDS ****
655 IF S<>0 THEN PO=0
```

```
 660 S=0
 665 P0=P0+1
 670 IF P0=P THEN S=1
 675 I1[1]=S
 680 CALL BACCEPT(C,F,I1[*],N0$[P0],A0$[P0])
 685 GOSUB 1900
 690 RETURN
 700 REM:************************************************
 701 REM:                  WRITE
 702 REM:************************************************
 705 GOSUB I1[1] OF 710,800,900,1000
 710 REM:**** NAME CHANGE ****
 715 CALL BACCPET(C,F,0,N1$,N2$)
 720 P0=0
 725 P9=P0+1
 730 IF N0$[P0]=N1$ THEN GOTO 750
 735 IF P0<> THEN GOTO 725
 740 S=-1
 745 RETURN
 750 REM:**** FOUND NAME, NOW CHANGE IT ****
 755 N0$[P0]=N2$
 760 S=1
 765 RETURN
 800 REM:**** ADDRESS CHANGE ****
 805 CALL BACCPET(C,F,0,N1$,A1$)
 810 GOSUB 1900
 815 P0=0
 820 P0=P0+1
 825 IF N0$[P0]=N1$ THEN GOTO 845
 830 IF P0<>P THEN GOTO 820
 835 S=-1
 840 RETURN
 845 REM:**** FOUND RECORD, NOW CHANGE ADDRESS ****
 850 RETURN
 845 REM:**** FOUND RECORD, NOW CHANGE ADDRESS ****
 850 A0$[P0]=A1$
 855 S=1
 860 RETURN
 900 REM:**** INSERT NAME ****
 905 CALL BACCEPT(C,F,0,N1$,A1$)
 910 GOSUB 1900
 915 IF P=50 THEN GOTO 945
 920 P=P+1
 925 N0$[P]=N1$
 930 A0$[P]=A1$
 935 S=1
 940 RETURN
 945 REM: **** LIST ALREADY FULL ****
 950 S=-1
 955 RETURN
1000 REM:**** DELETE NAME ****
1005 CALL BACCEPT(C,F,0,N1$)
1010 GOSUB 1900
1015 P0=0
1020 P0=P0+1
```

```
1025 IF NO$[PO]=N1$ THEN GOTO 1045
1030 IF PO<>P THEN GOTO 1020
1035 S=-1
1040 RETURN
1045 REM: **** FOUND IT, GET RID OF IT ****
1050 NO$[PO]=NO$[P]
1055 PRINT "### SLAVE: PTOP ERROR:";IO;"###"
1060 RETURN
1065 S=1
1070 RETURN
1100 REM:*********************************************
1101 REM:             BPCONTROL
1102 REM:*********************************************
1105 I1[1]=S
1110 CALL BACCEPT(C,F,I1[*])
1115 RETURN
1900 REM************************************************
1901 REM:          CONDITION CODE AND STATUS CHECK
1902 REM:***********************************************
1905 IF C>0 THEN GOTO 1925
1910 IF C<0 THEN GOTO 1940
1915 REM:**** EVERYTHING OKAY ****
1920 RETURN
1925 REM:**** CCG ****
1930 PRINT "### SLAVE: PTOP( CCG ) ###"
1935 RETURN
1940 REM:**** CCL ****
1945 I=O
1950 CALL BPCHECK(C,I,IO)
2000 REM:***********************************************
2001 REM:            DATA SECTION
2002 RE,:***********************************************
2005 DATA "CHRISTINE", "BRISTOL"
2010 DATA "MEL", "CAMBRIGE"
2015 DATA "CAROL","PALO ALTO"
2020 DATA "HISA, MISA & RICHIE","BRISTOL"
2025 DATA "LISBET","ZURICH"
2030 DATA "JOHN","BERKELEY"
2035 DATA "CAROLYN","ST. PAUL"
2040 DATA "TODD","SANTA CLARA"
2045 DATA "GARY","SAN FRANCISCO"
```

X.25/X.21 NETWORK CONFIGURATOR

APPENDIX H

INTRODUCTION

One of the features of the Network Configurator/Network Data Base is the ability to define the network configuration once, then store it away in an IMAGE/3000 data base, and have it automatically used whenever you use the network data base. In addition, you can later alter the configuration, and again store it away for future use.

The data associated with the configuration is stored in an IMAGE data base in the PUB group of the SYS account. The data in the data base is manipulated by means of the Network Configuration Utility (NETCONF), which also resides in PUB.SYS. Although the Distributed Systems Network (DSN) products have read-only access to the data base so that they can determine the options selected for a particular line, only a network manager can change the network configuration.

ENVIRONMENT

The Data Base

The network configuration information is held in an IMAGE/3000 data base in PUB.SYS. The data base consists of the following files:

```
NETCON  (Root File)
NETCON01
NETCON02
NETCON03
NETCON04
NETCON05
```

It will be necessary, since the configuration information is kept in a data base, to periodically make a backup copy onto magnetic tape, for purposes of catastrophe recovery. It is recommended that the backup be taken each time the network configuration is changed, since the data base is only updated by the Network Configuration Utility (NETCONF). By doing this, a secure backup will be held of the latest network configuration.

When backing up the data base, the DBSTORE operation must be done by a user of the PUB.SYS account. It is assumed that this user is also the network manager, as only the network manager would have access to the data base maintenance password.

The data base must be "RELEASED" using DBUTIL, so that all users (including DS/X.25) may have read access to the data base. Read access is necessary for the :DSLINE command to execute properly.

The NETCONF Utility

The DS Network Configuration Utility (NETCONF) resides in the PUB group of the SYS account. The purpose of NETCONF is to obtain from the network manager all of the information necessary to describe the network connection(s), the parameter values and options chosen at subscription time, and all information related to the way the connection(s) will be used.

The NETCONF utility can be run by any user with read access to the data base. Only the data base creator has a write access to the data base, and it is assumed that the creator is the network manager.

USING NETCONF

Data Base Organization

The network configuration data is arranged into two sets (or tables) of information.

The first set of data is known as the Remote Node (RN) table and contains the relationships between the destination logical node name, the logical device number (line identifier), the X.25/X.21 Public Data Network (PDN) address of the destination node (assigned by the PDN), and the remote node machine type. This table can have up to 2048 entries, and it must have one entry for every relationship. For example, if node FRED has a PDN address of 1234 and can be reached across two separate DS lines, then there must be one entry for each line.

Remote Node Name	Line Identifier (LDEV Number)	PDN Address
Fred	51	1234
Fred	52	1234

Similarly, if two different nodes can be reached across the same line (as is probable when that line is connected to a PDN) then there will be an entry for each node.

Remote Node Name	Line Identifier (LDEV Number)	PDN Address
Fred	51	1234
Joe	51	1235

For DS/X.25, the destination logical node name cannot be configured on the system as a device class name.

The second set of data is known as the Line Characteristics (LC) table and contains information pertaining to a particular line (logical device number). There must be an entry in this table for every line from this node and, unlike the RN table, each entry must be unique.

The relationship between the two tables is the line identifier (ldev number). For every line identifier referenced in the Remote Node table, there should be an entry in the LC table, and vice versa. NETCONF warns of any unsatisfied or illegal relationships when exiting; however, no attempt is made to insist on their being satisfied.

There must be an entry in the data base for every node to which you want to send information and from which you want to receive information. If a node tries to access your computer via X.25 and that node is not in the data base, the remote node will not be allowed access.

The Commands

The Network Configuration Utility (NETCONF) has eight first-level commands:

ADD

CHECK

DELETE

EXIT

HELP

LIST

PRINT

UPDATE

Any of these commands can be initiated after NETCONF has issued its identifying banner. The commands may (optionally) be abbreviated to one character, as any other input is ignored. The mode of NETCONF is conversational. After one of the commands has been specified, a series of prompts to the user is issued, as appropriate, for the relevant inputs.

To terminate NETCONF command execution during an interactive session on HP terminals, press Control-Y (hold the Control key and press Y). This action terminates the current command and prompts for another first-level command.

Refer to the information manual for your particular PDN for the recommended (or required) configuration parameters for X.25 connections.

NOTE

> The following description of the interactive dialogue, that is initiated by these commands, is presented in a format similar to the one used for the MPE Configuration Dialogue in Appendix A. For additional clarification of this format, refer to page xiv, "Conventions Used In This Manual".

THE A[DD] COMMAND

This command is used to add a new entry to either the Remote Node (RN) table or the Line Characteristic (LC) table. Note that only the creator of the data base can add entries to it. After specifying the ADD command, the dialogue proceeds as follows:

Step No. **Dialogue**

0 REMOTE NODE (RN) OR LINE CHARACTERISTICS (LC) TABLE?

Enter one of the following replies:

RN = When this is specified, you will be adding an entry to the Remote Node table, and the dialogue proceeds from there. Skip to Step 1.0.

LC = When this is specified, you will be adding an entry to the Line Characteristics table, and the dialogue proceeds from there. Skip to Step 2.0.

return = When you reply with a carriage return, you will receive the following prompt:

CONTINUING ADDING (YES OR NO)?

YES = This response takes you back to the ADD prompt (Step 0).

NO = This response takes you out of the ADD command and prompts for another first-level command.

INPUT MUST BE RN OR LC

If this message appears, the response was not one of the above. You will be prompted again with the ADD prompt (Step 0).

ADD

Adding to the RN Table

The following prompts cover the remote node characteristics.

Step No. **Dialogue**

1.0 REMOTE NODE NAME?

Enter a logical node name. This name can be up to eight alphanumeric characters (the first being an alphabetic character), and it must be the name by which the destination node is to be referred when using the DSLINE command.

NODE NAME SHOULD BE UP TO 8 ALPHANUMERIC CHARACTERS

This message appears when the node name is greater than eight alphanumeric characters or when the first character is numeric, You will be prompted again for a logical node name (Step 1.0).

1.1 REMOTE COMPUTER TYPE (HP3000 OR HP1000)?

return = The default Remote Computer type (HP 3000) is used.

HP3000 = The type of the Logical Node being addressed is an HP 3000.

HP1000 = The type of the Logical Node being addressed is an HP 1000.

INPUT MUST BE HP3000 OR HP1000

This message is received if the response was not one of the above. You will be prompted again for the Remote Computer type (Step 1.1).

1.2 LOGICAL DEVICE NUMBER TO BE USED?

Enter a line identifier (logical device number). This can be a numeric value between 1 and 255, and it must be the logical device number of the DS/X.25 line supervisor (IODSX).

Step No.	Dialogue
	LOGICAL DEVICE SHOULD BE IN THE RANGE OF 1 TO 255

This message appears when a line identifier that is not in the range of 1 to 255 has been specified. You will be prompted again for a Logical Device number (Step 1.2).

1.3 LINE TYPE (X25 OR X21)?

Enter one of the following replies:

return = The default (X25) line type is used.

X25 = When this is specified, you will be prompted for the X.25 Remote Node Address. Skip to Step 1.3.1.

X21 = When this is specified, you will be prompted for an X.21 Remote Node Address. Skip to Step 1.3.2.

INPUT MUST BE X25 OR X21

This message appears when the response was not one of the above. You will be prompted again for the Line Type (Step 1.3).

1.3.1 REMOTE X25 PDN ADDRESS?

Enter one of the following replies:

return = Either an X.25 network address is not needed because the connection will be across a point-to-point line, rather than a PDN; or the default network address will be used if the connection is across a PDN. (The default address is NULL). Skip to Step 1.4.

An X.25 PDN Network Address = This will be assigned by the relevant PDN across which you will be talking to the logical node. It should be a numeric address up to 15 digits in length, and it is the actual PDN address of the logical node. Skip to Step 1.4.

Step No. **Dialogue**

X25 ADDRESS SHOULD BE UP TO 15 DECIMAL DIGITS

This message appears if the specified address is greater than 15 decimal digits or if a non-numeric network address was entered. You will be prompted again for the X.25 PDN address (Step 1.3.1).

1.3.2 X21 PDN ADDRESS?

Enter one of the following replies:

return = The default X.21 Address (all blanks) is used.

An X.21 PDN Network Address = This address must be no more than 30 characters in length.

X21 ADDRESS SHOULD BE UP TO 30 CHARACTERS

This message appears if the address entered was larger than 30 characters. You will be prompted again for the X.21 PDN address (Step 1.3.2).

1.4 CONTINUING ADDING (YES OR NO)?

YES = This will take you back to the ADD prompt (Step 0).

NO or any input except YES = This will take you out of the ADD command and prompt for another first-level command.

ADDITION COMPLETE

This message appears when the Remote Node characteristics have been added to the Remote Node (RN) table.

DUPLICATE ENTRY - NEW ENTRY NOT ADDED

This message appears when there was already an entry in the Remote Node table with these relationships.

Step No.	Dialogue

DATA BASE IS FULL - NEW ENTRY NOT ADDED

This message appears when the data base is full. To correct this situation, exit from NETCONF and enlarge the size of the IMAGE data base.

Adding to the LC Table

The following prompts cover the general line characteristics.

Step No. **Dialogue**

2.0 LOGICAL DEVICE NUMBER?

Enter a Line Identifier (logical device number). This can be a numeric value between 1 and 255, and it must refer to the logical device number of the DS/X.25 line supervisor (IODSX).

LOGICAL DEVICE SHOULD BE IN THE RANGE OF 1 TO 255

This message appears if a line identifier not in the range 1 to 255 has been specified. You will be prompted again for a Logical Device number (Step 2.0).

DUPLICATE ENTRY - NEW ENTRY NOT ADDED

This message appears if there was already an entry in the LC table with the same Logical Device number.

2.1 LINE TYPE (X25 OR X21)?

Enter one of the following replies:

return = The default protocol (X.25) is used.

X25 = X.25 protocol will be used. Skip to Step 2.1.1.

X21 = X.21 protocol will be used. Skip to Step 2.1.2.

INPUT MUST BE X25 OR X21

This message appears if the response was not one of the above. You will be prompted again for a Logical Device number (Step 2.1).

Step No. **Dialogue**

2.1.1 CONNECTION DIRECT OR VIA PDN?

This prompt is issued only if the connection protocol is X.25. Enter one of the following responses:

return = The default connection type (DIRECT) is used. Skip to Step 2.1.1.1.

DIRECT = The line connection will be point-to-point. Skip to Step 2.1.1.1.

PDN = The line connection will be via Public Data Network. Skip to Step 2.1.1.2.

INPUT MUST BE DIRECT OR PDN

This message appears if the response was not one of the above. You will be prompted again for the connection type (Step 2.1.1).

2.1.2 LINE IS LEASED OR SWITCHED?

This prompt is issued only if the connection protocol is X.21. Enter one of the following responses:

return = The default LEASED is used (Step 2.2).

LEASED = The line type is LEASED and the connection is point-to-point. Skip to Step 2.2.

SWITCHED = The line type is SWITCHED and the connection is via PDN. Skip to Step 2.1.3.

INPUT MUST BE LEASED OR SWITCHED

This message appears if the response was not one of the above. You will be prompted again for the line type (Step 2.1.2).

2.1.1.1 MASTER (DCE) OR SLAVE (DTE) MODE?

This prompt is issued only if the connection protocol is X.25 and line connection is DIRECT. Enter one of the following responses:

DTE = The node is set up to act as a DTE. Note that one end of the connection must be set up as the DTE, while on the destination node it must be set up as a DCE. Skip to Step 2.2.

DCE = The node is set up to act as a DCE. Skip to Step 2.2.

INPUT MUST BE DCE OR DTE

This message appears if the response was not one of the above. You will be prompted again for the DTE or DCE mode (Step 2.1.1.1).

NOTE

When the connection protocol is X.25 and line connection is DIRECT, the following local X.25 addresses are used:

```
Master (DCE) mode -- Local X.25 address is 9
Slave (DTE) mode  -- Local X.25 address is 8
```

2.1.1.2 LOCAL X25 PDN ADDRESS?

This prompt is issued only if the connection protocol is X.25 and line connection is via PDN. Enter one of the following responses:

return = The default local address of all zeroes is used.

Local X25 PDN address = This is the actual local address (from address) assigned by PDN at subscription time. It should be a numeric address up to 15 digits in length.

X25 ADDRESS SHOULD BE UP TO 15 DECIMAL DIGITS

This message appears if the specified address is greater than 15 decimal digits or if a non-numeric network address has been specified. You will be prompted again for the local X.25 PDN address (Step 2.1.1.2).

Step No. **Dialogue**

2.1.3 NAME OF PDN?

This prompt is issued only if the connection protocol is X.21 and the line is SWITCHED, or if the connection protocol is X.25 and the line connection is via PDN. Enter one of the following responses:

Name of PDN = The PDN name must be no more than eight alphanumeric characters.

return = The default PDN name of all blanks is used.

PDN NAME SHOULD BE UP TO 8 ALPHANUMERIC CHARACTERS

This message appears if the PDN name is greater than eight alphanumeric characters. You will be prompted again for a PDN name (Step 2.1.3).

2.2 PRIMARY REMOTE NODE TO BE CONNECTED TO ON THIS LINE?

Reply by entering the Remote Node Name. This must correspond to one of the remote node names associated with this line identifier in the RN table. When a :DSLINE command is issued with a line identifier instead of a node name, it might not be unique. In such a case, the node to which the connection will be established is the one identified here.

NOTE

DS/X.25 users are encouraged to always use a node name in the :DSLINE command instead of a line identifier.

NODE NAME SHOULD BE UP TO 8 ALPHANUMERIC CHARACTERS

This message appears if the node name is greater than eight alphanumeric characters or if the first character was numeric. You will be prompted again for Primary Node name (Step 2.2).

The following prompts, covering low-level (Level 2) characteristics, are issued only if the connection protocol is X.25.

NOTE

For Direct Connect DS/X.25 lines, all Level 2 parameters must be configured exactly the same as the corresponding Level 3 parameters.

Step No. **Dialogue**

2.3 RESPONSE TIMER (MILLISECONDS) ?

Enter one of the following replies:

return = The default value of 200 is used.

Response Timer = This value must be an integer in the range of 1 to 9999, and it specifies the period at the end of which retransmission of a frame can be indicated. (In the case of a PDN connection, this is usually dictated by that PDN.)

RESPONSE TIMER SHOULD BE IN THE RANGE 1 TO 9999

This message appears if your reply was either non-numeric or not in the range of 1 to 9999. You will be prompted again for Response Timer (Step 2.3).

2.4 RETRY COUNT (1..255) ?

Enter one of the following replies:

return = The default value of 8 is used.

Retry Count = This must be a numeric value in the range of 1 to 255. It specifies the maximum number of retransmissions of frames that will be attempted following the running out of the response timer. (In the case of a PDN connection, this is usually dictated by that PDN.)

RETRY COUNT SHOULD BE IN THE RANGE 1 TO 255

This message appears if your response was either non-numeric or not in the range of 1 to 255. You will be prompted again for Retry Count (Step 2.4).

Step No. **Dialogue**

2.5 WINDOW SIZE (FRAMES) ?

Enter one of the following replies:

return = The default value of 2 is used.

Window size = Window size specifies the maximum number of sequentially numbered I-frames that a DTE/DCE may have outstanding (unacknowledged) at any given time. The minimum value of this parameter is 1, and the maximum value is 7. (In the case of a PDN connection, this is usually dictated by that PDN.)

NOTE

For optimum performance, tests have shown that both Level 2 and Level 3 window sizes should be 7.

WINDOW SIZE SHOULD BE IN THE RANGE 1 TO 7

This message appears if your response was either non-numeric or not in the range 1 to 7. You will be prompted again for Packet Size (Step 2.5).

The following prompts cover the upper-level (Level 3) characteristics.

2.6 LOW VC NUMBER (0..4095) ?

Enter one of the following replies:

return = The default value of 0 is assigned as the low virtual circuit number.

Virtual Circuit Number = This must be an integer in the range of 0 to 4095. It represents the low end of the virtual circuit identification numbers. (In the case of a PDN connection, this is usually dictated by that PDN.)

LOW VC SHOULD BE IN THE RANGE OF 0 TO 4095

This message appears if your response was not numeric or if it was not in the range of 0 to 4095. You will be prompted again for the Low VC Number (Step 2.6).

Step No. **Dialogue**

2.7 HIGH VC NUMBER (0..4095) ?

Enter one of the following replies:

return = The default value of the Low Virtual Circuit Number + 255 is assigned as the high virtual circuit number.

Virtual Circuit Number = This must be an integer in the range of 0 to 4095, and it represents the high end of the virtual circuit identification numbers. It has to be greater than the low virtual circuit number, but no more than 255 above that value. (In the case of a PDN connection, this is usually dictated by that PDN.)

HIGH VC SHOULD BE IN THE RANGE nnnn TO mmmm

This message appears if your response was non-numeric, was not in the range of 0 to 4095, was less than the low virtual circuit number, or was greater than the sum of the low virtual circuit number + 255. You will be prompted again for the high virtual circuit number (Step 2.7).

2.8 PACKET SIZE (32..1024) ?

Enter one of the following replies:

return = The default packet size (128 bytes) is used.

Packet Size = This must be a numeric value in the range of 32 to 1024. It represents the packet size (in bytes) that will be used across this connection. (In the case of a PDN connection, this is usually dictated by that PDN.)

PACKET SIZE SHOULD BE IN THE RANGE 32 TO 1024

This message appears if your response was either non-numeric or not in the range of 32 to 1024. You will be prompted again for the packet size (Step 2.8).

Step No.	Dialogue

2.9 MODULO COUNT (8 OR 128) ?

Enter one of the following replies:

return = The default of 8 is used.

Modulo Count = This is the counting scheme used for packets across this connection. (In the case of a Public Data Network (PDN) connection, This is usually dictated by that PDN.)

NOTE

The modulo count has no major effect on performance.

2.10 WINDOW SIZE (PACKETS) ?

Enter one of the following replies:

return = The default window size (2 packets) is used.

Window Size = This must be a numeric value in the range of 1 to 7 (for a modulo count of 8) or in the range of 1 to 15 (for a modulo count of 128). It represents the window size (in packets) that will be used across this connection. (In the case of a Public Data Network (PDN) connection, this is usually dictated by that PDN.)

NOTE

For optimum performance, tests have shown that both Level 2 and Level 3 window sizes should be 7 if a modulo count of 8 is being used at Level 3. For a modulo count of 128, any window size greater than 7 has approximately the same performance.

WINDOW SIZE SHOULD BE IN THE RANGE 1 TO 7

This message appears if your response was either non-numeric or not in the range of 1 to 7 when a modulo count of 8 is being used.

Step No. **Dialogue**

WINDOW SIZE SHOULD BE IN THE RANGE 1 TO 15

This message appears if your response was either non-numeric or not in the range of 1 to 15 when a modulo count of 128 is being used. After receiving either of these messages, you will be prompted again for the window size (Step 2.10).

2.11 CONTINUING ADDING (YES OR NO)?

YES = This will take you back to the ADD prompt (Step 0).

NO or any input except YES = This will take you out of the ADD command and prompt for another first-level command.

ADDITION COMPLETE

This message appears when the line characteristics have been added to the Line Characteristics (LC) table.

DATA BASE IS FULL - NEW ENTRY NOT ADDED

This message appears when the data base is full. To correct this situation, exit from NETCONF and enlarge the size of the IMAGE data base.

THE C[HECK] COMMAND

This command is used to check the relationships, and report any discrepancies, between the RN and LC tables. Three basic checks are performed; and since they are always done, there is no dialogue following the command.

The first check scans the Remote Node table. For every Line Identifier (Ldev number) that is used, it checks that there is a corresponding entry in the LC table. If there is no such entry, the following warning is printed:

```
Ldev nnn is not entered in the LC table
```

The second check scans the LC table. For each entry, it checks that the primary node name specified for a logical device has a corresponding entry in the Remote Node table. If there is no such entry, the following warning is printed:

```
aaaaaaaa ( using LDEV nnn ) is not entered
in the RN table
```

The third check scans the LC table. For each entry, it checks that all RN entries with the same LDEV have the same line type as the LC entry. For each entry in the RN table where the line types do not match, the following warning is printed:

```
aaaaaa (LDEV nnn) line type differs from LC line type
```

DELETE

THE D[ELETE] COMMAND

This command is used to remove data entries from the RN table or the LC table. Note that only the creator of the data base can delete entries. After specifying the DELETE command, the dialogue proceeds as follows:

Step No. **Dialogue**

0 REMOTE NODE (RN) OR LINE CHARACTERISTICS (LC) TABLE?

Enter one of the following replies:

RN = When this is specified, you will be deleting an entry from the Remote Node table, and the dialogue proceeds from there. Skip to Step 1.0.

LC = When this is specified, you will be deleting an entry from the Line Characteristics table, and the dialogue proceeds from there. Skip to Step 2.0.

return = When you reply with a carriage return, you will receive the following prompt:

CONTINUING DELETING (YES OR NO)?

YES = This response takes you back to the DELETE prompt (Step 0).

NO = This response takes you out of the DELETE command and prompts for another first-level command.

INPUT MUST BE RN OR LC

If this message appears, the response was not one of the above. You will be prompted again with the DELETE prompt (Step 0).

Deleting from the RN Table

The following prompts cover the remote node characteristics.

Step No. **Dialogue**

1.0 REMOTE NODE NAME?

Enter a remote node name. This name can be up to eight alphanumeric characters (the first being an alphabetic character), and it must refer to the same name by which the destination node is to be referred when using the DSLINE command.

NODE NAME SHOULD BE UP TO 8 ALPHANUMERIC CHARACTERS

This message appears when the node name is greater than eight alphanumeric characters or when the first character is numeric. You will be prompted again for a logical node name (Step 1.0).

NO SUCH ENTRY IN THE RN TABLE

This message appears if a legal remote node name has been specified, but there is no entry in the RN table for it. You will be prompted for another first-level command.

A valid remote node name has been specified, and you are about to delete an entry or entries from the RN table. NETCONF also prompts to enable you to delete a corresponding entry from the LC table. Since there can be multiple entries in the RN table for the name you have specified, NETCONF repeats the following sequence of prompts until all entries have been covered; whereupon you will be prompted for a first-level command.

DELETE

Step No. **Dialogue**

```
RN TABLE ENTRY WITH NODE NAME = xxxxxxxx USING LDEV = nnn

CONFIRM DELETION (YES OR NO) ?
```

Enter one of the following replies:

return = When you reply with a carriage return, this message appears:

```
ENTRY NOT DELETED
```

The delete is not confirmed, and processing proceeds. If there are further entries in the RN table satisfying the Remote Node Name specified, this step will be repeated; otherwise, you will be prompted for a first-level command.

NO or any input except YES = This reply results in the message:

```
ENTRY NOT DELETED
```

The delete is not confirmed, and processing proceeds. If there are further entries in the RN table satisfying the Remote Node Name specified, this step will be repeated; otherwise, you will be prompted for a first-level command.

YES = This reply results in the message:

```
ENTRY HAS BEEN DELETED
```

The entry has been deleted from the RN table, and processing proceeds. If there is an LC entry corresponding to this entry (having the same logical device number), processing proceeds to the next step; if there is not a corresponding LC entry and there are further entries in the RN table satisfying the remote node name specified, this step will be repeated. Otherwise, you will be prompted for a first-level command.

Step No. **Dialogue**

```
ASSOCIATED LC TABLE ENTRY WITH LDEV = nnn

CONFIRM DELETION (YES OR NO) ?
```

Enter one of the following replies:

return = The delete will not be confirmed, and processing proceeds. If there are further entries in the RN table satisfying the remote node name specified, the previous prompt is repeated; if there are not, you will be prompted for a first-level command.

NO or any input except YES = The delete will not be confirmed, and processing proceeds. If there are further entries in the RN table satisfying the logical node name specified, the previous prompt is repeated; if there are not, you will be prompted for a first-level command.

YES = The entry is deleted from the LC table, and processing proceeds. If there are further entries in the RN table satisfying the logical node name specified, the previous prompt is repeated; if there are not, you will be prompted for a first-level command.

DELETE

Deleting from the LC Table

Step No. **Dialogue**

2.0 LOGICAL DEVICE NUMBER?

Enter a line identifier (logical device number). This can be a numeric value between 1 and 255, and it must refer to the logical device number of the DS/X.25 line supervisor (IODSX).

LOGICAL DEVICE SHOULD BE IN THE RANGE OF 1 TO 255

This message appears if a line identifier not in the range 1 to 255 has been specified. You will be prompted again for a logical device number (Step 2.0).

NO SUCH ENTRY IN THE RN TABLE

This message appears if a legal line identifier was specified, but there is no entry in the LC table for it. You will be prompted for another first-level command.

A valid logical device number has been specified, and you are about to delete an entry from the LC table. NETCONF also prompts to enable you to delete a corresponding entry or entries from the RN table.

LC TABLE ENTRY WITH LOGICAL DEVICE NUMBER = nnn

CONFIRM DELETION (YES OR NO) ?

Enter one of the following replies:

return = The delete is not confirmed, and you will be prompted for a first-level command.

NO or any input except YES = The delete will not be confirmed, and processing proceeds. You will be prompted for a first-level command.

Step No. **Dialogue**

YES = The entry is deleted from the LC table, and processing proceeds. If there is an entry (or entries) in the RN table corresponding to this line identifier, processing proceeds to the next step; if not, you will be prompted for a first-level command.

ASSOCIATE RN TABLE ENTRIES USING LDEV = nnn

CONFIRM DELETION (YES OR NO) ?

Enter one of the following replies:

return = The delete is not confirmed, and you will be prompted for a first-level command.

NO or any input except YES = The delete is not confirmed, and you will be prompted for a first-level command.

YES = All entries in the RN table that use this line identifier (LDEV) are deleted, and you will be prompted for a first-level command.

EXIT

THE E[XIT] COMMAND

This command is used to terminate the execution of the Network Configurator. Prior to termination, a call is automatically made to the CHECK command, and any discrepancies in the relationship between the RN and LC tables are printed. If there are no discrepancies, NETCONF terminates. If there are discrepancies, processing proceeds as follows:

Step No. **Dialogue**

IS IT OK TO EXIT ?

Enter one of the following replies:

return = You will be prompted for a first-level command.

NO or any input except YES = You will be prompted for a first-level command.

YES = NETCONF terminates execution.

THE H[ELP] COMMAND

This command provides a basic description of each of the commands in the Network Configuration Utility (NETCONF) command set. Since the commands are only being described, there is no follow-up dialogue in the HELP command. Only a very basic description of functionality is provided by the HELP command; so when more detail is required, refer to the descriptions presented in this manual.

LIST

THE L[IST] COMMAND

This command provides a display on your terminal screen of the current content of the network configuration data base. The data is arranged into the Remote Node (RN) and Line Characteristics (LC) tables, under the following headings:

Remote Node Table

Node Name	System Type	Ldev No	Line Type	Remote PDN Address/Phone Number
xxxxxxxx	HP3000	nnn	X25	nnnnnnnnnnnnnn
.	.	.	.	.
xxxxxxxx	HP1000	nnn	X21	xxxxxxxxxxxxxxxxxxxxxxxx

Line Characteristics Table

Ldev No	Line Type	Connect Method	DCE/DTE	Remote Primary Node
nnn	X21	xxxxxxxx	DCE	xxxxxxxx
.	.	.	.	.
nnn	X25	xxxxxxxx	DTE	xxxxxxxx

Line Characteristics Table (X25)

Ldev No	Local X25 Address	Level 2			Level 3				
		T1 Timer	Retry Count	Win-dow	Low VC	High VC	Packet Size	Win-dow	Mod Cnt
nnn	nnnn ... nnn	nnnn	nnn	n	nnnn	nnnn	nnnn	nnnn	nnn

NOTE

The Line Characteristics Table (X25) will not be printed unless there are entries in the LC table that have the "X25" line type. If there are any X.25-related entries in the general LC table, then the LC (X25) table will contain entries only for those X.25-related LDEVs.

Since all of the information is automatically provided upon specifying the LIST command, there is no following dialogue.

PRINT

THE P[RINT] COMMAND

The PRINT command lists the current contents of the Remote Node (RN) and Line Characteristics (LC) tables to a line printer and validates the node name and logical device relationship between the two tables. It executes the LIST and CHECK commands, with the output device being a line printer rather than the $STDLIST. The formal designator is NETLIST and the default device name is LP. FILE equations are permitted, which enables you to specify a file or device to which the data base contents are to be printed.

THE U[PDATE] COMMAND

This command is used to update entries in the Remote Node (RN) or Line Characteristics (LC) table which already exist. Note that only the creator of the data base can update the data base. After specifying the UPDATE command, the dialogue proceeds as follows:

Step No. **Dialogue**

0 REMOTE NODE (RN) OR LINE CHARACTERISTICS (LC) TABLE?

Enter one of the following replies:

RN = When this is specified, you will be adding an entry to the Remote Node table, and the dialogue proceeds from there. Skip to Step 1.0.

LC = When this is specified, you will be adding an entry to the Line Characteristics table, and the dialogue proceeds from there. Skip to Step 2.0.

return = When you reply with a carriage return, you will receive the following prompt:

CONTINUING UPDATING (YES OR NO)?

YES = This response takes you back to the UPDATE prompt (Step 0).

NO = This response takes you out of the UPDATE command and prompts for another first-level command.

INPUT MUST BE RN OR LC

If this message appears, the response was not one of the above. You will be prompted again with the UPDATE prompt (Step 0).

Updating the RN Table

The following prompts cover the remote node characteristics.

Step No. **Dialogue**

1.0 REMOTE NODE NAME?

Enter a remote node name. This name can be up to eight alphanumeric characters (the first being an alphabetic character), and it must be the name by which the destination node is to be referred when using the DSLINE command.

NODE NAME SHOULD BE UP TO 8 ALPHANUMERIC CHARACTERS

This message appears when the node name is greater than eight alphanumeric characters or when the first character is numeric. You will be prompted again for a remote node name (Step 1.0).

NO SUCH ENTRY IN THE RN TABLE

This message appears if a legal remote node name was specified, but there is no entry in the RN table for it. You will be prompted for another first-level command.

Prior to issuing any prompts, NETCONF first prints all entries in the RN table that qualify with the remote node name specified. Since there can be multiple entries in the RN table for the remote node name that you have specified, you will be prompted for the logical device number associated with the remote node name.

1.2 LOGICAL DEVICE NUMBER ?

Enter one of the following replies:

<u>return</u> = You will be prompted again for an LDEV.

<u>A Line Identifier (Logical Device Number)</u> = This must be the line identifier that specifies which of the entries in the RN table for a particular logical node you wish to update.

Step No. **Dialogue**

LOGICAL DEVICE NUMBER SHOULD BE IN THE RANGE OF 1 TO 255

This message appears when a non-numeric line identifier, or a line identifier that is not in the range of 1 to 255, has been specified. You will be prompted again for a logical device number (Step 1.2).

NO SUCH ENTRY IN THE RN TABLE

This message appears if a valid line identifier was specified, but none of the qualifying NLA entries uses this line identification. You will be prompted for a first-level command.

All other prompts, responses, and error messages are the same as for the ADD command. For each variable in the entry, the current value is printed, followed by a prompt for a new value. A carriage return maintains the current value.

UPDATE

Updating the LC Table

Step No. **Dialogue**

2.0 LOGICAL DEVICE NUMBER?

Enter a line identifier (logical device number). This can be a numeric value between 1 and 255, and it must refer to the logical device number of the DS/X.25 line supervisor (IODSX).

LOGICAL DEVICE SHOULD BE IN THE RANGE 1 TO 255

This message appears if a line identifier not in the range of 1 to 255 has been specified. You will be prompted again for a logical device number (Step 2.0).

NO SUCH ENTRY IN LC TABLE

This message appears if a legal line identifier was specified, but there is no entry in the LC table for it. You will be prompted for another first-level command.

This procedure follows that of adding to the LC table. For each variable in the entry, the current value is printed, followed by a prompt for a new value. A carriage return maintains the current value. All other prompts, responses, and error messages are the same as for the ADD command.

USING X.25/X.29 CAPABILITIES

APPENDIX J

The DS/X.25 capability of DSN/DS is an enhancement to the DSN/DS subsystem that permits system-to-system communication via an X.25 Public Data Network (PDN). This enhancement is not a separate product, but rather it is an additional capability of DSN/DS to provide support for the X.25 protocol (an international standard for connection to PDNs). With this capability, it is now possible to connect to multiple remote systems over a single X.25 link to a PDN.

In conjunction with the DS/X.25 capability, DSN/DS also provides support for the X.29 protocol (an international standard for connecting asynchronous character-mode terminals to the system through an X.25 Public Data Network). With the X.25 and X.29 capabilities, you can run a session on a remote HP 3000 computer from your terminal by calling through a Packet Assembler/ Disassembler (PAD) on a Public Data Network. This feature allows you to benefit from the lower communication cost afforded by the use of a PDN rather than dialing a long distance call directly to the computer.

X.25 SYSTEM-TO-SYSTEM COMMUNICATION

As a DS/X.25 user, you can establish a maximum of eight remote sessions across one X.25 link to a Public Data Network (PDN). These eight remote sessions can be on eight different remote systems or all on one remote system, or any combination thereof, if the network data base has been set up accordingly.

To communicate with several different systems across one link, an entry is required in the Remote Node (RN) table for each system associated with one DS device (IODSX) ldev, with a unique remote address for each remote node. To establish multiple sessions on a remote node, there must be multiple unique remote node names in the RN table associated with one DS device ldev, with the same remote address for each node name.

WHEN TO USE A PAD

The following considerations may be helpful in determining which occasions are most efficient and/or economical for using a PAD on a Public Data Network instead of long-distance dialing:

- A PAD connection to the computer is advised when the cost of a long distance call directly to the computer would exceed the cost of the call to the PAD plus the packet charges for the amount of data sent. You should consult your PDN tariff charges to evaluate your particular situation.

- When a computer is configured to use PAD terminals, it is less costly to use the PAD terminals than to use direct dial-in ports for remote communication. The incremental cost of configuring an additional PAD terminal into the system is very small compared to the cost of having another dedicated dial-in port. Therefore, many systems may use PADs exclusively for terminal-to-host communication over long distances.

- The overall response time for interactive work with the computer may be longer for a connection through the PAD than for a direct dial-in connection. The major cause for the increased response time is the additional time needed by the PDN to process the data within the network. Also, terminal speeds are sometimes slower for terminals working through a PAD. The allowable terminal speeds are determined by the network. Refer to your network's user documentation for allowable speeds.

- The current CCITT X.3, X.28, and X.29 standards concerning the use of PADs do not provide for the use of terminals in block mode. Therefore, block mode programs and multicharacter data transfers for HP terminals (such as tape reads) will not work across PADs at the present time.

ESTABLISHING A REMOTE SESSION VIA PAD ON PDN

The means by which the user establishes a connection to the remote HP 3000 computer is determined by the network. While there is a standard interface between the user and the PAD (described in CCITT standard X.28), each network may be somewhat at variance with this standard. Therefore, each terminal user should consult the user documentation provided by the network for the appropriate commands to make the connection between the terminal and the remote computer.

Once the connection (virtual circuit) between the terminal and the computer is established, type a carriage return. The computer responds with a colon prompt (:). Then log on as if you were connected locally to the computer. This communication is designed to allow you to operate as if you were using a local terminal, with a few exceptions. (For example, block mode is not allowed, and tape and diskette transfers are not allowed.)

PAD PARAMETERS

The Packet Assembler/Disassembler (PAD) has a set of parameters that control its functions. The CCITT standard X.3 parameters 1 through 12 are currently supported on most networks. Some networks may also implement additional parameters which are not currently supported by DSN/DS. The X.3 parameters are listed below. Those marked with an asterisk indicate the values automatically set for the user when he establishes a session. The terminal user should not change these parameters.

Parameter Number	Function	Possible Values (Decimal)
1	PAD recall by escape from data transfer state	Possible .... 1* Not possible 0
2	Echo by PAD	Echo .... 1* No echo 0
3	Selection of data forwarding signal	No forwarding character .. 0 Carriage return .... 2* All nonprinting chars .. 126
4	Selection of idle timer delay	No idle timer used .... 0* Any number 1-255 will indicate delay in twentieths of a second

Parameter Number	Function	Possible Values (Decimal)
5	Ancillary device control	No use of X-ON (DC1) or X-OFF (DC3) 0
		Use of X-ON and X-OFF 1*
6	Suppression of PAD service signals	No service signals are sent to terminal 0
		Service signals sent 1*
7	PAD action on receipt of BREAK	Send host interrupt packet 1
		Reset 2
		Send host an indication of a break PAD message 4
		Escape from data transfer state 8
		Discard output to terminal 16
		Send interrupt packet, send indication of break, discard output 21*
		Do nothing 0
8	Discard output	Normal data delivery to terminal 0*
		Discard output to the terminal 1
9	Padding after carriage return	Set according to the data rate of terminal .. 0*
		Number of padding characters 1-7
10	Line folding	No folding 0*
		Number of characters per line 1-255
11	Binary speed	READ ONLY. SET BY PAD
12	Flow control of the PAD by terminal	No use of XON or XOFF for flow control 0
		Use of XON and XOFF 1*

TERMINAL RESTRICTIONS ON THE PUBLIC DATA NETWORK

There are certain restrictions on the use of a terminal running through the PAD. Some of these restrictions are:

- Block mode transfers will not work.
- Multicharacter transfers will not work. (See the manual for the particular terminal you are using for a definition of multicharacter transfers.)
- The terminal speed must be set manually; it cannot be set programmatically.
- Parity must be zero with no parity checking.
- Control sequences (such as Control-X and Control-Y, but not Control-S and Control-Q) must be forwarded with a carriage return.
- All data must be forwarded with a carriage return.
- The PAD terminal cannot be used as a console terminal.

WHICH TERMINALS TO USE

The following HP terminals will operate across a PAD (not in block/page mode):

HP 2382

HP 2621A, 2621B, 2621P, 2622, 2623, 2624A, 2624B, 2626A

HP 2635, 2640B, 2640B, 2642*, 2644*, 2645A*, 2647*, 2648*

(* Not including cassette tape or diskette transfers.)

The following HP desktop and personal computers will operate across a PAD in terminal emulation mode:

HP 85
HP 87
HP 125
HP 9826
HP 9836
HP 9835*
HP 9845*

(* These desktop computers should be set to work in character mode, not in line mode.)

CONFIGURING YOUR TERMINAL

Terminals operating across a PAD must be configured in the following manner:

1. The PAD does not use the HP standard ENQ/ACK handshake to allow the terminal to control the flow of data to it. At terminal speeds up to 2400 baud, the CRT terminals without attached or integrated printers need no flow control. However, printing terminals (terminals with attached printers or integrated printers) must be set to use XON/XOFF receive pacing. This setting allows the terminal to control the flow of data from the PAD, so as to avoid sending data characters to the terminal faster than the printer can print them. Attached printers must also be set to use the XON/XOFF flow control of the terminal to which they are connected.

2. Since terminals may have different line lengths, the initial configuration of the PAD does not specify a line length. The terminal user sets the line length before automatic wraparound for his particular terminal and application. Hewlett-Packard CRT terminals use the 80-character line length as the default. Printing terminals may use longer line lengths. Set the line length prior to logging on.

3. Set your terminal as follows:

   For HP 264x-series terminals, manually set the G and H straps in your terminal to the open position to inhibit the use of handshaking and DC2.

   For menu-driven terminals, set:

   ```
               INHIBIT HANDSHAKING (G) = YES
               INHIBIT DC2 (H) = YES
   ```

CHARACTERISTICS OF PAD

The terminal running across a PAD behaves slightly differently than a local terminal. You will notice the following behavioral differences:

- In a local environment, a line feed is sent immediately to the terminal when a read is completed. However, a terminal on the PAD receives the line feed at the beginning of the next write to the terminal. This method is more economical, since the line feed is sent as the first item in the next write data packet rather than its being sent in a packet by itself.
- Writes to the terminal are buffered until the next I/O request is processed. Thus it is possible that a write to the terminal will be delayed if a large amount of data processing is done between terminal I/O operations.
- Some network PADs will not echo escape sequences entered from the keyboard.

PROGRAM COMPATIBILITY WITH PAD

The following HP software products are compatible with the Packet Assembler/Disassembler (PAD) on a Public Data Network and will, therefore, operate correctly across a PAD:

All MPE commands

DSN/DS commands

All compilers and interpreters

EDIT/3000

TDP/3000 (not in screen mode)

FCOPY

MPE Utilities

MPE Segmenter

SORT/3000

There are other HP software products that will not work correctly across a PAD, however. Most of the programs in this category won't work because they operate in block mode, but others don't work because they programmatically change the terminal strap settings to settings that are incompatible with the PAD interface requirements for flow control. Until further notice, do NOT attempt to use the following HP software products across a PAD -- these programs can "hang" (or "lock up") the user's terminal:

RAPID/3000

DSG/3000

HPDRAW

V/3000, VPLUS/3000

HPSLATE

HPTOOLSET

HPWORD

HPEASYCHART

PSP/3000

IFS/2680

MM/3000

PM/3000

GA/3000

PLANT MAINTENANCE/3000

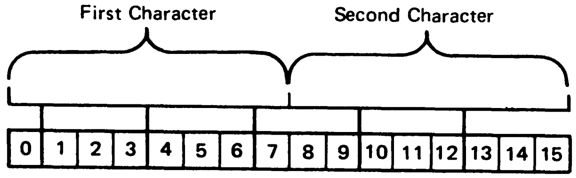
TDP/3000 (in screen mode)

Console operator commands (a PAD terminal cannot be the console).

ASCII CHARACTER SET

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
Δ	057000	000136
—	057400	000137
`	060000	000140
{	075400	000173
|	076000	000174
}	076400	000175
~	077000	000176
DEL	077400	000177



INDEX

INDEX (continued)

(NOT AN ORDER FORM. Copies of this manual may be ordered directly from a local sales office. Refer to the sales office listing on the following pages for addresses.)

READER COMMENT SHEET

HP Distributed Systems Network
DSN/DS HP 3000 to HP 3000
Reference Manual
32190-90001 Sep 1982

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

Is this manual technically accurate? Yes □ No □ (If no, explain under Comments, below.)

Are the concepts and wording easy to understand? Yes □ No □ (If no, explain under Comments, below.)

Is the format of this manual convenient in size, arrangement, and readability? Yes □ No □ (If no, explain or suggest improvements under Comments, below.)

Comments:

FROM:

Name

Company

Address

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO,CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Information Networks Division
Hewlett-Packard Company
19420 Homestead Road
Cupertino, California 95014

Attn: DataComm Documentation Manager

FOLD FOLD

SALES & SUPPORT OFFICES
Arranged Alphabetically by Country

Product Line Sales/Support Key

Key	Product Line
A	Analytical
CM	Components
C	Computer Systems Sales only
CH	Computer Systems Hardware Sales and Services
CS	Computer Systems Software Sales and Services
E	Electronic Instruments & Measurement Systems
M	Medical Products
MP	Medical Products Primary SRO
MS	Medical Products Secondary SRO
P	Personal Computation Products
*	Sales only for specific product line
**	Support only for specific product line

IMPORTANT: These symbols designate general product line capability. They do not insure sales or support availability for all products within a line, at all locations. Contact your local sales office for information regarding locations where HP support is available for specific products.

HP distributors are printed in italics.

ANGOLA

Telectra
Empresa Técnica de Equipamentos
Eléctricos, S.A.R.L.
R. Barbosa Rodrigues, 41-I DT.
Caixa Postal 6487
LUANDA
Tel: 35515,35516
E,M,P

ARGENTINA

Hewlett-Packard Argentina S.A.
Avenida Santa Fe 2035
Martinez 1640 **BUENOS AIRES**
Tel: 798-5735, 792-1293
Telex: 17595 BIONAR
Cable: HEWPACKARG
A,E,CH,CS,P

Biotron S.A.C.I.M. e I.
Av Paseo Colon 221, Piso 9
1399 ***BUENOS AIRES,***
Tel: 30-4846, 30-1851
Telex: 17595 BIONAR
M

Fate S.A. I.C.I.Electronica
Venezuela 1326
1095 ***BUENOS AIRES***
Tel: 37-9020, 37-9026/9
Telex: 9234 FATEN AR
P

AUSTRALIA

Adelaide, South Australia Office

Hewlett-Packard Australia Ltd.
153 Greenhill Road
PARKSIDE, S.A. 5063
Tel: 272-5911
Telex: 82536
Cable: HEWPARD Adelaide
A*,CH,CM,,E,MS,P

Brisbane, Queensland Office

Hewlett-Packard Australia Ltd.
49 Park Road
MILTON, Queensland 4064
Tel: 229-1544
Telex: 42133
Cable: HEWPARD Brisbane
A,CH,CM,E,M,P
Effective November 1, 1982:
10 Payne Road
THE GAP, Queensland 4061
Tel: 30-4133
Telex: 42133

Canberra, Australia Capital Territory Office

Hewlett-Packard Australia Ltd.
121 Wollongong Street
FYSHWICK, A.C.T. 2609
Tel: 80 4244
Telex: 62650
Cable: HEWPARD Canberra
CH,CM,E,P

Melbourne, Victoria Office

Hewlett-Packard Australia Ltd.
31-41 Joseph Street
BLACKBURN, Victoria 3130
Tel: 877 7777
Telex: 31-024
Cable: HEWPARD Melbourne
A,CH,CM,CS,E,MS,P

Perth, Western Australia Office

Hewlett-Packard Australia Ltd.
261 Stirling Highway
CLAREMONT, W.A. 6010
Tel: 383-2188
Telex: 93859
Cable: HEWPARD Perth
A,CH,CM,,E,MS,P

Sydney, New South Wales Office

Hewlett-Packard Australia Ltd.
17-23 Talavera Road
P.O. Box 308
NORTH RYDE, N.S.W. 2113
Tel: 887-1611
Telex: 21561
Cable: HEWPARD Sydney
A,CH,CM,CS,E,MS,P

AUSTRIA

Hewlett-Packard Ges.m.b.h.
Grottenhofstrasse 94
Verkaufsburo Graz
A-8052 **GRAZ**
Tel: 291-5-66
Telex: 32375
CH,E*

Hewlett-Packard Ges.m.b.h.
Stanglhofweg 5
A-4020 **LINZ**
Tel: 0732 51585
CH

Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O. Box 72
A-1222 **VIENNA**
Tel: (0222) 23-65-11-0
Telex: 134425 HEPA A
A,CH,CM,CS,E,MS,P

BAHRAIN

Green Salon
P.O. Box 557
BAHRAIN
Tel: 255503-255950
Telex: 84419
P

Wael Pharmacy
P.O. Box 648
BAHRAIN
Tel: 256123
Telex: 8550 WAEL BN
M, E

BELGIUM

Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 **BRUSSELS**
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CH,CM,CS,E,MP,P

BRAZIL

Hewlett-Packard do Brasil I.e.C. Ltda.
Alameda Rio Negro, 750
Alphaville 06400 **BARUERI SP**
Tel: (11) 421-1311
Telex: 01 133872 HPBR-BR
Cable: HEWPACK Sao Paulo
A,CH,CM,CS,E,M,P

Hewlett-Packard do Brasil I.e.C. Ltda.
Avenida Epitacio Pessoa, 4664
22471 **RIO DE JANEIRO**-RJ
Tel: (21) 286-0237
Telex: 021-21905 HPBR-BR
Cable: HEWPACK Rio de Janeiro
A,CH,CM,E,MS,P*

CANADA

Alberta

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher Street.S.E.
CALGARY, Alberta T2H 2H8
Tel: (403) 253-2713
A,CH,CM,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
11620A-168th Street
EDMONTON, Alberta T5M 3T9
Tel: (403) 452-3670
A,CH,CM,CS,E,MS,P*

British Columbia

Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
RICHMOND,
British Columbia V6X 2W7
Tel: (604) 270-2277
Telex: 610-922-5059
A,CH,CM,CS,E*,MS,P*

Manitoba

Hewlett-Packard (Canada) Ltd.
380-550 Century Street
WINNIPEG, Manitoba R3H 0Y1
Tel: (204) 786-6701
A,CH,CM,E,MS,P*

New Brunswick

Hewlett-Packard (Canada) Ltd.
37 Sheadiac Road
MONCTON, New Brunswick E2B 2VQ
Tel: (506) 855-2841
CH**

Nova Scotia

Hewlett-Packard (Canada) Ltd.
P.O. Box 931
900 Windmill Road
DARTMOUTH, Nova Scotia B2Y 3Z6
Tel: (902) 469-7820
CH,CM,CS,E*,MS,P*

Ontario

Hewlett-Packard (Canada) Ltd.
552 Newbold Street
LONDON, Ontario N6E 2S5
Tel: (519) 686-9181
A,CH,CM,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
A,CH,CM,CS,E,MP,P

Hewlett-Packard (Canada) Ltd.
2670 Queensview Dr.
OTTAWA, Ontario K2B 8K1
Tel: (613) 820-6483
A,CH,CM,CS,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
220 Yorkland Blvd., Unit #11
WILLOWDALE, Ontario M2J 1R5
Tel: (416) 499-9333
CH

Quebec

Hewlett-Packard (Canada) Ltd.
17500 South Service Road
Trans-Canada Highway
KIRKLAND, Quebec H9J 2M5
Tel: (514) 697-4232
A,CH,CM,CS,E,MP,P*

Hewlett-Packard (Canada) Ltd.
Les Galeries du Vallon
2323 Du Versont Nord
STE. FOY, Quebec G1N 4C2
Tel: (418) 687-4570
CH

CHILE

Jorge Calcagni y Cia. Ltda.
Arturo Burhle 065
Casilla 16475
SANTIAGO 9
Tel: 222-0222
Telex: Public Booth 440001
A,CM,E,M

Olympia (Chile) Ltda.
Av. Rodrigo de Araya 1045
Casilla 256-V
SANTIAGO 21
Tel: 2-25-50-44
Telex: 340-892 OLYMP CK
Cable: Olympiachile Santiagochile
CH,CS,P

CHINA, People's Republic of

China Hewlett-Packard Rep. Office
P.O. Box 418
1A Lane 2, Luchang St.
Beiwei Rd., Xuanwu District
BEIJING
Tel: 33-1947, 33-7426
Telex: 22601 CTSHP CN
Cable: 1920
A,CH,CM,CS,E,P

COLOMBIA

Instrumentación
H. A. Langebaek & Kier S.A.
Carrera 7 No. 48-75
Apartado Aereo 6287
BOGOTA *1, D.E.*
Tel: 287-8877
Telex: 44400 INST CO
Cable: AARIS Bogota
A,CM,E,M,PS,P

COSTA RICA

Cientifica Costarricense S.A.
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
SAN JOSÉ
Tel: 24-38-20, 24-08-19
Telex: 2367 GALGUR CR
CM,E,MS,P

CYPRUS

Telerexa Ltd.
P.O. Box 4809
14C Stassinos Avenue
NICOSIA
Tel: 62698
Telex: 2894 LEVIDO CY
E,M,P

DENMARK

Hewlett-Packard A/S
Datavej 52
DK-3460 **Birkerod**
Tel: (02) 81-66-40
Telex: 37409 hpas dk
A,CH,CM,CS,E,MS,P

Hewlett-Packard A/S
Navervej 1
DK-8600 **SILKEBORG**
Tel: (06) 82-71-66
Telex: 37409 hpas dk
CH,E

ECUADOR

CYEDE Cia. Ltda.
Avenida Eloy Alfaro 1749
Casilla 6423 CCI
QUITO
Tel: 450-975, 243-052
Telex: 2548 CYEDE ED
A,CM,E,P

Hospitalar S.A.
Robles 625
Casilla 3590
QUITO
Tel: 545-250, 545-122
Telex: 2485 HOSPTL ED
Cable: HOSPITALAR-Quito
M

EGYPT

International Engineering Associates
24 Hussein Hegazi Street
Kasr-el-Aini
CAIRO
Tel: 23829, 21641
Telex: IEA UN 93830
CH,CS,E,M

Informatic For Systems
22 Talaat Harb Street
CAIRO
Tel: 759006
Telex: 93938 FRANK UN
CH,CS,P

Egyptian International Office
for Foreign Trade
P.O.Box 2558
CAIRO
Tel: 650021
Telex: 93337 EGPOR
P

EL SALVADOR

IPESA de El Salvador S.A.
29 Avenida Norte 1216
SAN SALVADOR
Tel: 26-6858, 26-6868
Telex: Public Booth 20107
A,CH,CM,CS,E,P

FINLAND

Hewlett-Packard Oy
Revontulentie 7
SF-02100 **ESPOO** 10
Tel: (90) 455-0211
Telex: 121563 hewpa sf
A,CH,CM,CS,E,MS,P

Hewlett-Packard Oy
Aatoksenkatv 10-C

SALES & SUPPORT OFFICES
Arranged Alphabetically by Country

SF-40720-72 **JYVASKYLA**
Tel: (941) 216318
CH

Hewlett-Packard Oy
Kainvuntie 1-C
SF-90140-14 **OULU**
Tel: (981) 338785
CH

FRANCE

Hewlett-Packard France
Z.I. Mercure B
Rue Berthelot
F-13763 Les Milles Cedex
AIX-EN-PROVENCE
Tel: (42) 59-41-02
Telex: 410770F
A,CH,E,MS,P*

Hewlett-Packard France
Boite Postale No. 503
F-25026 **BESANCON**
28 Rue de la Republique
F-25000 **BESANCON**
Tel: (81) 83-16-22
CH,M

Hewlett-Packard France
Bureau de Vente de Lyon
Chemin des Mouilles
Boite Postale 162
F-69130 **ECULLY** Cédex
Tel: (7) 833-81-25
Telex: 310617F
A,CH,CS,E,MP

Hewlett-Packard France
Immeuble France Evry
Tour Lorraine
Boulevard de France
F-91035 **EVRY** Cédex
Tel: (6) 077-96-60
Telex: 692315F
E

Hewlett-Packard France
5th Avenue Raymond Chanas
F-38320 **EYBENS**
Tel: (76) 25-81-41
Telex: 980124 HP GRENOB EYBE
CH

Hewlett-Packard France
Centre d'Affaire Paris-Nord
Bâtiment Ampère 5 étage
Rue de la Commune de Paris
Boite Postale 300
F-93153 **LE BLANC MESNIL**
Tel: (01) 865-44-52
Telex: 211032F
CH,CS,E,MS

Hewlett-Packard France
Parc d'Activites Cadera
Quartier Jean Mermoz
Avenue du President JF Kennedy
F-33700 **MERIGNAC**
Tel: (56) 34-00-84
Telex: 550105F
CH,E,MS

Hewlett-Packard France
32 Rue Lothaire
F-57000 **METZ**
Tel: (8) 765-53-50
CH

Hewlett-Packard France
Immueble Les 3 B
Nouveau Chemin de la Garde
Z.A.C. de Bois Briand
F-44085 **NANTES** Cedex
Tel: (40) 50-32-22
CH**

Hewlett-Packard France
Zone Industrielle de Courtaboeuf
Avenue des Tropiques
F-91947 Les Ulis Cédex **ORSAY**
Tel: (6) 907-78-25
Telex: 600048F
A,CH,CM,CS,E,MP,P

Hewlett-Packard France
Paris Porte-Maillot
15, Avenue De L'Amiral Bruix
F-75782 **PARIS** 16
Tel: (1) 502-12-20
Telex: 613663F
CH,MS,P

Hewlett-Packard France
2 Allee de la Bourgonette
F-35100 **RENNES**
Tel: (99) 51-42-44
Telex: 740912F
CH,CM,E,MS,P*

Hewlett-Packard France
98 Avenue de Bretagne
F-76100 **ROUEN**
Tel: (35) 63-57-66 CH**,CS

Hewlett-Packard France
4 Rue Thomas Mann
Boite Postale 56
F-67200 **STRASBOURG**
Tel: (88) 28-56-46
Telex: 890141F
CH,E,MS,P*

Hewlett-Packard France
Pericentre de la Cépière
F-31081 **TOULOUSE** Cedex
Tel: (61) 40-11-12
Telex: 531639F
A,CH,CS,E,P*

Hewlett-Packard France
Immeuble Péricentre
F-59658 **VILLENEUVE D'ASCQ** Cedex
Tel: (20) 91-41-25
Telex: 160124F
CH,E,MS,P*

GERMAN FEDERAL REPUBLIC

Hewlett-Packard GmbH
Technisches Büro Berlin
Keithstrasse 2-4
D-1000 **BERLIN** 30
Tel: (030) 24-90-86
Telex: 018 3405 hpbln d
A,CH,E,M,P

Hewlett-Packard GmbH
Technisches Büro Böblingen
Herrenberger Strasse 110
D-7030 **BÖBLINGEN**
Tel: (07031) 667-1
Telex: bbn or
A,CH,CM,CS,E,MP,P

Hewlett-Packard GmbH
Technisches Büro Dusseldorf
Emanuel-Leutze-Strasse 1
D-4000 **DUSSELDORF**
Tel: (0211) 5971-1
Telex: 085/86 533 hpdd d
A,CH,CS,E,MS,P

Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Berner Strasse 117
Postfach 560 140
D-6000 **FRANKFURT** 56
Tel: (0611) 50-04-1
Telex: 04 13249 hpffm d
A,CH,CM,CS,E,MP,P

Hewlett-Packard GmbH
Technisches Büro Hamburg
Kapstadtring 5
D-2000 **HAMBURG** 60
Tel: (040) 63804-1
Telex: 021 63 032 hphh d
A,CH,CS,E,MS,P

Hewlett-Packard GmbH
Technisches Büro Hannover
Am Grossmarkt 6
D-3000 **HANNOVER** 91
Tel: (0511) 46-60-01
Telex: 092 3259
A,CH,CM,E,MS,P

Hewlett-Packard GmbH
Technisches Büro Mannheim
Rosslauer Weg 2-4
D-6800 **MANNHEIM**
Tel: (0621) 70050
Telex: 0462105
A,C,E

Hewlett-Packard GmbH
Technisches Büro Neu Ulm
Messerschmittstrasse 7
D-7910 **NEU ULM**
Tel: 0731-70241
Telex: 0712816 HP ULM-D
A,C,E*

Hewlett-Packard GmbH
Technisches Büro Nürnberg
Neumeyerstrasse 90
D-8500 **NÜRNBERG**
Tel: (0911) 52 20 83-87
Telex: 0623 860
CH,CM,E,MS,P

Hewlett-Packard GmbH
Technisches Büro München
Eschenstrasse 5
D-8028 **TAUFKIRCHEN**
Tel: (089) 6117-1
Telex: 0524985
A,CH,CM,E,MS,P

GREAT BRITAIN

Hewlett-Packard Ltd.
Trafalgar House
Navigation Road
ALTRINCHAM
Chesire WA14 1NU
Tel: (061) 928-6422
Telex: 668068
A,CH,CS,E,M

Hewlett-Packard Ltd.
Oakfield House, Oakfield Grove
Clifton
BRISTOL BS8 2BN, Avon
Tel: (027) 38606
Telex: 444302
CH,M,P

Hewlett-Packard Ltd.
(Pinewood)
Nine Mile Ride
EASTHAMPSTEAD
Wokingham
Berkshire, 3RG11 3LL
Tel: 3446 3100
Telex: 84-88-05
CH,CS,E

Hewlett-Packard Ltd.
Fourier House
257-263 High Street
LONDON COLNEY
Herts., AL2 1HA, St. Albans
Tel: (0727) 24400
Telex: 1-8952716
CH,CS,E

Hewlett-Packard Ltd
Tradax House, St. Mary's Walk
MAIDENHEAD
Berkshire, SL6 1ST
Tel: (0628) 39151
CH,CS,E,P

Hewlett-Packard Ltd.
Quadrangle
106-118 Station Road
REDHILL, Surrey
Tel: (0737) 68655
Telex: 947234 CH,CS,E

Hewlett-Packard Ltd.
Avon House
435 Stratford Road
SHIRLEY, Solihull
West Midlands B90 4BL
Tel: (021) 745 8800
Telex: 339105
CH

Hewlett-Packard Ltd.
West End House 41
High Street, West End
SOUTHAMPTON
Hampshire S03 3DQ
Tel: (703) 886767
Telex: 477138
CH

Hewlett-Packard Ltd.
King Street Lane
WINNERSH, Wokingham
Berkshire RG11 5AR
Tel: (0734) 784774
Telex: 847178
A,CH,E,M

GREECE

Kostas Karaynnis S.A.
8 Omirou Street
***ATHENS** 133*
Tel: 32 30 303, 32 37 371
Telex: 215962 RKAR GR
A,CH,CM,CS,E,M,P

PLAISIO S.A.
G. Gerardos
24 Stournara Street
ATHENS
Tel: 36-11-160
Telex: 221871
P

GUATEMALA

IPESA
Avenida Reforma 3-48, Zona 9
GUATEMALA CITY
Tel: 316627, 314786
Telex: 4192 TELTRO GU
A,CH,CM,CS,E,M,P

HONG KONG

Hewlett-Packard Hong Kong, Ltd.
G.P.O. Box 795
5th Floor, Sun Hung Kai Centre
30 Harbour Road
HONG KONG
Tel: 5-8323211
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG
E,CH,CS,P

CET Ltd.
1402 Tung Way Mansion
199-203 Hennessy Rd.
*Wanchia, **HONG KONG***
Tel: 5-729376
Telex: 85148 CET HX
CM

Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
HONG KONG
Tel: 5-455644
Telex: 74766 SCHMX HX
A,M

ICELAND

Elding Trading Company Inc.
Hafnarnvoli-Tryggvagotu
P.O. Box 895
IS-REYKJAVIK
Tel: 1-58-20, 1-63-03
M

INDIA

Blue Star Ltd.
Sabri Complex II Floor
24 Residency Rd.
***BANGALORE** 560 025*
Tel: 55660
Telex: 0845-430
Cable: BLUESTAR
A,CH,CM,CS,E

Blue Star Ltd.
Band Box House
Prabhadevi
***BOMBAY** 400 025*
Tel: 422-3101
Telex: 011-3751
Cable: BLUESTAR
A,M

Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
***BOMBAY** 400 025*
Tel: 422-6155
Telex: 011-4093
Cable: FROSTBLUE
A,CH,CM,CS,E,M

Blue Star Ltd.
Kalyan, 19 Vishwas Colony
*Alkapuri, **BORODA**, 390 005*
Tel: 65235
Cable: BLUE STAR
A

Blue Star Ltd.
7 Hare Street
***CALCUTTA** 700 001*
Tel: 12-01-31
Telex: 021-7655
Cable: BLUESTAR
A,M

Blue Star Ltd.
133 Kodambakkam High Road
***MADRAS** 600 034*
Tel: 82057
Telex: 041-379
Cable: BLUESTAR
A,M

Blue Star Ltd.
Bhandari House, 7th/8th Floors
91 Nehru Place
***NEW DELHI** 110 024*
Tel: 682547
Telex: 031-2463
Cable: BLUESTAR
A,CH,CM,CS,E,M

Blue Star Ltd.
15/16:C Wellesley Rd.
***PUNE** 411 011*
Tel: 22775
Cable: BLUE STAR
A

Blue Star Ltd.
2-2-47/1108 Bolarum Rd.
***SECUNDERABAD** 500 003*
Tel: 72057
Telex: 0155-459
Cable: BLUEFROST
A,E

Blue Star Ltd.
T.C. 7/603 Poornima
Maruthankuzhi
***TRIVANDRUM** 695 013*
Tel: 65799
Telex: 0884-259
Cable: BLUESTAR
E

INDONESIA

BERCA Indonesia P.T.
P.O.Box 496/JKT.
Jl. Abdul Muis 62
JAKARTA
Tel: 373009
Telex: 46748 BERSAL IA
Cable: BERSAL JAKARTA
P

BERCA Indonesia P.T.
Wisma Antara Bldg., 17th floor
JAKARTA
A,CS,E,M

BERCA Indonesia P.T.
P.O. Box 174/SBY.
Jl. Kutei No. 11
SURABAYA
Tel: 68172
Telex: 31146 BERSAL SB
Cable: BERSAL-SURABAYA
A,E,M,P*

IRAQ

Hewlett-Packard Trading S.A.
Service Operation
Al Mansoor City 9B/3/7
BAGHDAD
Tel: 551-49-73
Telex: 212-455 HEPAIRAQ IK
CH,CS

IRELAND

Hewlett-Packard Ireland Ltd.
82/83 Lower Leeson St.
DUBLIN 2
Tel: (1) 60 88 00
Telex: 30439
A,CH,CM,CS,E,M,P

Cardiac Services Ltd.
Kilmore Road
Artane
***DUBLIN** 5*
Tel: (01) 351820
Telex: 30439
M

ISRAEL

Eldan Electronic Instrument Ltd.
P.O. Box 1270
JERUSALEM 91000
16, Ohaliav St.
JERUSALEM 94467
Tel: 533 221, 553 242
Telex: 25231 AB/PAKRD IL
A

Electronics Engineering Division
Motorola Israel Ltd.
16 Kremenetski Street
P.O. Box 25016
TEL-AVIV 67899
Tel: 3-338973
Telex: 33569 Motil IL
Cable: BASTEL Tel-Aviv
CH,CM,CS,E,M,P

ITALY

Hewlett-Packard Italiana S.p.A.
Traversa 99C
Via Giulio Petroni, 19
I-70124 **BARI**
Tel: (080) 41-07-44
M

Hewlett-Packard Italiana S.p.A.
Via Martin Luther King, 38/111
I-40132 **BOLOGNA**
Tel: (051) 402394
Telex: 511630
CH,E,MS

Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43G/C
I-95126 **CATANIA**
Tel: (095) 37-10-87
Telex: 970291
C,P

Hewlett-Packard Italiana S.p.A.
Via G. Di Vittorio 9
I-20063 **CERNUSCO SUL NAVIGLIO**
Tel: (2) 903691
Telex: 334632
A,CH,CM,CS,E,MP,P

Hewlett-Packard Italiana S.p.A.
Via Nuova San Rocco a
Capodimonte, 62/A
I-80131 **NAPLES**
Tel: (081) 7413544
Telex: 710698
A,CH,E

Hewlett-Packard Italiana S.p.A.
Viale G. Modugno 33
I-16156 **GENOVA PEGLI**
Tel: (010) 68-37-07
Telex: 215238
E,C

Hewlett-Packard Italiana S.p.A.
Via Turazza 14
I-35100 **PADOVA**
Tel: (049) 664888
Telex: 430315
A,CH,E,MS

Hewlett-Packard Italiana S.p.A.
Viale C. Pavese 340
I-00144 **ROMA**
Tel: (06) 54831
Telex: 610514
A,CH,CM,CS,E,MS,P*

Hewlett-Packard Italiana S.p.A.
Corso Svizzera, 184
I-10149 **TORINO**
Tel: (011) 74 4044
Telex: 221079
CH,E

JAPAN

Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1-21-8, Asahi-cho
ATSUGI, Kanagawa 243
Tel: (0462) 28-0451
CM,C*,E

Yokogawa-Hewlett-Packard Ltd.
Towa Building
2-2-3, Kaigandori, Chuo-ku
KOBE, 650, Hyogo
Tel: (078) 392-4791
C,E

Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi Yasoji Bldg 4F
3-4 Chome Tsukuba
KUMAGAYA, Saitama 360
Tel: (0485) 24-6563
CH,CM,E

Yokogawa-Hewlett-Packard Ltd.
Asahi Shinbun Dai-ichi Seimei Bldg., 2F
4-7 Hanabata-cho
KUMAMOTO-SHI,860
Tel: (0963) 54-7311
CH,E

Yokogawa-Hewlett-Packard Ltd.
Shin Kyoto Center Bldg. 5F
614 Siokoji-cho
Nishiiruhigashi, Karasuma
Siokoji-dori, Shimogyo-ku
KYOTO 600
Tel: 075-343-0921
CH,E

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Building
1-4-73, San-no-maru
MITO, Ibaragi 310
Tel: (0292) 25-7470
CH,CM,E

Yokogawa-Hewlett-Packard Ltd.
Sumitomo Seimei Nagoya Bldg.
2-14-19, Meieki-Minami,
Nakamura-ku
NAGOYA, 450 Aichi
Tel: (052) 571-5171
CH,CM,CS,E,MS

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg., 4th Floor
5-4-20 Nishinakajima,
Yodogawa-ku
OSAKA, 532
Tel: (06) 304-6021
Telex: YHPOSA 523-3624
A,CH,CM,CS,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
1-27-15, Yabe,
SAGAMIHARA Kanagawa, 229
Tel: 0427 59-1311

Yokogawa-Hewlett-Packard Ltd.
Shinjuku Dai-ichi Seimei 6F
2-7-1, Nishi Shinjuku
Shinjuku-ku, **TOKYO** 160
Tel: 03-348-4611-5
CH,E

Yokogawa-Hewlett-Packard Ltd.
3-29-21 Takaido-Higashi
Suginami-ku **TOKYO** 168
Tel: (03) 331-6111
Telex: 232-2024 YHPTOK
A,CH,CM,CS,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
Daiichi Asano Building 4F
5-2-8, Oodori,
UTSUNOMIYA, 320
Tochigi
Tel: (0286) 25-7155
CH, CS, E

Yokogawa-Hewlett-Packard Ltd.
Yasudaseimei Yokohama
Nishiguchi Bldg.
3-30-4 Tsuruya-cho
Kanagawa-ku
YOKOHAMA, Kanagawa, 221
Tel: (045) 312-1252
CH,CM,E

JORDAN

Mouasher Cousins Company
P.O. Box 1387
AMMAN
Tel: 24907, 39907
Telex: 21456 SABCO JO
CH,E,M,P

KENYA

ADCOM Ltd., Inc., Kenya
P.O.Box 30070
NAIROBI
Tel: 331955
Telex: 22639
E,M

KOREA

Samsung Electronics Computer Division
76-561 Yeoksam-Dong
Kangnam-Ku
C.P.O. Box 2775
SEOUL
Tel: 555-7555, 555-5447
Telex: K27364 SAMSAN
A,CH,CM,CS,E,M,P

KUWAIT

Al-Khaldiya Trading & Contracting
P.O. Box 830 Safat
KUWAIT
Tel: 42-4910, 41-1726
Telex: 22481 Areeg kt
CH,E,M

Photo & Cine Equipment
P.O. Box 270 Safat
KUWAIT
Tel: 42-2846, 42-3801
Telex: 22247 Matin-KT
P

LEBANON

G.M. Dolmadjian
Achrafieh
P.O. Box 165.167
BEIRUT
Tel: 290293
*MP***

LUXEMBOURG

Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 **BRUSSELS**
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CH,CM,CS,E,MP,P

MALAYSIA

Hewlett-Packard Sales (Malaysia) Sdn. Bhd.
1st Floor, Bangunan British American
Jalan Semantan, Damansara Heights
KUALA LUMPUR 23-03
Tel: 943022
Telex: MA31011
A,CH,E,M,P*

Protel Engineering
Lot 319, Satok Road
P.O.Box 1917
*Kuching, **SARAWAK***
Tel: 53544
Telex: MA 70904 PROMAL
Cable: PROTELENG
A,E,M

MALTA

Philip Toledo Ltd.
Notabile Rd.
MRIEHEL
Tel: 447 47, 455 66
Telex: 649 Media MW
P

MEXICO

Hewlett-Packard Mexicana, S.A. de C.V.
Av. Periferico Sur No. 6501
Tepepan, Xochimilco
MEXICO D.F. 16020
Tel: 676-4600
Telex: 17-74-507 HEWPACK MEX
A,CH,CS,E,MS,P

Effective November 1, 1982:
Hewlett-Packard Mexicana, S.A. de C.V.
Ejercito Nacional #570
Colonia Granada
11560 **MEXICO**, D.F.
CH**

Hewlett-Packard Mexicana, S.A. de C.V.
Rio Volga 600
Pte. Colonia del Valle
MONTERREY, N.L.
Tel: 78-42-93, 78-42-40, 78-42-41
Telex: 038-2410 HPMTY ME
CH

Effective Nov. 1, 1982
Ave. Colonia del Valle #409
Col. del Valle
Municinio de garza garcia
MONTERREY, N.V.

ECISA
Taihe 229, Piso 10
*Polanco **MEXICO D.F.** 11570*
Tel: 250-5391
Telex: 17-72755 ECE ME
M

MOROCCO

Dolbeau
81 rue Karatchi
CASABLANCA
Tel: 3041-82, 3068-38
Telex: 23051, 22822
E

Gerep
2 rue d'Agadir
Boite Postale 156
CASABLANCA
Tel: 272093, 272095
Telex: 23 739
P

NETHERLANDS

Hewlett-Packard Nederland B.V.
Van Heuven Goedhartlaan 121
NL 1181KK **AMSTELVEEN**
P.O. Box 667
NL1180 AR **AMSTELVEEN**
Tel: (20) 47-20-21
Telex: 13 216
A,CH,CM,CS,E,MP,P

Hewlett-Packard Nederland B.V.
Bongerd 2
NL 2906VK **CAPPELLE, A/D** Ijessel
P.O. Box 41
NL2900 AA **CAPELLE**, Ijssel
Tel: (10) 51-64-44
Telex: 21261 HEPAC NL
A,CH,CS

NEW ZEALAND

Hewlett-Packard (N.Z.) Ltd.
169 Manukau Road
P.O. Box 26-189
Epsom, **AUCKLAND**
Tel: 687-159
Cable: HEWPACK Auckland
CH,CM,E,P*

Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
Kilbirnie, **WELLINGTON 3**
P.O. Box 9443
Courtenay Place, **WELLINGTON 3**
Tel: 877-199
Cable: HEWPACK Wellington
CH,CM,E,P

Northrop Instruments & Systems Ltd.
369 Khyber Pass Road
P.O. Box 8602
AUCKLAND
Tel: 794-091
Telex: 60605
A,M

Northrop Instruments & Systems Ltd.
110 Mandeville St.
P.O. Box 8388
CHRISTCHURCH
Tel: 486-928
Telex: 4203
A,M

Northrop Instruments & Systems Ltd.
Sturdee House
85-87 Ghuznee Street
P.O. Box 2406
WELLINGTON
Tel: 850-091
Telex: NZ 3380
A,M

NORTHERN IRELAND

Cardiac Services Company
95A Finaghy Road South
***BELFAST** BT 10 OBY*
Tel: (0232) 625-566
Telex: 747626
M

NORWAY

Hewlett-Packard Norge A/S
Folke Bernadottes vei 50
P.O. Box 3558
N-5033 **FYLLINGSDALEN** (Bergen)
Tel: (05) 16-55-40
Telex: 16621 hpnas n
CH,CS,E,MS

Hewlett-Packard Norge A/S
Österndalen 18
P.O. Box 34
N-1345 **ÖSTERºAS**
Tel: (02) 17-11-80
Telex: 16621 hpnas n
A,CH,CM,CS,E,M,P

OMAN

Khimjil Ramdas
P.O. Box 19
MUSCAT
Tel: 722225, 745601
Telex: 3289 BROKER MB MUSCAT
P

SALES & SUPPORT OFFICES

Arranged Alphabetically by Country

Suhail & Saud Bahwan
P.O. Box 169
MUSCAT
Tel: 734 201-3
Telex: 3274 BAHWAN MB

PAKISTAN

Mushko & Company Ltd.
1-B, Street 43
Sector F-8/1
ISLAMABAD
Tel: 26875
Cable: FEMUS Rawalpindi
A,E,M

Mushko & Company Ltd.
Oosman Chambers
Abdullah Haroon Road
KARACHI 0302
Tel: 511027, 512927
Telex: 2894 MUSKO PK
Cable: COOPERATOR Karachi
A,E,M,P*

PANAMA

Electrónico Balboa, S.A.
Calle Samuel Lewis, Ed. Alfa
Apartado 4929
PANAMA 5
Tel: 64-2700
Telex: 3483 ELECTRON PG
A,CM,E,M,P

Foto Internacional, S.A.
Colon Free Zone
Apartado 2068
COLON 3
Tel: 45-2333
Telex: 8626 IMPORT PG
P

PERU

Cía Electro Médica S.A.
Los Flamencos 145, San Isidro
Casilla 1030
LIMA 1
Tel: 41-4325, 41-3703
Telex: Pub. Booth 25306
A,CM,E,M,P

PHILIPPINES

The Online Advanced Systems Corporation
Rico House, Amorsolo Cor. Herrera Street
Legaspi Village, Makati
P.O. Box 1510
Metro MANILA
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE
A,CH,CS,E,M

Electronic Specialists and Proponents Inc.
690-B Epifanio de los Santos Avenue
Cubao, QUEZON CITY
P.O. Box 2649 Manila
Tel: 98-96-81, 98-96-82, 98-96-83
Telex: 40018, 42000 ITT GLOBE MACKAY BOOTH
P

PORTUGAL

Mundinter
Intercambio Mundial de Comércio S.a.r.l
P.O. Box 2761
Av. Antonio Augusto de Aguiar 138
P-LISBON
Tel: (19) 53-21-31, 53-21-37
Telex: 16691 munter p
M

Soquimica
Av. da Liberdade, 220-2
1298 LISBON Codex
Tel: 56 21 81/2/3
Telex: 13316 SABASA P

Telectra-Empresa Técnica de Equipmentos Eléctricos S.a.r.l.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
P-LISBON 1
Tel: (19) 68-60-72
Telex: 12598
CH,CS,E,P

PUERTO RICO

Hewlett-Packard Puerto Rico
P.O. Box 4407
CAROLINA, Puerto Rico 00628
Calle 272 Edificio 203
Urb. Country Club
RIO PIEDRAS, Puerto Rico 00924
Tel: (809) 762-7255
A,CH,CS

QATAR

Nasser Trading & Contracting
P.O. Box 1563
DOHA
Tel: 22170, 23539
Telex: 4439 NASSER DH
M

Computearbia
P.O. Box 2750
DOHA
Tel: 883555
Telex: 4806 CHPARB
P

Eastern Technical Services
P.O. Box 4747
DOHA
Tel: 329 993
Telex: 4156 EASTEC DH

SAUDI ARABIA

Modern Electronic Establishment
Hewlett-Packard Division
P.O. Box 281
Thuobah
AL-KHOBAR
Tel: 864-46 78
Telex: 671 106 HPMEEK SJ
Cable: ELECTA AL-KHOBAR
CH,CS,E,M,P

Modern Electronic Establishment
Hewlett-Packard Division
P.O. Box 1228
Redec Plaza, 6th Floor
JEDDAH
Tel: 644 38 48
Telex: 402712 FARNAS SJ
Cable: ELECTA JEDDAH
CH,CS,E,M,P

Modern Electronic Establishment
Hewlett Packard Division
P.O. Box 2728
RIYADH
Tel: 491-97 15, 491-63 87
Telex: 202049 MEERYD SJ
CH,CS,E,M,P

SCOTLAND

Hewlett-Packard Ltd.
Royal Bank Buildings
Swan Street
BRECHIN, Angus, Scotland
Tel: (03562) 3101-2
CH

Hewlett-Packard Ltd.
SOUTH QUEENSFERRY
West Lothian, EH30 9GT
GB-Scotland
Tel: (031) 3311188
Telex: 72682
A,CH,CM,CS,E,M

SINGAPORE

Hewlett-Packard Singapore (Pty.) Ltd.
P.O. Box 58 Alexandra Post Office
SINGAPORE, 9115
6th Floor, Inchcape House
450-452 Alexandra Road
SINGAPORE 0511
Tel: 631788
Telex: HPSGSO RS 34209
Cable: HEWPACK, Singapore
A,CH,CS,E,MS,P

Dynamar International Ltd.
Unit 05-11 Block 6
Kolam Ayer Industrial Estate
SINGAPORE 1334
Tel: 747-6188
Telex: RS 26283
CM

SOUTH AFRICA

Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 120
Howard Place
Pine Park Center, Forest Drive, Pinelands
CAPE PROVINCE 7405
Tel: 53-7954
Telex: 57-20006
A,CH,CM,E,MS,P

Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 37099
92 Overport Drive
DURBAN 4067
Tel: 28-4178, 28-4179, 28-4110
Telex: 6-22954
CH,CM

Hewlett-Packard So Africa (Pty.) Ltd.
6 Linton Arcade
511 Cape Road
Linton Grange
PORT ELIZABETH 6001
Tel: 041-302148
CH

Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 33345
Glenstantia 0010 TRANSVAAL
1st Floor East
Constantia Park Ridge Shopping Centre
Constantia Park
PRETORIA
Tel: 982043
Telex: 32163
CH,E

Hewlett-Packard So Africa (Pty.) Ltd.
Private Bag Wendywood
SANDTON 2144
Tel: 802-5111, 802-5125
Telex: 4-20877
Cable: HEWPACK Johannesburg
A,CH,CM,CS,E,MS,P

SPAIN

Hewlett-Packard Española S.A.
c/Entenza, 321
E-BARCELONA 29
Tel: (3) 322-24-51, 321-73-54
Telex: 52603 hpbee
A,CH,CS,E,MS,P

Hewlett-Packard Española S.A.
c/San Vicente S/N
Edificio Albia II,7 B
E-BILBAO 1
Tel: (4) 23-8306, (4) 23-8206
A,CH,E,MS

Hewlett-Packard Española S.A.
Calle Jerez 3
E-MADRID 16
Tel: (1) 458-2600
Telex: 23515 hpe
A,CM,E

Hewlett-Packard Española S.A.
c/o Costa Brava 13
Colonia Mirasierra
E-MADRID 34
Tel: (1) 734-8061, (1) 734-1162
CH,CS,M

Hewlett-Packard Española S.A.
Av Ramón y Cajal 1-9
Edificio Sevilla 1,
E-SEVILLA 5
Tel: 64-44-54, 64-44-58
Telex: 72933
A,CS,MS,P

Hewlett-Packard Española S.A.
C/Ramon Gordillo, 1 (Entlo.3)
E-VALENCIA 10
Tel: 361-1354, 361-1358
CH,P

SWEDEN

Hewlett-Packard Sverige AB
Sunnanvagen 14K
S-22226 LUND
Tel: (046) 13-69-79
Telex: (854) 17886 (via SPÅNGA office)
CH

Hewlett-Packard Sverige AB
Vastra Vintergatan 9
S-70344 OREBRO
Tel: (19) 10-48-80
Telex: (854) 17886 (via SPÅNGA office)
CH

Hewlett-Packard Sverige AB
Skalholtsgatan 9, Kista
Box 19
S-16393 SPÅNGA
Tel: (08) 750-2000
Telex: (854) 17886
A,CH,CM,CS,E,MS,P

Hewlett-Packard Sverige AB
Frötallisgatan 30
S-42132 VÄSTRA-FRÖLUNDA
Tel: (031) 49-09-50
Telex: (854) 17886 (via SPÅNGA office)
CH,E,P

SWITZERLAND

Hewlett-Packard (Schweiz) AG
Clarastrasse 12
CH-4058 BASLE
Tel: (61) 33-59-20
A

Hewlett-Packard (Schweiz) AG
Bahnhoheweg 44
CH-3018 BERN
Tel: (031) 56-24-22
CH

Hewlett-Packard (Schweiz) AG
47 Avenue Blanc
CH-1202 GENEVA
Tel: (022) 32-48-00
CH,CM,CS

Hewlett-Packard (Schweiz) AG
19 Chemin Château Bloc
CH-1219 LE LIGNON-Geneva
Tel: (022) 96-03-22
Telex: 27333 hpag ch
Cable: HEWPACKAG Geneva
A,E,MS,P

Hewlett-Packard (Schweiz) AG
Allmend 2
CH-8967 WIDEN
Tel: (57) 31 21 11
Telex: 53933 hpag ch
Cable: HPAG CH
A,CH,CM,CS,E,MS,P

SYRIA

General Electronic Inc.
Nuri Basha
P.O. Box 5781
DAMASCUS
Tel: 33-24-87
Telex: 11216 ITIKAL SY
Cable: ELECTROBOR DAMASCUS
E

Middle East Electronics
Place Azmé
Boite Postale 2308
DAMASCUS
Tel: 334592
Telex: 11304 SATACO SY
M,P

TAIWAN

Hewlett-Packard Far East Ltd.
Kaohsiung Office
2/F 68-2, Chung Cheng 3rd Road
KAOHSIUNG
Tel: 241-2318, 261-3253
CH,CS,E

Hewlett-Packard Far East Ltd.
Taiwan Branch
5th Floor
205 Tun Hwa North Road
TAIPEI
Tel:(02) 751-0404
Cable:HEWPACK Taipei
A,CH,CM,CS,E,M,P

Ing Lih Trading Co.
3rd Floor, 7 Jen-Ai Road, Sec. 2
TAIPEI 100
Tel: (02) 3948191
Cable: INGLIH TAIPEI
A

THAILAND

Unimesa
30 Patpong Ave., Suriwong
BANGKOK 5
Tel: 234 091, 234 092
Telex: 84439 Simonco TH
Cable: UNIMESA Bangkok
A,CH,CS,E,M

Bangkok Business Equipment Ltd.
5/5-6 Dejo Road
BANGKOK
Tel: 234-8670, 234-8671
Telex: 87669-BEQUIPT TH
Cable: BUSIQUIPT Bangkok
P

TRINIDAD & TOBAGO

Caribbean Telecoms Ltd.
50/A Jerningham Avenue
P.O. Box 732
PORT-OF-SPAIN
Tel: 62-44213, 62-44214
Telex: 235,272 HUGCO WG
A,CM,E,M,P

TUNISIA

Tunisie Electronique
31 Avenue de la Liberte
TUNIS
Tel: 280-144
E,P

Corema
1 ter. Av. de Carthage
TUNIS
Tel: 253-821
Telex: 12319 CABAM TN
M

TURKEY

Teknim Company Ltd.
Iran Caddesi No. 7
Kavaklidere, ANKARA
Tel: 275800
Telex: 42155 TKNM TR
E

E.M.A.
Medina Eldem Sokak No.41/6
Yuksel Caddesi
ANKARA
Tel: 175 622
M

UNITED ARAB EMIRATES

Emitac Ltd.
P.O. Box 1641
SHARJAH
Tel: 354121, 354123
Telex: 68136 Emitac Sh
CH,CS,E,M,P

UNITED KINGDOM

see: GREAT BRITAIN

NORTHERN IRELAND

SCOTLAND

UNITED STATES

Alabama

Hewlett-Packard Co.
700 Century Park South
Suite 128
BIRMINGHAM, AL 35226
Tel: (205) 822-6802
CH,MP

Hewlett-Packard Co.
P.O. Box 4207
8290 Whitesburg Drive, S.E.
HUNTSVILLE, AL 35802
Tel: (205) 881-4591
CH,CM,CS,E,M*

Alaska

Hewlett-Packard Co.
1577 "C" Street, Suite 252
ANCHORAGE, AK 99501
Tel: (907) 276-5709
CH*

Arizona

Hewlett-Packard Co.
2336 East Magnolia Street
PHOENIX, AZ 85034
Tel: (602) 273-8000
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
2424 East Aragon Road
TUCSON, AZ 85706
Tel: (602) 889-4631
CH,E,MS**

Arkansas

Hewlett-Packard Co.
P.O. Box 5646
Brady Station
LITTLE ROCK, AR 72215
111 N. Filmore
LITTLE ROCK, AR 72205
Tel: (501) 664-8773, 376-1844
MS

California

Hewlett-Packard Co.
99 South Hill Dr.
BRISBANE, CA 94005
Tel: (415) 330-2500
CH,CS

Hewlett-Packard Co.
7621 Canoga Avenue
CANOGA PARK, CA 91304
Tel: (213) 702-8300
A,CH,CS,E,P

Hewlett-Packard Co.
5060 Clinton Avenue
FRESNO, CA 93727
Tel: (209) 252-9652
MS

Hewlett-Packard Co.
P.O. Box 4230
1430 East Orangethorpe
FULLERTON, CA 92631
Tel: (714) 870-1000
CH,CM,CS,E,MP

Hewlett-Packard Co.
320 S. Kellogg, Suite B
GOLETA, CA 93117
Tel: (805) 967-3405
CH

Hewlett-Packard Co.
5400 W. Rosecrans Boulevard
LAWNDALE, CA 90260
P.O. Box 92105
LOS ANGELES, CA 90009
Tel: (213) 970-7500
Telex: 910-325-6608
CH,CM,CS,MP

Hewlett-Packard Co.
3200 Hillview Avenue
PALO ALTO, CA 94304
Tel: (415) 857-8000
CH,CS,E

Hewlett-Packard Co.
P.O. Box 15976 (95813)
4244 So. Market Court, Suite A
SACRAMENTO, CA 95834
Tel: (916) 929-7222
A*,CH,CS,E,MS

Hewlett-Packard Co.
9606 Aero Drive
P.O. Box 23333
SAN DIEGO, CA 92123
Tel: (714) 279-3200
CH,CM,CS,E,MP

Hewlett-Packard Co.
2305 Camino Ramon "C"
SAN RAMON, CA 94583
Tel: (415) 838-5900
CH,CS

Hewlett-Packard Co.
P.O. Box 4230
Fullerton, CA 92631
363 Brookhollow Drive
SANTA ANA, CA 92705
Tel: (714) 641-0977
A,CH,CM,CS,MP

Hewlett-Packard Co.
Suite A
5553 Hollister
SANTA BARBARA, CA 93111
Tel: (805) 964-3390

Hewlett-Packard Co.
3003 Scott Boulevard
SANTA CLARA, CA 95050
Tel: (408) 988-7000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
5703 Corsa Avenue
WESTLAKE VILLAGE, CA 91362
Tel: (213) 706-6800
E*,CH*,CS*

Colorado

Hewlett-Packard Co.
24 Inverness Place, East
ENGLEWOOD, CO 80112
Tel: (303) 771-3455
Telex: 910-935-0785
A,CH,CM,CS,E,MS

Connecticut

Hewlett-Packard Co.
47 Barnes Industrial Road South
P.O. Box 5007
WALLINGFORD, CT 06492
Tel: (203) 265-7801
A,CH,CM,CS,E,MS

Florida

Hewlett-Packard Co.
P.O. Box 24210 (33307)
2901 N.W. 62nd Street
FORT LAUDERDALE, FL 33307
Tel: (305) 973-2600
CH,CS,E,MP

Hewlett-Packard Co.
4080 Woodcock Drive, #132
Brownett Building
JACKSONVILLE, FL 32207
Tel: (904) 398-0663
C*,E*,MS**

Hewlett-Packard Co.
1101 W. Hibiscus Ave., Suite E210
MELBOURNE, FL 32901
Tel: (305) 729-0704
E*

Hewlett-Packard Co.
P.O. Box 13910 (32859)
6177 Lake Ellenor Drive
ORLANDO, FL 32809
Tel: (305) 859-2900
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
6425 N. Pensacola Blvd.
Suite 4, Building 1
P.O. Box 12826
PENSACOLA, FL 32575
Tel: (904) 476-8422
A,MS

Hewlett-Packard Co.
5750B N. Hoover Blvd., Suite 123
TAMPA, FL 33614
Tel: (813) 884-3282
A*,CH,CM,CS,E*,M*

Georgia

Hewlett-Packard Co.
P.O. Box 105005
ATLANTA, GA 30348
2000 South Park Place
ATLANTA, GA 30339
Tel: (404) 955-1500
Telex: 810-766-4890
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
P.O. Box 816 (80903)
2531 Center West Parkway
Suite 110
AUGUSTA, GA 30904
Tel: (404) 736-0592
MS

Hewlett-Packard Co.
200-E Montgomery Cross Rds.
SAVANNAH, GA 31401
Tel:(912) 925-5358
CH**

Hewlett-Packard Co.
P.O. Box 2103
WARNER ROBINS, GA 31099
1172 N. Davis Drive
WARNER ROBINS, GA 31093
Tel: (912) 923-8831
E

Hawaii

Hewlett-Packard Co.
Kawaiahao Plaza, Suite 190
567 South King Street
HONOLULU, HI 96813
Tel: (808) 526-1555
A,CH,E,MS

Illinois

Hewlett-Packard Co.
211 Prospect Road, Suite C
BLOOMINGTON, IL 61701
Tel: (309) 662-9411
CH,MS**

Hewlett-Packard Co.
1100 31st Street, Suite 100
DOWNERS GROVE, IL 60515
Tel: (312) 960-5760
CH,CS

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800
A,CH,CM,CS,E,MP

Indiana

Hewlett-Packard Co.
P.O. Box 50807
7301 No. Shadeland Avenue
INDIANAPOLIS, IN 46250
Tel: (317) 842-1000
A,CH,CM,CS,E,MS

Iowa

Hewlett-Packard Co.
1776 22nd Street, Suite 1
WEST DES MOINES, IA 50265
Tel: (515) 224-1435
CH,MS**

Hewlett-Packard Co.
2415 Heinz Road
IOWA CITY, IA 52240
Tel: (319) 351-1020
CH,E*,MS

Kansas

Hewlett-Packard Co.
1644 S. Rock Road
WICHITA, KA 67207
Tel: (316) 684-8491
CH

Kentucky

Hewlett-Packard Co.
10300 Linn Station Road
Suite 100
LOUISVILLE, KY 40223
Tel: (502) 426-0100
A,CH,CS,MS

Louisiana

Hewlett-Packard Co.
8126 Calais Bldg.
BATON ROUGE, LA 70806
Tel: (504) 467-4100
A**,CH**

Hewlett-Packard Co.
P.O. Box 1449
KENNER, LA 70062
160 James Drive East
DESTAHAN, LA 70047
Tel: (504) 467-4100
A,CH,CS,E,MS

Maryland

Hewlett-Packard Co.
7121 Standard Drive
HANOVER, MD 21076
Tel: (301) 796-7700
Telex: 710-862-1943
Eff. Dec. 1, 1982
3701 Koppers St.
BALTIMORE, MD 21227
Tel: (301) 644-5800
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
2 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 948-6370
A,CH,CM,CS,E,MP

Massachusetts

Hewlett-Packard Co.
32 Hartwell Avenue
LEXINGTON, MA 02173
Tel: (617) 861-8960
A,CH,CM,CS,E,MP

Michigan

Hewlett-Packard Co.
23855 Research Drive
FARMINGTON HILLS, MI 48024
Tel: (313) 476-6400
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
4326 Cascade Road S.E.
GRAND RAPIDS, MI 49506
Tel: (616) 957-1970
CH,CS,MS

Hewlett-Packard Co.
1771 W. Big Beaver Road
TROY, MI 48084
Tel: (313) 643-6474
CH,CS

Minnesota

Hewlett-Packard Co.
2025 W. Larpenteur Ave.
ST. PAUL, MN 55113
Tel: (612) 644-1100
A,CH,CM,CS,E,MP

Mississippi

Hewlett-Packard Co.
P.O. Box 5028
1675 Lakeland Drive
JACKSON, MS 39216
Tel: (601) 982-9363
MS

Missouri

Hewlett-Packard Co.
11131 Colorado Avenue
KANSAS CITY, MO 64137
Tel: (816) 763-8000
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
P.O. Box 27307
1024 Executive Parkway
ST. LOUIS, MO 63141
Tel: (314) 878-0200
A,CH,CS,E,MP
Effective September 1982:
13001 Hollenberg Drive
BRIDGETON, MO 63044

Nebraska

Hewlett-Packard
7101 Mercy Road
Suite 101, IBX Building
OMAHA, NE 68106
Tel: (402) 392-0948
CM,MS

Nevada

Hewlett-Packard Co.
Suite D-130
5030 Paradise Blvd.
LAS VEGAS, NV 89119
Tel: (702) 736-6610
MS**

New Jersey

Hewlett-Packard Co.
W120 Century Road
PARAMUS, NJ 07652
Tel: (201) 265-5000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
60 New England Av. West
PISCATAWAY, NJ 08854
Tel: (201) 981-1199
A,CH,CM,CS,E

New Mexico

Hewlett-Packard Co.
P.O. Box 11634
ALBUQUERQUE, NM 87112
11300 Lomas Blvd.,N.E.
ALBUQUERQUE, NM 87123
Tel: (505) 292-1330
Telex: 910-989-1185
CH,CS,E,MS

New York

Hewlett-Packard Co.
5 Computer Drive South
ALBANY, NY 12205
Tel: (518) 458-1550
Telex: 710-444-4691
A,CH,E,MS

Hewlett-Packard Co.
P.O. Box 297
9600 Main Street
CLARENCE, NY 14031
Tel: (716) 759-8621
Telex: 710-523-1893
CH

Hewlett-Packard Co.
200 Cross Keys Office
FAIRPORT, NY 14450
Tel: (716) 223-9950
Telex: 510-253-0092
CH,CM,CS,E,MS

Hewlett-Packard Co.
7641 Henry Clay Blvd.
LIVERPOOL, NY 13088
Tel: (315) 451-1820
A,CH,CM,E,MS

Hewlett-Packard Co.
No. 1 Pennsylvania Plaza
55th Floor
34th Street & 8th Avenue
NEW YORK, NY 10119
Tel: (212) 971-0800
CH,CS,E*,M*

SALES & SUPPORT OFFICES

Arranged Alphabetically by Country

Hewlett-Packard Co.
250 Westchester Avenue
WHITE PLAINS, NY 10604
CM,CH,CS,E

Hewlett-Packard Co.
3 Crossways Park West
WOODBURY, NY 11797
Tel: (516) 921-0300
Telex: 510-221-2183
A,CH,CM,CS,E,MS

North Carolina

Hewlett-Packard Co.
4915 Water's Edge Drive
Suite 160
RALEIGH, NC 27606
Tel: (919) 851-3021
C,M

Hewlett-Packard Co.
P.O. Box 26500
5605 Roanne Way
GREENSBORO, NC 27450
Tel: (919) 852-1800
A,CH,CM,CS,E,MS

Ohio

Hewlett-Packard Co.
9920 Carver Road
CINCINNATI, OH 45242
Tel: (513) 891-9870
CH,CS,MS

Hewlett-Packard Co.
16500 Sprague Road
CLEVELAND, OH 44130
Tel: (216) 243-7300
Telex: 810-423-9430
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
962 Crupper Ave.
COLUMBUS, OH 43229
Tel: (614) 436-1041
CH,CM,CS,E*

Hewlett-Packard Co.
P.O. Box 280
330 Progress Rd.
DAYTON, OH 45449
Tel: (513) 859-8202
A,CH,CM,E*,MS

Oklahoma

Hewlett-Packard Co.
P.O. Box 32008
Oklahoma City, OK 73123
1503 W. Gore Blvd., Suite #2
LAWTON, OK 73505
Tel: (405) 248-4248
C

Hewlett-Packard Co.
P.O. Box 32008
OKLAHOMA CITY, OK 73123
304 N. Meridian Avenue, Suite A
OKLAHOMA CITY, OK 73107
Tel: (405) 946-9499
A*,CH,E*,MS

Hewlett-Packard Co.
Suite 121
9920 E. 42nd Street
TULSA, OK 74145
Tel: (918) 665-3300
A**,CH,CS,M*

Oregon

Hewlett-Packard Co.
1500 Valley River Drive
Suite 330
EUGENE, OR 97401
Tel: (503) 683-8075
C

Hewlett-Packard Co.
9255 S. W. Pioneer Court
WILSONVILLE, OR 97070
Tel: (503) 682-8000
A,CH,CS,E*,MS

Pennsylvania

Hewlett-Packard Co.
1021 8th Avenue
King of Prussia Industrial Park
KING OF PRUSSIA, PA 19406
Tel: (215) 265-7000
Telex: 510-660-2670
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
111 Zeta Drive
PITTSBURGH, PA 15238
Tel: (412) 782-0400
A,CH,CS,E,MP

South Carolina

Hewlett-Packard Co.
P.O. Box 21708
Brookside Park, Suite 122
1 Harbison Way
COLUMBIA, SC 29210
Tel: (803) 732-0400
CH,E,MS

Hewlett-Packard Co.
Koger Executive Center
Chesterfield Bldg., Suite 124
GREENVILLE, SC 29615
Tel: (803) 748-5601
C

Tennessee

Hewlett-Packard Co.
P.O. Box 22490
224 Peters Road
Suite 102
KNOXVILLE, TN 37922
Tel: (615) 691-2371
A*,CH,MS

Hewlett-Packard Co.
3070 Directors Row
MEMPHIS, TN 38131
Tel: (901) 346-8370
A,CH,MS

Hewlett-Packard Co.
230 Great Circle Road
Suite 216
NASHVILLE, TN 32228
Tel: (615) 255-1271
MS**

Texas

Hewlett-Packard Co.
Suite 310W
7800 Shoalcreek Blvd.
AUSTIN, TX 78757
Tel: (512) 459-3143
E

Hewlett-Packard Co.
Suite C-110
4171 North Mesa
EL PASO, TX 79902
Tel: (915) 533-3555, 533-4489
CH,E*,MS**

Hewlett-Packard Co.
5020 Mark IV Parkway
FORT WORTH, TX 76106
Tel: (817) 625-6361
CH,CS*

Hewlett-Packard Co.
P.O. Box 42816
HOUSTON, TX 77042
10535 Harwin Street
HOUSTON, TX 77036
Tel: (713) 776-6400
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
3309 67th Street
Suite 24
LUBBOCK, TX 79413
Tel: (806) 799-4472
M

Hewlett-Packard Co.
417 Nolana Gardens, Suite C
P.O. Box 2256
McALLEN, TX 78501
Tel: (512) 781-3226
CH,CS

Hewlett-Packard Co.
P.O. Box 1270
RICHARDSON, TX 75080
930 E. Campbell Rd.
RICHARDSON, TX 75081
Tel: (214) 231-6101
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
P.O. Box 32993
SAN ANTONIO, TX 78216
1020 Central Parkway South
SAN ANTONIO, TX 78232
Tel: (512) 494-9336
CH,CS,E,MS

Utah

Hewlett-Packard Co.
P.O. Box 26626
3530 W. 2100 South
SALT LAKE CITY, UT 84119
Tel: (801) 974-1700
A,CH,CS,E,MS

Virginia

Hewlett-Packard Co.
P.O. Box 9669
2914 Hungary Spring Road
RICHMOND, VA 23228
Tel: (804) 285-3431
A,CH,CS,E,MS

Hewlett-Packard Co.
3106 Peters Creek Road, N.W.
ROANOKE, VA 24019
Tel: (703) 563-2205
CH,E**

Hewlett-Packard Co.
5700 Thurston Avenue
Suite 111
VIRGINIA BEACH, VA 23455
Tel: (804) 460-2471
CH,MS

Washington

Hewlett-Packard Co.
15815 S.E. 37th Street
BELLEVUE, WA 98006
Tel: (206) 643-4000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
Suite A
708 North Argonne Road
SPOKANE, WA 99206
Tel: (509) 922-7000
CH,CS

West Virginia

Hewlett-Packard Co.
4604 MacCorkle Ave., S.E.
CHARLESTON, WV 25304-4297
Tel: (304) 925-0492
A,MS

Wisconsin

Hewlett-Packard Co.
150 S. Sunny Slope Road
BROOKFIELD, WI 53005
Tel: (414) 784-8800
A,CH,CS,E*,MP

URUGUAY

Pablo Ferrando S.A.C. e L.
Avenida Italia 2877
Casilla de Correo 370
MONTEVIDEO
Tel: 80-2586
Telex: Public Booth 901
A,CM,E,M

Guillermo Kraft del Uruguay S.A.
Av. Lib. Brig. Gral. Lavalleja 2083
MONTEVIDEO
Tel: 234588, 234808, 208830
Telex: 22030 ACTOUR UY
P

VENEZUELA

Hewlett-Packard de Venezuela C.A.
3A Transversal Los Ruices Norte
Edificio Segre
Apartado 50933
CARACAS 1071
Tel: 239-4133
Telex: 25146 HEWPACK
A,CH,CS,E,MS,P

Colimodio S.A.
Este 2 - Sur 21 No. 148
Apartado 1053
***CARACAS** 1010*
Tel: 571-3511
Telex: 21529 COLMODIO
M

ZIMBABWE

Field Technical Sales
45 Kelvin Road, North
P.B. 3458
SALISBURY
Tel: 705 231
Telex: 4-122 RH
C,E,M,P

Headquarters offices

If there is no sales office listed for your area, contact one of these headquarters offices.

NORTH/CENTRAL AFRICA

Hewlett-Packard S.A.
7 Rue du Bois-du-Lan
CH-1217 **MEYRIN 2**, Switzerland
Tel: (022) 98-96-51
Telex: 27835 hpse
Cable: HEWPACKSA Geneve

ASIA

Hewlett-Packard Asia Ltd.
6th Floor, Sun Hung Kai Center
30 Harbor Rd.
G.P.O. Box 795
HONG KONG
Tel: 5-832 3211
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG

CANADA

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Telex: 610-492-4246

EASTERN EUROPE

Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O.Box 72
A-1222 **VIENNA**, Austria
Tel: (222) 2365110
Telex: 1 3 4425 HEPA A

NORTHERN EUROPE

Hewlett-Packard S.A.
Uilenstede 475
NL-1183 AG **AMSTELVEEN**
The Netherlands
P.O.Box 999
NL-1180 AZ **AMSTELVEEN**
The Netherlands
Tel: 20 437771

OTHER EUROPE

Hewlett-Packard S.A.
7 Rue du Bois-du-Lan
CH-1217 **MEYRIN 2**, Switzerland
Tel: (022) 98-96-51
Telex: 27835 hpse
Cable: HEWPACKSA Geneve
(Offices in the World Trade Center)

MEDITERRANEAN AND MIDDLE EAST

Hewlett-Packard S.A.
Mediterranean and Middle East
Operations
Atrina Centre
32 Kifissias Ave.
Maroussi, **ATHENS**, Greece
Tel: 682 88 11
Telex: 21-6588 HPAT GR
Cable: HEWPACKSA Athens

EASTERN USA

Hewlett-Packard Co.
4 Choke Cherry Road
Rockville, MD 20850
Tel: (301) 258-2000

MIDWESTERN USA

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800

SOUTHERN USA

Hewlett-Packard Co.
P.O. Box 105005
450 Interstate N. Parkway
ATLANTA, GA 30339
Tel: (404) 955-1500

WESTERN USA

Hewlett-Packard Co.
3939 Lankersim Blvd.
LOS ANGELES, CA 91604
Tel: (213) 877-1282

OTHER INTERNATIONAL AREAS

Hewlett-Packard Co.
Intercontinental Headquarters
3495 Deer Creek Road
PALO ALTO, CA 94304
Tel: (415) 857-1501
Telex: 034-8300
Cable: HEWPACK

Part No. 32190-90001
Printed in U.S.A. 9/82
E0982

HEWLETT
PACKARD