HP AdvanceNet

HEWLETT
PACKARD

RJE/1000-II
Remote Job Entry Subsystem

User's/Programmer's Reference Manual

HP AdvanceNet

RJE/1000-II
REMOTE JOB ENTRY SUBSYSTEM

User's/Programmer's Reference Manual

HEWLETT
PACKARD

19420 HOMESTEAD ROAD, CUPERTINO, CA 95014

Part No. 91781-90001
E1084

Printed in U.S.A. 10/84

Credits

Software Engineers:	Norm Newlon, Jim Parkes, Perry Scott, Malcolm H. Teas
Technical Support Engineers:	Robert Jon Owyang, Al Yung
Technical Writer:	Richard Rolph
Production Engineer:	Irv Solomon
Product Managers:	Hassan Alam, John Zoglin
Project Manager:	Paul Witort

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.

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 dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual 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.................................Oct 1984.......... Software Revision Code 2427

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page of the manual. Within the manual, changes since the most recent edition are indicated by printing the date of the update on the bottom of the page, and by marking the changes with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates are retained. No information is incorporated into a reprinting unless it appears as a prior update. To verify that your manual contains the most current information, check that the date printed at the bottom of the page matches the date listed below for that page.

Effective Pages	Date
all	Oct 1984

PREFACE

Purpose

This manual describes RJE/1000-II, the second Hewlett-Packard Remote Job Entry subsystem for HP 1000 computer systems. This subsystem allows the HP 1000 computer and attached terminal to emulate the IBM 2780 Data Transmission Terminal and the IBM 3780 Data Communication Terminal making it possible for you to exchange data between your HP 1000 system and a variety of host systems.

RJE/1000-II is designed to run under the control of either RTE-6/VM or RTE-A Real Time Executive operating systems and is supported on the E,F, and A series HP 1000 computers.

This manual is intended to provide sufficient information to allow you, the subsystem user, to configure RJE/1000-II to meet your remote job entry needs, to install the product on your HP 1000 system, and to use the product optimally.

Audience

This manual assumes that you have an understanding of the RTE operating system and its file system on your HP 1000, and a familiarity with the characteristics of the host system. Knowledge of the IBM Binary Synchronous Communications (BSC) line protocol conventions would be useful but is not required.

Typical uses for RJE/1000-II are expected to include simple file transfers to and from various host systems through terminal access, and programmatic scheduling for data transfer triggered from end-user application programs.

Organization

This manual is organized as follows:

SECTION 1 introduces RJE/1000-II. This section summarizes RJE concepts and describes IBM 2780, and 3780 characteristics. RJE/1000-II product capabilities, subsystem organization, and hardware and software requirements are described. The subsystem commands for RJE/1000-II are introduced.

SECTION 2 describes the installation procedures for RJE/1000-II, and provides examples of the subsystem involvement in the SYSGEN procedures for RTE-6/VM, and for RTE-A.

SECTION 3 covers configuring and initializing the subsystem. Development of the configuration file is described in detail with examples. Procedures for use of the configuration file and initialization of the subsystem are explained.

SECTION 4 discusses methods for using and controlling RJE/1000-II. The subsystem commands are described in detail as are considerations for reception of data from the host system.

SECTION 5 covers programmatic use of the subsystem and provides some examples.

SECTION 6 briefly describes the diagnostic facilities that are available with RJE/1000-II. Basic trouble shooting procedures are also covered. Information is provided to help you recover from system or subsystem failure situations, and help you supply necessary information to System Engineers in cases of unrecoverable failure.

APPENDIX A provides a summary of information and error messages and codes. A listing of the subsystem messages that are displayed on your terminal are provided along with detailed explanations of the messages. Error recovery procedures are also provided with each of the error messages.

APPENDIX B shows a complete listing of the console application program that is provided with the RJE/1000-II subsystem on the product medium. The program is written in PASCAL.

APPENDIX C provides an example printer postprocessor program. This program is also supplied on the product medium. It allows you to send received data to a typical line printer. A punch postprocessor program not supplied on the product media is also listed.

APPENDIX D provides the listing of an example FORTRAN program that makes use of the programmatic RJE,STATUS request. An example output from the program is included.

APPENDIX E lists information that must be considered when generating a host system to support RJE/1000-II. Example portions of system generations containing RJE-II support information are shown for the IBM 3704/3705 Communications Controller Emulation Program and for IBM JES2.

APPENDIX F provides a listing of modems that can be used with RJE/1000-II.

APPENDIX G supplies a description of PSI installation verification procedures.

APPENDIX H contains a table of ASCII, EBCDIC, and Hollerith code character sets.

APPENDIX I describes binary synchronous protocol.

APPENDIX J provides a chart comparing IBM 2780 and 3780 characteristics

IBM References

- *Component Description: IBM 2780 Data Transmission Terminal*
  IBM Part # GA27-3005
- *Component Information for the IBM 3780 Data Communication Terminal*
  IBM Part # GA27-3063
- *OS/VS2 System Programming Library: System Generation Reference*
  IBM Part # GC26-3792
- *System Programming Library: JES2 Installation, Initialization, and Tuning*
  IBM Part # SC23-0046
- *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual*
  IBM Part # GC30-3008
- *IBM Binary Synchronous Communications*
  IBM Part # GA27-3004

HP References

- *Data Communications Fundamentals Handbook*
  HP Part # 5957-4634
- *M/E/F Series Programmable Serial Interface (PSI) (MODEM) Installation and Service Manual*
  HP Part # 12826-91001
- *L/A Series Programmable Serial Interface (PSI) (MODEM) Installation and Service Manual*
  HP Part # 12042-91001
- *RTE-A Quick Reference Guide*
  HP Part # 92077-90020
- *RTE-A User's Manual*
  HP Part # 92077-90002

HP References (continued)

- *RTE-A Programmer's Reference Manual*
  HP Part # 92077-90007

- *RTE-A System Generation and Installation Manual*
  HP Part # 92077-90034

- *RTE-6/VM Quick Reference Guide*
  HP Part # 92084-90003
- *RTE-6/VM Terminal User's Reference Manual*
  HP Part # 92084-90004

- *RTE-6/VM Programmer's Reference Manual*
  HP Part # 92084-90005

- *RTE-6/VM System Manager's Reference Manual*
  HP Part # 92084-90009

- *RTE-6/VM On-Line Generator Reference Manual*
  HP Part # 92084-90010

CONVENTIONS USED IN THIS MANUAL

NOTATION **DESCRIPTION**

nonitalics

Words in syntax statements which are not in italics must be entered exactly as shown. Punctuation characters other than brackets, braces and ellipses must also be entered exactly as shown. For example:

```
EXIT;
```

italics

Words in syntax statements which are in italics denote a parameter which must be replaced by a user-supplied variable. For example:

CLOSE *filename*

[ ]

An element inside brackets in a syntax statement is optional. Several elements stacked inside brackets means the user may select any one or none of these elements. For example:

$\begin{bmatrix}\mathrm{A}\\ \mathrm{B}\end{bmatrix}$ User *may* select A or B or neither.

{ }

When several elements are stacked within braces in a syntax statement, the user must select one of those elements. For example:

$\begin{Bmatrix}\mathrm{A}\\ \mathrm{B}\\ \mathrm{C}\end{Bmatrix}$ User *must* select A or B or C.

. . .

A horizontal ellipsis in a syntax statement indicates that a previous element may be repeated. For example:

[,*itemname*]...;

In addition, vertical and horizontal ellipses may be used in examples to indicate that portions of the example have been omitted.

,

A shaded delimiter preceding a parameter in a syntax statement indicates that the delimiter *must* be supplied whenever (a) that parameter is included or (b) that parameter is omitted and any *other* parameter which follows is included. For example:

itema[,*itemb*][,*itemc*]

means that the following are allowed:

itema
itema,*itemb*
itema,*itemb*,*itemc*
itema,,*itemc*

CONVENTIONS (continued)

Δ — When necessary for clarity, the symbol Δ may be used in a syntax statement to indicate a required blank or an exact number of blanks. For example:

```
SET[(modifier)]Δ(variable);
```

underlining — When necessary for clarity in an example, user input may be underlined. For example:

```
NEW NAME? ALPHA
```

In addition, brackets, braces or ellipses appearing in syntax or format statements which must be entered as shown will be underlined. For example:

```
LET var[[subscript]] = value
```

shading — Shading represents inverse video on the terminal's screen. In addition, it is used to emphasize key portions of an example.

▭ — The symbol ▭ may be used to indicate a key on the terminal's keyboard. For example, [RETURN] indicates the carriage return key.

[CONTROL]*char* — Control characters are indicated by [CONTROL] followed by the character. For example, [CONTROL]Y means the user presses the control key and the character Y simultaneously.

CONTENTS

Section 1
OVERVIEW OF RJE/1000-II

Section 2
INSTALLING RJE/1000-II

Section 3
CONFIGURING AND INITIALIZING RJE/1000-II

CONTENTS (continued)

Section 4
INTERACTIVE USE OF RJE/1000-II

CONTENTS (continued)

Section 5
PROGRAMMATIC USE OF RJE/1000-II

Section 6
TROUBLESHOOTING RJE/1000-II

Appendix A
ERROR MESSAGES AND CODES

Appendix B
CONSOLE APPLICATION PROGRAM

CONTENTS (continued)

Appendix C
POSTPROCESSOR EXAMPLES

Appendix D
PROGRAMMATIC STATUS REQUEST

Appendix E
HOST SYSTEM GENERATION CONSIDERATIONS

Appendix F
MODEM OPTIONS

Appendix G
PSI INSTALLATION VERIFICATION

Appendix H
CHARACTER SETS

Appendix I
BINARY SYNCHRONOUS COMMUNICATIONS

Appendix J
IBM 2780/3780 COMPARISON

OVERVIEW OF RJE/1000-II

SECTION 1

INTRODUCTION

RJE/1000-II (RJE-II) is the second Hewlett-Packard Remote Job Entry subsystem for HP 1000 computer systems. It is a software product that allows you to use your HP 1000 to transfer data with any of a variety of host processors. RJE/1000-II accomplishes this by emulating IBM 2780 and 3780 workstations. Unlike the actual workstations which require the presence of specific hardware input/output devices, RJE-II is a disc file oriented system. Files are sent from or received to disc files that you specify.

The data transfer can take place over a switched public telephone network, over a private leased line, or within a private local point-to-point environment. RJE/1000-II supports transfer rates up to 19.2 Kbps. The actual transfer rates you may be able to use depends on the data communication link environment you are using.

RJE-II allows your HP 1000 system to emulate the major functions of an IBM 2780 or IBM 3780 workstation. RJE-II can transmit batch jobs to, and receive output from, a host processor that can support standard IBM 2780/3780 devices. Additionally, RJE-II can exchange files between an HP 1000 and many other processors that emulate standard IBM 2780/3780 devices.

Throughout this manual, the system with which you are exchanging data on the other end of the data communications link is referred to as the "host system".

RJE-II is primarily a "link-level services" product. It is not designed to provide end-to-end processing. RJE-II does, though, include the capability of automatic implementation of user written routines that can provide ene-to-end support.

SUBSYSTEM FEATURES

RJE/1000-II allows your HP 1000 to appear as a real 2780 or 3780 workstation. Although RJE-II supports a majority of the 2780 and 3780 features, there are some differences and enhanced capabilities associated with the subsystem as follows:

- Whereas the actual IBM 2780 and 3780 workstations each include a card punch, card reader, and line printer, RJE/1000-II assigns disc files as a "virtual" reader, punch, and printer. In addition, RJE-II provides handling of "unrouted" files; files that are not flagged as a specific output type.

  "Virtual" reader files are those you direct RJE-II to transmit to the host system. "Virtual" printer and punch, and unrouted files are those used to store data received from the host. Data directed to these "virtual" output files may be routed to physical devices under your control.

- For both 2780 and 3780 emulation, data transmission is performed in EBCDIC. Received text data is translated into ASCII before being stored on the HP 1000.

- Similar to the actual workstations, RJE-II recognizes and executes horizontal tab control.

- Vertical forms control is not handled by the RJE-II subsystem. RJE-II instead provides automatic scheduling of postprocessing so that you can specialize your output formatting.

- RJE-II is capable of transmitting and receiving transparent data at any time.

- RJE-II automatically performs truncation of trailing blanks in non-transparent data when in 3780 mode. In 2780 mode this capability can be selected by an indication in the subsystem configuration. You do not need to include EM control characters in the data. Trailing blanks are not truncated in transparent data.

- RJE-II automatically performs imbedded blank compression in 3780 mode.

- Blocking of transmitted data is also performed automatically by RJE-II. The blocking factors are set in the subsystem configuration. RJE-II allows up to 255 records per block of either transparent or non-transparent data, and in either 2780 or 3780 mode.

  The blocking factors for the actual IBM 2780 and 3780 devices are given as the "Standard Value" in the appropriate explanations in Section 3, Configuring and Initializing RJE/1000-II.

- Unlike the actual 2780 workstation, you can configure RJE-II to generate WACK (Wait before ACKnowledgement) and TTD (Temporary Text Delay) bisync protocol control characters in 2780 mode. For actual IBM 3780s, these control characters are always automatically generated when necessary. You can configure RJE/1000-II to adhere to standard 3780 function.

In addition to the above emulation characteristics, RJE-II provides for the following capabilities:

- Multiple simultaneous users of the subsystem.

- Multiple subsystems running on a single HP 1000.

- Subsystem diagnostics: event logging, trace capabilities on various software and firmware activities as well as monitoring of the datacomm link.

- On-line alteration of event logging, trace functions, and output routing.

- Both interactive and programmatic access to the full range of subsystem capabilities.

- Support of 15 native languages through translation tables.

For a comparison of the IBM 2780 and 3780 characteristics, see Appendix J. Detailed information on the IBM 2780 can be found in *Component Description: IBM 2780 Data Transmission Terminal*, GA27-3005. Information on the IBM 3780 can be found in *Component Information for the IBM 3780 Data Communication Terminal*, GA27-3063.

SUBSYSTEM ORGANIZATION

The major components of the subsystem consist of a memory resident datacomm monitor, identified as RJExx; down-loaded Z80 microprocessor firmware running on a programmable serial interface (PSI) card; and two transient user-interface processes, RINIT and RJE, which can be activated either interactively from any system terminal, or programmatically through RTE EXEC calls.

Overall organization of the RJE-II subsystem is shown in Figure 1-1.

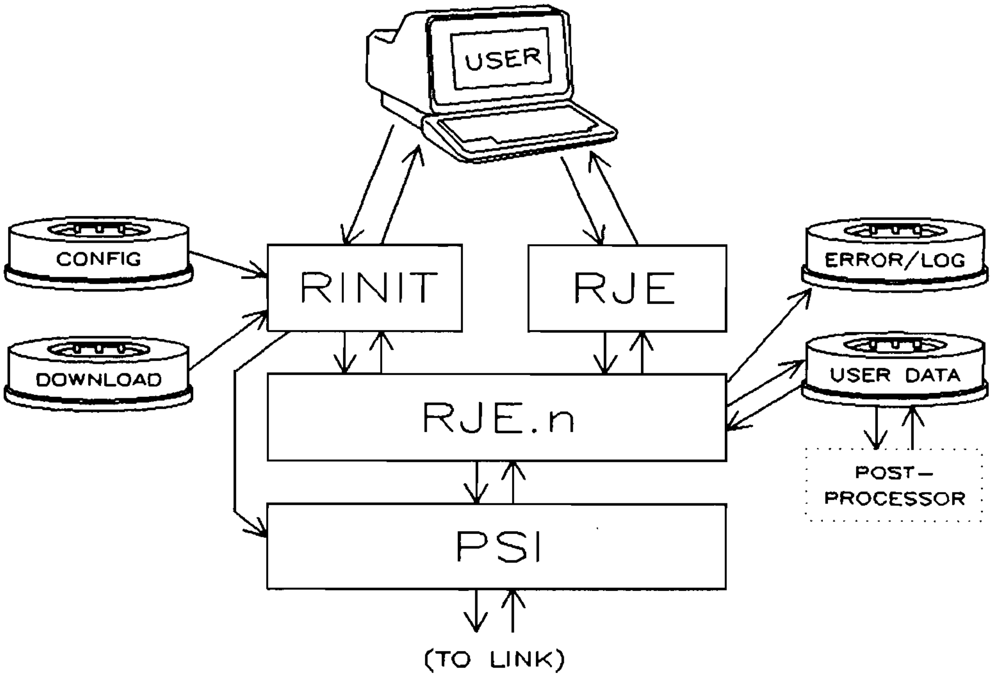


Figure 1-1, RJE/1000-II Subsystem Organization

Each installation of RJE/1000-II consists of the major components as well as a subsystem configuration file, subsystem log file, user files, and a clone of the datacomm monitor program.

Within a single HP 1000, there can be up to eight installations of RJE/1000-II. Each is a seperate entity with its own components. Coordination and control between the various RJE-II subsystem installations is by means of a table, called RJTAB, in the operating system area. RJTAB contains class numbers for communication with each of the subsystem datacomm monitor clones.

When an RJE-II subsystem is "brought up", the subsystem number, *n*, identified in the configuration file, is used in naming the datacomm monitor clone, making it RJE.*n*. The subsystem number functions as an index into RJTAB and is used to retrieve the class number for communication with the appropriate clone of RJExx.

Both the datacomm monitor clone and the PSI card remain active through the "life" of the subsystem. The transient process programs, RINIT and RJE, are cloned by the operating system when runstrings calling them are entered. A clone exists for each runstring until the requested processing (e.g. initializing the subsystem or sending a list of files) completes. At this point the clone terminates.

SUBSYSTEM OPERATION

To make the RJE-II subsystem operational, it must first be installed on your HP 1000 system and then configured and initialized. Installation of the product is covered in detail in Section 2, Installing RJE/1000-II.

Configuration

You configure the product through the use of a configuration file that you create with an editor. The configuration file consists of thirty-one lines of information that allow you to specify some of the critical characteristics of the subsystem. You can find complete information on the contents, format, and development of the configuration file in Section 3, Configuring And Initializing RJE/1000-II.

Initialization

Once the configuration file has been created, the subsystem is initialized and activated by running the transient process RINIT. You supply the name of the configuration file in the RINIT runstring using the syntax:

```
[RU,]RINIT,configfile
```

where *configfile* is the RTE file name for your configuration file.

If successful, RINIT performs the following functions:

- Creates a memory image clone of RJExx. The clone is identified as RJE.*n*.
- Resets and downloads the Z80 firmware to the PSI card and initializes it.
- Configures the PSI card and the datacomm monitor clone with information from the subsystem configuration file.
- Sends SIGNON card image (if specified in the configuration) to the host.

Utilization

You use the subsystem both interactively and programmatically through runstrings of the other transient process, RJE. The general syntax for the RJE runstrings is:

```
[RU,]RJE,keyword,n,password,parameters
```

There are six acceptable values for *keyword*, each of which invokes an entirely different function of the RJE program as follows:

• SEND	transmits data to your host system
• STATUS	provides current and long term, device and configuration status information on the emulator, PSI card, data communications link, and virtual reader and output devices
• ABORT	terminates current specific subsystem activity and, as soon as possible, makes the data communications link available to your HP 1000 and the host system
• CLOSE	brings down the RJE-II subsystem, and the data communications link
• TRACE	creates on-line changes that affect the PSI and Monitor trace control words, and log file specification
• ROUTE	creates on-line changes that affect the output file masks and postprocessor names

For each RJE runstring you enter, a request is issued to the datacomm monitor (RJE.*n*) via class I/O. The request queues up on the datacomm monitor's input queue until it can be processed. When the request is complete, RJE.*n* sends a response to the RJE program. The RJE program then returns the completion status (success or failure) of the runstring to you and then terminates. The return of your operating system prompt indicates success. In failure situations, you are sent error information.

There is virtually no limit to the number of RJE runstrings which may be issuing requests to the datacomm monitor. Multiple users may issue RJE requests simultaneously. A single user may issue multiple RJE requests by running RJE without wait (by using the CI XQ command). The requests queue up on the datacomm monitor (RJE.*n*) and are performed one at a time.

Detailed information on the RJE runstrings is available in Section 4, Interactive Use of RJE/1000-II, and Section 5, Programmatic Use of RJE/1000-II.

Sending Files

When you use the subsystem to transmit files to the host system, you use an RJE runstring with the SEND keyword. The file names that you include in the runstring are checked and passed to the datacomm monitor. When the monitor detects it has files to transmit, it selects the files one at a time and sends them to the PSI card. The PSI card completes the blocking of the data, translates the data into EBCDIC, compresses imbedded blanks, applies the necessary bisync protocol characters, and transmits each block to the host.

Files can be sent interactively, by entering the RJE,SEND runstring from your terminal, or programmatically, by issuing the equivalent runstring through RTE EXEC calls. More information on interactive and programmatic SEND requests can be found in Section 4, Interactive Use of RJE/1000-II, and Section 5, Programmatic Use of RJE/1000-II.

Receiving Files

Whenever RJE/1000-II is operational, it is capable of receiving and automatically storing files from the host system. Once you have initialized your subsystem, no intervention is required on your part to receive data.

NOTE

All non-transparent (text) data is transmitted and must be received in EBCDIC. Non-transparent data received by the PSI card is subjected to EBCDIC-to-ASCII translation. Non-EBCDIC data will not be usable after translation. Transparent (binary) data is not translated.

When the PSI card detects that data is going to be received from the host, it informs the datacomm monitor. The received blocks of data are checked by the PSI card, stripped of bisync protocol characters, translated from EBCDIC to ASCII (if non-transparent data), decompressed, and sent to the datacomm monitor. The monitor writes each block of data to the file system on your HP 1000. When the last block of data is written to a file, the monitor opens the next file, closes the file that received the data, and automatically schedules any postprocessing you have configured. You can find more information on data reception and postprocessing by RJE-II in Section 4 under "Data Reception".

Error Handling

RJE/1000-II has significant error handling and diagnostic capabilities. These include continuous log file posting of significant non-link, and link related events, tracing of the datacomm monitor and PSI card activities, and a status display indicating virtual device statistics, current and long term file transfer information, and current and long term data communications link statistics. An example STATUS display can be found in figure 4-1.

Link related error conditions from which the PSI card cannot recover are reported to the datacomm monitor and hence to the subsystem log. If the error is associated with a particular RJE request, it is posted to your terminal LU.

A complete list of error messages and codes can be found in Appendix A, and you can find a more complete description of the error handling and diagnostics capabilities of RJE-II in Section 6, Troubleshooting RJE/1000-II.

SUBSYSTEM ENVIRONMENT

RJE/1000-II has specific hardware, software, data communications link, and host system environment requirements in order to function. An overview of the requirements is presented here. You can find the details of the environment needs in Section 2, Installing RJE/1000-II.

Hardware Requirements

RJE/1000-II is designed to run on E,F, and A series HP 1000 computer systems. The standard programmable serial interface (PSI) card with download/selftest firmware (12260A for E and F series or 12043A for A series systems) is required for each subsystem present. Multiple subsystems can be installed on one HP 1000 system. One PSI card is required per subsystem. The minimum system must include 128 pages of memory, a disc storage unit, and an operators console or other terminal device.

Software Requirements

RJE/1000-II runs under RTE-6/VM and RTE-A operating systems. The subsystem requires the presence of the "new file system" and is thus not supportable on older versions of RTE.

Your HP 1000 system generation must include the standard smart-card drivers: DDD63 and IDS00 on A series systems, DDV63 and DVN00 on E, and F series systems.

Link Requirements

The environment requirements to establish the link with the host system depend on the type of line you are using: switched public network, leased, or local point-to-point. Generally, a pair of compatible synchronous modems is required. You specify in the configuration file if they are full duplex or half-duplex. The types of modems you can use are listed Appendix F. Other link requirements are detailed in Section 2, Installing RJE/1000-II.

Host Requirements

There are a number of characteristics of your host system with which you should be aware. Some of the host characteristics you must know to properly configure the subsystem are:

- Workstation Type: 2780 or 3780 3703
- Workstation Mode: Master or Slave
- Block Size
- Blocking Factors (transparent and non-transparent)

The Blocking Factors may not be a necessary consideration when communicating with IBM host systems. They are a configurable item for RJE-II to enable your subsystem to communicate with those 2780/3780 emulators requiring control of these factors.

The Block Size translates to Transmission Buffer Size in IBM terminology.

To assist you in understanding some of these characteristics, an example of a typical host system generation to support RJE/1000-II is included in Appendix E.

INSTALLING RJE/1000-II

SECTION 2

This section discusses the procedures used to install RJE/1000-II on your HP 1000 system. There are two phases.

- System Generation adds the necessary I/O drivers, equipment tables (EQTs), logical units (LUs), class numbers, ID segments, and system area.
- RJE-II is then loaded on-line using a Hewlett-Packard supplied transfer file.

Relevant portions of sample system generation answer files for both RTE-6/VM and RTE-A, and on-line loading procedures are supplied in this section.

SYSTEM GENERATION

A new system generation is needed to support a PSI used for RJE communication to a host.

PSI Driver

The product media contains the specific device driver required for RJE/1000-II to interact with the subsystem firmware on the PSI card. The driver is specific to the particular model of the HP 1000 system, as indicated in Table 2-1.

TABLE 2-1. RJE/1000-II DRIVERS

HP 1000	Device Driver:	Interface Driver:
E/F-Series A-Series	DDV63 DDD63	DVN00 IDS00

The E/F-Series HP 1000 computers use DVN00 as the physical interface driver for the PSI and DDV63 as the logical device driver. These must be generated in the same driver partition. For A-Series HP 1000 computers the PSI interface driver is IDS00, and the logical device driver is DDD63. The interface/device driver combination is added during the program input phase of system generation on the E/F-Series computers and can be added during the driver partition phase on the A-Series.

RJE/1000-II SOFTWARE MODULES

The RJE/1000-II subsystem for the HP 1000 consists of the modules described in Table 2-2.

TABLE 2-2. RJE/1000-II MODULES

Installation CMD File	**Function**
RJE.CMD	Subsystem installation file (CI transfer file)
System Generation Modules	**Function**
DDD63.REL	RTE-A device driver
DDV63.REL	RTE-6/VM device driver
RJTAB.REL	Subsystem table area
Main Process Modules	**Function**
RINIT.REL	Configures and initializes the subsystem
RJEXX.REL	Datacomm Monitor
RJE.DAT	User interface
Link Libraries	**Function**
RJELB.LIB	Subsystem library area
PASCAL.LIB	PASCAL library area
Support Modules	**Function**
RJE.HELP	Subsystem help file
CON.PAS	Example console application program PASCAL source code
CON.REL	Example console application program relocatable module
FMT.FTN	Example printer postprocessor FORTRAN source code
FMT.REL	Example printer postprocessor relocatable module
STAT.FTN	Example programmatic status request FORTRAN source code

TABLE 2-2. RJE/1000-II MODULES (cont.)

Support Modules (cont.)	Function
STD3780.TXT	Example subsystem configuration file (3780 emulation)
A91781	Software numbering file (SNF)

During the subsystem intialization process, all relocatable modules (xxxx.REL) are linked and form type 6 executable modules (xxxx.RUN) and are referred to in the manual without the type extensions.

The RINIT module is active only during subsystem initialization. RJExx is cloned for each installation of the subsystem. The clone, identified as RJE.*n*, is active for the life of the subsystem. RJE is also cloned with each user interface request. Multiple clones of RJE can exist. They are active during execution of the request.

All modules listed here are supplied on the product media. Not listed are the 15 native language translation relocatable modules. These and all the other modules are listed in the Software Numbering File that resides on the product media. The name of this file on the media is A91781.

Native Language Character Sets

Included in the Software Numbering File is a list of the native language EBCDIC/ASCII character set tables available with RJE/1000-II. Browse A91871 to see names of the relocatable files for the languages that are supported.

To invoke a character set table other than HP-native, you edit the subsystem installation file, RJE.CMD, listed in Figure 2-1, replacing native.rel with the appropriate native language relocatable file.

```
*   91781-17001 REV.2427 <time stamp>
*
*
link   rje.run::programs +ro rjelb.lib pascal.lib   rje.rel
link   con.run::programs +ro rjelb.lib pascal.lib   con.rel
*
*     For character sets other than HP-native, replace 'native.rel'
*     with the appropriate native language support module.
*
link rinit.run::programs +ro rjelb.lib pascal.lib rinit.rel nativ.rel
link rjexx.run::programs +ro rjelb.lib pascal.lib rjexx.rel nativ.rel +su
*
*  Install help file
*
co rje.help rje::help d
*
*  Install PSI download file
*
co rje.dat @.@::system d
*
*  Done !
```

Figure 2-1, Subsystem Installation File (RJE.CMD)

For example, to use the Swedish translation table, the two lines in RJE.CMD that you change would read:

```
link rinit.run::programs . . . swedi.rel
link rjexx.run::programs . . . swedi.rel trace.rel +su
```

OPERATING SYSTEM REQUIREMENTS

RJE/1000-II requires the use of the "new" file system. The current revision of the RTE-6/VM and RTE-A operating system must be used.

HARDWARE INSTALLATION

PSI Installation

The PSI can be installed in any I/O slot of an A Series HP 1000. In a Series E or F HP 1000 system a PSI must be installed above the privileged fence printed circuit assembly (PCA), if one exists. Detailed information can be found in these documents:

M/E/F-Series PSI Installation and Service Manual, 12826-91001
L/A-Series PSI Installation and Service Manual, 12042-91001
Multi-Use PSI Installation Manual, 5955-7630

The position of the switch settings on the PSI for the E/F-Series does not affect RJE/1000-II operation. However, one bank of switches on the A-Series PSI determines the PSI select code; for more information on this switch bank, see the *L/A-Series PSI Installation and Service Manual*, 12042-91001 or the *Multi-Use PSI Installation Manual*, 5955-7630.

NOTE

> Switch number one on this A-Series switch bank is the Firmware Read Switch. This switch must be open to operate RJE/1000-II.

PSI installation verification procedures are described in Appendix G.

Modem Requirements

The PSI (product number 12260A or 12043A) comes with a five-meter cable that connects the PSI to the modem. The modems themselves may operate at speeds up to 19.2K bits per second. The RJE/1000-II subsystem also operates at speeds up to 19.2K bits per second. Although the protocol is half-duplex, you can use full-duplex modems to reduce line turnaround time. The modem you chose must meet the following requirements:

- Synchronous
- Timing internal to modem
- EIA RS232C, RS-449, or CCITT V.24 compatible
- Does not use New Sync
- Does not use Reverse Channel
- Must be matched to the modem at the host site

A list of compatible modems and recommended strappings is in Appendix F.

SYSTEM GENERATION ON THE HP 1000

System generation on the HP 1000 requires RJE/1000-II's relocatable modules as listed in "RJE/1000-II Software Modules" in this section. These modules are made available by the On-Line Loading procedures described in this section.

In the following section you are assumed to have some experience in generating systems. Example system generations follow, for RTE-6/VM in Table 2-3, and for RTE-A in Table 2-4.

RJE/1000-II Requirements

RJE/1000-II uses the entry point RJTAB in the RTE system area for subsystem communication. Only RJE/1000-II uses this area.

To optimize the subsystem performance, you can reserve partitions for the datacomm monitor (RJE.*n*) to increase it's chances of being memory-resident. See "Partition Definition," which follows.

NOTE

Best preformance is achieved when RJE.*n* is memory resident.

Partition Definition

On an RTE-6/VM system, partition definition may occur during system generation or at boot-up time; on an RTE-A system, partition definition occurs during or after boot-up.

The following list gives the approximate size in pages of each RJE/1000-II module:

APPROXIMATE RJE PROGRAM SIZE	8 pages per outstanding request
APPROXIMATE RJE.*n* PROGRAM SIZE	19 pages per subsystem
APPROXIMATE RINIT PROGRAM SIZE	14 pages per system

RJE/1000-II On-Line Loading

Once the system generation and boot-up are complete, you use a transfer file, identified as RJE.CMD, to load the RJE/1000-II modules and search the subsystem libraries and the Pascal library. You type TR RJE.CMD at the CI> prompt. The LINK loader then searches for the RJE/1000-II modules, the RJE/1000-II libraries, and the Pascal library.

LINK expects the SNAP file to be SNAP.SNP::SYSTEM. The Pascal library is expected in PASCAL.LIB::LIBRARIES. LINK places all run programs on the PROGRAMS directory as @.RUN::PROGRAMS and moves the subsystem help file (RJE.HELP) to the system HELP directory as RJE::HELP.

NOTE

> When loading RJE/1000-II on-line under RTE-6/VM or RTE-A, you should be operating in CI and in the working directory, ::RJE.

The procedures for loading RJE/1000-II on-line are as follows:

- if the product medium is magnetic tape:

  1 create the directory RJE (CI>crdir ::RJE)

  2 copy product from medium to the directory ::RJE
  (CI>tf co 8{@.@} @::RJE) where 8 is the tape LU

  3 generate your HP 1000 system

  4 make ::RJE your working directory (CI>wd ::RJE)

  5 load the product on your HP 1000 system (CI>tr rje.cmd)

- if the product medium is floppy disc:

  1 follow the instructions supplied with the product medium. These procedures create the directory ::RJE on the system disc, and copy the contents from the floppy discs to this directory.

  2 proceed as in steps 3, 4, and 5 under the magnetic tape directions.

RTE-6/VM Generation

RT6GN is the program used to generate an RTE-6/VM system. More detailed system generation information is in the following manuals:

RTE-6/VM System Manager's Reference Manual, 92084-90009
RTE-6/VM On-Line Generator Reference Manual, 92084-90010

Two Equipment Table (EQT) entries and two Logical Units (LUs) are required. One EQT is used for transmission of files to the host system and the other EQT is used for receiving printer/punch/unrouted files and trace data. This creates a logical full-duplex I/O channel between the PSI and the HP 1000 processor. Each EQT requires thirty-one extension words. The EQTs are added during the equipment table entry phase of system generation.

RJE/1000-II uses two LUs to address the EQT entries. These are added during the device reference table entry phase of system generation.

An interrupt trap cell, corresponding to the PSI I/O slot number (select code), is set up to handle PSI hardware interrupts. This is done during the interrupt table entry phase of system generation.

The example that follows uses variables for the select code, the trace/receive and transmit EQT entry values, and the EQT's corresponding LU values. The PSI is assumed to be in the PCA slot corresponding to select code *sc*. The transmit and receive/trace EQT entry values are *e1* and *e2*, respectively; these correspond with LU values of *l1* and *l2*.

NOTE

To allow the RJE-II software to identify *l1* and *l2* as PSI LUs, enter the following two commands in the RTE-6 Welcome File. Both of these commands need to be entered for each of the PSI LUs:

```
CN,lu,33B,3
CN,lu,0
```

For RJE-II to run with optimal efficiency under RTE-6/VM, you should configure the maximum amount of System Available Memory (SAM) possible.

The following variables, chosen by you, appear in Table 2-3 and are defined as follows:

sc PSI I/O select code.

e1, e2 transmit and receive/trace EQTs respectively.

l1, l2 transmit and receive/trace LUs respectively.

q integer value of system requirements for class numbers and ID segments before adding RJE/1000-II.

s the number of RJE/1000-II subsystems.

u the maximum expected number of outstanding RJE user requests (ie. the maximum expected number of jobs queued up awaiting transmission to the host plus other concurrent pending user requests).

TABLE 2-3. RTE-6/VM EXAMPLE:
RELEVANT PORTIONS OF THE GENERATION ANSWER FILE

```
     .
     .
     .
*
*   PROGRAM INPUT PHASE
*
REL,%DVN00::RTE6          *STANDARD INTERFACE DRIVER
REL,%PVM00::RTE6          *PRE-DRIVER - HANDLES INTERRUPTS
REL,%$DVTN::RTE6          *DEVICE DRIVER ADDRESS TABLE FOR DVN00
REL,DDV63.REL::RJE        *DEVICE DRIVER FOR RJE-II
     .
     .      (Other Device Drivers and Other System Drivers)
     .
REL,RJTAB.REL::RJE        *COMMON AREA IN SYSTEM MAP
*                             FOR RJE/1000-II
     .
     .
     .
*
* EQUIPMENT TABLE ENTRY
*
     .
     .
     .
sc,DVN00,X=31            * EQT e1 - RJE-II TRANSMIT
sc,DVN00,X=31            * EQT e2 - RJE-II RECEIVE/TRACE
     .
     .
     .
*
* DEVICE REFERENCE TABLE
*
```

TABLE 2-3. RTE-6/VM EXAMPLE: RELEVANT PORTIONS OF THE GENERATION ANSWER FILE (cont'd)

```
       .
       .
       .
e1                          * LU l1 - RJE-II TRANSMIT
e2                          * LU l2 - RJE-II RECEIVE/TRACE
*
* INTERRUPT TABLE
*
       .
       .
       .
sc,EQT,e1                   * RJE-II PSI
       .


       .
       .
*
* MISCELLANEOUS SYSTEM PARAMETERS
*
       .
       .
       .
q+(2*s+u)                   * CLASS NUMBERS - RJE-II USES THE
                            *   NUMBER OF PENDING USER REQUESTS PLUS
                            *   TWO TIMES THE NUMBER OF SUBSYSTEMS

q+(s+u)                     * LONG ID SEGMENTS - RJE-II USES THE
                            *   THE NUMBER OF SUBSYSTEMS PLUS THE
                            *   THE NUMBER OF USERS
*
* DEFINE PARTITIONS
*
       .
       .
       .
        (See "Partition Definition," in this section.)
       .
       .
       .
```

After generation is completed, you load the RJE/1000-II modules on-line. For more information see "RJE/1000-II On-line Loading" in this section.

RTE-A Generation

The HP 1000 A-series interface printed circuit assemblies (PCAs) are different from the M/E/F-Series I/O PCAs. The select code of the A-Series PCAs does not depend on physical location. Instead, the select code is set by switches on the PSI, as described in the *L/A-Series (PSI) (MODEM) Installation and Service Manual*, 12042-91001.

NOTE

> Switch number one on this A-Series switch bank is the Firmware Read Switch. This switch must be open to operate RJE/1000-II.

The RJE/1000-II PSI may be placed in any slot below the CPU; however, there must not be an open slot between PCAs.

The program that generates an A 600, A 700, or A 900 system is RTAGN. Details about system generation are in the following manual:

RTE-A System Generation and Installation, 92077-90034

One interface table entry (IFT) is made for the PSI card. Two device table entries (DVT) are made to accomplish the logical full duplex channel, as with the RTE-6/VM case. The drivers IDS00 and DDD63 contain system generation records which set up many of the default IFT and DVT parameters. A sample answer file is shown in Table 2-4.

In order to reduce system size, the drivers IDS00 and DDD63 would ordinarily be installed during the driver partition phase. The sample answer file shows each of them in a partition by itself. Alternatively, IDS00 and DDD63 can be arranged with other drivers to make maximum use of partition space, as explained in the System Generation and Installation manual. IDS00 is approximately 1230 decimal words in length, and DDD63 is less than 150 decimal words.

The RJE/1000-II table area, RJTAB, is not partitionable; RTE-A users must generate RJTAB into the system area. RJTAB is 9 decimal words long.

For RJE-II to run with optimal efficiency under RTE-A, you should configure an additional 2 pages of System Available Memory (SAM).

These terms appear in Table 2-4 and are defined as follows:

sc PSI I/O select code.

lu1, lu2 transmit and receive/trace LUs, respectively.

q integer value of system requirements for class numbers and ID segments <u>before</u> adding RJE/1000-II.

s number of RJE/1000-II subsystems.

u the maximum expected number of outstanding RJE user requests.
(ie. the maximum expected number of jobs queued up awaiting transmission to the host, plus other concurrent pending user requests).

TABLE 2-4. RTE-A EXAMPLE: RELEVANT PORTIONS OF THE GENERATION ANSWER FILE

```
     .
     .
     .
*  SYSTEM RELOCATION PHASE
     .
     .                                  (system modules)
     .
RE,RJTAB.REL::RJE
     .
     .                                  (privileged driver modules)
     .                                  (libraries)
*  END OF SYSTEM RELOCATION PHASE
END
*
*  DRIVER PARTITION PHASE
     .
     .
     .
RE,%IDS00::RTEA
END
RE,DDD63.REL::RJE
END
     .
     .
     .
*   END OF DRIVER PARTITIONS
END
*
```

TABLE 2-4. RTE-A EXAMPLE RELEVANT PORTIONS OF THE GENERATION ANSWER FILE (cont'd)

```
*   TABLE GENERATION PHASE
     .
     .
     .
*                                            * PSI INTERFACE
IFT,%IDS00::RTEA,EIDS00,SC:sc,TX:15,QU:FI
*
*                                            * PSI TRANSMIT LU
DVT,DDD63.REL::RJE,,LU:lu1
*                                            * PSI RECEIVE/TRACE LU
DVT,DDD63.REL::RJE,,LU:lu2
     .
     .
     .
*    END OF TABLE GENERATION PHASE
END
*
     .
     .
     .
CLAS,q+(2*s+u)                * CLASS NUMBERS - RJE-II USES THE
                              *   NUMBER OF PENDING USER REQUESTS PLUS
                              *   TWO TIMES THE NUMBER OF SUBSYSTEMS

ID,q+(s+u)                    * LONG ID SEGMENTS - RJE-II USES THE
                              *   THE NUMBER OF SUBSYSTEMS PLUS THE
                              *   THE NUMBER OF USERS
     .
     .
     .
```

After completing system generation and boot-up, you can define partitions for the RJE/1000-II modules. For details see "Partition Definition" in this section.

Once the boot-up process is complete, you load the RJE/1000-II modules on-line. "RJE/1000-II On-Line Loading" in this section describes the transfer file, RJE.CMD, used to load the modules.

/HP/RTEA/%IDS00

CONFIGURING AND INITIALIZING RJE/1000-II

SECTION 3

Once your HP 1000 system has been generated to include the RJE/1000-II subsystem, the next step is to develop the subsystem configuration file. This section describes how to create the configuration file and how to initialize the subsystem using the configuration file.

CREATING THE CONFIGURATION FILE

The RJE/1000-II configuration file is a thirty-one line, Type 3 or 4 (text) file that you create with a text editor prior to initializing the subsystem. This file takes into account your own remote job entry needs as well as the requirements and limitations established for your site by your host machine operating system. A knowledge of both demands is necessary to establish a correct configuration file for your RJE/1000-II usage.

Format

The format of the configuration file is rigidly defined as follows:

- Thirty-one lines of information must be supplied in the file. (For host systems not requiring a SIGNON card, only 30 lines are needed.)

- No lines (except the SIGNON where permitted) may be omitted. No extra lines may be inserted.

- Each line of the file contains one configuration element.

- With two exceptions, each element is defined by a field starting with the first non-blank character in the line, and ending with a blank or comma. The two exceptions are the first element of the file, all of which is a comment string, and the last element, which is used as the signon card image to those host systems requiring it.

- Elements that you can leave blank are specified in the element descriptions under "The Configuration File Elements". *Comments that you wish to leave on the lines defining these elements should be preceeded by a comma.* Leading blanks are ignored but non-blank characters other than commas are selected to define the element.

  Comments are defined as characters following embedded blanks or commas. The one exception to this definition occurs on the log file naming line. You can have the subsystem automatically purge your current log file at start up by including a space followed by p after the log file name.

- Both upper and lower case characters may be used in creating the configuration elements. In the following examples, upper case characters have been used for clarity.

- The file is created by simply typing in a correct value for each element on the correct line.

- The elements must appear in the order shown in Figure 3-1 and as defined under "The Configuration File Elements".

STD3780.TXT

The following figure lists the example configuration file named STD3780.TXT that is supplied with the RJE/1000-II product. You can use this file as a starting point for the development of a configuration that meets your needs.

Line numbers are included in the comment section of each line, where allowed, for your convenience.

```
STD3780.TXT     91781-17002 REV.2427       RJE-II Example Configuration File
NOECHO                         ,(2) echo during init
1                              ,(3) subsystem number
                               ,(4) subsystem password
RJEXX                          ,(5) subsystem monitor file name
30                             ,(6) PSI read LU
31                             ,(7) PSI write LU
RJE.DAT::SYSTEM                ,(8) PSI download file name
3780                           ,(9) workstation type
SLAVE                          ,(10) workstation mode
MO                             ,(11) modem connect type
240                            ,(12) modem connect timeout
HALF                           ,(13) duplex
216                            ,(14) nontransparent blocking factor
6                              ,(15) transparent blocking factor
512                            ,(16) block size
80                             ,(17) maximum record size
240                            ,(18) inactivity timeout
7                              ,(19) retry limit
NO                             ,(20) don't append trailing blanks
YES                            ,(21) generate WACK/TTD
0                              ,(22) Monitor trace control
0                              ,(23) PSI trace control
/SYSTEM/RJE/RJELOG p           ,(24) system log file (auto purge)
/USR/PAYROLL/PRDATA            ,(25) printer output file mask
/USR/PAYROLL/VFC/CHECKS.RUN    ,(26) printer postprocessor
/USR/CADCAM/SCHEMATIC          ,(27) punch output file mask
/USR/CADCAM/PHOTOPLOT.RUN      ,(28) punch postprocessor
RS::17                         ,(29) unrouted output file mask
/PROGRAM/SPOOL_RJE_MSGS.RUN    ,(30) unrouted postprocessor
/*SIGNON       REMOTEnn
```

Figure 3-1. STD3780.TXT Configuration File

Note that this configuration specifies 3780 emulation. To do 2780 emulation you must change the specified "Workstation Type". You will also have to change the "Non-transparent" and "Transparent Blocking Factors", and "Block Size" elements. Suggested values for both 2780 and 3780 emulation are provided, where appropriate, in the element definitions following.

THE CONFIGURATION FILE ELEMENTS

The following is a description of each of the configuration file elements. For each element, a range of acceptable values and a standard value are displayed. For those elements that are different for 2780 and 3780 emulation, values for both are displayed. If a single standard value is supplied, it will work for both 2780 and 3780 emulation.

Comment String

The first element of the configuration file is a comment string. The information in this string is displayed on your terminal at initialization and is returned as part of the RJE,STAT command. You are limited to eighty characters in this string, including blanks and special characters. You should include a reference to which configuration file is being used for initialization, and to which host system you are connecting.

Example

```
RJE-II Configuration File "STD3780.TXT" for connect to HQ main frame
```

Echo During Initialization

During the initialization process, the contents of the configuration file along with "prompts" issued by the initialization program, RINIT, may optionally be echoed to your terminal LU. The RINIT prompts indicate which configuration element is expected each time a line is read from the file. This echoing is of value when setting up a configuration file, or when checking a configuration file for validity. If you enter ECHO on the second line of the configuration file, the information is echoed. If you enter NOECHO, (the usual case) only the comment string and the echo value prompt and value are echoed.

Range	Standard Value
ECHO or NOECHO	NOECHO

Subsystem Number

The subsystem number must be supplied when multiple RJE/1000-II subsystems are configured on your HP 1000 system. The number refers to an entry in the RJTAB table as explained in Section 1, under "Subsystem Organization". This value is used when the subsystem monitor program is cloned and when an RJE request is issued to the subsystem. See Section 4, Interactive Use of RJE/1000-II for more discussion of the subsystem number in the RJE request runstrings.

Range	Standard Value
an integer from 1 to 8	1

Subsystem Password

The subsystem password is an optional 16-bit element that can be used to restrict access to the RJE/1000-II subsystem.

You can specify a character string, or an integer value for the password. If you enter a character string, you can use any combination of alphabetic characters, special characters (except commas and spaces), and control characters. The subsystem recognizes only the first two bytes of the string.

If you enter an integer value, you can specify it as an octal integer, by entering "B" after the digits, or as a decimal integer. The acceptable value range for an integer password is -32768 to 32767.

If you specify a password in the configuration file, you must include it with every runstring of the RJE program. You can leave this configuration line blank (comments on the line must be preceeded by at least one comma), in which case the password can be left out of the RJE runstrings. Entering 0 (zero) for the password in the configuration file has the same effect as leaving the element blank.

Range	Standard Value
If character string: 2 characters If integer value: -32768 to 32767	User supplied

Subsystem Monitor (RJExx) File Name

This configuration element refers to the program file containing the datacomm monitor module. The file is supplied on the RJE/1000-II product media and transferred to your system during the subsystem installation under the identification RJEXX.RUN::PROGRAMS. During initialization, the RINIT program accesses this file and creates a clone of the job identified as RJE.*n*, where *n* is the subsystem number. To locate the file, RINIT follows the normal CI search sequence. If you do not have a working directory when you run RINIT, you must include a fully qualified path name for this configuration file element.

Range	**Standard Value**
Determined by file naming restrictions	RJEXX

PSI Read And Write LUs

Each of the RJE/1000-II subsystems running on your HP 1000 system requires two LUs. These are the PSI card Logical Unit numbers established during system generation. The two LUs are entered on lines 6 and 7 of the configuration file. The values entered are system LU numbers. Identification of these LUs follow the same rules as for other system Logical Unit numbers. Enter each as an appropriate integer.

Range	**Standard Value**
A unique integer value from 2 to 255	Determined by your system generation

PSI Download File Name

The PSI download file is another file supplied with the product media and must be accessible from the working directory of the user or process that issues the RINIT command. The file is supplied as RJE.DAT and is copied to the ::SYSTEM directory during the online loading procedure. RINIT follows the normal CI search sequence to locate this file. If the file is not on your working directory, you must supply a fully qualified path name on this configuration file line.

Range	**Standard Value**
Determined by file naming restrictions	RJE.DAT::SYSTEM

Workstation Type

RJE/1000-II can emulate either an IBM 2780 or an IBM 3780 workstation. You must specify the type of workstation being emulated on this configuration file line. This is one of the configuration file elements that must match the host system configuration for your remote. Assure that a match exists between the host system generation and your subsystem configuration entry, 2780 or 3780.

Range	Standard Value
2780 or 3780	3780

Workstation Mode

This configuration element refers to the position of your remote in the line bid hierarchy. The workstation can assume either a "primary" or "secondary" mode, in IBM's terminology. You can enter MASTER or SLAVE . These terms are in more common usage. The mode selected affects the line bid timeout period. For RJE-II, MASTER creates a one second timeout period, and SLAVE creates a three second timeout. The mode must be the opposite of that generated for the host system to which you are connected. For IBM systems, the host processor uses a three second (slave) timeout value.

Range	Standard Value
MASTER or SLAVE	MASTER

Note that in the example configuration file supplied with RJE-II, STD3780.TXT, the workstation is configured as SLAVE. If you are connecting to an IBM host processor, this element should be changed to MASTER.

Modem Connect Type

This configuration file element controls what type of link connection sequence is expected by the PSI card firmware. The three permissible values are MO, AAR, and AA. MO stands for "manual originate" and is used when the RJE-II subsystem initiates the communication session with the host. AAR stands for "auto answer with ring" and is used when the host system initiates the session over switched lines. AA should be specified to indicate "auto answer without ring" when the host initiates the conversation over a leased line.

Range	Standard Value
MO, AAR, or AA	MO

Modem Connect Timeout

This timeout value, specified in seconds, is the amount of time the system will wait for a response from the host system once link initiation is attempted. For manual dial installations, a large value (perhaps a minute or more) may be necessary to allow time for you to dial the phone. If there is no host response within the period specified, the subsystem initialization will fail and all subsystem processes will terminate. A value of zero indicates an infinite timeout period. With this specification, if there is no host response to RJE-II's attempts to establish the link, the initialization procedure will suspend indefinitely. Recovery prodecures for this situation can be found in Section 6 under "RINIT Errors and Recovery".

Range	Standard Value
An integer value, in seconds from 0 to 255	240

Duplex

This configuration file element refers to the type of modem you are using. Although RJE/1000-II uses bisync protocol which is inherently half-duplex, full-duplex modems are often used to reduce the overhead associated with communication link turnaround. Enter HALF or FULL on this line to match your modem type.

Range	Standard Value
HALF or FULL	HALF

Nontransparent Blocking Factor

This element specifies the maximum number of records that are packed into a nontransparent (text) block to be transmitted, space permitting. A selection must be made which is compatible with what the host system is capable of receiving. For 2780 emulation, RJE/1000-II allows a maximum value of 255 records per block. The standard value is 7. For 3780 emulation, this value has a maximum of 255. The standard value is 216.

Range	Standard Value
For 2780: an integer value from 1 to 255	For 2780. 7
For 3780: an integer value from 1 to 255	For 3780: 216

Transparent Blocking Factor

Analogous to the Nontransparent Blocking Factor configuration file element, this factor refers to the number of records that are packed into a block of data to be transmitted in transparent (binary) mode. For 2780 emulation a maximum of 255 records per block are allowed. The standard values are either 1 or 4. For 3780 emulation, you are allowed to enter a maximum value of 255. The standard values are 1 or 6.

Range	Standard Value
For 2780: an integer value from 1 to 255	For 2780: 1 or 4
For 3780: an integer value from 1 to 255	For 3780: 1 or 6

NOTE

For additional information on nontransparent and transparent blocking factors refer to IBM publications:

- *Component Description: IBM 2780 Data Transmission Terminal, GA27-3005*
- *Component Information for the IBM 3780 Data Communication Terminal, GA27-3063*

Block Size

Enter the data communication transmission block size on this configuration line. It may be necessary that this value match the value specified in the host system generation. Generally, larger block sizes are used for highly reliable communication links with long propagation delays. Smaller blocks are used to get data across noisier links. The maximum block size that may be used for either 2780 or 3780 emulation is 512 bytes.

Range	Standard Value
An integer value, in bytes, from 1 to 512	For 2780: 400 For 3780: 512

Maximum Record Size

This file element refers to the length, in characters, of the longest record that will be transmitted to the host system. Longer records will be truncated without warning. For both 2780 and 3780 emulation, the longest record size that may be specified is 255 characters. It may be necessary that this element match the host system configuration.

Range	Standard Value
An integer value from 1 to 255	80

Inactivity Timeout

This element, indicated in seconds, identifies the period of inactivity that will be tolerated on the communications link before the line will be dropped.

The time used here is installation dependent. Shorter times would be useful for reducing charges on switched links, although the value is most often set to a fairly large value between 120 and 300 seconds. An infinite timeout, indicated by entering a value of 0, would usually be used with a leased line.

Range	Standard Value
An integer value, in seconds, from 0 to 65535	240

Retry Limit

This element controls how many retransmission attempts, associated with successive occurences of the same data communication error type, will be made before the line is dropped.

Any integer value between 0 and 255 may be specified. A value of seven is typical, but noisy communications line conditions may require a higher value. Although the retry limit can be raised as a temporary workaround for noise problems, continued necessity of a higher value indicates a communications link problem which should be corrected.

Range	Standard Value
An integer value from 0 to 255	7

Append Trailing Blanks

This configuration element affects 2780 emulation during transmission in nontransparent (text) mode. Normally, under these conditions, no trailing blanks are added for efficient utilization of the datacomm link. The standard setting for this element is therefore NO. Some RJE emulators require transmissions of fixed length records only. To communicate with this less sophiscated type of host, specify YES for this file element. In this case, records shorter than your specified maximum record size are padded out by appending blank characters. Under these conditions, the maximum record size must exactly match that expected by the host. For 3780 emulation you can enter either YES or NO. Trailing blank truncation is automatic and this configuration element is ignored.

Range	Standard Value
YES or NO	NO

Generate WACK/TTD

This file element applies especially to 2780 emulation mode. Actual 2780 workstations do not generate the bisync protocol characters, "Wait before ACKnowledgement" (WACK) and "Temporary Text Delay" (TTD). (The function of these characters is defined in APPENDIX I). RJE/1000-II provides compatability with the traditional 2780 protocol for hosts that expect adherence to the old convention.

To emulate traditional 2780 protocol enter NO on this file line. Under this configuration the PSI card will delay sending an "ACK" in response to a bid from the host system until sufficient buffer space is available to receive the data. If it is ever temporarily unable to transmit, the PSI card will simply pause rather than send a "TTD".

If this element is set to YES, "WACK" and "TTD" protocol characters will be sent when necessary.

Normal 3780 protocol always allows "WACKs" and "TTDs". If doing 3780 emulation you should set this element to YES. If you are doing 3780 emulation and you set this element to NO, "WACKs" and "TTDs" will not be generated.

Range	Standard Value
YES or NO	YES

Datacomm Monitor Trace Control

This element, usually entered as an integer value, lets you set bits in a 16 bit map to turn on the monitor trace and diagnostic options that are available on RJE/1000-II. A zero control word disables all tracing and diagnostics. Setting bits will enable trace and diagnostics options as follows:

- bit 0 tracing of received class headers
- bit 1 tracing of received class buffers
- bit 2 tracing of executed "otherwise" clauses in the subsystem code's Pascal CASE statements
- bit 4 tracing of monitor to PSI data buffers
- bit 14 continuous file posting on the data output streams
- bit 15 continuous file posting on the log device

All bits not specified here are reserved. Setting these reserved bits can cause unpredictable results

More information is available on these trace and diagnostics facilities in Section 6, Troubleshooting RJE/1000-II.

Range	Standard Value
An integer value	0

Example: 100001B turns on all tracing of received class headers and continuous file posting on the log

PSI Trace Control

This integer value controls enabling of PSI card tracing options. Only the eight least-significant bits are used. The others are ignored.

The trace options enabled by the four low order bits are intended for Hewlett-Packard internal use only.

A zero entry for this file element disables all trace options. Setting a bit enables the corresponding function as follows:

- bit 0 trace backplane state machine transitions
- bit 1 trace frontplane state machine transitions
- bit 2 trace transmit midplane state machine transitions
- bit 3 trace receive midplane state machine transitions
- bit 4 trace received bisync control characters
- bit 5 trace received data
- bit 6 trace transmitted bisync control characters
- bit 7 trace transmitted data

More information on the trace functions enabled by setting bits 4 through 7, including an example PSI trace display, is provided in Section 6, Troubleshooting RJE/1000-II.

Range	**Standard Value**
An integer value	0

Example: 360B turns on bits 4, 5, 6, and 7 defined above

System Log Destination

This configuration file element specifies where system event logging and optional trace data will be sent.

If you enter only a file name for this element, the log file is created under your working directory or the working directory of the program that issued the RINIT command. To create the log file under another directory, supply a fully qualified path name.

CAUTION

> Each time the RJE-II subsystem is started, it must create a new log file. If only an RTE operating system file reference appears on this line, you must purge or rename the file, if it exists, prior to each subsystem initialization. If RJE-II attempts to open the file and finds that it already exists, the initialization is aborted.

RJE/1000-II provides you with the option of having the log file purged automatically. If this configuration file element is specified as <RTE file reference>,P[URGE], and the system finds that a file with the given name exists, the file will be purged as part of the initialization process, and a new file will be opened with the name you specify. After each use of the subsystem, if the log file contains data which may be useful later, you should store it elsewhere or rename it.

Use of the purge feature somewhat reduces the reliability of available failure recovery procedures, but its added convenience may make it worthwhile in most routine applications.

If the RJE-II subsystem is used for an extended period of time without being shut down and restarted, the system log file could become quite large. If RJE-II is to be left running indefinitely without interruption, you should periodically redirect the system log in order to limit its size. The means for doing this is explained in Section 4 under the RJE,TRACE command explanation.

Range	Standard Value
determined by file naming restrictions	/SYSTEM/RJE/RJELOG

Output File Masks And User Postprocessors

The next six configuration file elements are pairs of elements used to control the handling of the three types of output considered by RJE/1000-II: jobs routed to the "printer", jobs routed to the "punch", and "unrouted" jobs. Each element pair consists of an output file mask, and the file name of a postprocessor program for manipulating each of the three output types. Refer to Sections 4, Interactive Use of RJE/1000-II, and 5, Programmatic Use of RJE/1000-II, for more information about RJE-II's output handling operation.

Each output file mask element must ultimately contain a fully qualified file name or the output files would be placed on the system session working directory. CI file references to files under your working directory are automatically expanded by RJE-II to a fully qualified path name relative to the root directory. For files under a directory other than your working directory, you must supply the full path name. For both types of naming methods, the expanded fully qualified path name must be less than 64 characters, including punctuation, separators, type extensions, and a four-character sequence number.

NOTE

> The length of the name given to each output file mask element must allow for a four-character sequence number which is appended by the RJE-II subsystem to construct the actual output filename. If you are using the "old file system" (FMGR), you must specify only one or two characters for the filename. If you are using the "new file system" (CI), you must specify only one to twelve characters in the filename, not including type extensions. If you use file names that are too long, the subsystem will not be able to create a sequence of files properly and many more disk accesses will result as each new file is created.

Each user postprocessor should fully identify the path names to the Type 6 (executable) program files you have established to manipulate your output. By having three different file elements, RJE-II allows you to specify seperate programs to handle the three classes of job output. If you don't intend to do postprocessing on any one or more of the classes of jobs, the corresponding element in the configuration file must be entered as a blank line. Comments may remain if identified with a leading comma.

The six configuration file elements are listed here in sequence for your convenience in developing the subsystem configuration:

- **Print Output File Mask**
- **Print Postprocessor**
- **Punch Output File Mask**
- **Punch Postprocessor**
- **Unrouted Output File Mask**
- **Unrouted Postprocessor**

Signon Card Image

In order to complete initialization of your RJE-II subsystem you may need to sign on to the host system. RJE/1000-II provides the option of including the signon card image that is required by some hosts as part of the configuration file. The eighty character card image that you enter on this last line of the configuration file will automatically be sent as the first element of information after the link is established with the host system. No comment may appear on this line.

For those host systems that don't require a signon card image, a blank line may be entered for this file element or the last line of the file may be omitted. In this case, no signon information is sent to the host.

This file element should also be left blank when configuring a subsystem that connects to a host that allows multiple-card signon sequences. Handling of this situation is discussed under "Initializing The Subsystem" in this manual section.

Determine the signon card requirements of your host system and enter the appropriate maximum 80 character string, or a blank line on this element line.

Range	Standard Value
Maximum 80 character string including blanks and special characters	Determined by host system

Example:

For IBM JES2 the signon card has the following syntax:

```
/*SIGNON        RMTnnnn
```

The card image that you enter for this configuration element must start in column 1. Your site identification (RMT*nnnn*) starts in column 16. Replace *nnnn* with the remote number configured for your site on your host system generation.

Some hosts require the signon card image to be in upper case. Some hosts may also have passwords assigned to your remote and/or to the line connecting your system to the host. Use of these passwords may be necessary in the signon sequence with the host system. Check with the host manager for use of remote and line passwords.

This completes the development of the configuration file. Use of this file and completion of the initialization process is accomplished through use of the RINIT command.

INITIALIZING THE SUBSYSTEM

Once you have successfully completed and saved your configuration file, you can initialize RJE/1000-II on your HP 1000 system. The RJE-II subsystem is started, configured, and, for some hosts, signed on to the host system with a single invocation of the RINIT command.

Running RINIT

At your terminal or through your program you run RINIT with a runstring of the form given in the following syntax:

Syntax

```
RINIT,configfile
```

Parameter

configfile The file name for a configuration file you have developed and stored

Discussion

RINIT reads your configuration file and completes the initialization process including resetting, downloading, and configuring the PSI card. RINIT also performs RPing, scheduling, and configuring the datacomm monitor, RJE.*n*.

Even if you specify NOECHO in the configuration file, a number of messages are displayed on your screen when you run RINIT:

- the configuration file comment line
- the ECHO prompt and value:
  ```
  ECHO?
  NOECHO
  ```
- PSI setup messages:
  ```
  Testing PSI...
  Downloading PSI...
  ```

- FmpOpen message for the log file (on the system console):

  ```
  xxxxxx.xxxxxx L00 LG FmpOpen  file:logfile name
  ```

  Any message which cannot be sent to the log file is sent to the system console. The FmpOpen message above cannot be sent to the log file because the log file is not open at that point, and thus appears on the system console.

If you specify ECHO in the configuration file, a prompt is issued to your screen for each configuration file element. The prompt is then followed by the value you have specified for that element. For example, when RINIT checks the Workstation Type configuration element the prompt and value display would look like:

```
Link type 2780/3780 ?
3780                             ,(9) workstation type
```

One such pair of lines are displayed for each configuration file element. In addition, during the process of bringing up the datacomm link, RINIT prints the message: Please establish the RJE link.. At this point you dial the phone or otherwise activate the link.

If a signon card image is included in the configuration file, RINIT also completes the signon to the host system. A signon card image is usually omitted from the configuration file in two situations:

- when a dedicated link exists between your remote and the host system, and specification of the remote identification to the host is not necessary. In this case, establishment of the link is sufficient to allow use of the RJE-II subsystem for data transfer.

- when the signon card image is supplied by running an RJE,SEND request which sends the file containing the necessary card image or images. If you include the card image as part of the configuration file, RINIT performs the RJE,SEND request. If multiple card images are required by your host, you must run the RJE,SEND command yourself either programmatically, or from your terminal after again receiving your operating system prompt.

Once the host successfully receives the required signon information, you get your operating system prompt back and RINIT terminates. The subsystem is now ready to receive data from the host automatically, or to send data to the host under your control.

If any unrecoverable errors occur during the initialization process, RJE-II supplies messages describing the problem and RINIT terminates, abandoning initialization and "cleaning up" as necessary. This includes resetting the PSI card if it has been used, dropping the link if it has been established, closing all open files, and adjusting the class tables. Information on initialization error conditions can be found in Section 6, Troubleshooting RJE/1000-II.

INTERACTIVE USE OF RJE/1000-II

SECTION

4

INTRODUCTION

With RJE/1000-II installed and initialized on your HP 1000 you may now use the subsystem to transmit and receive data from your host system. This section discusses the runstrings you will use to manage interactive utilization of the subsystem.

This manual section also details the information you need to handle data reception from the host system. Included is information about:

- output file masks
- transparent and non-transparent data reception
- postprocessing received data

RJE RUNSTRINGS

Interactive use of RJE/1000-II is simplified by the use of the single interface program, RJE. You enter RJE runstrings with the general syntax:

```
[RU,]RJE,keyword,n,password,parameters
```

The six values of *keyword*: SEND, STATUS, ABORT, CLOSE, TRACE, and, ROUTE determine the basic function of the runstring. For each of the keywords, the basic function is modified by use of the available *parameters*. The subsystem number, *n*, and the *password* are both positional parameters. You must include the delimiting commas even if you chose to omit *n* and/or *password*.

If *n* is left out of the runstring, the subsystem number defaults to 1 (one). You may omit *password* from the runstring only if, in the configuration file, you entered 0 for the password element, or left that line blank.

Error messages associated with the RJE request you are making can be disabled by using a negative *n* (-*n*) in the runstring (the same effect can be created when making programmatic RJE requests). These messages are normally sent to the terminal from which the RJE request was entered. This does not affect log file entries or error information generated by the RTE operating system.

The following are acceptable runstrings:

```
CI>RU,RJE,SEND,3,rr,JCL1,a,MAP.DAT,b,JCL2,a

CI>RJE TR,,,0 360B RJELOG2
```

The syntax and use description for each of the keywords in the RJE program follows.

RJE,SEND

Data transmission from your HP 1000 to the host system is triggered by using the RJE program with the SEND keyword.

Syntax

[RU,]RJE,SE[ND],*n*,*password*,*fileref*$_1$,*mode*$_1$[,*fileref*$_2$,*mode*$_2$]...

Parameters

n — the subsystem number. Must match the number in the configuration file for the subsystem being used.

password — the subsystem password. Must match the password in the configuration file for the subsystem being used.

fileref$_x$ — the file reference to the file being transmitted. If the file is under your working directory, only the file name need be supplied. If the file is under another directory, you must supply the fully qualified path name.

mode$_x$ — the mode of the file being transmitted. Enter a for ASCII, and b for binary.

NOTE

RJE,SEND requires that you include at least one *fileref,mode* indication, but you have the option of including up to eight such pairs. **For each *fileref* you must also include the *mode* of that file.**

Discussion

Method of Operation

For each RJE,SEND request you enter, one job stream is sent to the host. The RJE clone that handles the processing passes the file reference and mode information specified in the runstring to the datacomm monitor. The clone then suspends until the monitor has transmitted each of the files, in order, to the host.

The actual transmission starts when the link to the host is idle. If you are sending or receiving large files from the host, there may be considerable delay before your files are actually sent and your operating system prompt is returned. If such a delay in terminal availability is not acceptable, and you expect no errors to occur in transmission, you can run the program without wait.

If your transmission is successful, the clone receives an indication that the last block has been sent, and your operating system prompt is returned. If a transmission error occurs and the transmission is interrupted, the RJE clone transfers status information to you indicating which file was being transmitted when the error occured, and what type of error it was. More information on error messages and codes is available in Section 6, Troubleshooting RJE/1000-II, and Appendix A.

Multiple File Capability

RJE/1000-II allows you to include from one to eight files as part of each job stream through use of multiple *fileref,mode* parameter pairs in the runstring. The file references should include as much information as is necessary for the subsystem to locate your files. If the files you are transmitting are under your working directory, you only need to include the file name. For any partial path name under your working directory RJE-II automatically expands the reference to include the fully qualified path for its use. The fully qualified path name is limited to 64 characters including punctuation, special characters, separators, type extensions, and the appended four-character sequence number.

If the files exist on another directory, you must include a fully qualified path name for each *fileref*.

Files that are located on FMGR "group" or "private" cartridges are not accessible by the RJE-II subsystem.

CAUTION

> Using the multiple file capability of RJE,SEND may be convenient, since it eliminates the need for merging job stream components prior to transmission. However, if an FMP error is encountered on a reader file during transmission, an end-of-text block character (ETB) is sent to the host at the end of the error block. When the host sends an acknowledgement, an end-of-transmission character (EOT) is sent to the host and the remainder of the RJE,SEND request is flushed. Depending on what was in the partial data stream sent, the host may attempt to process whatever it received even though it was incomplete.

When RJE,SEND is scheduled it checks if all the files referenced in the runstring exist before it continues processing, but it does not check for the integrity of the files.

Variable Mode Capability

The requirement of including the mode (ASCII or binary) with each file reference, allows you the capability of including a mixture of both ASCII and binary files in a single job stream. You can, for example, include JCL in ASCII format as part of an otherwise all binary transmission.

Example:

```
CI>RJE,SEND,1,,JCLSTRT,a,gridspec,b,JCLEND,a
```

The binary data file `gridspec` is transmitted with starting and ending ASCII JCL statements.

File Translation

The RJE-II subsystem always uses EBCDIC as the transmission line code. Text files on your HP 1000 are assumed to be ASCII. All files sent in non-transparent mode are therefore translated to EBCDIC before transmission. Translation is performed according to one of several interchangeable character set tables. This method of translation makes possible support of various local languages. Files sent in binary mode are not translated by the subsystem. More information on character sets is available in Appendix H.

Configuration Considerations

If you are using the subsystem in 2780 emulation mode, you may send text files either with trailing blanks truncated for transmission efficiency, or with trailing blanks appended so that fixed size records are sent. This capability is controlled in the configuration file, and is discussed in Section 3, Configuring And Initializing RJE/1000-II, under "Append Trailing Blanks".

RJE,STATUS

You can obtain status information on the current monitor, PSI card, and data communications link states as well as some long term transmission statistics through the use of the RJE program with the STATUS keyword.

Syntax

```
[RU,]RJE,ST[ATUS],n,password[,type]
```

Parameters

n the subsystem number.

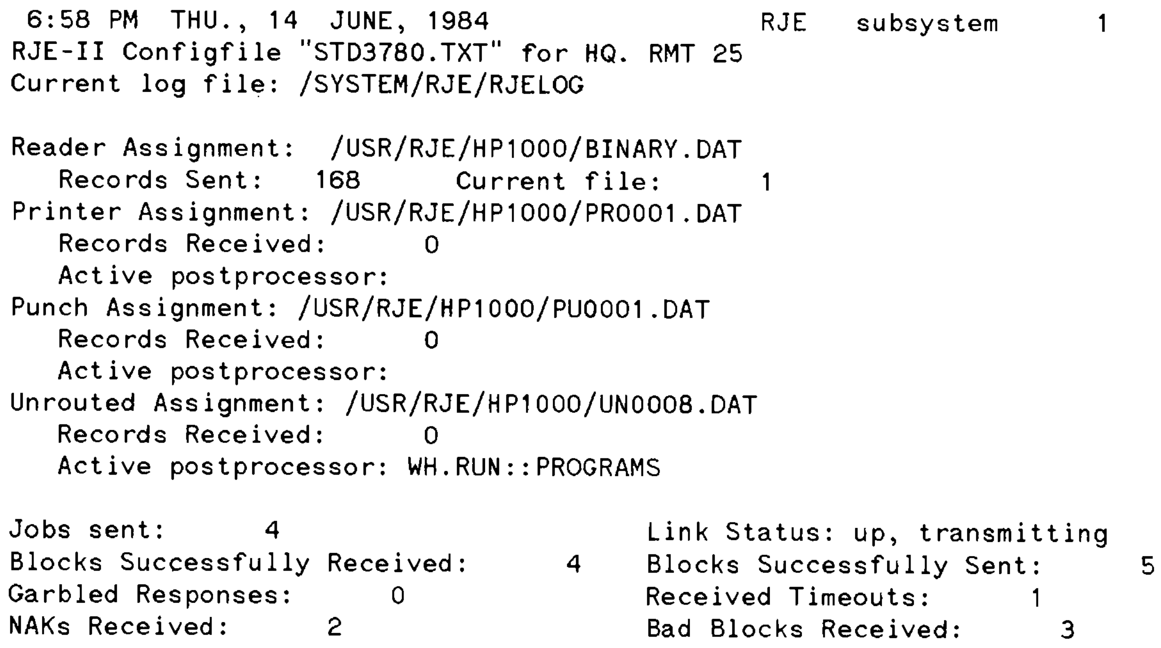
password the subsystem password.

type specifies the type of status information requested. *This parameter is reserved for programmatic status calls. Its use suppresses display of messages on your terminal.*

Discussion

You can issue STATUS requests both interactively and programmatically. If you use it programmatically, the syntax and the function of the program are different than when you use it interactively. More information on the programmatic use of RJE,STATUS can be found in the Section 4, Programmatic Use Of RJE/1000-II.

When you use it interactively, RJE,STATUS returns a display to your terminal. An example STATUS display is shown in Figure 4-1.



```
 6:58 PM  THU., 14  JUNE, 1984                RJE   subsystem     1
RJE-II Configfile "STD3780.TXT" for HQ. RMT 25
Current log file: /SYSTEM/RJE/RJELOG

Reader Assignment:  /USR/RJE/HP1000/BINARY.DAT
   Records Sent:   168      Current file:      1
Printer Assignment: /USR/RJE/HP1000/PR0001.DAT
   Records Received:      0
   Active postprocessor:
Punch Assignment: /USR/RJE/HP1000/PU0001.DAT
   Records Received:      0
   Active postprocessor:
Unrouted Assignment: /USR/RJE/HP1000/UN0008.DAT
   Records Received:      0
   Active postprocessor: WH.RUN::PROGRAMS

Jobs sent:      4                       Link Status: up, transmitting
Blocks Successfully Received:      4    Blocks Successfully Sent:      5
Garbled Responses:     0                Received Timeouts:       1
NAKs Received:      2                   Bad Blocks Received:      3
```

Figure 4-1. Interactive RJE,STATUS Display

The first line of the STATUS display contains time and date information and the subsystem number. The second line displays the comment string from your subsystem configuration file. The Current log file is the identification of the file to which you are currently sending log and trace information.

The next two lines define the **current reader file position**. The Reader Assignment is the disc file that is currently being "read in" by the subsystem. Current file identifies the sequence number of the file being transmitted from the list of files supplied in the SEND request. This number is always 1 unless multiple files are being sent with one SEND request. Records Sent tells you how many records have been transmitted from the Current file of the Reader Assignment.

The Printer, Punch, and Unrouted Assignments are the file names for the other logical devices being used by your subsystem. The Printer, Punch, and Unrouted Assignments include the appended four digit sequence number of the specific file being used. RJE-II also displays the number of records that have been received for each assignment, and the names of any Active postprocessor programs.

The remaining STATUS information, dealing with transmission statistics, accumulates from each resetting of the PSI card.

- Jobs sent essentially refers to the number of job streams that have been sent to the host. It specifically refers to the number of end-of-text (ETX) bisync protocol characters that have been acknowledged by the host.

- The Link Status is displayed with one of five possible states indicated:

  - down
  - up, idle
  - up, transmitting
  - up, receiving
  - RVI pending

- Garbled Responses are accumulated when you are receiving data and an unexpected response is received from the host. These are most often caused by excessive noise on the data communications link.

- Receive Timeouts occur when no response is received from the host before the timeout period is reached.

- Bad Blocks are those which contain an invalid bisync Cyclic Redundancy Check (CRC).

RJE,ABORT

To abort current specific subsystem activity and return the data communications link to control mode (both the host and the HP 1000 can bid for link control), submit an RJE request with the ABORT keyword.

Syntax

```
                                  ⎧CU[RRENT]⎫
[RU,]RJE,AB[ORT],n,password,⎨QU[EUE]  ⎬
                                  ⎩RE[CEIVE]⎭
```

Parameters

n — the subsystem number

password — the subsystem password

CU[RRENT] — aborts the currently active SEND request

QU[EUE] — aborts the currently active SEND request and all other SEND requests on the datacomm monitor's logical reader queue

RE[CEIVE] — aborts data being received from the host at the time the request is issued.

Discussion

RJE,ABORT is used to interrupt the flow of data between your HP 1000 and the host system, and return the data communications link to the control mode as soon as possible. While the ABORT request itself completes very quickly, the end effect of the request may take several seconds or more to complete. The total delay is dependent on the type of ABORT requested and on the status of your HP 1000 system. **When you submit an ABORT request, you must include one of the options (CU, QU, or RE) or a warning message will be issued and the request will not be processed.**

Current

If the CURRENT (CU) parameter is specified, transmission of files included in the currently active RJE,SEND request is terminated. The termination occurs at the current reader file position. (See "RJE,STATUS" for explanation of current reader file position). The last block of data sent to the host is terminated by an ETB bisync protocol character. This is followed by an EOT indicating that the link is to return to control mode. The remainder of the active RJE,SEND request is then purged from the datacomm monitor queue.

The ETB-EOT sequence are the last characters sent to the host for the currently active RJE,SEND request. When the host system receives the ETB-EOT sequence it is expected to ignore all data associated with the aborted SEND request. The host may then rebid for link control and should resume normal operation.

When the link returns to control mode, the RJE-II subsystem also resumes normal operation. If SEND requests are in the logical reader queue, RJE-II waits one second. It then checks to see if the link is still in control mode. If so, it bids for link control and sends the awaiting files. Anytime RJE-II sends an EOT it delays one second before bidding for the link. This allows the host to get control of the link and send any data it has back to your HP 1000.

Queue

If the QUEUE (QU) parameter is specified, the currently active SEND request is aborted, just as when using the CURRENT parameter. In addition, all other SEND requests currently on the datacomm monitor's logical reader queue are purged.

After the ABORT,QUEUE request completes, both RJE-II and the host system will be in control mode and able to bid or accept a bid for link control.

General Considerations for Current and Queue

The expected use of the CURRENT and QUEUE versions of the ABORT request is to terminate and eliminate SEND requests you have decided not to transmit. If the ABORT request is successful, the SEND request receives a message indicating which file was aborted. If you are using the subsystem interactively, the message is displayed on your terminal. If the ABORT request is unsuccessful, an error message is displayed on your screen. For example, if you submit the ABORT request after the last record has been sent to the host, the request will have no effect and you receive a message: `No SEND request outstanding`. See Appendix A, "RJE Error Messages And Codes", messages 24-31 for other ABORT messages.

In all uses of the CURRENT and QUEUE versions of ABORT, there is a time delay between submission of the ABORT request and termination of the aborted SEND request. This delay is caused by the time necessary for the data pipeline between your HP 1000 and the link to empty so the abort sequence can be transmitted to the host. If you are receiving data from the host at the time you submit the ABORT request, the delay is increased while the PSI waits to be able to bid for and get control of the link, and empty the pipeline.

Note that in all cases, when you abort a currently active SEND request, those data blocks already in the pipeline between the monitor and the link will be transmitted to the host. This is true even if you are receiving data and the SEND data is not yet being transmitted.

Receive

The RECEIVE (RE) parameter is used primarily to stop data reception so you can send files to the host. If RE is specified, an EOT bisync character is sent to the host as soon as possible. If the host system is operating under standard bisync protocol rules, this causes the host to stop transmitting and return the link to control mode. When the host receives the EOT it should delay for a short period before rebidding for the link. This delay should allow RJE-II to get the link and begin sending files. To have this desired effect, though, you should submit the SEND request to get data in the pipeline before you submit the ABORT,RE request. Otherwise the PSI will not bid for the link immediately, and the host may control the link and begin resending the transmission you just aborted.

RJE,CLOSE

To drop the data communications link with the host system and terminate your current subsystem session, you submit an RJE request with the CLOSE keyword.

Syntax

```
[RU,]RJE,CL[OSE],n,password[,NOW]
```

Parameters

n the subsystem number.

password the subsystem password.

NOW specifies that the data communications link and subsystem are to be terminated immediately.

Discussion

When you submit an RJE request with the CLOSE keyword, the datacomm link with the host is dropped and the subsystem is brought down. RJE-II closes all open files, restores the system to its initial state, and terminates the datacomm monitor.

When the optional NOW parameter is omitted, these shutdown procedures are delayed until all SEND requests have been completed and the link to the host goes idle. If NOW is included, the CLOSE procedures, including cleanup, are accomplished immediately regardless of current data communications activity. *You should make sure that there are no other pending requests on your subsystem before you initiate the* CLOSE,NOW *process.*

NOTE

> Some host systems require that you "sign off" before terminating your workstation emulator and dropping the link. This procedure allows the host system to clean up its resources and will enable you to "sign on" more efficiently for your next session. For IBM systems, signing off requires the transmission of a SIGNOFF card image. The format of the SIGNOFF card is:
>
> ```
> /*SIGNOFF
> ```
>
> The sign off statement must start in column 1 of your card image line. Check for sign off requirements from your host system.

RJE,TRACE

With the subsystem running, you can perform on-line modifications to the PSI and Monitor trace control words and the log file specification that you made in the configuration file. You do this by using the RJE program with the TRACE keyword, and associated parameters. These configuration elements are discussed in greater detail in Section 3, Configuring And Initializing RJE/1000-II.

Syntax

```
[RU,]RJE,TR[ACE],n,password,MONtrace,PSItrace,logfile
```

Parameters

n — the subsystem number

password — the subsystem password

MONtrace — an integer value used to set bits which control tracing of the subsystem datacomm monitor activity.

PSItrace — an integer value used to set bits which control tracing of PSI card activity.

logfile — an RTE file specification for the subsystem log destination.

Discussion

Values for all the parameters indicated in the TRACE syntax must be supplied with each invocation of this runstring. The values for the trace control words and log file destination that you specify in the TRACE request immediately replace the current values, taken either from the configuration file when the subsystem was initialized, or a previous TRACE request. They remain in effect until changed with another TRACE request, or until the subsystem is brought down. Using RINIT to bring the system back up reestablishes the values from the configuration file.

NOTE

> You should take some care in the changes you make in the log file specification. An incorrect specification can defeat the failure recovery features built into RJE/1000-II. Frequent modification can fragment the record of the subsystem activity.

In installations where the RJE-II subsystem is left running indefinitely, the RJE,TRACE request can be used to redirect the system log file periodically. This technique keeps the log file from growing without bound and consuming excessive disc space. This procedure may be carried out by the system operator, by an application program, or by means of a transfer file. For example, the following transfer files, calling one another and making use of the CI time-scheduling command, AT, will switch between two log files every twelve hours.

AMTRACE

```
pu rjelog.am
rje trace 1 sc 0 0 rjelog.am
at 12:00:00 noon ci pmtrace
```

PMTRACE

```
pu rjelog.pm
rje trace 1 sc 0 0 rjelog.pm
at 12:00:00 midnight di amtrace
```

This approach guarantees that there will always be a record of subsystem activity spanning at least twelve hours, but never exceeding twenty-four.

RJE,ROUTE

You can make on-line changes to the output handling specifications established in the subsystem configuration file by using the RJE program with the ROUTE keyword. While the system is running, you may use this runstring to alter the output file mask and postprocessor specifications for each of the logical output devices; printer, punch, and unrouted.

Syntax

```
                                ⎧PR⎫
[RU,]RJE,RO[UTE],n,password,⎨PU⎬,mask[,postprocessor]
                                ⎩UN⎭
```

Parameters

n — the subsystem number.

password — the subsystem password.

PR, PU, or UN — the logical device specification. Enter either PR for printer, PU for punch, or UN for unrouted.

mask — the output file mask specification. Must adhere to file system naming restrictions and allow for appended four digit sequence number.

postprocessor — the file name for the postprocessor used to handle data received for the device specified.

Discussion

The syntax and permissible values you may specify for the output file mask and postprocessor names in the ROUTE request are exactly the same as indicated under "Output File Masks And User Postprocessors" in Section 3. These values are found on lines twenty five through thirty of the configuration file. You may determine the current values by submitting a STATUS request.

The ability of the subsystem to make the changes you indicate in the ROUTE request depends on the status of the output files whose mask or postprocessing you are attempting to modify. If the open file in the current mask sequence is empty, the specified changes take place immediately. That file is purged and the first file in the new mask sequence is opened. Files created with the original file mask that contain data are left intact. If you have requested postprocessing changes, they take place the next time the file receives data.

If the open file in the current mask sequence contains data, the specified changes take place the next time the file sequence is incremented by the subsystem. Incrementing of output file names is described under "Data Reception" in this manual section.

You must specify the parameters up to and including *mask* in the program runstring. Leaving the *postprocessor* specification out of the runstring eliminates all postprocessing for the file mask you are modifying.

The changes you make with the ROUTE request will stay in effect until output for the same logical device is modified with another ROUTE request, or until the subsystem is brought down. Running RINIT when you bring the subsystem back up reestablishes the values from the configuration file.

DATA RECEPTION

Data received from your host system is handled automatically by RJE/1000-II. This is based on information you supply in the configuration file or in a ROUTE request. Each file received from the host system is routed to one of three output files corresponding to the three logical output device specifications recognized by RJE-II. No intervention is required by you to ready the subsystem to receive data.

Output Destination Selection

The selection of a printer, punch, or unrouted destination for received files is based on the recognition of a component selection escape sequence by the datacomm monitor. The escape sequence must be the first text character in the first block of text in a received transmission. RJE-II responds to these escape sequences like an actual 2780 or 3780 workstation.

If you are using 2780 mode, the component selection is as follows:

- A punch file is selected by esc 4
- A printer file is selected by any valid printer control escape sequence as displayed in the following table:

TABLE 4-1. VALID PRINTER SELECT CODES

Received EBCDIC Escape Sequence	Actual 2780 Printer Operation
esc /	Single Space
esc S	Double Space
esc T	Triple Space
esc A	Skip to Channel 1
esc B	Skip to Channel 2
esc C	Skip to Channel 3
esc D	Skip to Channel 4
esc E	Skip to Channel 5
esc F	Skip to Channel 6
esc G	Skip to Channel 7
esc H	Skip to Channel 8

- An unrouted file is selected by no escape sequence, or any escape sequence other than those listed for punch or printer.

If you are using 3780 mode, the component selection is as follows:

- A printer file is selected by a DC1 sequence
- A punch file is selected by a DC2 or DC3 sequence
- An unrouted file is selected by the absence of any of the characters used to select the printer or punch

Output File Masks

In the configuration file you identified an output file mask for files routed to the logical printer, to the logical punch, and for unrouted files. The output file masks are simply RTE file references. The files are created on your working directory unless a fully qualified path name to another directory is specified in the configuration file or as part of a ROUTE request.

The subsystem utilizes the masks to create a sequence of files to receive the data from the host system. The names of the actual files consist of the output file mask with an appended four digit sequence number.

When the subsystem is first started, all output file sequence numbers are set to zero. Similarly, if a ROUTE request is used to change an output file mask, the new output file sequence number will be zero. The first file received from the host will be directed to the output file with sequence number zero. The second file will be directed to sequence number one. Thus, if data intended for the logical punch were directed with the mask "/data/rjpunch", the first file received from the host system would be placed in the file "/data/rjpunch0000", the second in "/data/rjpunch0001", the third in "/data/rjpunch0002", etc. In identifying your output file masks you must allow for the appended sequence digits to adhere to the file naming length restrictions for the file handling system you are using (FMGR or CI). Fully qualified path names are limited to 64 characters including the appended 4 digit sequence number.

You should be aware that RJE-II requires only the reception of an ETX or an EOT bisync protocol character to create an output file. Some host systems may transmit sequences of just these characters causing RJE-II to create sequences of empty output files.

If a file name that results from the output file mask and appended sequence digits already exists, the subsystem increments the sequence number until an unused file name is found. As a result, RJE/1000-II will never write over existing data, but the system efficiency suffers as the number of these collisions increases. In addition, your ability to identify and locate a particular file within a group of similarly labeled files may be reduced. If you create files "rjpunch0000" through "rjpunch0025 during one subsystem session, then purge "rjpunch0005", the next time you run the subsystem and receive punch files, "rjpunch0005" will be used as the first storage location, followed by "rjpunch0026", etc. Although you will have a continuous sequence of files, adjacent files may be unrelated. *All files whose names will appear in a one of the subsystem queues should be renamed or purged before starting RJE/1000-II to avoid these problems.*

NOTE

When a file with the sequence number "9999" is received in any of the three file queues, the subsystem issues itself a CLOSE,NOW request. The link to the host system is dropped and the subsystem is disabled until you resolve the situation. This feature of RJE/1000-II is necessary to avoid potential data loss. The product will not accept data from the host unless it can locate an output file in which the data can be placed.

Transparent Data Reception

Files received by RJE/1000-II may contain either text or binary data. The two modes, selected by the host system's use of transparent (binary) or non-transparent (text) transmission protocol, result in significantly different processing by your HP 1000.

When receiving transparent files, the data is kept exactly as it is received. RJE-II does not translate the data. If you are running the subsystem as a 3780 emulator, transparent data consists of concatenated records without any separator characters. Multiple record 3780 transparent blocks create only one record in the output file. A postprocessor is required to split these blocks.

If you are running the subsystem as a 2780 emulator, RJE-II blocks the received records according to the location of the ITB characters in the data stream.

Note that punch data streams are sent from IBM hosts as transparent data.

Nontransparent Data Reception

Data that is transmitted from the host in non-transparent mode is in EBCDIC. Since files on the HP 1000 are assumed to be in ASCII, RJE/1000-II automatically translates text data from EBCDIC to ASCII before routing it to one of the output files.

Formatting of received non-transparent data is ignored by the RJE/1000-II subsystem except for tabulation. If any record of a non-transparent printer file contains an "escape horizontal tab" sequence, the rest of that record is assumed to be the tabulation specifications. Accordingly, the datacomm monitor does tabulation of the data and inserts required blanks for the rest of the file (or until the next "escape horizontal tab" sequence) as it routes the data to the specified printer output file. If no "escape horizontal tab" sequence is encountered, a default tabulation pattern is applied. The default pattern is a horizontal tab character in every position. Functionally, the default value redefines the horizontal tab character as a space character.

For either transparent or non-transparent data, when the last record for a given file is received from the host, the PSI card triggers the datacomm monitor to close the output file to which the data is being routed. The monitor closes the file, increments the output file name sequence and opens the next file in the sequence. At this point, the monitor also schedules any postprocessing that you have specified in the configuration file or with a ROUTE request.

Postprocessors

With the automatic scheduling of postprocessing, RJE/1000-II allows you to manipulate the data immediately upon reception. Once the received file is closed, the datacomm monitor triggers the operating system to make a copy of the postprocessing program that you have specified. The monitor then supplies the name of the file it has just closed as input to the postprocessing program.

An example postprocessing program is supplied with RJE/1000-II to enable you to print your data on a typical line printer. The postprocessor example is shown in Appendix C. An example punch data postprocessor is also displayed in Appendix C although it is not supplied on the RJE-II product media.

PROGRAMMATIC USE OF RJE/1000-II

SECTION 5

INTRODUCTION

Section 4 discusses the use of runstrings which you can use interactively to gain access to the RJE-II capabilities. For highly customized applications, you can also issue these runstrings from programs you write. RJE-II itself also incorporates the use of special programmatic status calls and an automatic postprocessor scheduling facility.

Customizing RJE/1000-II

Although these features do not constitute direct "programmatic access" to the data communications link, they do provide a basis for the implementation of such custom application services as:

- fully automated operation
- customized user interface
- job management
- auto-restart after link or power failure
- integration of RJE/1000-II into specialized application software

This manual section provides information on the interaction between RJE-II and user programs. This information can be used by a systems programmer to implement the types of applications listed above.

To make use of this section, you should be familiar with RTE concepts such as EXEC calls and parameter passing conventions. Information on these topics is provided in *RTE-A Programmer's Reference Manual*, 92077-90007, or *RTE-6/VM Programmer's Reference Manual*, 92084-90005.

Appendices B, C, and D contain example programs demonstrating the use of the topics covered in this section.

Working Directory Considerations

File references for a programmatically scheduled RJE request are relative to the working directory of the scheduling program. This means that if an application program is detached from your session when it issues an RJE request, file references are made relative to the system-session working directory. If this is not desired, you must specify fully qualified pathnames in your file references.

Error Message Passing

Textual error messages of the type displayed when you use the subsystem interactively are also created during programmatic use. These messages are output to the terminal identified as the log-LU of the program that scheduled the RJE request. *To disable display of any error messages associated with the* RJE *requests in your application program, you can use negative values of the subsystem number in these requests.*

The RJE and RINIT programs also return machine readable error/status information via the PRTN mechanism. Two integer values are passed back and picked up in RMPAR_1 and RMPAR_2. The meanings of these integer values are defined in Appendix A. If EXEC scheduling is used, the scheduling program must execute a call to RMPAR immediately after the RJE process terminates. RMPAR recovers five integers, only the first two of which are significant. If FmpRunProgram is used for scheduling, the return values are recovered automatically and placed in the "parms" array specified in the FmpRunProgram call. In either case, the returned values are meaningful only if the RJE or RINIT program was scheduled with wait.

PROGRAMMATIC STATUS CALLS

If you are using the subsystem interactively, RJE,STATUS displays a formatted output of selected status information, as shown in Figure 4-1. For advanced applications, you may need more detailed information. For this purpose, an extended form of the RJE,STATUS runstring is used.

As indicated briefly in Section 4, under "RJE,STATUS", there is a fourth runstring parameter reserved for programmatic use. This fourth parameter is used to select the type of status data you need. Six different types are available. For each type there is a two-character selector. Either the character selector listed below, or an integer equivalent can be entered in the STATUS runstring.

- MC -- monitor configuration
- LS -- link statistics
- RD -- logical "reader" device status
- PR -- logical "printer" device status
- PU -- logical "punch" device status
- UN -- device status for unrouted data streams

If one of the legal type selectors is present as the fourth runstring parameter, the function of the RJE,STATUS runstring is redefined. No status information is output to the scheduling LU. Instead a machine readable buffer is dispatched to the calling program via an EXEC 14 call. After the programmatically scheduled RJE request completes, the scheduling program can retrieve the buffer with another EXEC 14 call.

The sizes and formats of the returned buffers vary, depending upon the type of status requested. PASCAL definitions of these buffers appear in the Console Application Program listed in Appendix B. A FORTRAN programmatic example, STAT.FTN, which makes use of programmatic status calls, is included on the RJE-II product media. STAT.FTN and an example of the status display it generates are listed in Appendix D.

The following sections detail the meanings and formats of the various data buffers. Note that the first word of each buffer is refered to as word 1, not word 0.

Monitor Configuration

The MC option used in the RJE,STATUS runstring yields information on the configuration of the datacomm monitor (RJE.*n*).

Word	Meaning
1	subsystem number [integer]
2	subsystem password [integer]
3	the "read" LU of the PSI card [integer]
4	reserved for HP internal use
5	the "write" LU of the PSI card [integer]
6	reserved for HP internal use
7	link type [integer] (1=2780, 0=3780)
8	maximum block size [integer]
9	maximum record size [integer]
10	append (zero) or truncate (nonzero) [integer]
11	start of configuration file comment line [80-byte packed ASCII string]
50	last word of buffer (last two characters of comment line)
51	start of log file name [80-byte packed ASCII string]
90	last two characters in log file name buffer

Link Statistics

The "LS" status option retrieves current statistics and other information from the PSI card firmware.

Word	Meaning
1	reserved for HP internal use
2	reserved for HP internal use
3	"board type" and "link status" [two packed bytes]
4	PSI firmware revision code [integer]
5	files sent (ETX's ACK'ed) [integer]
6	garbled responses received [integer]
7	receive timeouts [integer]
8	NAK's received [integer]
9	bad blocks received [integer]
10	blocks successfully received [integer]
11	blocks successfully sent [integer]

The "board type" field is used to identify the PSI card code as RJE firmware. The upper byte of word three will contain the value 8 if the correct RJE download file (RJE.DAT) was specified. The lower byte of this word is a bit array used to indicate the status of the datacomm link. Only one bit is set at any given time. The meanings of the bits are:

Bit	Meaning
(lsb)0	link is down
1	link is down/open link pending
2	link is up and idle
3	link is up, transmitting
4	link is up, receiving
5	RVI pending
6	reserved for HP internal use
(msb)7	reserved for HP internal use

The firmware revision code is an integer value which identifies the particular version of the PSI download code present in your system.

Device Status

The remaining four status options all return a buffer of the same format. Some of the items are applicable only to output devices, or only to the "reader" (SEND) device.

Word	Meaning
1	reserved for HP internal use
2	sequence number of current file [integer]
3	number of records received or sent [integer]
4	start of output file name [80 byte packed ASCII string]
44	start of postprocessor file [80 byte packed ASCII string] (output file only)
84	reserved for HP internal use
85	class number of RJE,SEND program (reader only) [integer]
86	time of last open [double-integer, TimeNow format]
88	time of last close [double-integer, TimeNow format]

The file sequence number, word two, contains the four-digit sequence number for output files. For reader (SEND) files, this number indicates which file in a multiple-file SEND request is currently being transmitted.

OUTPUT POSTPROCESSORS

RJE/1000-II supports automatic scheduling of output postprocessors. This support permits custom postprocessing steps to occur automatically upon data reception, and to offload time consuming presentation services processing from the real-time datacomm monitor.

You perform postprocessor specification at subsystem configuration time, as explained in Section 3 under "Output File Masks And User Postprocessors", or change these specifications with an RJE,ROUTE request as explained in Section 4 under "RJE,ROUTE".

When a job reception is complete, the postprocessor associated with the job's output type (printer, punch, or unrouted) is cloned with FmpRpProgram and EXEC-scheduled. One parameter, the name of the newly created file, is passed to the postprocessor in its runstring. Postprocessors are scheduled without wait.

If the postprocessing time for any output file may exceed the time required to receive the next job, multiple clones of the postprocessor may exist simultaneously. This should be taken into account when programming the postprocessor. For instance, a postprocessor which writes to a line printer should utilize an LU lock in order to prevent interleaved output from multiple copies of the program. The printer postprocessor listed in Appendix C and supplied on the RJE-II product medium performs in this fashion.

Postprocessors run in system session, since they are scheduled by RJE.*n* which runs in the system session.

NOTE

If you are operating under an RTE-6 system, the ID segment created for each execution of a postprocessor remains on the ID segment table. On the RTE-6 operating system, a program must "off" itself (CI>OF) to remove its ID segment from use. Without this, the operating system would run out of ID segments. To OF itself, the program makes a call to PNAME to get the name the operating system uses for that program. This name is entered into a buffer (called MESBUF in the postprocessor program FMT.FTN) and then used in the call to MESSS. This call makes the operating system use the MESBUF buffer as if it were a command line entered by a user. Since the MESBUF buffer is initialized with the OF,*xxxxxx*,ID command, the RTE-6 system "offs" the program making this call.

This call procedure only need be done in RTE-6. The code for this may be lifted right from FMT.FTN. The necessary part is the four lines beginning with the label 9999. Also necessary are the declarations for the MESBUF buffer, and the OPSY function (which can identify the operating system). In addition, the DATA statement to initialize the MESBUF buffer is necessary.

TROUBLESHOOTING RJE/1000-II

SECTION 6

INTRODUCTION

This manual section presents an overview of the subsystem design, and discusses various subsystem failure recovery procedures.

OVERVIEW OF SUBSYSTEM DESIGN

RJE/1000-II is a data communications subsystem comprised of several routines implemented on both the HP 1000 system and on the PSI. These routines are designed to operate concurrently and to communicate data and status information to each other. The PSI processes the link activity during connection between the RJE-II subsystem and the host system. The 2780/3780 workstation emulation is implemented with concurrently operating processes consisting of the PSI and the datacomm monitor clone, RJE.*n*. The processes, together with the functions performed by the user interface program, RJE, and the services provided by the operating system, constitute the RJE/1000-II subsystem.

The operations of the host system are outside of Hewlett-Packard control but must be considered as part of the complete communications system. To properly configure and operate the RJE-II subsystem, you need a basic knowledge of the host system support of 2780/3780 data communications. Problem situations arise from a lack of correspondence between the configurations on the host side and on your RJE-II subsystem side. Information to help you develop sufficient correspondence can be found in Appendix E, Host System Generation Considerations.

PSI Services

The link handler running on the PSI card controls the 2780/3780 bisynchronous protocol. The PSI is RAM-based, so that the link handling firmware is loaded onto the PSI (i.e., downloaded) from the disc resident download file on your HP 1000. Identification of the download file must be provided in the subsystem configuration file as described in Section 3, Configuring and Initializing RJE/1000-II. Maintenance of the line connection and the integrity of data transmission are implemented by the code resident on the PSI. The PSI installation verification procedure is described in Appendix G.

UNEXPECTED LINK TERMINATION

An unexpected break in the communication link can occur in several situations. First, if the modem signals that Data Set Ready is down, the PSI will generate a "link down" interrupt to RJE-II. Second, if the link has been idle for a host-configured time interval, the host may close the line. Third, if noise on the transmission line causes the transmission retry count to be exceeded, the link will go down. Fourth, if the host is no longer responding to the HP 1000 line bid, the PSI will produce a link down interrupt.

When the link goes down unexpectedly, the host handles data sets it is currently transmitting according to its configuration. It may rewind each file to the last job entry subsystem checkpoint, or it may rewind each file to its beginning. Some hosts may flush any input buffers for transmissions coming from your HP 1000.

On its side, your HP 1000 closes the files associated with currently active output data streams and aborts any active reader devices (RJE,SEND requests). The input files from the aborted readers remain intact, and are listed in the subsystem log.

NOTE

> RJE/1000-II purges any empty output files that it closes. For any output file mask, the file with the highest appended four digit sequence number should contain the last of the information received from the host. If you have questions about the currency of your files you can issue an RTE dl command with the a (last access) or c (creation date) option.

After closing the files, RJE.*n* resets the PSI card, returns all class numbers to the operating system, closes the log file, and restores RJTAB to its original condition. Finally, RJE.*n* terminates.

Recovery

To recover from an unexpected link termination, after you have resolved the cause of the link drop, you must perform the same procedures as you do when you are bringing the subsystem up for a normal session. For RJE-II, this simply means running RINIT. You should remember that if you have included the purge option for the log file in the subsystem configuration, each time you run RINIT the current log will be purged. If you wish to save the current log file and use it for diagnostic purposes, be sure to rename it or change the log file name in your configuration file.

Once RINIT completes and the link is back up, some host systems will retransmit the interrupted files from the last checkpoint or from the beginning, depending on their configurations. Many systems to which you may connect don't have these automated characteristics. You should assure that files being transmitted from either side of the link are restarted after the link is re-established. For RJE-II you must re-issue your SEND requests in order to transmit the jobs which were aborted. Examination of the log file will tell you which files were not transmitted.

POWER FAILURE RECOVERY

If a power failure occurs on the HP 1000, the host system responds in the same way as if an unexpected link termination had occurred.

On the HP 1000 side, recovery procedures vary slightly depending on whether or not your system has a battery back-up. In either case, if a power failure occurs, the PSI loses its configuration and download information. Any power failure recovery procedures must include downloading the PSI. Identical to the "unexpected link down" situation, this means that you run RINIT. If you have battery back-up, this may be done as soon as you have restored power to your system. If you don't have battery back-up, you will have to first re-boot your HP 1000 system, and then issue the RINIT runstring.

RINIT ERRORS AND RECOVERY

There are a number of possible error situations that may arise during subsystem initialization. For each of these situations an appropriate error message is generated and displayed on the terminal from which RINIT was started. These messages are listed in Appendix A under "Initialization Error Messages". In addition, the line number of the configuration file element being processed at the time of the error is displayed.

Under one situation, the error message is not displayed. If, for any reason during the initialization process, the link between your RJE-II installation and the host cannot be established, the associated error message, `The link did not come up`, is displayed when the modem connect timeout period elapses. If you have set an infinite modem connect timeout in the subsystem configuration, the error message is not displayed. Once the `Please establish the RJE link` message is displayed, unless you have a dialed-up connection, it should take less than one minute for the link to be established and the initialization process to terminate.

If the link cannot be made and you have set an infinite timeout, you will have to "off" the RINIT process. Get the break mode prompt and enter: `of RINIT,ID` (for RTE-A) or `of RINIT,8` (for RTE-6).

In addition, you may have to "off" the datacomm monitor clone. Do a WHZAT to see if the monitor clone is still present. If so, enter `of RJE.`*n*`,ID` (for RTE-A) or `of RJE.`*n*`,8` (for RTE-6), where *n* is the subsystem number.

Once your attempt to initialize the subsystem is completely cleaned up, resolve the reason for the inability to establish the link and run RINIT again.

ABORTING RJE/1000-II

Two situations you may encounter that can require aborting the subsystem are:

- An inability to write to the log file, mostly because it is full. (Normally if you cannot write to the log file, messages go to the system terminal. To get this error situation you must also be unable to write to the system terminal.)

- A "fatal" error occurs on one of the output files and you cannot write to it.

The recovery from both of these situations depends on the file system you are using. If you are working under the new file system, you may be able to free up sufficient disc space to complete the "write" function by purging unneeded files from the system. If the problem is with old file system files, you do not have this capability. You must abort the datacomm monitor (CI>of,RJE.*n*). Aborting the datacomm monitor should be avoided if at all possible since you can lose up to 4k of data in the PSI and RJE.*n* buffers and system-available memory buffers.

If it becomes necessary to terminate the subsystem and issuing an RJE,CLOSE or RJE,CLOSE,NOW request does not work, you may have to abort the subsystem. This is accomplished with the following sequence of operations:

- Terminate the datacomm monitor by issuing the RTE command: CI>of,RJE.*n*

- Terminate any pending RJE requests that are queued up on the datacomm monitor. (You can determine the existence of pending RJE requests by using the RTE command: CI>wh).

  If you OF an RJE process before it has received it's response from the datacomm monitor, you receive an error indication. See Appendix A under "No Abort Error Message" for the message text and an explanation.

- Reset the PSI card (:CN,*LU*,35B,RS) where *LU* is the configured Logical Unit for the subsystem PSI card. This command must be issued for both the Read LU and the Write LU.)

  RTE-6 users should make sure there is an entry in the switch table to allow access to these system LUs from your session. This capability is necessary if the PSI must be reset from an RTE-6 session.

Before issuing an RINIT request to recover the subsystem, check the log file to get the names of the files that were being transmitted, if any, under any pending RJE,SEND requests.

NOTE

Depending on the phone system and the host configuration, resetting the PSI may or may not close the phone line or the connection between the host and your system. If in doubt, you should call the host installation to ensure that your remote is logged off.

DIAGNOSTIC CAPABILITIES

As part of the RJE-II subsystem, there are a number of facilities available to assist you in diagnosing problems that may arise.

Help Facility

RJE-II follows the same conventions for help requests as other current RTE software products. You can specify HELP,RJE or ?,RJE to get a display of the RJE-II help file.

Trace Capabilities

Diagostic information on the function of the datacomm monitor and the PSI and datacomm link is available with RJE-II. Control words to enable these diagnostic trace capabilities are accessible through the subsystem configuration file or, while the subsystem is running, through use of the RJE,TRACE request.

Monitor Tracing

The monitor trace control word allows tracing of the datacomm monitor class-I/O activity and execution of "otherwise" clauses in the subsystem's PASCAL case statements. The datacomm monitor trace control word also enables continuous file posting on the data output streams or on the log file. Ordinarily, files written by RJE-II are posted only when the associated data control block becomes full, rather than after each write. This standard procedure considerably increases disc I/O efficiency, and should be used in normal circumstances. However, a recurring system or subsystem failure may make it desirable to post information to output files after each write, to reduce the possibility of data "written" to the DCB never making it out to the file. You may enable this mode of operation individually for the trace/log file and the output data files. Setting bit 14 of the Monitor trace control word enables continuous posting of the "printer", "punch", and "unrouted" data files (enter 40000B). Bit 15 controls continuous posting of the log file (enter 100000B).

PSI Tracing

The PSI trace control word can be used to enable the display of some line trace information. Figure 6-1 shows an example PSI trace record with the transmitted and received bisync control characters, and the transmitted and received data. The communication between RJE-II and an IBM JES2 host is displayed. A file copy utility job (IEBGENER) is sent to and run on the host system, and a file is returned to RJE-II. This trace data is enabled by turning on bits 4, 5, 6, and 7 of the PSI trace control word (enter 360B).

```
840920.160505 L00 PR FmpOpen file: /USR/RJE/TEST/PR0000
840920.160506 L00 PU FmpOpen file: /USR/RJE/TEST/PU0000
840920.160506 L00 UN FmpOpen file: /USR/RJE/TEST/UN0000
840920.160508 L14 RJE,SEND request
/USR/RJE/TEST/!"$%*4
840920.160509 L00 RD FmpOpen file: /USR/RJE/TEST/!"$%*4
840920.160509 L01 RD FmpClose file: /USR/RJE/TEST/!"$%*4
840920.160509 L07 RJE,SEND complete
840920.160512 L01 PR FmpClose file: /USR/RJE/TEST/PR0000
840920.160513 L00 PR FmpOpen file: /USR/RJE/TEST/PR0001
840920.160513 L02 PR FmpOpen error     -2 /USR/RJE/TEST/PR0001
840920.160513 L00 PR FmpOpen file: /USR/RJE/TEST/PR0002
840920.160514 L02 PR FmpOpen error     -2 /USR/RJE/TEST/PR0002
840920.160514 L00 PR FmpOpen file: /USR/RJE/TEST/PR0003
840920.160514 L02 PR FmpOpen error     -2 /USR/RJE/TEST/PR0003
840920.160514 L00 PR FmpOpen file: /USR/RJE/TEST/PR0004
840920.160526 L14 RJE,SEND request
/USR/RJE/TEST/JOB6.STD:::4:1:22
840920.160526 L00 RD FmpOpen file: /USR/RJE/TEST/JOB6.STD:::4:1:22
840920.160527 L01 RD FmpClose file: /USR/RJE/TEST/JOB6.STD:::4:1:22
840920.160527 L11 PSI trace data
 ** TRACE START **

PSI WRITE,CLOCK=42
32  32  32  32  32  2D  FF  FF  FF  FF
SYN SYN SYN SYN SYN ENQ PAD PAD PAD PAD

PSI READ, CLOCK=42
10  70
DLE <*>

PSI WRITE,CLOCK=42
32  32  32  32  32  02  61  5C  E2  C9  C7  D5  D6  D5  1D  C7  D9  C5
SYN SYN SYN SYN SYN STX /   *   S   I   G   N   O   N   GS  G   R   E

PSI WRITE,CLOCK=42
D4  D6  E3  C5  F5  1D  7A  1E  03  31  6E  FF  FF  FF  FF
M   O   T   E   5   GS  :   RS  ETX <*> >   PAD PAD PAD PAD

PSI READ, CLOCK=42
10  61
DLE /

PSI WRITE,CLOCK=72
32  32  32  32  32  37  FF  FF  FF  FF
SYN SYN SYN SYN SYN EOT PAD PAD PAD PAD

PSI READ, CLOCK=72
2D
ENQ
```

Figure 6-1. PSI Trace Data

```
PSI WRITE,CLOCK=73
32  32  32  32  32  10  70  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE <*> PAD PAD PAD PAD

PSI READ, CLOCK=73
02  11  27  61  F1  F6  4B  F0  F3  4B  F5  F6  1D  4A  5B  C8  C1  E2
STX DC1 ESC /   1   6   .   0   3   .   5   6   GS  [   $   H   A   S

PSI READ, CLOCK=74
D7  F2  F0  F0  40  D9  D4  E3  F5  1D  45  E2  E3  C1  D9  E3  C5  C4
P   2   0   0   SP  R   M   T   5   GS  <*> S   T   A   R   T   E   D

PSI READ, CLOCK=74
40  D6  D5  40  D3  C9  D5  C5  F6  1D  44  D5  D6  C4  C5  40  D5  F1
SP  O   N   SP  L   I   N   E   6   GS  <*> N   O   D   E   SP  N   1

PSI READ, CLOCK=74
1E  27  E3  40  1E  27  E3  40  1E  27  E3  40  1E  27  E3  40  1E  32
RS  ESC T   SP  RS  ESC T   SP  RS  ESC T   SP  RS  ESC T   SP  RS  SYN

PSI READ, CLOCK=74
03  7B  ED
ETX #   <*>

PSI WRITE,CLOCK=74
32  32  32  32  32  10  61  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE /   PAD PAD PAD PAD

PSI READ, CLOCK=220
37  FF
EOT PAD

PSI WRITE,CLOCK=220
32  32  32  32  32  2D  FF  FF  FF  FF
SYN SYN SYN SYN SYN ENQ PAD PAD PAD PAD

PSI READ, CLOCK=220
10  70
DLE <*>

PSI WRITE,CLOCK=220
32  32  32  32  32  02  61  61  D9  D1  C5  F1  D2  F2  40  40  D1  D6
SYN SYN SYN SYN SYN STX /   /   R   J   E   1   K   2   SP  SP  J   O

PSI WRITE,CLOCK=221
C2  40  6B  7D  E4  E2  C5  D9  60  D9  D1  C5  7D  6B  C3  D3  C1  E2
B   SP  ,   '   U   S   E   R   -   R   J   E   '   ,   C   L   A   S

PSI WRITE,CLOCK=221
E2  7E  C1  6B  D4  E2  C7  C3  D3  C1  E2  E2  7E  C1  1E  61  61  C1
S   =   A   ,   M   S   G   C   L   A   S   S   =   A   RS  /   /   A
```

Figure 6-1. PSI Trace Data (continued)

```
PSI WRITE,CLOCK=221
40  C5  E7  C5  C3  40  D7  C7  D4  7E  C9  C5  C2  C7  C5  D5  C5  D9
SP  E   X   E   C   SP  P   G   M   =   I   E   B   G   E   N   E   R
840920.160530 L07 RJE,SEND complete
840920.160531 L01 PR FmpClose file: /USR/RJE/TEST/PR0004
840920.160531 L00 PR FmpOpen file: /USR/RJE/TEST/PR0005
840920.160534 L11 PSI trace data

PSI WRITE,CLOCK=221
1E  61  61  E2  E8  E2  C9  D5  40  C4  C4  40  C4  E4  D4  D4  E8  1E
RS  /   /   S   Y   S   I   N   SP  D   D   SP  D   U   M   M   Y   RS
.
.  The middle of the job being sent to the host has been omitted.
.

PSI WRITE,CLOCK=222
C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  1E  61
A   A   A   A   A   A   A   A   A   A   A   A   A   A   A   A   RS  /

PSI WRITE,CLOCK=223
5C  1E  03  D5  16  FF  FF  FF  FF
*   RS  ETX N   BS  PAD PAD PAD PAD

PSI READ, CLOCK=223
10  61
DLE /

PSI WRITE,CLOCK=255
32  32  32  32  32  37  FF  FF  FF  FF
SYN SYN SYN SYN SYN EOT PAD PAD PAD PAD

PSI READ, CLOCK=255
2D
ENQ

PSI WRITE,CLOCK=257
32  32  32  32  32  10  70  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE <*> PAD PAD PAD PAD

PSI READ, CLOCK=257
02  11  27  61  F1  F6  4B  F0  F4  4B  F1  F4  40  D1  D6  C2  40  F5
STX DC1 ESC /   1   6   .   0   4   .   1   4   SP  J   O   B   SP  5

PSI READ, CLOCK=257
F9  F8  F6  40  5B  C8  C1  E2  D7  F1  F0  F0  40  D9  D1  C5  F1  D2
9   8   6   SP  $   H   A   S   P   1   0   0   SP  R   J   E   1   K
.
.  The middle of the job being returned to the HP 1000 has been omitted.
.
```

Figure 6-1. PSI Trace Data (continued)

```
PSI WRITE,CLOCK=441
32  32  32  32  32  10  6B  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE ,   PAD PAD PAD PAD

PSI READ, CLOCK=461
2D
ENQ

PSI WRITE,CLOCK=462
32  32  32  32  32  10  6B  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE ,   PAD PAD PAD PAD

PSI READ, CLOCK=462
2D
ENQ

PSI WRITE,CLOCK=462
32  32  32  32  32  10  70  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE <*> PAD PAD PAD PAD

PSI READ, CLOCK=462
02  32  03  40  01
STX SYN ETX SP  SOH
840920.160602 L14 RJE,SEND request
/USR/RJE/TEST/SIGNOFF:::3:1:5
840920.160602 L00 RD FmpOpen file: /USR/RJE/TEST/SIGNOFF:::3:1:5
840920.160603 L01 RD FmpClose file: /USR/RJE/TEST/SIGNOFF:::3:1:5
840920.160603 L07 RJE,SEND complete
840920.160606 L11 PSI trace data

PSI WRITE,CLOCK=462
32  32  32  32  32  10  61  FF  FF  FF  FF
SYN SYN SYN SYN SYN DLE /   PAD PAD PAD PAD

PSI READ, CLOCK=580
37  FF
EOT PAD

PSI WRITE,CLOCK=580
32  32  32  32  32  2D  FF  FF  FF  FF
SYN SYN SYN SYN SYN ENQ PAD PAD PAD PAD

PSI READ, CLOCK=580
10  70
DLE <*>

PSI WRITE,CLOCK=581
32  32  32  32  32  02  61  5C  E2  C9  C7  D5  D6  C6  C6  1E  03  5C
SYN SYN SYN SYN SYN STX /   *   S   I   G   N   O   F   F   RS  ETX *
```

Figure 6-1. PSI Trace Data (continued)

```
PSI WRITE,CLOCK=581
D2  FF  FF  FF  FF
K   PAD PAD PAD PAD

PSI READ, CLOCK=581
10  61
DLE /

PSI WRITE,CLOCK=611
32  32  32  32  32  37  FF  FF  FF  FF
SYN SYN SYN SYN SYN EOT PAD PAD PAD PAD

PSI READ, CLOCK=611
10  37  FF
DLE EOT PAD
 ** TRACE END **
840920.160607 L09 RJE link is down
    15
840920.160617 L01 PR FmpClose file: /USR/RJE/TEST/PR0006
840920.160619 L01 PU FmpClose file: /USR/RJE/TEST/PU0000
840920.160620 L01 UN FmpClose file: /USR/RJE/TEST/UN0000
840920.160621 L13 RJExx termination
```

Figure 6-1. PSI Trace Data (continued)

Control Bit Table

The following table is included to help you determine appropriate Monitor and PSI trace control words.

TABLE 6-1. CONTROL BIT VALUES

Control Bit	Decimal	Octal	Hexadecimal
0	1	1	1
1	2	2	2
2	4	4	4
3	8	10	8
4	16	20	10
5	32	40	20
6	64	100	40
7	128	200	80
8	256	400	100
9	512	1000	200
10	1024	2000	400
11	2048	4000	800
12	4096	10000	1000
13	8192	20000	2000
14	16384	40000	4000
15	-32768	100000	8000

ERROR MESSAGES AND CODES

APPENDIX A

This appendix contains error codes and messages produced by errors associated with the interface program RJE, and with the transient process RINIT (Initialization Error Messages). In addition, this appendix provides a list of log file codes and messages.

RJE ERROR MESSAGES AND CODES

RMPAR$_1$	RMPAR$_2$	Message
0	0	(no message) *The request was successfully processed.*
0	n	(no message) RJE,SEND*was successful.* n *is the number of files sent.*
1	0	Illegal subsystem number. *The specified subsystem index is out of range. Permissible values are one to eight.*
2	0	Subsystem is not running. *The subsystem index given in the request refers to a copy of the RJE system which is not running. Use RINIT to initialize RJE and sign-on to the host, then repeat the RJE request.*
4	0	Syntax error, use "HELP RJE" for help. *The RJE runstring did not match any of the expected runstring formats.*
5	0	Incorrect security code. *The subsystem security code given in the RJE request does not match the one in the configuration file. See your system manager.*
6	0	Too many parameters. *More parameters were found than expected for this type of subsystem request.*
7	0	Not enough parameters. *Fewer parameters were found than the minimum expected for this type of subsystem request.*

RMPAR$_1$	RMPAR$_2$	Messages
8	0	RJE subsystem terminating. *A* CLOSE *request was issued while the subsystem was already in the process of shutting down.*
10	0	Device specification must be PR, PU, or UN. *The* RJE ROUTE *runstring was issued with something other than* PR, PU, or UN *in the device field.*
11	0	Unrecognized option. *An* RJE CLOSE *request was issued with an option keyword other than* NOW *or programmatic status was requested with an option other than* PR, PU, UN, RD, MC, or LS.
12	0	Power failure, request aborted. *A power failure occured during the processing of the request. Use* RINIT *to restart the subsystem.*
13	0	Link down, request aborted. *The datacomm link is down. Use* RINIT *to restart the system.*
15	0	Unspecified file reference. *A file reference in the* RJE SEND *request was omitted.*
18	0	Link close error. *The interface card response to a close link request was incorrect. This is an RJE internal error; contact your Hewlett Packard service representative.*
22	0	(A)SCII or (B)INARY must be specified for each file. *ASCII or Binary must be specified for each file present in the* RJE SEND *request.*
23	file-n	RJE terminating, send aborted at <filename> *The link has gone down while data was being transmitted. The indicated file in the* SEND *request has not been successfully sent.*
24	0	Initialization in progress *A subsystem process was requested while* RINIT *was running. Reissue the request after initialization is complete.*

RMPAR$_1$	RMPAR$_2$	Message
25	0	Reader aborted per user request at <filename> *An* ABORT,QUEUE *request was issued. Transmission of the indicated file was aborted, and all* SEND *requests in the logical reader queue have been purged.*
26	0	Job aborted per user request at <filename> *An* ABORT,CURRENT *request was issued. Transmission of the indicated file was interrupted and associated files in the* SEND *request were purged from the logical reader queue. Other* SEND *requests in the queue will be processed normally.*
28	0	No SEND requests outstanding *An* ABORT,CURRENT *or an* ABORT,QUEUE *request was issued with no active or outstanding* SEND *requests on the subsystem.*
29	0	Not receiving *An* ABORT,RECEIVE *request was issued when the subsystem was not receiving data from the host system.*
30	0	EOT received, send aborted at <filename> *An EOT bisync character was received from the host system. The currently active* SEND *request was aborted while the filename indicated was being transmitted. The entire* SEND *request must be resubmitted to complete the transmission.*
31	0	RVI received, send aborted at <filename> *An RVI bisync character was received from the host system. The currently active* SEND *request was aborted while the filename indicated was being transmitted. The entire* SEND *request must be resubmitted to complete the transmission.*
<0	file-n	FMP -xx, <filename> *An FMP error occured on one of the files referenced in the runstring. If the runstring was of a type which allows multiple file references (*SEND *or* ROUTE*) the second RMPAR value indicates the sequence number of the offending file.*

INITIALIZATION ERROR MESSAGES

Error Number	Message
101	RJE/1000 II runs on RTE-A and RTE-6/VM only. *An attempt has been made to bring up the RJE subsystem on a computer running an incompatible operating system.*
102	The LU specified is not an IBM PSI (type 63B). *The logical unit number(s) specified in your configuration file do not refer to a downloadable serial interface card. Consult your system manager.*
103	The LU specified is locked to another program.. *Another IBM datacomm product (RJE, MRJE, or PMF) is already using the interface card specified in your configuration file.*
104	The PSI failed self test. *Hardware error. Double-check your installation procedures. The operation of the card during the self-test download sequence may be monitored using the indicator LEDs on the interface card, as explained in the troubleshooting section of this manual. If the problem persists, contact your Hewlett-Packard service representative.*
105	The PSI failed to start. *The download file is incorrect or corrupt, or a hardware error has occured. Contact your Hewlett-Packard service representative.*
106	Subsystem is already active. *A copy of RJE/1000-II with the "subsystem number" specified in your configuration file is already running.*
107	Value is out of bounds. *The value given on the indicated line of the configuration file is illegal. Refer to the section of this manual on Configuring and Initializing the Subsystem.*
108	The download file is not RJE PSI firmware. *The PSI download file specified in your configuration file is incorrect.*
109	The link did not come up. *The modem connected to the PSI card never raised "data set ready" (DSR). Check the configuration of the modems on both ends of the link, as well as your cabling.*
110	PSI configuration error. *An attempt by* RINIT *to configure the PSI card has failed. For additional information see* "Soft Fail Codes" *in this appendix.*

RMPAR Returns

When an initialization error occurs, RINIT returns the error number in $RMPAR_1$. $RMPAR_2$ contains the configuration file line number which was being evaluated when the error was encountered.

If the initialization is successful, RINIT returns a value of 0 to both $RMPAR_1$ and $RMPAR_2$.

Soft Fail Codes

When you receive an `Initialization error: 110` message on your screen you may also get a `PSI soft fail: C0xx` message.

The soft fail conditions you may see when you get a PSI configuration error are as follows:

- C016 - PSI reset while trying to be configured or otherwise needs download data
- C017 - Modem link went down
- C018 - PSI card didn't respond to a subsystem request

If you receive one of these fail indications it may indicate a hardware failure. Double check your configuration procedures. If the problem persists, contact your Hewlett-Packard service representative.

NO ABORT ERROR MESSAGE

Occasionally, the RJE-II subsystem may print the following message on the system console:

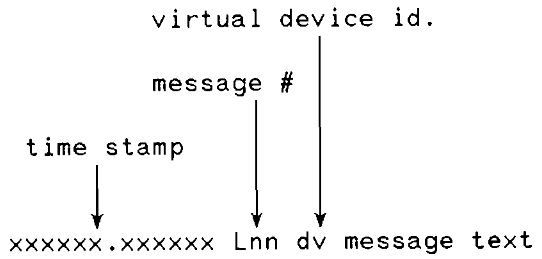
```
RJE.n No Abort.  P nnnnnn  A nnnnnn  B nnnnnn
```

You will usually see this message if you OF an RJE process before it has received it's response from the datacomm monitor RJE.*n*.

The message is generated whenever RJE.*n* executes the abort return from an RTE,EXEC call. The P value indicates the program counter (PC) value in octal. A and B are the octal values of the A and B registers, respectively. Consult *RTE-A Programmer's Reference Manual*, 92077-90007, or *RTE-6/VM Programmer's Reference Manual*, 92084-90005 for the meanings of the A and B registers.

LOG MESSAGES

The general format of the log messages is:



```
          virtual device id.

          message #

    time stamp

xxxxxx.xxxxxx Lnn dv message text
```

The "L" in the message number identifies the message as a log message and may be used to programmatically key recovery procedures off of the error log.

The possible values for dv are:

- RD indicating virtual reader file
- PR indicating virtual printer file
- PU indicating virtual punch file
- UN indicating an unrouted file
- LG indicating the log file
- PP indicating a post processor job

The dv specification is made in Fmp and Rp error messages only.

The following is a list of the messages that the subsystem enters in its log. -fec indicates an Fmp error code.

```
xxxxxx.xxxxxx L00 dv FmpOpen file:  <file name>
xxxxxx.xxxxxx L01 dv FmpClose file:  <file name>
xxxxxx.xxxxxx L02 dv FmpOpen error -fec  <file name>
xxxxxx.xxxxxx L03 dv FmpRead error -fec  <file name>
xxxxxx.xxxxxx L04 dv FmpWrite error -fec  <file name>
xxxxxx.xxxxxx L05 dv FmpClose error -fec  <file name>
xxxxxx.xxxxxx L06 Job files flushed (this message is followed by a list of
                                     files flushed by the subsystem)
xxxxxx.xxxxxx L07 RJE,SEND complete
xxxxxx.xxxxxx L08 dv RP error  <file name>
xxxxxx.xxxxxx L09 RJE link is down
xxxxxx.xxxxxx L10 PSI reset, may be due to power-fail
xxxxxx.xxxxxx L11 PSI trace data (this message is followed by PSI trace data)
xxxxxx.xxxxxx L13 RJExx termination
xxxxxx.xxxxxx L14 RJE,SEND request (this message is followed by a list of files)
```

CONSOLE APPLICATION PROGRAM

APPENDIX B

```
$PASCAL ',4,109 91781-16004 REV.2427 000000'$
$RECURSIVE OFF$ $HEAP 0$
{
  NAME:          CON.PAS
  SOURCE:        91781-18004
  RELOC:         91781-16004
  PGMR:          N.N.
  PURPOSE:       RJE Console Service Routine
  DATE:          <840704.1113>
}

PROGRAM con;

{
   This source file provides a simple example of controlling
   RJE/1000 II through programmatic EXEC and FMP calls. It
   is an installation-specific implementation of a "console"
   function. Although the 2780/3780 workstation definition does
   not include an operator's console, the function can be simulated
   by placing a host (i.e. JES2) command in a file, sending the
   file to the host with "RJE SEND", and retrieving the resulting
   output file.

   Data may arrive asynchronously from the host at any time.
   In order to capture the correct output file, this program
   waits until the reader queue is empty and the link is idle
   before issuing the SEND request. It then polls the printer
   file assignment until it increments (indicating that a file
   has returned) and assumes that this file is the command
   response. Note that this approach is not necessarily valid
   if link traffic is heavy. }

LABEL 99;
TYPE                                        { primitive types }
   integer      = -32768..32767;
   dbl_int      = -2147483648..2147483647;
   byte         = 0..255;
   xlu_arr      = array[1..2] OF integer;
   string_descriptor = dbl_int;
   rmpar_type   = array[1..5] OF integer;
   byte2        = packed array[1..2] OF byte;
   string80     = packed array[1..80] OF char;
```

```
{ For programmatic RJE status calls, there are three possible buffer
  formats... }

TYPE
   device_status = record           { description of a virtual device }
                  header:     integer;     {HP internal use only}
                  file_num:   integer;     {number of files tx/rx}
                  rec_num:    integer;     {number of records in file}
                  q_file:     string80;    {output file name}
                  pgm_file:   string80;    {post-process type 6 file}
                  reserved:   integer;     {HP internal use only}
                  class:      integer;     {assigning program's class #}
                  open_time:  dbl_int;     {$TIME of last open}
                  close_time: dbl_int;     {$TIME of last close}
               END;

TYPE
   monitor_config  = record    { RJExx configuration data }
                  monitor_num:   integer;  {subsystem number}
                  monitor_sec:   integer;  {security code}
                  lu_r:          xlu_arr;  {PSI read LU}
                  lu_w:          xlu_arr;  {PSI write LU}
                  linktype:      integer;  {2780 or 3780}
                  maxblock:      integer;  {max block size}
                  maxrec:        integer;  {max record size}
                  truncate:      integer;  {0= append}
                  host_id:       string80; {config file comment string}
                  log_file:      string80; {current log file name}
                 END;

TYPE
   link_statistics = packed record   { PSI data }
                  header   : dbl_int;   {HP internal use only}
                  brdtype  : byte;      {this is rje firmware}
                  linkstat : byte;      {link status}
                  revcode  : integer;   {revision code}
                  lts_file : integer;   {files sent (Etx's Ack'd)}
                  lts_gr   : integer;   {garbled responses}
                  lts_rt   : integer;   {receive time-outs}
                  lts_nakr : integer;   {naks received}
                  lts_bbr  : integer;   {bad blocks received}
                  lts_bsr  : integer;   {blocks successfully received}
                  lts_bss  : integer;   {blocks successfully sent}
                 END;
 TYPE
    status_response_type = record CASE integer OF
                  0: (LS: link_statistics);
                  1: (DV: device_status);
                  2: (MC: monitor_config);
                 END;
```

```
CONST status_response_length = -89;
      comma = ',';
      reader_stat  = 'RD';
      printer_stat = 'PR';
      link_stat    = 'LS';
      active_bit = 10;
      idle = 4;
      temp_file_name = 'rjtemp::0';
      ascii_specifier = 'A';

VAR temp_file, output_file, userin, userout : text;
    length, error : integer;
    run_string, runname, jes_command, current_printer_file, line : string80;
    passwd, n : string80;
    parms : rmpar_type;
    sd_runstring, sd_runname : string_descriptor;
    status_buf : status_response_type;
```

```
PROCEDURE exec14(    ecode, rcode : integer;
                 VAR buffer : status_response_type;
                     length : integer);
          $ALIAS 'exec'$ EXTERNAL;

PROCEDURE recover_buffer(VAR buff : status_response_type);
     BEGIN
      exec14(14,1,buff,status_response_length);
     END;

FUNCTION StrDsc80(buff : string80;
                  start, n : integer) : string_descriptor;
         $ALIAS 'StrDsc'$ EXTERNAL;

FUNCTION TrimLen(sd : string_descriptor) : integer;
         EXTERNAL;

PROCEDURE append (VAR destination : string80; source : string80);
VAR d_len, s_len, ptr : integer;
BEGIN
     d_len := TrimLen(StrDsc80(destination,1,80));
     s_len := TrimLen(StrDsc80(source,1,80));
     for ptr := 1 to s_len DO destination[d_len+ptr] := source[ptr];
END;

PROCEDURE initialize_stat_runstring; $DIRECT$
BEGIN
     run_string := 'RU,RJE,STAT,';
     append(run_string,n);
     append(run_string,comma);
     append(run_string,passwd);
     append(run_string,comma);
END;

PROCEDURE initialize_send_runstring; $DIRECT$
BEGIN
     run_string := 'RU,RJE,SEND,';
     append(run_string,n);
     append(run_string,comma);
     append(run_string,passwd);
     append(run_string,comma);
     append(run_string,temp_file_name);
     append(run_string,comma);
     append(run_string,ascii_specifier);
END;

FUNCTION FmpRunProgram(    runstring : string_descriptor;
                       VAR     parms : rmpar_type;
                       VAR   runname : string_descriptor) : integer;
         EXTERNAL;
```

```
PROCEDURE set_up_string_descriptors; $DIRECT$
  BEGIN
   sd_runname := StrDsc80(runname,1,80);
   sd_runstring := StrDsc80(run_string,1,80);
  END;

PROCEDURE get_user_inputs; $DIRECT$
  BEGIN
   reset(userin,'1');
   rewrite(userout,'1');
   prompt(userout,'Subsystem number? ');
   readln(userin,n);
   prompt(userout,'Security code? ');
   readln(userin,passwd);
   prompt(userout,'Host command? ');
   readln(userin,jes_command);
  END;

PROCEDURE create_temp_file; $DIRECT$
  BEGIN
   rewrite(temp_file,temp_file_name,'NOCCTL');
   writeln(temp_file,jes_command);
   close(temp_file);
 END;

PROCEDURE wait_until_reader_is_idle; $DIRECT$
  BEGIN
   initialize_stat_runstring;
   append(run_string,reader_stat);
   REPEAT
    BEGIN
     error := FmpRunProgram(sd_runstring,parms,sd_runname);
     IF (error <> 0) or (parms[1] <> 0) THEN GOTO 99;
     recover_buffer(status_buf);
    END
   UNTIL status_buf.dv.file_num = 0;
  END;

PROCEDURE wait_until_link_is_idle; $DIRECT$
  BEGIN
   initialize_stat_runstring;
   append(run_string,link_stat);
   REPEAT
    BEGIN
     error := FmpRunProgram(sd_runstring,parms,sd_runname);
     IF (error <> 0) or (parms[1] <> 0) THEN GOTO 99;
     recover_buffer(status_buf);
    END
    UNTIL status_buf.ls.linkstat = idle;
  END;
```

```
PROCEDURE get_current_printer_filename; $DIRECT$
   BEGIN
     initialize_stat_runstring;
     append(run_string,printer_stat);
     error := FmpRunProgram(sd_runstring,parms,sd_runname);
     IF (error <> 0) or (parms[1] <> 0) THEN GOTO 99;
     recover_buffer(status_buf);
     current_printer_file := status_buf.dv.q_file;
   END;

PROCEDURE send_console_command; $DIRECT$
    BEGIN
      initialize_send_runstring;
      error := FmpRunProgram(sd_runstring,parms,sd_runname);
      IF (error <> 0) or (parms[1] <> 0) THEN GOTO 99;
    END;

 PROCEDURE wait_until_job_received; $DIRECT$
    BEGIN
      REPEAT
        BEGIN
         initialize_stat_runstring;
         append(run_string,printer_stat);
         error := FmpRunProgram(sd_runstring,parms,sd_runname);
         IF (error <> 0) or (parms[1] <> 0) THEN GOTO 99;
         recover_buffer(status_buf);
        END
      UNTIL current_printer_file <> status_buf.dv.q_file;
    END;

 PROCEDURE output_response_to_terminal; $DIRECT$
    VAR c_pos : integer;
    BEGIN
      reset(output_file,current_printer_file);
      WHILE not eof(output_file) DO
        BEGIN
         readln(output_file,line);
         length := TrimLen(StrDsc80(line,1,80)); { skip first two chars }
         for c_pos := 3 to length DO
          write(userout,line[c_pos]);
         writeln(userout);
        END;
      close(output_file);
    END;
   BEGIN    { main }
      set_up_string_descriptors;
      get_user_inputs;
      create_temp_file;
      wait_until_reader_is_idle;
      wait_until_link_is_idle;
      get_current_printer_filename;
      send_console_command;
      wait_until_job_received;
      output_response_to_terminal;

99: { Jump here for error exit. }

IF (error <> 0) OR (parms[1] <> 0)
   THEN writeln(userout,'Error encountered.');

END
```

POSTPROCESSOR EXAMPLES

APPENDIX

C

PRINTER POSTPROCESSOR EXAMPLE

Following is an example printer data postprocessing routine. This program is included, in both source and relocatable forms, as modules on the RJE-II product media. A hardcopy listing of the program is included in this appendix to assist you in handling printer data received by your RJE-II installation.

```
FTN77,1
C
C  NAME:          FMT.FTN
C  SOURCE:        91781-18021
C  RELOC:         91781-16021
C  PGMR:          M.H.T.
C  PURPOSE:       Example Post-processor
C  DATE:          <840920.0940>
C
$ALIAS OPSY = ′.OPSY′,DIRECT
$FILES 0,2
      PROGRAM fmt(),91781-16021 REV.2427 <840920.0940>
*
*     This is an example postprocessor program for the RJE/1000-II product.
*     FMT is an IBM-compatible pretty-printer that takes the output file
*     from RJE then formats and prints it on the printer.
*
*     FMT formats the text for a print window that is 60 lines long
*     (page_len), 132 columns wide (page_wid), has a 3 line top margin
*     (page_head), and a 3 line bottom margin (page_tail). These values
*     are set in the PARAMETER statement below and can be changed.
*
*     The output destination is determined by two statements. The
*     first is the DATA statement for luary (LU array). This sets luary
*     to LU 6. The other statement is the first OPEN statement (with
*     FILE equal to ′6′). These two statements set the printer LU number.
*
*     FMT also implements a virtual carriage control tape. In the DATA
*     statements for the tape array below, any 1 in a bit position 0-11
*     is interpreted as a hole in the carriage control tape. Currently
*     the virtual tape is set only for the form-feed character.
*
```

```
************************************************************
*     VARIABLES:
*
*     Line_cnt is the count of lines printed.
*     Page_len and page_wid are the length and width of the
*        print window.
*     Page_head and page_tail determine the number of lines
*        between pages.
*     String and buffer are for finding and using the input filename.
*     Line is the record read from the data file.
*     Esc is the escape character.
*     Tape is the array of numbers (specified in octal in the DATA
*        statements) that are the carriage control tape.
*     Luary is the printer LU number.
*     Track is the track number to skip to.
*     Skip is the number of lines to skip (when not using the tape).
*     Ios is the IO status or error code.
*
      CHARACTER esc, string*40, line*200
      INTEGER page_len, page_wid, luary(2), page_head, page_tail
      PARAMETER (page_len=60, page_wid=132, page_head=3, page_tail=3)
      INTEGER buffer(40), ios, line_cnt, tape(page_len), track, skip
      INTEGER mesbuf(8),opsy
      EQUIVALENCE (buffer, string)
*
```

```
*********************************************************
*     INITIALIZATIONS:
*
*     Initialize the variable line_cnt.
*
      DATA line_cnt / 1 /
*
*     Initialize the variable luary to the printer LU number.
*
      DATA luary(1) / 6 /
*
*     Initialize the virtual carriage control tape. Currently only
*     the form-feed character is in place (in track 1). A one means a hole
*     (an active location in the tape).
*
      DATA tape(1) / Z'0001' /
      DATA (tape(i),i=2,page_len)/ 59*Z'0000' /  !59=page_len-1
*
*     Initialize the string variable with blanks to avoid problems
*     with odd byte-length filenames. FORTRAN needs a terminating blank
*     at the end of the filename in the open statement. This data
*     statement insures a blank will follow the filename.
*
      DATA string / '                                        ' /
*
*     Initialize RTE-6 MESSS buffer
*
      DATA mesbuf/16HOF, nnnnnn,8,NP,/
*
*     Initialize the escape character.
*
      esc = char(27)
*
```

```
************************************************************
*     OPEN INPUT AND OUTPUT:
*
*     Get the input filename from the parameters, it is the only parameter.
*     Then correct for (remove) the possible comma on the end of the filename
      CALL getst(buffer, -80, ios)
      IF (string(ios:ios) .EQ. ',') THEN
          string(ios:ios) = ' '
      ENDIF
*
*     Now try to open the file, if there is a file open error then
*     abort the program.
*
      OPEN (UNIT=200, FILE=string, IOSTAT=ios, STATUS='old')
      IF (ios .NE. 0) THEN
         WRITE (1,*) 'Error on input file open, err#=', ios
         GOTO 9999
      ENDIF
*
*     Lock and open the printer LU number as the output file. If there
*     is an error then abort the program.
*
      CALL lurq (000001b, luary)
      OPEN (UNIT=201, FILE='6', IOSTAT= ios, STATUS='unknown')
      IF (ios .NE. 0) THEN
         WRITE (1,*) 'Error on output file open, err#=', ios
         GOTO 9999
      ENDIF
*
************************************************************
*     PROCESS THE FILE:
*
*     Write the top margin of the first page to the printer. The specific
*     device driver-printer combination that this was written for prints
*     two blank lines at the top of the first page printed. The minus two
*     compensates for this help.
*
      DO i=1, (page_head-2), 1
         WRITE (UNIT=201, FMT=10) ' '
      ENDDO
*
*     The read loop. Read a line, translate the escape sequences to either
*     number of lines to skip or track of carriage control tape to skip to.
*     Then print the line with vertical spacing suppressed. Use single
*     line spacing to do vertical format according to the escape sequences.
*     The exit from the loop is via the END condition in the READ statement
*     (exit to label 100).
*
```

```
*     *****START OF READ LOOP*****
*
      DO WHILE (.TRUE.)
         track = 0                            !Reset track number.
         READ (UNIT=200, FMT=10, END=100) line !On eof goto line 100
*
*     Translate the escape sequences into skip amounts or track numbers.
*     The comments to the right show the IBM standard for the 3780 type
*     remote unit.
*
         IF (line(1:1) .EQ. esc) THEN
            IF (line(2:2) .EQ. '/') THEN      !esc/ then single space
               skip = 1
            ELSE IF (line(2:2) .EQ. 'S') THEN !escS then double space
               skip = 2
            ELSE IF (line(2:2) .EQ. 'T') THEN !escT then triple space
               skip = 3
            ELSE IF (line(2:2) .EQ. 'A') THEN !escA then skip track1
               track = 1
            ELSE IF (line(2:2) .EQ. 'B') THEN !escB then skip on track 2
               track = 2
            ELSE IF (line(2:2) .EQ. 'C') THEN !If escC then skip on track 3
               track = 3
            ELSE IF (line(2:2) .EQ. 'D') THEN !If escD then skip on track 4
               track = 4
            ELSE IF (line(2:2) .EQ. 'E') THEN !If escE then skip on track 5
               track = 5
            ELSE IF (line(2:2) .EQ. 'F') THEN !If escF then skip on track 6
               track = 6
            ELSE IF (line(2:2) .EQ. 'G') THEN !If escG then skip on track 7
               track = 7
            ELSE IF (line(2:2) .EQ. 'H') THEN !If escH then skip on track 8
               track = 8
            ELSE IF (line(2:2) .EQ. 'I') THEN !If escI then skip on track 9
               track = 9
            ELSE IF (line(2:2) .EQ. 'J') THEN !If escJ then skip on track 10
               track = 10
            ELSE IF (line(2:2) .EQ. 'K') THEN !If escK then skip on track 11
               track = 11
            ELSE IF (line(2:2) .EQ. 'L') THEN !If escL then skip on track 12
               track = 12
            ELSE IF (line(2:2) .EQ. 'M') THEN !If escM then form feed
               track = 1
            ELSE                        !Any other, just single space.
               skip = 1
            ENDIF
         ELSE                           !If not esc seq. just single space.
            skip = 1
         ENDIF
*
```

```
*     Print the line to the printer with no vertical spacing, the
*     spacing will come after print.
*
         line (2:2) = '*'                  !Print line w/o vertical spacing.
         WRITE (UNIT=201, FMT=10) line (2:(page_wid+2))
*
*     Do the carriage control (vertical spacing) now. If track is not
*     zero then skip to that track number. Otherwise use the carriage
*     control character.
*
         IF (track .EQ. 0) THEN                 !If not using tape then
            DO i=1, skip, 1
               WRITE (UNIT=201, FMT=10) '  '    !Skip line on page
               line_cnt = line_cnt + 1          !Correct line_cnt
               IF (line_cnt .GT. page_len) THEN   !If end of page, do margin
                  DO j=1, (page_tail+page_head), 1
                     WRITE (UNIT=201, FMT=10) '  '
                  ENDDO
                  line_cnt = 1                   !Correct line_cnt.
               ENDIF
            ENDDO
         ELSE                                      !On tape skip then
            DO WHILE (.NOT. BTEST(tape(line_cnt),(track-1))) !while not hole
               WRITE (UNIT=201, FMT=10) '  '
               line_cnt = line_cnt + 1
               IF (line_cnt .GT. page_len) THEN  !If end of page, do margin
                  DO j=1, (page_head+page_tail), 1
                     WRITE (UNIT=201, FMT=10) '  '
                  ENDDO
                  line_cnt = 1                   !Correct line_cnt
               ENDIF
            ENDDO
         ENDIF
      ENDDO
*
*     **********END OF READ LOOP*************
*
```

```
*     End of input file. Realign paper and close files.
*
100   DO I=(line_cnt-1), page_len, 1        !Space till end of print window
         WRITE (UNIT=201, FMT=10) '  '
      ENDDO
      DO I=1, (page_tail), 1          !Write the tail margin of last page.
         WRITE (UNIT=201, FMT=10) '  '
      ENDDO
      DO I=1, (page_head + page_len + page_tail + 1), 1   !Eject a page
          WRITE (UNIT=201, FMT=10)  '  '
      ENDDO
      CLOSE (UNIT=200, STATUS='keep')   !Keep the file, do not purge it
      CLOSE (UNIT=201, STATUS='keep')
      CALL lurq (000000b, luary)        !Release the printer LU number.
C
C     If running under RTE-6, terminate by calling MESSS('OF,PNAME,8,NP',N).
C     This will free this program's ID segment.
C
9999  IF (OPSY() .EQ. -17) THEN
          CALL pname(mesbuf(3))
          CALL messs(mesbuf,16)
      ENDIF
10    FORMAT (A)
      STOP
      END
```

PUNCH POSTPROCESSOR EXAMPLE

Following is an example punch data postprocessing routine. This program is not included as one of the product media modules. The listing for the program is included in this appendix for your assistance in developing postprocessing routines to handle data received by your RJE-II installation.

```
FTN7X,L
$ALIAS OPSY = '.OPSY', DIRECT
$FILES 0,2
      PROGRAM CARDS()
*
*************************************************************
*     EXPLANATION:
*         CARDS is an example punch postprocessor. It takes a 480 byte
*     input record (six eighty-byte card images) and unblocks it
*     to six eighty-byte output records. The input file is a punch
*     file from the RJE monitor. The output filename is the input
*     filename with a "C" appended to its end. These numbers can be
*     adjusted for differant configurations.
*
*************************************************************
*     VARIABLES:
*     Tlog is the transmition log for the getstring statement.
*     Ios is an variable to hold an error code.
*     Index is a counter for the output records.
*     The namebuf array holds the input file name, it is equivalent to
*         the namestring.
*     The mesbuf is  buffer for the RTE-6 "programmatic suicide".
*     Namsestring is the array for the input filename string.
*     Outfile id the output filename.
*     Largebuf is the input buffer for the large record.
*     The smallbuf array is the output buffer and is equivalent to largebuf.
*
      INTEGER tlog, ios, index, namebuf(40), mesbuf(8), opsy
      CHARACTER namestr*80, outfile*80, largebuf*512, smallbuf(6)*80
      EQUIVALENCE (namebuf, namestr), (largebuf, smallbuf)
*
*************************************************************
*     INITIALIZATIONS:
*
*     Initialize the namstr to blanks to avoid problems with fortran
*     open statement.
*
      DATA namestr /'                                                '/
*
*     Initialize the mesbuf for RTE-6 programmatic suicide.
*
      DATA mesbuf /16HOF, nnnnnn,8,NP,/
*
```

```
************************************************************
*     OPEN INPUT AND OUTPUT:
*
*     Get the input filename (the only parameter).
*

      CALL GETST (namebuf, -80, tlog)
*
*     Under some conditions a comma can be appended to the end of the
*     namestr. This removes the comma and corrects tlog.
*
      IF (namestr(tlog:tlog) .EQ. ',') THEN
          namestr(tlog:tlog) = ' '
          tlog = tlog - 1
      ENDIF
*
*     Open the input filename, goto 9999 if there is an error.
*
      OPEN (UNIT=200, FILE=namestr, IOSTAT=ios, STATUS='old')
      IF (ios .NE. 0) THEN
          WRITE (1,*) 'Error on input file open, err#=', ios
          GOTO 9999
      ENDIF
*
*     Create the output filename by appending  a 'C' to the end of the
*     input filename. This keeps the names consistant.
*
      outfile = namestr(:tlog) // 'C    '
*
*     Open the output filename, goto 9999 if there is an error.
*
      OPEN (UNIT=201, FILE=outfile, IOSTAT=ios, STATUS='new')
      IF (ios .NE. 0) THEN
          WRITE (1,*) 'Error on output file open, err#=', ios
          GOTO 9999
      ENDIF
*
************************************************************
*     PROCESS THE FILE:
*
*     Read the long record from the input file, then write it out as
*     six output records. If the block size is smaller than 512 bytes
*     then the number of records may need to be changed.
*         If you are using the default (manual example) definitions
*     of the link block sizes and record sizes then the numbers
*     used here will work.
*
1000  READ (UNIT=200, FMT=10, END=9000) largebuf
10    FORMAT (A)
      DO index = 1, 6, 1
          WRITE (UNIT=201, FMT=20) smallbuf(index)
20        FORMAT (A80)
      ENDDO
      GOTO 1000
```

```
*
***********************************************************
*     CLEANUP AND STOP:
*
*     Close the output files.
*
9000  CLOSE (UNIT=200, STATUS='keep')
      CLOSE (UNIT=201, STATUS='keep')
*
*     If this is a RTE-6 system then do programmatic suicide and off
*     out own ID segment. Otherwise just stop normally.
*
9999  IF (OPSY() .EQ. -17) THEN
          CALL pname(mesbuf(3))
          CALL messs(mesbuf,16)
      ENDIF
      STOP
      END
```

PROGRAMMATIC STATUS REQUEST

APPENDIX D

Following is a hard copy of a program you may use to programmatically make an RJE,STATUS request. This program is included as one of the modules on the RJE-II product medium. The program listing and an example of the output produced by execution of the program are included in this appendix for your convenience.

STAT.FTN LISTING

```
ftn7x,1

C Name:    STAT.FTN
C Source:  91781-18022
C Reloc:   91781-16022
C Pgmr:    E.W.
C Purpose: Example FORTRAN Use of RJE/1000-II

      program stat(),91781-16022 REV.2427 <840920.1525>

C This source illustrates the control of RJE/1000-II through simple
C FORTRAN calls. The program produces a status display similar to
C the one produced by the RJE,STATUS command. The runstring is
C "STAT,<lu>". The output is directed to the logical unit number
C given in the runstring. The subsystem number and password are
C hardcoded into the program so that they do not need to be included
C in the runstring.

      integer irje(3),ipram(5),ibuf(5),jbuf(100)
      integer iparms(5)
      integer irmane(3)
      integer*4 lopen,lclose
      character*19 rstring
      character*16 tbuf
      character*2 device(4)
      character*1 underscore
      equivalence (lopen,jbuf(86)),(lclose,jbuf(88))
C assumes subsystem #1, security code = "sc"
      data rstring/′ru,rje,stat,1,sc,mc′/
      data device/′rd′,′pr′,′pu′,′un′/,underscore/′_′/

C send output to the indicated LU, as long as it is less than 64
      call rmpar(iparms)
      lu =iparms (1)
      if (lu .le. 0 .or. lu .gt. 63) lu =1
```

```
C get the 'monitor configuration' status
     call FmpRunProgram(rstring,iparms,irname)
C abort on error
     if (iparms(1) .ne. 0) stop
C recover the status buffer
     call exec(14,1,jbuf,-120)
C print formatted display of monitor configuration data
     write(lu,'(/" [ RJE STATUS ] ")')
     write(lu,'(" read lu.........",i4,5x,"_")') jbuf(3)
     write(lu,'(" write lu........",i4,5x,"_")') jbuf(5)
     write(lu,'(" link type.......",i4)') jbuf(7)
     write(lu,'(" max block size..",i4,5x,"_")') jbuf(8)
     write(lu,'(" max record size.",i4,5x,"_")') jbuf(9)
     write(lu,'(" truncate........",i4)') jbuf(10)

C get the 'link status' data
     rstring(18:19)='ls'
     call FmpRunProgram(rstring,iprams,irname)
C abort on error
     if (iparms(1) .ne. 0) stop
C recover the status buffer
     call exec(14,1,jbuf,11)
C print the formatted link status information
     iboard = jbuf(3)/256
     if (iboard .eg. 8) then
        write(lu,'(" hardware correct")')
     else
        write(lu,'(" hardware or firmware incompatible with rje!!!")')
     end if
     lstat = jbuf(3) - iboard * 256
     if ( lstat .eq. 1 .or. lstat .eq. 2 ) then
        write(lu,'(" link down")')
        else
        if ( lstat .eq. 4 ) then
            write(lu,'(" link up and idle")')
            else
            if ( lstat .eq. 8 ) then
                 write(lu,'(" link up and transmitting")')
                 else
                     if ( lstat .eq. 16 ) then
                         write(lu,'(" link up and receiving")')
                         else
                             write(lu,'(" unknown link status")')
                     endif
            endif
        endif
     endif
```

```
      write(lu,'(" firmware rev code..............",i6.6)') jbuf(4)
      write(lu,'(" number of files sent...........",i6.6)') jbuf(5)
      write(lu,'(" number of garbled responses....",i6.6)') jbuf(6)
      write(lu,'(" number of receive time outs....",i6.6)') jbuf(7)
      write(lu,'(" number of NAKS received........",i6.6)') jbuf(8)
      write(lu,'(" number of bad blocks received..",i6.6)') jbuf(9)
      write(lu,'(" number of good blocks received.",i6.6)') jbuf(10)
      write(lu,'(" number of good blocks sent.....",i6.6)') jbuf(11)

C get 'device status' for each of the four virtual devices
      do idevice=1,4

      rstring(18:19) = device (idevice)
      call FmpRunProgram(rstring,iparms,irmane)
C abort on error
      if (iparms(1) .ne. 0) stop
C recover the status buffer
      call exec(14,1,jbuf,90)
C output the formatted data for this device
      write(lu,'(/" ************************** status of ",a," *********
     >****************")') rstring(18:19)
      write(lu,'(" file name .............",20a2)') (jbuf(i),i=4,23)
      write(lu,'(" file sequence number...........",i6.6,"  _")') jbuf(2)
      write(lu,'(" number of records..............",i6.6)') jbuf(3)
      call NumericTime(lopen,tbuf)
      tbuf(16:16) = underscore
      write(lu,'(" time of last open......",16a,5x)') tbuf
      call NumericTime(lclose,tbuf)
      write(lu,'(" time of last close.....",14a)') tbuf
      write(lu,'(" post processor name....",20a2)') (jbuf(i),i=44,63)

      end do

      end
```

STAT.FTN OUTPUT EXAMPLE

```
[ RJE STATUS ]
read lu.........  82     write lu........  83      link type.......   0
max block size.. 512     max record size.  80      truncate........   1
hardware correct
link up and idle
firmware rev code.............002427
number of files sent..........000002
number of garbled responses....000000
number of receive time outs....000000
number of NAKS received........000000
number of bad blocks received..000000
number of good blocks received.000030
number of good blocks sent.....000019
************************* status of RD ************************
file name ............
file sequence number.........000000   number of records............000000
time of last open......840920.152419   time of last close.....840920.152426
post processor name....
************************* status of PR ************************
file name ............/USR/RJE/TEST/PR0005.PS
file sequence number.........000005   number of records............000000
time of last open......840920.152451   time of last close.....840920.152450
post processor name..../USR/RJE/TEST/DUMMY.RUN
************************* status of PU ************************
file name ............/USR/RJE/TEST/PU0000.PS
file sequence number.........000000   number of records............000000
time of last open......840920.152322   time of last close.....870130.033808
post processor name..../USR/RJE/TEST/DUMMY.RUN
************************* status of UN ************************
file name ............/USR/RJE/TEST/UN0000.PS
file sequence number.........000000   number of records............000000
time of last open......840920.152323   time of last close.....870130.033808
post processor name..../USR/RJE/TEST/DUMMY.RUN
```

HOST SYSTEM GENERATION CONSIDERATIONS

APPENDIX E

INTRODUCTION

3705 This appendix is written to provide you with some quick reference assistance in establishing support for IBM 2780/3780 binary synchronous communications on an IBM or IBM plug compatible host system.

Information is provided to establish this support on systems using JES2 under OS/VS2 (MVS). Information is also provided to establish the proper data communications link support on an IBM 3705 Communications Controller. A 3705 or similar device is necessary to support data communications on an IBM host system. Example system generation statements and brief explanations are provided.

This appendix does not include information on the operating system statements you must provide to support the communications controller. For information on these statements and for a detailed understanding of all the host system generation procedures you should refer to the following documentation:

OS/VS2 System Programming Library: System Generation Reference, GC26-3792

System Programming Library: JES2 Installaton, Initialization, and Tuning, SC23-0046

IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual, GC30-3008

GENERAL CONSIDERATIONS

Following is a list of elements to be considered when establishing communications between RJE-II and another computer system. This list applies specifically to communication between RJE-II and an IBM JES2 host system, but the items should be considered when establishing communication with any other system.

Host Side Considerations

• **Type of host operating system**	used in the 3704/3705 emulation program (EP)
• **Type of communication controller**	used in the EP
• **Subchannel address**	used in the LINE macro of the EP and under the JES2 generation in the LINE definition
• **Line number**	used under the JES2 generation in the LINE and REMOTE definitions.
• **Remote number**	used under the JES2 generation in the REMOTE definition. Also used by RJE-II for the SIGNON card image in the configuration file.
• **3705 address**	used in the LINE macro of the EP

Line Considerations

• **Type of line (dialed or leased)**	used in the EP, JES2 generation, and RJE-II generation
• **Type of line (full or half-duplex)**	used in the EP, JES2 generation, and RJE-II generation
• **Type of line control (must be BSC)**	used in the EP
• **Line speed**	used in the EP
• **Line code (must be EBCDIC)**	used in the EP and JES2 generation
• **Data transmission characteristics (transparency, data compression, tabulation)**	used in the JES2 generation under the LINE and REMOTE definitions. RJE-II automatically handles these characteristics.

Modem Considerations

- **Type of modem (full or half-duplex)**

  used in the EP, JES2 generation, and RJE-II configuration

- **Compatibility**

  yes

Workstation Considerations

- **Workstation type (2780 or 3780)**

  used in the EP, JES2 generation, and RJE-II configuration

- **Workstation mode (slave or master)**

  IBM hosts assume the master role by default. RJE-II should usually be configured as slave when connecting to an IBM system.

  This configuration item affects the line bid collision timeout value. The site designated as the slave delays for a longer period of time after a line bid collision. This allows the master site to gain control of the line first when there is a collision.

3705 EMULATION PROGRAM (EP) GENERATION

To support a binary syncronous communications line used by RJE-II and your host system, you must generate a program to run on the host communication controller that performs emulation of IBM 2701, 2702, or 2703 transmission control units. This can be done by generating an Emulation Program (EP), or, if the program is to also perform network control functions, by generating a Network Control Program (NCP) with the Partitioned Emulation Program (PEP) Extension. For clarity, the EP generation is described in this appendix. To generate the PEP refer to *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual*, GC30-3008.

The EP generation consists of defining the five macros and associated parameters:

- BUILD defines the physical environment in which the EP is to run including the control unit being emulated, the type of communication controller being used for the emulation, and the diagnostics capabilities.

- CSB defines the communication scanner being used in the communications controller and its environment.

- GROUP defines common characteristics of all the data communication lines within one group. For example, the GROUP macro is used to define whether the lines are switched or non-switched, and whether the lines are BSC or start-stop.

- LINE defines the characteristics of one particular line attached to the communication controller including the address of the line, some of the transmission characteristics (speed, transmission code, etc.), some of the modem characteristics (duplex, clocking, etc.), and the type of remote terminal to which the line is connected.

- GENEND defines the end of the 3705 control program file (input deck) and must be the last macro coded in the program.

Definitions of the parameters that you may use under each of these macros follows. These parameters apply to BSC or start-stop (asynchronous) lines only, and are used in support of 2780/3780 emulation. You must define **highlighted** parameters. Other parameters described are allowed with restrictions as indicated. Undefined parameters need not be specified to provide complete support for RJE/1000-II.

BUILD Macro

BUILD

Required for Emulation Program

HICHAN=(*addr1*[,*addr2*])

Required. Defines the high end of the range of subchannel addresses associated with the channel adapter(s) in the communication controller. This address must be equal to or greater than the highest EP subchannel address that you specify in the ADDRESS parameter of any LINE macro.

LOCHAN=(*addr1*[,*addr2*])

Required. Defines the low end of the range of addresses associated with the channel adapter(s) in the communications controller. This address must be the lowest EP subchannel address, regardless of whether it is used for dynamic dump data transfer or for an EP line.

NOTE

For additional information on addresses you may specify for HICHAN and LOCHAN, and how to handle the use of multiple type 4 channel adapters, see *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual, GC30-3008.*

LOADLIB=*dsname*

Required for OS/VS generation. Does not apply to DOS/VS generation. Identifies the name of the cataloged, partitioned data set (file) containing the load module produced by the program generation procedure.

OBJLIB=*dsname*

Required for OS/VS generation. Does not apply to DOS/VS generation. Identifies the name of the cataloged, partitioned data set (file) containing the output from all EP assemblies produced during stage two of the generation procedure.

CA=(*adapter1*[,*adapter2*])

Required if you are using any adapter other than a type 1 for a 3704 or 3705-1 or a type 4 for a 3705-2 (default values). Identifies the type(s) of channel adapter(s) being used in your host communication controller.

DYNADMP=YES,NSC

Recommended in order to include the dynamic dump facility in the EP. This facility allows transfer of the storage contents of the communication controller to the host processor without interrupting execution of the EP. The NSC specification invokes the use of the native subchannel address for dynamic dump communication.

JOBCARD=YES

Recommended (default value). Applies to OS/VS generation only. Causes the EP generation procedure to create a job card for the stage two input stream. If you specify NO, you must supply your own job card.

LESIZE=*size*

Applies to OS/VS generation only. Specifies the region size in K bytes to be used by all linkage editor job steps during stage two of the EP generation.

LINETRC=YES

Recommended. Enables you to perform channel and line interrupt tracing on the host side.

TEST=YES

Recommended. Enables you to perform control panel-initiated line testing on the host side.

```
      3705-2
MODEL=3705
      3704
```

Required. Identifies the type of communications controller you are using to run the EP.

NEWNAME=EP001

Recommended (default value). Identifies a name for the generated program. You can use any valid string of 8 characters or less.

```
OPCSB2=YES
       NO
```

Affects type 2 channel adapters only. If you specify YES, a 20-byte data buffer is provided for communication lines serviced by a type 2 adapter and for which you specify CHNPRI=HIGH in the LINE macro.

TYPGEN=EP

Required (default value). Indicates that the program you are generating performs only emulation functions.

```
TYPSYS=DOS
       OS
```

Required if your host operating system is DOS (OS is the default value). Identifies the type of operating system under which stage two of the control program generation procedure is to run.

UNIT=*unit type*

Applies to OS/VS program generations only. Identifies the type of disc device used during stage two of the EP generation for assembler and linkage editor utility data sets.

UT1=*dsname*

Applies to OS/VS program generations only. Identifies a sequential data set (file) to be used as a work space for the EP assembly steps (SYSUT1). If omitted, a temporary data set is automatically created during each assembly step.

UT2=*dsname*

The same as UT1 for EP assembly steps (SYSUT2).

UT3=*dsname*

Also applies to OS/VS program generation only. Identifies a sequential data set to be used as a work space for the EP assembly (SYSUT3) and linkage editor (SYSUT1) steps. If omitted, a temporary data set is automatically created during each assembly step.

CSB Macro

CSB — **Required for Emulation Program.**

SPEED=*(rate1,...,rate4)* — **Required.** Identifies the bit rates for each of up to four oscillators installed in the scanner in the communications controller.

WRAPLN=*line addr* — **Required.** Identifies the interface address for the line you want to use to send test data to the communications controller. The address specified here is used in the "wraparound" test which checks out the interface hardware on a different line. *line addr* is a hexadecimal address that must appear in the ADDRESS operand of one of the LINE macros. The range available for use as *line addr* can be found in *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual, GC30-3008.*

MOD=*n* — Identifies the communications controller module in which the scanner is found, as follows:

Scanner Location	MOD=
3704	0 (default value)
3705 base module	0 (default value)
3705 first expansion module	1
3705 second expansion module	2
3705 third expansion module	3

GROUP Macro

GROUP

Required

DIAL=NO
    YES

Indicates if the lines in the group are switched (DIAL=YES), or not switched (DIAL=NO, or omit the parameter).

LNCTL=BSC

Required for bisync line. Identifies the line control protocol as either bisync or start-stop (asynchronous). RJE-II uses BSC lines.

LINE Macro

LINE

Required. One LINE macro must be included for each line attached to the communications controller.

ADDRESS=(*lnaddr*,*subchaddr1*)

Required. For the line being defined by this LINE macro, this parameter identifies the line interface address and the corresponding subchannel address(es). For information on addressing requirements and restrictions see: *IBM 3704 and 3705 Control Program Generation Utilities Guide and Reference Manual, GC30-3008.*

SPEED=*rate*

Required. Identifies the data transfer rate, in bits per second, for the line being defined. If all the lines in the group have the same speed, this parameter may be defined under the GROUP macro.

AUTO=*address*

Required if the line being defined has the auto call facility. *address* identifies the automatic calling unit (ACU) interface address.

NOTE

Each of the remaining parameters defined for the LINE macro may instead be defined under the GROUP macro if the parameter value is the same for all lines in the group.

BUFSIZE=*n*

Applies to lines serviced by a type 3 communications scanner only. For this type of line, *n* identifies the size of the emulation mode buffers. It also identifies the amount of data (up to 32 bytes) transferred between the communications controller and the host processor over the channel connection.

Values for *n* (in bytes) you may specify are:
4,8,16,32,64,96,128,160,192,or 224
For *n*>32, the channel transfer is still 32 bytes.

CHNPRI=NORMAL

Recommended (default value) unless the line being defined is to operate at a data rate of 19.2 Kbps or higher and the majority of the other lines operate at 2400 bps of lower. This parameter identifies the relative priority of the subchannel for this line to the other subchannels used by the emulation program. If the data rate difference exists, specify CHNPRI=HIGH.

CLOCKNG=EXT

Recommended for BSC line (default value for BSC line). Identifies the source of clocking for the line being defined. EXT indicates that the modem is the clocking source. Specifying CLOCKNG=INT, indicates that the scanner provides clocking.

CODE=EBCDIC

Required for communications with RJE-II. Identifies the transmission line code as EBCDIC.

```
   2701
CU=2702
   2703
```

Identifies the transmission control unit that is being emulated by the EP. 2703 is the default value.

```
DATRATE=LOW
        HIGH
```

If you have a dual-rate modem, use this parameter to specify which rate applies to this line. Omit this parameter if this line is using a single data rate modem.

```
DISABLE=NO
        YES
```

Applies to modems that require a long "disable timeout" when disconnecting from the line. Specifying YES causes a timeout value of 25.6 seconds. For most modems, specify NO (default value), or omit the parameter.

```
DUPLEX=HALF
       FULL
```

Required. Must match the type of modem and line you are using. This should also match the corresponding RJE-II configuration file element.

```
        0
INTPRI=1
        2
        3
```

Identifies the relative interrupt priority for this line, with 3 being the highest (the default value is 1). Lines with higher data rates should have higher relative priorities.

A method for determining interrupt priorities can be found in *IBM 3704 and 3705 Control Program Generation Utilities Guide and Reference Manual, GC30-3008.*

```
      OPTION1
MODEM=OPTION2
      NTT
```

Indicates when and how the communication line is enabled. The default value (OPTION2) indicates that the line is disabled after the controller has been loaded (IPL) or the System Reset key has been pressed. The line is then enabled by command from the access method. For information on the other options, refer to the 3705 Reference Manual.

```
PAD=YES
    NO
```

Applies to emulation of an IBM 2703 Transmission Controller only. A YES value (default value) indicates that the communications controller is to verify that the first four bits of trailing PAD characters received from the line are all "1"s.

QUIET=NO
YES

Indicates which "line quiet" timeout period is to be used by the emulation program. A YES value indicates that the long timeout period of 25.6 seconds is to be used. **A NO value (default value) is recommended** and indicates that the normal timeout of 3 seconds is to be used.

RING=NO
YES

Applies to some modems used outside the U.S. and Canada and to switched lines only. Indicates if the modem used for this line has the "ring indicator interface" lead. A NO value (default value) indicates that the modem does not have this lead.

TERM=3780
2780

Required. Indicates the type of workstation with which the host is communicating on this line. This value should match the Workstation Type element in the RJE-II configuration file.

GENEND Macro

GENEND

Required for Control Program generation.

HSPDSEL=([*msk1*],...,[*msk4*])

Applies to type 3 scanners only. *mskn* are eight bit masks used to identify the high speed lines. These lines are identified to increase the amount of scanning they receive. See the 3705 Reference Manual for the effects of the different bit patterns.

Example Partial 3705 Generation

```
          BUILD  MODEL=3705-2,CA=TYPE4,LOCHAN=80,HICHAN=FF
                 DYNADMP=YES,NSC,LINETRC=YES,TEST=YES,NEWNAME=EP001
*******************************************************************
          CSB    MOD=0,TYPE=TYPE3,WRAPLN=020,SPEED=(150,600,1200)
*******************************************************************
          GROUP  DIAL=NO,LNCTL=BSC,CU=2701
*******************************************************************
          LINE   ADDRESS=(020,AC),SPEED=19200,CLOCKNG=EXT,
                 CODE=EBCDIC,DUPLEX=HALF,TERM=3780
      .
      .
      .

*******************************************************************
          GENEND HSPDSEL=(10000000)
```

This configuration would be used under an OS operating system, on a 3705-2 communications controller emulating a 2701 control unit. The TYPE 3 communications scanner being used is located in the 3705 base module. Its three oscillators have bit rates of 150, 600, and 1200. The group of lines being defined are leased BSC lines. One of the lines (address= hex 020, subchannel address= hex AC) has a data rate of 19.2 Kbps. It's a half duplex line transmitting and receiving EBCDIC code with a 3780 terminal, or terminal emulator.

IBM JES2 GENERATION CONSIDERATIONS

To generate support for an RJE-II work station under IBM JES2, the program consists primarily of defining the characteristics of the data communications lines, and the remote station facilities and characteristics using the following macros:

•LINE*nnnn* defines the characteristics of a single data communications line servicing a remote station. *nnnn* identifies the line number.

•RMT*nnnn* defines the characteristics of the remote terminal located at REMOTE *nnnn*.

•R*nnnn*.PR*m* defines the characteristics of a printer at REMOTE *nnnn*. For RJE-II, you define a single logical printer (*m*=1).

•R*nnnn*.PU*m* defines the characteristics of a card punch at REMOTE *nnnn*. For RJE-II, you define a single logical punch (*m*=1).

•R*nnnn*.RD*m* defines the characteristics of a card reader at REMOTE *nnnn*. For RJE-II you define a single logical reader (*m*=1).

In addition to the above macros, changes in the following JES2 initialization parameters should be considered so that the host system can properly service your workstation:

•&NUMLNES=*nnnn* defines the number of data communications lines that are available for JES2 to use (2000 is the maximum). These lines include those used for RJE communications as well as the lines defined as logical lines for SNA RJE terminals and SNA NJE application-to-application sessions.

•&NUMRJE=*nnnn* defines the number of remote terminals that the JES2 subsystem on your host will service (1000 is the maximum). You should make sure that this value is high enough to be able to handle the inclusion of your RJE-II work station.

•&NUMTPBF=*nnnn*,*mmm* defines the total number (*nnnn*) of teleprocessing buffers generated by JES2 to service RJE and NJE lines (2000 is the maximum).

Also defines at what percent (*mmm*) of teleprocessing buffer usage the host system operator will be alerted via a console message.

•&TPBFSIZ=*nnnn* defines the size of the JES2 teleprocessing buffers in bytes. This value is dependent on the type of terminal you are emulating.

For complete definitions of these macros and parameters and the other macros and parameters that are defined in a complete JES2 generation, see: *System Programming Library: JES2 Installation, Initialization and Tuning, SC23-0046.*

An example partial JES2 generation follows displaying the macros and parameters necessary to define the remote and line that is used by your RJE/1000-II installation. A number of available parameters, particularly for the remote card reader, card punch, and printer definitions, are not specified. The default values are appropriate for communications between the host and your RJE-II station.

LINE Definition

LINE=*nnnn*

Required.

COMP

Required to allow the line being defined to perform data compression (default value). Enter NOCOMP to disable data compression. The value used here overrides specifications made in the RMT definition or in the definition of the workstation in the communications controller EP.

HDUPLEX

Required for communications with RJE-II (default value). Indicates that the data communications on the line being defined is half duplex. A full duplex line is identified by entering FDUPLEX.

TRANSP

Required to invoke the transparency feature for the line being defined. The default is NOTRANSP, which disallows the transparency feature.

UNIT=*cau*

Required. Identifies the subchannel address of the data communications line being defined. The default value is selected by JES2 and is the first available BSC line address.

EBCDIC

Required for communication with RJE-II (default value). Defines the data communications line code as EBCDIC.

REMOTE Definition

RMT*nnnn*

Required. *nnnn* is the remote number.

terminal type

Required. Identifies the type of terminal being emulated by your RJE-II installation. Specify 2780 or 3780. **This must match the "Workstation Type" element in the RJE-II configuration file.**

COMP

Recommended if you are doing 3780 emulation. Indicates that the remote being defined has the data compression/expansion feature. RJE-II in 3780 mode automatically compresses imbedded blanks and the host configuration should match this capability. In 2780 mode RJE-II acts like a real 2780 hardware terminal and does not perform compression. For 2780, you should specify NOCOMP (default value), or omit the parameter.

Note that COMP and TABS (which allows the tabulation feature) cannot both be specified.

LINE=*nnnn*

Required for dedicated, leased lines. Identifies the number of the data communications line that is connected to your RJE-II installation. **This parameter applies to dedicated lines only.** If this parameter is omitted, JES2 assumes that the remote being defined is using a nondedicated (SIGNON) line.

The available range for *nnnn* is 1-2000 and cannot be greater than the value assigned to the &NUMLNES parameter.

MRF

Required for 2780 emulation if you specify values greater than 1 for either the "Nontransparent" or "Transparent Blocking Factor"s in the RJE-II configuration file. Indicates that the remote being defined has the multiple record feature. The default value is NOMRF.

NUMPU=1

Required if you want your RJE-II subsystem to receive punch streams. Defines the number of "card punches" at the remote. For RJE-II, the maximum you can specify is 1. The default value is 0.

NOTE

There are also NUMPR and NUMRD parameters to define the numbers of printers and readers respectively at the remote. These need not be specified, though, since the default values for both of these parameters is 1.

TABS

Required if you want to send tabulated data from the host to RJE-II. (default is NOTABS).

Note that TABS and COMP (which allows the data compression/expansion feature) cannot both be specified.

TRANSP

Required if you want transparent data to be transmitted between the host and RJE-II (default is NOTRANSP). This value will not take effect unless TRANSP is also specified in the LINE definition.

Remote Reader Definition

R*nnnn*.RD1

Required. *nnnn* is the number of the remote at which this "reader" is found.

CLASS=*v*

Identifies the job class as recognized by JES2. This value is overridden by a CLASS specification on JCL JOB cards that you send to the host. The default value for *v* is A.

START

Recommended (default value). Indicates that the "card reader" at the remote being defined is to be started or enabled automatically when JES2 starts processing. If you specify DRAIN, the device (in the case of RJE-II, the logical device) must be enabled by a command from the host system operator.

To communicate with RJE-II, none of the parameters under the Remote Reader Definition other than R*nnnn*.RD1 **actually need to be specified. Use of default values allow proper operation. See:** *System Programming Library: JES2 Installation, Initialization, and Tuning, SC23-0046* **for additional information.**

Remote Printer Definition

R*nnnn*.PR1

Required. *nnnn* is the remote number.

PRWIDTH=*nnn*

Identifies the number (*nnn*) of characters that would be printed on the line of an actual remote terminal (2780 or 3780) printer. For RJE-II, this identifies the size of the blocks of data that it stores on disc in the "printer" files. For postprocessing purposes, this value should equal the width of the line for the printer you are using. The default value is 120.

To communicate with RJE-II, none of the other parameters under the Remote Printer Definition need to be specified. Use of the default values allows proper operation.

Remote Punch Definition

R*nnnn*.PU1

Required. *nnnn* is the remote number.

To communicate with RJE-II, none of the parameters other than R*nnnn*.PU1 **under the Remote Punch Definition need to be specified. Use of the default values allows proper operation.**

Example Partial JES2 Generation

```
     .
     .
     .
LINE 71  COMP,HDUPLEX,TRANSP,EBCDIC,UNIT=020
     .
     .
     .
************************************************************
     .
     .
     .
RMT71    LINE=71
RMT71    3780,COMP,NUMPU=1,TRANSP
R71.RD1  START,CLASS=A
R71.PR1  PRWIDTH=132
R71.PU1
     .
     .
     .
```

MODEM OPTIONS

APPENDIX

F

BELL MODEMS

The following Synchronous modem recommendations and options are for use with the 12260A or 12043A Programmable Serial Interface (PSI). Further definition of these options and capabilities can be obtained from the relevant *Bell System Technical Reference* publication, which is available from your local Bell System Representative or CCITT reference.

It should be noted that although the following modem configurations are fully supported, many other configurations will also work. RJE/1000-II requires the use of synchronous modems which must:

- provide transmit and receive clocks for the PSI card
- ensure ground isolation between the communicating systems
- support the following handshake signals: DTR,DSR, RTS, CTS, CD, RI

Modem Configuration

Modems for RJE-II must be connected in a point-to-point configuration. Public (switched) or private (leased) lines may be used.

BELL 201C Modem

Private Leased Line, Full or Half Duplex, Point-to-Point

Type of Modem	Bell System Type 201C Data Set (Also called DATAPHONE 2400)
Type of Line	Public Telephone Network, Private Leased Line
Transmission Rate	2400 bits-per-second

TABLE F-1. BELL 201C MODEM OPTIONS

Option Number	Description	Comments
A1	EIA Interface.	
B3 B4	With alternate voice. Without alternate voice.	Customer Defined
C6	Without new synch.	Both MASTER and SLAVE
D8	4-wire circuit.	
E9 E10	4-wire private line continuous carrier. 0 millisecond delay. 4-wire private line transmitter internally timed.	Either

BELL 208A Modem

Private Leased Line, Point-to-Point, Full Duplex

Type of Modem	Bell System Type 208A Data Set (Also called DATAPHONE 4800)
Type of Line	Private Leased Line
Transmission Rate	4800 bits-per-second

TABLE F-2. BELL 208A MODEM OPTIONS

Option Number	Description
A1	Transmitter timing internal.
B3	Continuous carrier.
C6	Continuous Request-to-Send.
D7	One second holdover used.
E10	Without new synch.
F11	Continuous carrier ON when analog loop is present.

BELL 208B Modem

Switched Line, Half Duplex, Point-to-Point

Type of Modem	Bell System Type 208B Data Set (Also called DATAPHONE 4800)
Type of Line	Public Telephone Network (Switched)
Transmission Rate	4800 bits-per-second

TABLE F-3. BELL 208B MODEM OPTIONS

Option Number	Description	Comments
A1	Transmitter internally timed.	
B3 B4	Without 801 Automatic Calling Unit With 801 Automatic Calling Unit.	See Note 1.
C6	Data Set Ready (CC) ON when analog loop is present.	
D8	With automatic answer.	
E9 E10	Desk mounting. Rack or cabinet mounting.	Either

Note 1: Switch controlled 50 or 150 msec Request-to-Send Clear-to-Send delay.

BELL DATAPHONE II 2024A Modem

Private Leased Line, Point-to-Point

Type of Modem	Bell System Type DPII 2024A Data Set
Type of Line	Private Leased Line or Public Telephone Network (Dial Back-Up).
Transmission Rate	2400 bits-per-second

NOTE

For Dataphone II modems:

- Only one control on each line.
- C is usually the control computer site.
- T is usually the remote computer, host, or terminal site.
- C and T are for diagnostic purposes only.

TABLE F-4. BELL 2024A MODEM OPTIONS

Option Number	Description	Comments
A1 A2	Point-to-point control. Point-to-point tributary or extended point-to-point tributary.	MASTER: A1 SLAVE : A2
B1	Internal timing (default).	Both MASTER and SLAVE
C5	Continuous carrier, continuous RTS.	Both MASTER and SLAVE
D7	Data auxiliary set not used.	Both MASTER and SLAVE
E5	Maximum address: 16.	MASTER
SA	RS-232 rise time.	Both MASTER and SLAVE
	Local address [1]	MASTER: 101 SLAVE : 011
	Network address	MASTER: 65 SLAVE : 01

[1] If rack mount, Level II or III, contact the Telephone Company marketing for assistance.

BELL DATAPHONE II 2048A Modem

Private Leased Line, Point-to-Point

Type of Modem	Bell System Type DPII 2048A Data Set
Type of Line	Private Leased Line or Public Telephone Network (Dial Back-Up).
Transmission Rate	4800 bits-per-second

TABLE F-5. BELL 2048A MODEM OPTIONS

Option Number	Description	Comments
A1 A2	Point-to-Point Control Point-to-Point Tributary	MASTER: A1 SLAVE: A2
B1	Internal Timing (Default).	Both MASTER and SLAVE.
C5	Continuous Carrier.	Both MASTER and SLAVE.
D7	Data auxiliary set not used.	Both MASTER and SLAVE.
E5	Maximum Address - 16	MASTER
SA	RS-232 Rise Time	Both MASTER and SLAVE.
	[1] Local Address.	MASTER: 101 SLAVE: 011
	Network Address	MASTER: 65 SLAVE: 01

[1] If rack mount, Level II or III, contact the Telephone Company marketing for addressing assistance.

BELL DATAPHONE II 2096A Modem

Private Leased Line, Point-to-Point

Type of Modem	Bell System Type DPII 2096A Data Set
Type of Line	Private Leased Line or Public Telephone Network (Dial Back-Up).
Transmission Rate	9600 bits-per-second

TABLE F-6. BELL 2096A MODEM OPTIONS

Option Number	Description	Comments
A1 A2	Point-to-Point Control Point-to-Point Tributary	MASTER: A1 SLAVE: A2
B1	Internal Timing (Default).	Both MASTER and SLAVE.
C5	Continuous Carrier.	Both MASTER and SLAVE.
D7	Data auxiliary set not used.	Both MASTER and SLAVE.
E5	Maximum Address - 16	MASTER
SA	RS-232 Rise Time	Both MASTER and SLAVE.
	[1] Local Address.	MASTER: 101 SLAVE: 011
	Network Address	MASTER: 65 SLAVE: 01

[1] If rack mount, Level II or III, contact the Telephone Company marketing for addressing assistance.

PSI INSTALLATION VERIFICATION

APPENDIX

G

Light emitting diodes (LEDs) located on the PSI can be used to verify installation. There are four LEDs located on the back of the PSI for RJE/1000-II. The LEDs are located on the left corner of the PSI for the E/F-Series HP 1000, and on the right corner for the A-Series HP 1000.

These items can be verified:

- driver configuration.
- software installation.
- the contents of the RAM file loaded onto the PSI.

This is essentially everything at the HP 1000 side of the communications link.

Table G-1, PSI Verification Steps, relates the LED status to RJE/1000-II status in a typical user command sequence. The RJE/1000-II Event column, after the RINIT request indication, displays the messages you see on your terminal as the subsystem is initialized. The PSI LED column indicates the state of each LED, where 1=on and 0=off. The digits 3210 in the PSI LED column refer the number of each LED on the PSI card. The Time Required column shows the approximate time needed to move from the previous state to the present state. The relative positions of the entries in the RJE/1000-II Event and PSI LED columns indicate their sequencing.

TABLE G-1. PSI VERIFICATION STEPS

RJE/1000-II Event	PSI LED 3210	Time Required	Installation Step Verified
system boot-up or :CN,LU,35B,RS	1111	-----	PSI Installation in HP 1000 backplane and self-test passed.
	1110	-----	PSI self-test executing. Everything is verified except for the modem line driver circuits.
	1111	4 sec	PSI self-test passed without error.
RINIT request Testing PSI...	1110	-----	DVN00 (on E/F-Series) or IDS00 (on A-Series) has been installed correctly. RJE/1000-II uses the driver subsystem (DVN00, EQT's, LUs, etc.) to make this step work. Also, this step verifies the correct read or write LUs in line 6 and 7 of the configuration file.
download file name? Downloading PSI...	1111	4 sec	PSI is downloaded by RINIT. Verifies the download file in line 8 of the configuration file
link type? . . . generate WACK/TTD?	1001	5 sec	Start up the downloaded firmware. This is an intermediate state that may never be noticed.
trace cntl? route cntl? Please establish the RJE link.	1101	-----	The PSI firmware is configured. This step verifies the correctness of the link parameters in the configuration file. (configuration file lines 9 through 21)
	1101	-----	Data Terminal Ready (DTR) on modem is high. This verifies that the cable between the PSI and the modem is installed.

Once the PSI firmware is configured, the LEDs have the 1101 pattern. This pattern should continue until another system boot-up occurs or the PSI is reset with a CN,LU,35B,RS command. It is normal for LED #1 to blink. It is normally off more than on.

If a user notices a deviation from the patterns in Table G-1, the user should notify the installation SE or CE.

CHARACTER SETS

APPENDIX

H

Table H-1 presents all of the ASCII, EBCDIC, and Hollerith codes that RJE/1000-II can handle. This table is the domestic version of these codes. Character sets for the other twelve supported languages are available.

Note that all line transmission must be done in EBCDIC, and that data transmitted or received in transparent mode is not translated.

TABLE H-1. ASCII, EBCDIC, and Hollerith Codes

HOW TO USE THIS TABLE

- The table is sorted by character code, each code being represented by its decimal, octal, and hexadecimal equivalent.
- Each row of the table gives the ASCII and EBCDIC meaning of the character code, the ASCII↔EBCDIC conversion code, and the Hollerith representation (punched card code) for the ASCII character.

The following examples describe several ways of using the table:

Example 1: Suppose you want to determine the ASCII code for the $ character. Scan down the ASCII graphic column until you locate $, then look left on that row to find the character code – 36 (dec), 044 (oct), and 24 (hex). This is the code used by an ASCII device (terminal, printer, computer, etc.) to represent the $ character. Its Hollerith punched card code is 11-3-8.

Example 2: The character code 5B (hex) is the EBCDIC code for what character? Also, when 5B is converted to ASCII (for example, by FCOPY with the EBCDICIN option), what is the octal character code? First, locate 5B in the hex character code column and move right on that row to the EBCDIC graphic which is $. The next column to the right gives the conversion to ASCII, 044. As a check, find 044 (oct) in the character code column, look right to the ASCII graphic column and note that $ converted to EBCDIC is 133 (oct) which equals 5B (hex).

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
0	000	00	NUL	000	12-0-1-8-9	NUL	000
1	001	01	SOH	001	12-1-9	SOH	001
2	002	02	STX	002	12-2-9	STX	002
3	003	03	ETX	003	12-3-9	ETX	003
4	004	04	EOT	067	7-9	PF	234
5	005	05	ENQ	055	0-5-8-9	HT	011
6	006	06	ACK	056	0-6-8-9	LC	206
7	007	07	BEL	057	0-7-8-9	DEL	177
8	010	08	BS	026	11-6-9		227
9	011	09	HT	005	12-5-9		215
10	012	0A	LF	045	0-5-9	SMM	216
11	013	0B	VT	013	12-3-8-9	VT	013
12	014	0C	FF	014	12-4-8-9	FF	014
13	015	0D	CR	015	12-5-8-9	CR	015
14	016	0E	SO	016	12-6-8-9	SO	016
15	017	0F	SI	017	12-7-8-9	SI	017
16	020	10	DLE	020	12-11-1-8-9	DLE	020
17	021	11	DC1	021	11-1-9	DC1	021
18	022	12	DC2	022	11-2-9	DC2	022
19	023	13	DC3	023	11-3-9	TM	023
20	024	14	DC4	074	4-8-9	RES	235
21	025	15	NAK	075	5-8-9	NL	205
22	026	16	SYN	062	2-9	BS	010
23	027	17	ETB	046	0-6-9	IL	207
24	030	18	CAN	030	11-8-9	CAN	030
25	031	19	EM	031	11-1-8-9	EM	031
26	032	1A	SUB	077	7-8-9	CC	222
27	033	1B	ESC	047	0-7-9	CU1	217
28	034	1C	FS	034	11-4-8-9	IFS	034
29	035	1D	GS	035	11-5-8-9	IGS	035
30	036	1E	RS	036	11-6-8-9	IRS	036
31	037	1F	US	037	11-7-8-9	IUS	037
32	040	20	SP	100	Blank	DS	200
33	041	21	!	117	12-7-8	SOS	201
34	042	22	"	177	7-8	FS	202
35	043	23	#	173	3-8		203
36	044	24	$	133	11-3-8	BYP	204
37	045	25	%	154	0-4-8	LF	012
38	046	26	&	120	12	ETB	027
39	047	27	'	175	5-8	ESC	033
40	050	28	(	115	12-5-8		210
41	051	29	)	135	11-5-8		211
42	052	2A	•	134	11-4-8	SM	212
43	053	2B	+	116	12-6-8	CU2	213
44	054	2C	,	153	0-3-8		214
45	055	2D		140	11	ENQ	005
46	056	2E	.	113	12-3-8	ACK	006
47	057	2F	/	141	0-1	BEL	007
48	060	30	0	360	0		220
49	061	31	1	361	1		221
50	062	32	2	362	2	SYN	026
51	063	33	3	363	3		223
52	064	34	4	364	4	PN	224
53	065	35	5	365	5	RS	225
54	066	36	6	366	6	UC	226
55	067	37	7	367	7	EOT	004
56	070	38	8	370	8		230
57	071	39	9	371	9		231
58	072	3A		172	2-8		232
59	073	3B	;	136	11-6-8	CU3	233
60	074	3C	<	114	12-4-8	DC4	024
61	075	3D		176	6-8	NAK	025
62	076	3E	>	156	0-6-8		236
63	077	3F	?	157	0-7-8	SUB	032
64	100	40	@	174	4-8	SP	040
65	101	41	A	301	12-1		240
66	102	42	B	302	12-2		241
67	103	43	C	303	12-3		242
68	104	44	D	304	12-4		243
69	105	45	E	305	12-5		244
70	106	46	F	306	12-6		245
71	107	47	G	307	12-7		246
72	110	48	H	310	12-8		247
73	111	49	I	311	12-9		250
74	112	4A	J	321	11-1	¢	133
75	113	4B	K	322	11-2	.	056
76	114	4C	L	323	11-3	<	074
77	115	4D	M	324	11-4	(	050
78	116	4E	N	325	11-5	+	053
79	117	4F	O	326	11-6	|	041
80	120	50	P	327	11-7	&	046
81	121	51	Q	330	11-8		251
82	122	52	R	331	11-9		252
83	123	53	S	342	0-2		253
84	124	54	T	343	0-3		254
85	125	55	U	344	0-4		255
86	126	56	V	345	0-5		256
87	127	57	W	346	0-6		257
88	130	58	X	347	0-7		260
89	131	59	Y	350	0-8		261
90	132	5A	Z	351	0-9	!	135
91	133	5B	[	112	12-2-8	$	044
92	134	5C	\	340	0-2-8	•	052
93	135	5D	]	132	11-2-8	)	051
94	136	5E	^	137	11-7-8	;	073
95	137	5F	_	155	0-5-8	¬	136

TABLE H-1. ASCII, EBCDIC, and Hollerith Codes (continued)

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
96	140	60	`	171	1-8	-	055
97	141	61	a	201	12-0-1	/	057
98	142	62	b	202	12-0-2		262
99	143	63	c	203	12-0-3		263
100	144	64	d	204	12-0-4		264
101	145	65	e	205	12-0-5		265
102	146	66	f	206	12-0-6		266
103	147	67	g	207	12-0-7		267
104	150	68	h	210	12-0-8		270
105	151	69	i	211	12-0-9		271
106	152	6A	j	221	12-11-1	|	174
107	153	6B	k	222	12-11-2	,	054
108	154	6C	l	223	12-11-3	%	045
109	155	6D	m	224	12-11-4	_	137
110	156	6E	n	225	12-11-5	>	076
111	157	6F	o	226	12-11-6	?	077
112	160	70	p	227	12-11-7		272
113	161	71	q	230	12-11-B		273
114	162	72	r	231	12-11-9		274
115	163	73	s	242	11-0-2		275
116	164	74	t	243	11-0-3		276
117	165	75	u	244	11-0-4		277
118	166	76	v	245	11-0-5		300
119	167	77	w	246	11-0-6		301
120	170	7B	x	247	11-0-7		302
121	171	79	y	250	11-0-8	`	140
122	172	7A	z	251	11-0-9	:	072
123	173	7B	{	300	12-0	#	043
124	174	7C	|	152	12-11	@	100
125	175	7D	}	320	11-0	'	047
126	176	7E	~	241	11-0-1	=	075
127	177	7F	DEL	007	12-7-9	"	042
128	200	80		040	11-0-1-8-9		303
129	201	81		041	0-1-9	a	141
130	202	82		042	0-2-9	b	142
131	203	83		043	0-3-9	c	143
132	204	84		044	0-4-9	d	144
133	205	85		025	11-5-9	e	145
134	206	86		006	12-6-9	f	146
135	207	87		027	11-7-9	g	147
136	210	88		050	0-8-9	h	150
137	211	89		051	0-1-8-9	i	151
138	212	8A		052	0-2-8-9		304
139	213	8B		053	0-3-8-9		305
140	214	8C		054	0-4-8-9		306
141	215	BD		011	12-1-8-9		307
142	216	8E		012	12-2-8-9		310
143	217	8F		033	11-3-8-9		311
144	220	90		060	12-11-0-1-8-9		312
145	221	91		061	1-9	j	152
146	222	92		032	11-2-8-9	k	153
147	223	93		063	3-9	l	154
148	224	94		064	4-9	m	155
149	225	95		065	5-9	n	156
150	226	96		066	6-9	o	157
151	227	97		010	12-8-9	p	160
152	230	98		070	8-9	q	161
153	231	99		071	1-B-9	r	162
154	232	9A		072	2-B-9		313
155	233	9B		073	3-8-9		314
156	234	9C		004	12-4-9		315
157	235	9D		024	11-4-9		316
158	236	9E		076	6-8-9		317
159	237	9F		341	11-0-1-9		320
160	240	A0		101	12-0-1-9		321
161	241	A1		102	12-0-2-9	~	176
162	242	A2		103	12-0-3-9	s	163
163	243	A3		104	12-0-4-9	t	164
164	244	A4		105	12-0-5-9	u	165
165	245	A5		106	12-0-6-9	v	166
166	246	A6		107	12-0-7-9	w	167
167	247	A7		110	12-0-8-9	x	170
168	250	A8		111	12-1-8	y	171
169	251	A9		121	12-11-1-9	z	172
170	252	AA		122	12-11-2-9		322
171	253	AB		123	12-11-3-9		323
172	254	AC		124	12-11-4-9		324
173	255	AD		125	12-11-5-9		325
174	256	AE		126	12-11-6-9		326
175	257	AF		127	12-11-7-9		327

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
176	260	B0		130	12-11-8-9		330
177	261	B1		131	11-1-8		331
178	262	B2		142	11-0-2-9		332
179	263	83		143	11-0-3-9		333
180	264	B4		144	11-0-4-9		334
181	265	B5		145	11-0-5-9		335
182	266	B6		146	11-0-6-9		336
183	267	B7		147	11-0-7-9		337
184	270	B8		150	11-0-8-9		340
185	271	B9		151	0-1-8		341
186	272	BA		160	12-11-0		342
187	273	B8		161	12-11-0-1-9		343
188	274	BC		162	12-11-0-2-9		344
189	275	BD		163	12-11-0-3-9		345
190	276	BE		164	12-11-0-4-9		346
191	277	BF		165	12-11-0-5-9		347
192	300	C0		166	12-11-0-6-9	{	173
193	301	C1		167	12-11-0-7-9	A	101
194	302	C2		170	12-11-0-8-9	B	102
195	303	C3		200	12-0-1-8	C	103
196	304	C4		212	12-0-2-8	D	104
197	305	C5		213	12-0-3-8	E	105
198	306	C6		214	12-0-4-8	F	106
199	307	C7		215	12-0-5-8	G	107
200	310	C8		216	12-0-6-8	H	110
201	311	C9		217	12-0-7-8	I	111
202	312	CA		220	12-11-1-8		350
203	313	CB		232	12-11-2-8		351
204	314	CC		233	12-11-3-8	ʃ	352
205	315	CD		234	12-11-4-8		353
206	316	CE		235	12-11-5-8	⑂	354
207	317	CF		236	12-11-6-8		355
208	320	D0		237	12-11-7-8	}	175
209	321	D1		240	11-0-1-8	J	112
210	322	D2		252	11-0-2-8	K	113
211	323	D3		253	11-0-3-8	L	114
212	324	D4		254	11-0-4-8	M	115
213	325	D5		255	11-0-5-8	N	116
214	326	D6		256	11-0-6-8	O	117
215	327	D7		257	11-0-7-8	P	120
216	330	D8		260	12-11-0-1-8	O	121
217	331	D9		261	12-11-0-1	R	122
218	332	DA		262	12-11-0-2		356
219	333	DB		263	12-11-0-3		357
220	334	DC		264	12-11-0-4		360
221	335	DD		265	12-11-0-5		361
222	336	DE		266	12-11-0-6		362
223	337	DF		267	12-11-0-7		363
224	340	E0		270	12-11-0-8	\	134
225	341	E1		271	12-11-0-9		237
226	342	E2		272	12-11-0-2-8	S	123
227	343	E3		273	12-11-0-3-8	T	124
228	344	E4		274	12-11-0-4-8	U	125
229	345	E5		275	12-11-0-5-8	V	126
230	346	E6		276	12-11-0-6-8	W	127
231	347	E7		277	12-11-0-7-8	X	130
232	350	E8		312	12-0-2-8-9	Y	131
233	351	E9		313	12-0-3-8-9	Z	132
234	352	EA		314	12-0-4-8-9		364
235	353	EB		315	12-0-5-8-9		365
236	354	EC		316	12-0-6-8-9	⑁	366
237	355	ED		317	12-0-7-8-9		367
238	356	EE		332	12-11-2-8-9		370
239	357	EF		333	12-11-3-8-9		371
240	360	F0		334	12-11-4-8-9	0	060
241	361	F1		335	12-11-5-8-9	1	061
242	362	F2		336	12-11-6-8-9	2	062
243	363	F3		337	12-11-7-8-9	3	063
244	364	F4		352	11-0-2-8-9	4	064
245	365	F5		353	11-0-3-8-9	5	065
246	366	F6		354	11-0-4-8-9	6	066
247	367	F7		355	11-0-5-8-9	7	067
248	370	F8		356	11-0-6-8-9	8	070
249	371	F9		357	11-0-7-B-9	9	071
250	372	FA		372	12-11-0-2-8-9	|	372
251	373	F8		373	12-11-0-3-8-9		373
252	374	FC		374	12-11-0-4-8-9		374
253	375	FD		375	12-11-0-5-8-9		375
254	376	FE		376	12-11-0-6-8-9		376
255	377	FF		377	12-11-0-7-8-9	EO	377

BINARY SYNCHRONOUS COMMUNICATIONS

APPENDIX

I

HANDSHAKING

With binary synchronous communications (BSC) and other line protocols of this type, "handshaking" is the term commonly used to describe the interaction between stations. Typically the following is exchanged:

- Message available for transmission
- Start of text transmission
- Acknowledgment or rejection of text
- Detection of errors
- Retransmission after error detection
- End of transmission

A simplified handshaking sequence is summarized in Figure I-1. This shows handshaking between a terminal and computer, but it could also be between two terminals or two computers. In this sequence, a terminal tells a computer it has a message to transmit. The computer recognizes the terminal and tells it to proceed with the message. When the computer receives the message, it detects an error in the text and requests a retransmission. The retransmission is error-free and the computer asks the terminal for another message. The terminal does not have anything more to transmit and so informs the computer. If the computer had a message to transmit to the terminal it could now do so. Since it does not, it disconnects from the terminal.

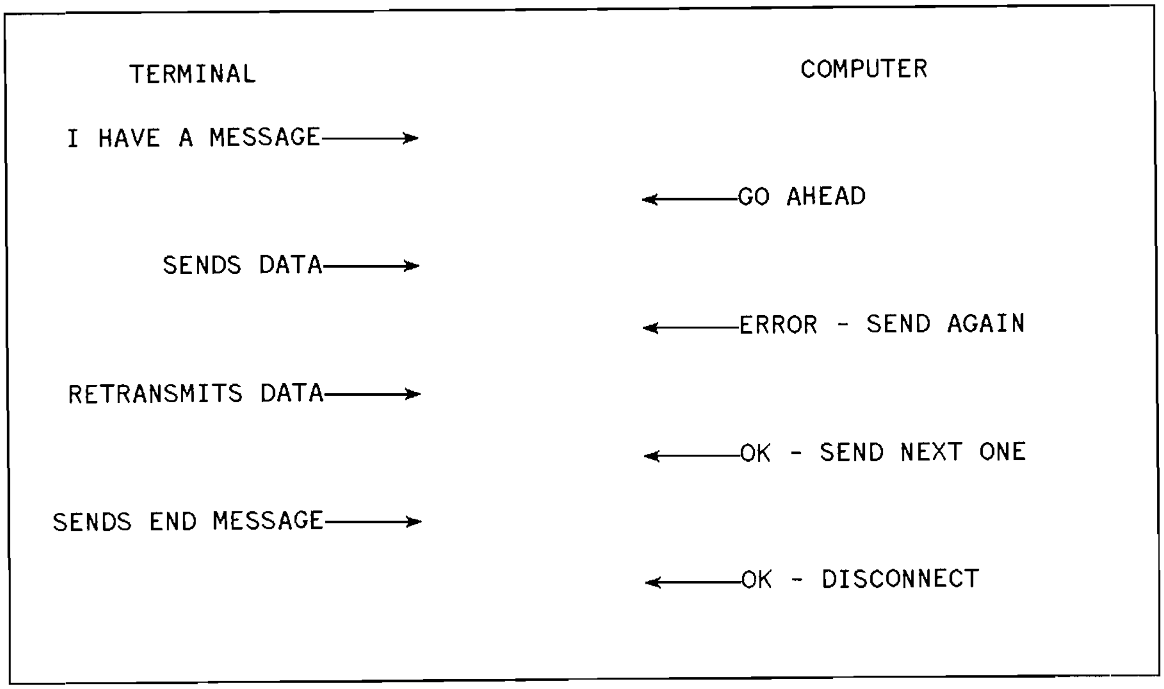


Figure I-1. Simplified Handshaking Sequence

MESSAGE SEQUENCE

A message exchange is initiated when a location sends a synchronization (SYN) sequence and an enquiry (ENQ) to another location. If the other location can accept a message, it acknowledges (ACK) the enquiry. Throughout the message exchange sequence, each acknowledgment is alternately numbered one and zero. When an acknowledgment is not received by the sender, the next one will be out of sequence and an error detected.

As shown in Figure I-2, the SYN synchronous characters are sent to assure that the transmitting and receiving stations are synchronized for data exchange. These are followed by the ENQ control character used to request the communications link and initiate the exhange. The IBM host responds with a synchronizing sequence and an ACK0 even positive acknowledgement.

For error detection RJE-II uses Cyclic Redundancy Checking (CRC). Each message block includes a CRC check sum. If an error is detected in a received block, the receiving system responds with a negative acknowledgement, NAK. The sending system then retransmits the message block in error. If the retransmission is error free, the receiving system returns an odd positive acknowledgement, ACK1. Synchronization is performed at the start of each message block. Each block is terminated with an ETB except for the last block which is terminated with an ETX. The end of transmission is indicated by sending an EOT control character. At that time, the communications link is available for control by either remote or host stations.

The bisync contention protocol contains many control characters including those for error control, record separation, and data compression. The line protocol characters used in the above example represent only a few of the BSC control characters. Table I-1 lists character mnemonics along with their meaning and function.

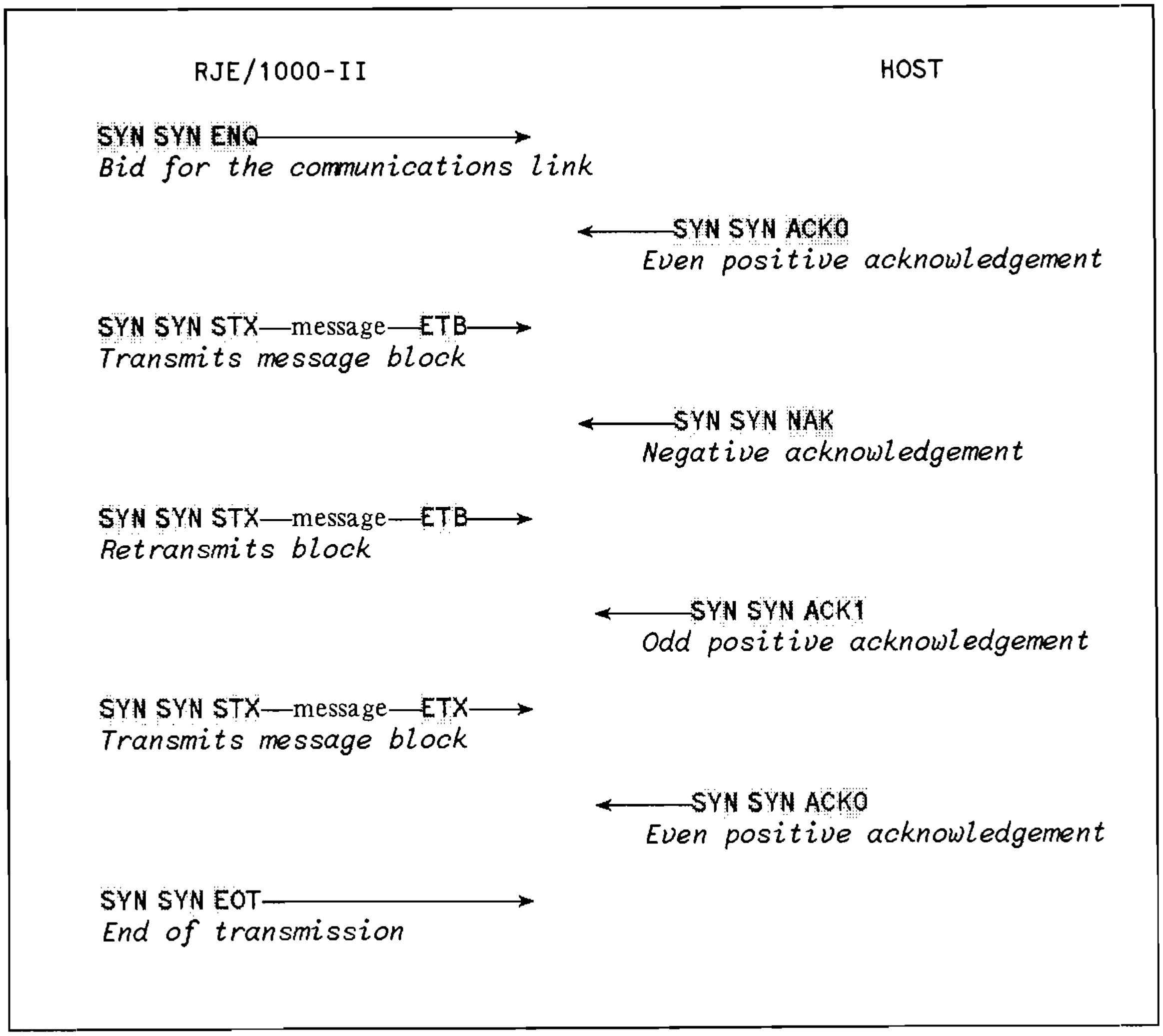

RJE/1000-II
HOST
SYN SYN ENQ
Bid for the communications link
SYN SYN ACK0
Even positive acknowledgement
SYN SYN STX—message—ETB
Transmits message block
SYN SYN NAK
Negative acknowledgement
SYN SYN STX—message—ETB
Retransmits block
SYN SYN ACK1
Odd positive acknowledgement
SYN SYN STX—message—ETX
Transmits message block
SYN SYN ACK0
Even positive acknowledgement
SYN SYN EOT
End of transmission

Figure I-2. Message Sequence

TABLE I-1. BSC CONTROL CHARACTERS

Character	Meaning	Function
SYN	Synchronous Idle	Establishes and maintains character synchronization prior to the message block. Also used as time fill that maintains synchronization in the absence of control characters and data.
STX	Start of Text	Transmitted before a block of text characters.
ETB	End of Transmission Block	Indicates the end of the text block starting with STX or SOH. BCC is sent after ETB, requiring the receiver to respond with ACK, NAK or optionally WACK or RVI.
US or ITB	End of Intermediate Transmission Block	Divides a message for error checking purposes without the turnaround required by ETB. BCC follows ITB and resets the block-check count to zero. STX or SOH is not required for the following text blocks, but STX is required if a header is followed by text.
ETX	End of Text	Terminates a block of characters, begun with SOH or STX, after the last block in a sequence of blocks. BCC immediately follows ETX, requiring a receiver status reply.
EOT	End of Transmission	Concludes transmission, resets all stations to control mode (either transmitter or receiver). Also a non-transmit response to a poll and an abort signal for a malfunction.
ENQ	Enquiry	Bids for the line in a point-to-point and multipoint connection: requests retransmission of last acknowledgment; or requests that preceding block to be ignored. Also indicates completion of poll or selection sequence.
ACK (two bytes)	Affirmative Acknowledgment	Previous block accepted and error-free, receiver reaxy for next block. Also a positive response to selection (multi-point) or line bid (point-to-point).
SOH	Start of Heading	Transmitted before a block of heading characters. These contain information such as the routing and priority of the message.
NAK	Negative Acknowledgment	Previous block unacceptable and retransmission required. Also a negative response to a selection or line bid.

TABLE I-1. BSC CONTROL CHARACTERS

Character	Meaning	Function
TTD (two bytes)	Temporary Text Delay	Transmitter not ready to commence transmission but wants to maintain connection. Sent two seconds after message received to avoid three second timeout.
RVI (two bytes)	Reverse Interrupt	Sent to a transmitter by a receiver in place of ACK, indicating the receiver has a high priority message waiting transmission.
WACK (two bytes)	Wait Before Transmitting Positive Acknowledgment	Previous block accepted and error-free, but receiver not ready for next block. Will continue to respond with WACK until ready to receive. Also a positive response to a text or heading block selection sequence (multi-point), line bid (point-to-point) or identification line bid sequence (switched network).
DLE	Data Link Escape	Prefix for control characters during transparent mode, when control characters have no control meaning unless prefixed by DLE. The two-character sequences ACK, WACK, and RVI have DLE as the first character.
DLE and EOT	Disconnect Sequence for a Switched Line	Transmitted on a switched line when all message exchanges are complete. Can optionally be transmitted at any time instead of EOT to cause a disconnect.
Pad	None	Added before (leading pad) a transmission and after (trailing pad) a transmission to ensure the first and last characters are properly transmitted.

IBM 2780/3780 COMPARISON

Table J-1 summarizes the features of the IBM 2780 Data Transmission Terminal and the IBM 3780 Data Communication Terminal.

TABLE J-1. IBM 2780/3780 COMPARISON

Feature	IBM 2780	IBM 3780
Peripheral Devices	400 cpm card reader 355 cpm card punch (optional) 300 lpm line printer (optional)	600 cpm card reader 359 cpm card punch (optional) 425 lpm line printer
Half/Full Duplex Protocol	Half duplex only	Half duplex only
Permissible Communications Networks	Point-to-Point (leased or switched) Multipoint (2- or 4- wire)	Point-to-Point (leased or switched) Multipoint (2- or 4- wire)
Permissible Transmission Codes	ASCII (non-transparent mode only) EBCDIC (transparent or non-transparent modes) Six-bit transcode	ASCII (non-transparent mode only) EBCDIC (transparent or non-transparent modes)
Error checking	ASCII: Odd parity VRC and 8-bit LRC EBCDIC: 16-bit CRC Six-bit transcode: 12-bit CRC	ASCII: Odd parity VRC and 8-bit LRC EBCDIC: 16-bit CRC
Permissible Transmission Rates	1200, 2000, 2400, 4800 bps	1200, 1800, 2400, 3600, 4800, 7200 bps
Buffer(s)	One 400-byte buffer used for either input or output	Two 512-byte buffers used simultaneously for input or output
Accept Horizontal Tabulation Codes?	Yes (optional)	YES
Accept Vertical Forms Control Codes?	YES	YES

TABLE J-1. IBM 2780/3780 COMPARISON (cont'd)

Feature	IBM 2780	IBM 3780
Transmit and Receive in Transparent Mode?	YES (optional; applies only to EBCDIC mode)	YES (optional; applies only to EBCDIC mode)
Short-Record Truncation?	YES (user must supply EM control characters in the data)	YES (done automatically; user does not need to supply EM or IRS control characters in data)
Blanks Compression?	NO	YES (done automatically; user does not need to supply any special characters in the data)
Maximum Blocking Factor?	2 (optionally 7)	Non-Transparent Mode: 255 Transparent Mode: 1 (optionally 6)
Generate WACK and TTD Sequences when Temporarily Unable to Transmit or Receive?	NO	YES
Intermediate Block Terminator Character?	ASCII: US BCC EBCDIC: IUS BCC Six-bit transcode: ITB BCC	ASCII: RS (no BCC) EBCDIC: IRS (no BCC) Receive Mode: Will accept IUS BCC
Recognize WACK?	YES (receive mode only)	YES.
Component Select Codes (Point-to-Point Only)	ESC followed by any carriage control code specifies printer ESC 4 specifies punch	DC1 specifies printer DC2 or DC3 specifies punch
SOH at Start of Received Text	Treated as an STX	Not recognized

INDEX

READER COMMENT SHEET

HP AdvanceNet

RJE/1000-II
User's/Programmer's Reference Manual

91781-90001 October 1984

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. Please explain your answers under Comments, below, and use additional pages if necessary.

Is this manual technically accurate?	□ **Yes**	□ **No**
Are the concepts and wording easy to understand?	□ **Yes**	□ **No**
Is the format of this manual convenient in size, arrangement, and readability?	□ **Yes**	□ **No**

Comments:

This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

FROM: **Date** ______

Name ______

Company ______

Address ______

______

______

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Publications Manager
Hewlett-Packard Company
Information Networks Division
19420 Homestead Road
Cupertino, CA 95014

FOLD

FOLD