Guide to IBM Remote Job Entry for PWB/UNIX Users

A. L. Sabsevitz

Bell Laboratories
Piscataway, New Jersey 08854

1. PREFACE

A set of background processes supports remote job entry (RJE) from a PWB/UNIX* computer to IBM System/360 and /370 host computers. "Hasp" is the common name used for the collection of programs and for the file organization that provides this facility. The *PWB/UNIX User's Manual* page *hasp*(VIII) summarizes their design and operating procedures. That manual also contains a terse description of the *send*(I) command, which is the user's primary interface to RJE. These are the definitive sources for information about RJE. Although the word "Hasp" may be used in this guide, it represents all IBM RJE subsystems of the PWB/UNIX system.

This guide is a tutorial overview of RJE†. It is addressed to the user who needs to know how to use the system, but does *not* need to know details of its implementation. The two following sections constitute an introduction to RJE.

2. PRELIMINARIES

To become a PWB/UNIX user, you must receive a login name that identifies you to the PWB/UNIX system. You should also get a copy of the *PWB/UNIX User's Manual*. This is a fairly complete description of the system and includes a section entitled "How to Get Started," which introduces you to PWB/UNIX; you should read that section before proceeding with this guide.

In order to begin using RJE, you need only become familiar with a subset of basic commands. You must understand the directory structure of the file system, and you should know something about the attributes of files: see *chdir*(I), *chmod*(I), *chown*(I), *cp*(I), *ln*(I), *ls*(I), *mkdir*(I), *mv*(I), *rm*(I). You must know how to enter, edit, and examine text files: see *cat*(I), *ed*(I), *pr*(I). You should know how to communicate with other users and with the system: see *mail*(I), *mesg*(I), *who*(I), *write*(I). And, finally, you might have to know how to describe your terminal to the system: see *ascii*(V), *stty*(I), *tabs*(I).

3. BASIC RJE

Let's suppose that you have used the editor, *ed*(I), to create a file "jobfile" that contains your control statements (JCL) and input data. This file should look exactly like a card deck, except that alphabetic characters, for convenience, may be in either upper or lower case. Here is an example:

```
% cat jobfile
//gener job (9999,r740),pgmrname,class=x usr=(mylogin,myplace)
//step exec pgm=iebgener
//sysprint dd sysout=a
//sysin dd dummy
//sysut2 dd sysout=a
//sysut1 dd *
   first card of data
      :
      :
   last card of data
/*
```

---

* UNIX is a Trademark/Service Mark of the Bell System

† The original versions of this manual and of RJE itself were written by T. G. Lyons.

To submit this job for execution, you must invoke the *send*(I) command:

```
% send jobfile
```

The system will reply:

```
10 cards.
Queued as /usr/hasp/xmit311.
```

Note that *send* tells you how many cards it submitted and reports the position that your job has been assigned in the queue of all jobs waiting to be transmitted to the host system. Until the transmission of the job actually begins, you can prevent the job from being transmitted by doing a "chmod 0" on the queued file to make it unreadable. For our example, you could say: "chmod 0 /usr/hasp/xmit311".

When your job is accepted by the host system, a job number will be assigned to it, and an acknowledgement message will be generated. This indicates that your job has been scheduled on the host system. Later, after the job has executed, its output will be returned to the PWB/UNIX system. You will be notified automatically of both of these events: if you are logged in when RJE detects these events, and if you are permitting messages to be sent to your terminal (see *mesg*(I)), the following two messages will be sent to you (still using the example above) when the job is scheduled and when the output is returned, respectively:

Two bells
S12.18.42 JOB 384 ON RM4.RD1 -- GENER PGMRNAME
Bell

Two bells
12:21:54 /a1/user/rje/prnt0 384.gener ready
Bell

The job-acknowledgement message is passed on directly from the host system, as indicated by the fact that it appears in upper case. The output-ready message is generated by RJE and appears in lower case. Two bells, with an interval of one second between them, precede each message. They should be interpreted as a warning to stop typing on your terminal, so that the imminent message is not interspersed with your typing.

If you are not logged in when one of these events occurs, or if you do not allow messages to be sent to your terminal, then the notification will be posted to you via the *mail*(I) command. You can prevent messages directly by executing the *mesg*(I) command, or indirectly by executing another command, such as *pr*(I), which prohibits messages for as long as it is active. You may inspect (by invoking the *mail* command) your *.mail* file at any time for messages that have been diverted. For this example, this might look as follows:

```
% mail

From rje Mon Aug  1 12:20:36 1977
S12.18.42 JOB 384 ON RM4.RD1 -- GENER PGMRNAME

From rje Mon Aug  1 12:21:55 1977
12:21:54 /a1/user/rje/prnt0 384.gener ready

Save?
```

Note that there may exist a discrepancy between the host and PWB/UNIX clocks.

The job-acknowledgement message performs two functions. First, it confirms the fact that your job has been scheduled for eventual execution. Second, it assigns a number to the job in such a way that the number and the name together will uniquely identify the job for some period of time.

The output-ready message provides the name of a PWB/UNIX file into which output has been written and identifies the job to which the output belongs:

```
% ls -l prnt0
-r--r-xr-- 1 rje       1184 Aug  1 12:21 prnt0
```

Note that rje retains ownership of the output and allows you only read access to it. It is intended that you will inspect the file, perhaps extract some information from it and then promptly delete it (see *rm*(I)):

```
% rm -f prnt0
```

The retention of machine-generated files, such as RJE output, is discouraged. It is your responsibility to remove files from your RJE directory. Files of RJE output may not exceed 256K bytes. In addition, only files of 64K bytes are guaranteed to be accepted in their entirety. Limits of 64K, 128K, or 192K may be automatically enforced if file space gets scarce. Output beyond the current limit will be discarded, with no provision for retrieving it. The user should also be aware of the fact that RJE attempts to keep roughly 1000 "blocks" free on any file system it uses. Warning messages or suspension of certain functions will occur as this limit is approached.

The most elementary way to examine your output is to *cat* it to your terminal. The Appendix shows the result of listing the output of our sample job in this way. Printouts are stored with standard tabs to conserve space, so you must ensure that the tabs are set on your terminal at every eighth column across the entire line; *tabs*(I) will do that for you. Because PWB/UNIX has no high-volume printing capability, you should route to the host's printer any large listings of which you desire a hard copy.

The structure of an output listing will generally conform to the following sequence:

```
HASP log
jcl information
data sets
HASP end
```

"Burst" pages are discarded. Single, double, and triple spacing is reflected in the output file, but other forms controls, such as the skip to the top of a new page, are suppressed. Page boundaries are indicated by the presence of a space *character* at the end of the last line of each page.

The big file scanner *bfs*(I) or the context editor *ed*(I) provide a more flexible method than *cat*(I) for examining printed output; *bfs* can handle files of any size and is more efficient than *ed* for scanning files.

RJE is also capable of receiving punched output as formatted files (see *ebcdic*(V)); this format allows an exact representation of an arbitrary card deck to be stored on the PWB/UNIX machine. However, there are few commands that can be used to manipulate EBCDIC files. You will probably want to route your punched output to one of the host's output devices.

4. SEND COMMAND

The *send*(I) command is capable of more general processing than has been indicated in the previous section. In the first place, it will concatenate a sequence of files to create a single job stream. This allows files of JCL and files of data to be maintained separately on the PWB/UNIX machine. In addition, it recognizes any line of an input file that begins with the character "~" as being a *control* line that can call for the inclusion, inside the current file, of some other file. This allows you to "send" a top-level skeleton that "pulls" in subordinate files as needed. Some of these may be "virtual" files that actually consist of the output of PWB/UNIX commands or Shell procedures. Furthermore, the *send* command is able to collect input directly from a terminal, and can be instructed to prompt for required information.

Each source of input can contain a format specification that determines such things as how to expand tabs and how long can an input line be. The manual page for *fspec*(V) explains how to define such formats. When properly instructed, *send* will also replace arbitrarily defined "keywords" by other text strings or by EBCDIC character codes. (These two substitution facilities are useful in other applications besides RJE; for that reason, *send* may be invoked under the name *gath* to produce standard output *without* submitting an RJE job.)

Two aspects of *send* with which everyone should be acquainted are the ability to specify to which computer a job is to be submitted, and the ability to verify a job prior to submission. To run our sample job on a host machine known to RJE as "A", we would issue the command:

```
% send A jobfile
```

When no host is explicitly cited, *send* makes a reasonable choice.

To verify the text of a collected job stream, without actually submitting it, set the "-lq" flags:

```
% send -lq jobfile
```

The complete list of arguments and flags that control the execution of *send* can be found in *send*(1).

5. JOB STREAM

It is assumed that the job stream submitted as the result of a single execution of *send* consists of a single *job*, i.e., the file that is queued for transmission should contain one JOB card near the beginning and no others. A priority control card may legitimately precede the JOB card. The JOB card must conform to the local installation's standard. At BISP, it has the following structure:

```
//name job (acct[,...]),pgmrname[,keywds=?] [usr=...]
```

6. USER SPECIFICATION

The "usr=..." field is required if any print or punch output is to be delivered to the PWB/UNIX user.

```
usr=(login,place[,[level][,retry]])
```

where *login* is the PWB/UNIX login name of the user, *level* is the desired level of notification (see end of this section for an explanation), *retry* is a one-character code specifying the number of attempts to retransmit an entire job if the transmission to the host computer is interrupted by an unrecoverable error (default is three attempts; the digits "1" through "9" specify that number of retries; "0", "y", or "Y" invoke the default; any other entry in this field limits the number of attempts to one, i.e., no retry), and *place* is as follows:

A. If *place* is the name of a *directory* (writable by others), then the output file is placed there as a unique *prnt* or *pnch* file (up to 500 of each allowed). The mode of the file will be 454.

B. If *place* is the name of an existing, writable (by others), non-executable (by others) file, then the output file replaces it. The mode of the file will be 454.

C. If *place* is the name of a non-existent file in a writable (by others) directory, then the output file is placed there. The mode of the file will be 454.

D. If *place* is the name of an executable (by others) file, then the RJE output is set up as standard input to *place*, and *place* is executed. Five string arguments are passed to *place*. For example, if *place* is a shell procedure, the following arguments are passed as $1 ... $5:

   1. Flag indicating whether file space is scarce in the file system where *place* resides. 0 indicates that space is *not* scarce, while 1 indicates that it is.
   2. Job name.
   3. Programmer's name.
   4. Job number.
   5. Login name from the "usr=..." specification.

   A ":" is passed if a value is not present.

E. In all other cases, the output will be thrown away.

The *place* value must not be a full pathname, unless it refers to an executable file (see D above). For cases A, B, and C above (and case D, if a full pathname is not supplied), the name of the user's login directory will be used to form a full pathname.

The "usr=..." field may occur anywhere within the first 100 card images sent and within the first 200 output images received by the PWB/UNIX system. The only restrictions are:

- Column one must contain a "/" or a "*".
- "usr=..." must begin after column 4 and must be preceded by a space.

Therefore, the "usr=..." field may be placed on the JOB card, a comment card, passed as data, etc.

For redirection of output by the host, a "usr=..." card, if not already present, must be supplied by the user. This can be done by placing a job step that creates this card before your output steps.

Messages generated by RJE or passed on from the host are assigned a level of importance ranging from 1 to 9. The levels currently in use are:

3 transmittal assurance
5 job acknowledgement
6 output ready message
7 transmit format error

The optional "level" field of the "usr=..." specification must be a one- or two-digit code. A message from the host with importance "x" (where x comes from the above list) is compared with each of the two decimal digits "mw". If $x \geq w$ and if the user is logged in and is accepting messages, the message will be written to his or her terminal. Otherwise, if $x \geq m$, the message will be mailed to the user. In all other cases, the message will be discarded. The default "level field" is "54". You should specify level "1" if you want to receive complete notification, and level "59" to divert the last three messages in the above list to your mailbox.

7. CONTROL CARDS

A number of control cards are recognized by the host's HASP subsystem. Two are of particular interest to RJE users, because they control the disposition of output. These are the ROUTE and OUTPUT cards, and their use is illustrated below. If used, both should be inserted into a job stream *immediately* after the JOB card.

The ROUTE card can be used to direct the entire printed or punched output of a job to a specified destination. Two cards are required to direct both outputs:

```
/*route  punch local
/*route  print  rmt55
```

The ROUTE card has a fixed format. "Print" or "punch" must begin in column 10, and the destination field in column 16.

The proper use of the OUTPUT card is a bit more complicated. It allows you to associate parameters with all SYSOUT data sets whose forms numbers match the one specified on the OUTPUT card. The forms number is fictitious and may consist of up to four characters. A copy count and destination are among the parameters that may be associated with SYSOUT data sets in this way:

```
//name job ...
/*output py d=rmt56
/*output abcd d=local,n=2
//step exec ...
//prt1 dd sysout=a
//prt2 dd sysout=(a,,abcd)
//prt3 dd sysout=(a,,py)
//pnch dd sysout=(b,,abcd)
```

In the above example, one copy of *prt1* would be directed to the default destination and one copy of *prt3* to *rmt56*. Two copies each would be made of *prt2* and *pnch*, and they would remain at the *local* site.

8. MONITORING RJE

RJE is designed to be an autonomous facility that does not require manual supervision. RJE is initiated by the PWB/UNIX operator after system "reboots" and continues in execution indefinitely. Experience has proved it to be reasonably robust, although it is vulnerable to system crashes and reconfigurations.

Users have a right to assume that, if the PWB/UNIX system is up for production use, RJE should also be up. This implies more than an ability to execute the *send*(I) command, which should be available at all times. It means that queued jobs should be submitted to the host for execution and their output returned to the PWB/UNIX system. If a user cannot obtain any throughput from RJE, the user should so advise the PWB/UNIX operators.

The *rjestat*(I) command, invoked without the "-" argument, will report the status of all RJE links for which a given PWB/UNIX system is configured. It may sometimes also print a message of the day from RJE.

```
% rjestat

15:12:24 RJE to B is operating normally.

15:12:25 A is not responding to RJE.
 (8 files queued since 14:34:26)
```

A parenthetical statement, such as the last line above, will summarize any backlog of queued files waiting to be transmitted to the host machine. A backlog that persists for 20 minutes or more often is an indication that there exists a problem with the corresponding RJE link.

A host machine may be reported to be not responding to RJE because it is down, or because of its operator's failure to initialize the associated line, or because of a communications hardware failure.

Appendix—Sample Output Listing

```
% cat rje/prnt0
14.40.31 JOB 384  $HASP373 GENER  STARTED - INIT 26 - CLASS X - SYS RRMA
14.40.32 JOB 384  $HASP395 GENER  ENDED


------ JES2 JOB STATISTICS ------


 1 AUG 77 JOB EXECUTION DATE


          54 CARDS READ


          76 SYSOUT PRINT RECORDS


           0 SYSOUT PUNCH RECORDS


      0.01 MINUTES EXECUTION TIME
   1        //GENER JOB (9999,R740),PGMRNAME,CLASS=X                    JOB 384
            ...       USR=(MYLOGIN,MYPLACE)
   2        //IEBGENER EXEC PGM=IEBGENER
   3        //SYSPRINT DD DUMMY
   4        //SYSIN DD DUMMY
   5        //SYSUT2 DD SYSOUT=A
   6        //SYSUT1 DD *
            //
IEF236I ALLOC. FOR GENER IEBGENER
IEF237I DMY ALLOCATED TO SYSPRINT
IEF237I DMY ALLOCATED TO SYSIN
IEF237I JES ALLOCATED TO SYSUT2
IEF237I JES ALLOCATED TO SYSUT1
IEF142I GENER IEBGENER - STEP WAS EXECUTED - COND CODE 0000
IEF285I   JES2.JOB0384.SO0102                    SYSOUT
IEF285I   JES2.JOB0384.SI0101                    SYSIN
IEF373I STEP /IEBGENER/ START 77242.1440
IEF374I STEP /IEBGENER/ STOP 77242.1440 CPU  0MIN 00.13SEC SRB 0MIN 00.01SEC VIRT 36K SYS 188K

******* SERVICE UNITS=0000174          SERVICE RATE=0000268 SERVICE UNITS/SECOND
******* PERFORMANCE GROUP=005
******* EXCP COUNT BY UNIT ADDRESS
IEF375I  JOB /GENER / START 77242.1440
IEF376I  JOB /GENER / STOP  77242.1440 CPU    0MIN 00.13SEC SRB    0MIN 00.01SEC

******* SERVICE UNITS=0000174          SERVICE RATE=0000268 SERVICE UNITS/SECOND
******* APPROXIMATE PROCESSING TIME=           .01 MINUTES
******* EXCPS=000000000
******* PROJECTED CHARGES=             .01
```

first line of data

:

last line of data

```
-OS/VS2 REL 3.7 JES2- END  JOBNAME=GENER   BIN=R740      JOB #=384  PGMRNAME
-OS/VS2 REL 3.7 JES2- END  JOBNAME=GENER   BIN=R740      JOB #=384  PGMRNAME
-OS/VS2 REL 3.7 JES2- END  JOBNAME=GENER   BIN=R740      JOB #=384  PGMRNAME
% rm -f rje/prnt0
```

ADDENDUM #1

UNIVAC RJE

I. PREFACE {1.} *

A set of PWB/UNIX † background processes supports remote job entry (RJE) from a PWB/UNIX computer to UNIVAC 1100-series host computers. "Uvac" is the common name used for the collection of programs and for the file organization that provides this facility.

II. BASIC RJE {3.}

Alphabetic characters, for convenience, may be in either upper or lower case. The master space "@" may be represented as "`". Here is an example:

```
% cat jobfile
`run   echo,acct-no,project-id     .    usr= (mylogin,myplace)
`ed,i    .eit
   first card of data
        :
        :
   last card of data
`fin
```

The system will reply:

```
8 cards.
Queued as /usr/uvac/xmit311.
```

The job acknowledgement messages are:

Two bells
$12.18.42 001.ECHO STARTED ACCT-NO
Bell

Two bells
12:21:54 /a1/user/rje/prnt0 .echo ready
Bell

To route your printout to the host's printer, insert the following control card into the run stream following the RUN card:

```
`sym     print$,,pr
```

"Burst" pages are *not* discarded and the reception of punched output is *not* supported.

III. JOB STREAM {5.}

The RUN card must conform to the local installation's standard. In general, it has the following format:

```
@run    name,acct-no,project-id    [ . usr= ...]
```

IV. USER SPECIFICATION {6.}

The "usr= ..." field may occur anywhere within the first 100 card images sent and within the first 200 output images received by the PWB/UNIX system. The only restrictions are:

- Column 1 must contain a "@" or a "`".
- "usr= ..." must begin after column 4 and must be preceded by a space.

Therefore, the "usr= ..." field may be placed on the RUN card, a message card, passed as data, etc.

---

* Numbers enclosed in curly braces are section numbers of *Guide to IBM Remote Job Entry for PWB/UNIX Users* by A. L. Sabsevitz, September 1977.

† UNIX is a Trademark of Bell Laboratories.

V. CONTROL CARDS {7.}

This section of the guide is not applicable to UNIVAC RJE.

VI. MONITORING RJE {8.}

The interactive status terminal capability of the *rjestat*(I) command is *not* implemented.