Table of Contents

List of Figures

Basic Concepts

NOTE The runtime directory that is referred to in this document is the directory that contains the executable files for the AT&T BSC/RJE Emulator+. The default name for the directory is **/usr/bscrje**, but the System Administrator may choose any name for it.

RJE Jobs

The most basic RJE concept is the job. Any command that sends something to the host creates a job. Before a job is sent, the BSC/RJE Emulator+ assigns it a local job number. A message containing this job number is sent to the user's terminal. This number remains associated with the job until it has been completely transmitted to the host. To cancel the job before it is sent, or to determine the job's position on the local queue of jobs waiting to be sent, you must know this local job number. (Note that when the host receives the job, it assigns its own job number to it, and this new number serves to identify the job as long as it is on the host system. The BSC/RJE Emulator+ does not use the host job number.)

The Job Queue

Jobs that are waiting to be sent to the host are kept on a local job queue. At most 24 jobs can be on the queue at any one time. The job queue consists of files named R*nnnn*, where *nnnn* is the local job number.

Job Output from the Host

Job output received from the host is initially placed in the **output** subdirectory of the BSC/RJE Emulator+ runtime directory. (This directory is normally called **/usr/bscrje**, and it is referred to as such throughout this document. Note also that there may be more than one emulator running and therefore more than one runtime directory.) When the file has been completely received, it is submitted to the **disptch** process for automatic routing, assuming the automatic routing feature has been enabled by the System Administrator. (See question 9 in "What the **startrje** Questions Mean" in the *System Administrator's Guide*.) Automatic routing allows each user to specify a directory or file in which the output from a job is to be placed. (For details see the section "Automatic Routing of Job Output" in "Receiving Job Files" below.)

Configuring Your Environment

Before using the AT&T BSC/RJE Emulator+, you should configure your environment as follows:

Step 1. Set the **BR** environment variable equal to the *rjepipe* your System Administrator specified when the BSC/RJE Emulator+ was started, and export it. (Check with your System Administrator for the full pathname of the *rjepipe* you should use.) For example:

```
BR=/usr/bscrje/hostx
export BR
```

Note that if you do not set this variable it will default to HOSTA.

Step 2. Set your **PATH** variable to include the runtime directory of the BSC/RJE Emulator+, and export it. For example:

```
PATH=/usr/bscrje:$PATH
export PATH
```

Note that these changes to your environment can be added to your **.profile** file so that they will be executed automatically each time you log in.

Using the AT&T BSC/RJE Emulator+

The **rje** Command

After your System Administrator has started the AT&T BSC/RJE Emulator+ and you have configured your environment, you can use your terminal to send files to and receive files from a remote host. You can also cancel jobs queued locally and display local status and job information. You can perform all these functions by executing the **rje** command as a UNIX system shell command either interactively from the command line or from a script. All **rje** commands have the following format:

rje −*func* [**−h** *rjepipe*] *cmdopts...*

Figure 1 lists the basic options, describes them, and gives the environment variables and default values used when they are not specified.

Option	Description	Default Environ. Variable	Default Value
−*func*	*func* designates the function of the command. It must be the first option after **rje**. Valid user options are: **−s** send **−c** cancel **−x** display queue **−l** list RJE statistics **−o** console operator	—	—
−h *rjepipe*	*rjepipe* specifies the pipe to the BSC/RJE Emulator+ process, **bscrje**. Standard option for all **rje** commands.	**BR**	HOSTA
cmdopts	*cmdopts* are the options specific to each command. For details see the command descriptions.	—	—

Figure 1: Basic **rje** Command Options

The −h *rjepipe* Option

The user should take particular note of this option and of the corresponding default environment variable **BR**. Whenever you issue an **rje** command, you invoke the **rje** process. The **rje** process uses the named pipe known as the *rjepipe* to communicate your command to the BSC/RJE Emulator+ process, **bscrje**. The name of the *rjepipe* is designated by the System Administrator when the BSC/RJE Emulator+ is started. If you do not specify the **−h** *rjepipe* option, the **rje** process must seek elsewhere for the name of the *rjepipe* it should use.

The following list describes in order the ways in which the **rje** process attempts to obtain the name of the *rjepipe* it should use to communicate with the **bscrje** process.

1. The user may specify the pipe name in the **rje** command by using the **−h** *rjepipe* option. To do this the user must know the exact name of the pipe that the System Administrator used to start the BSC/RJE Emulator+. For example the user might enter:

   ```
   rje −s −h /usr/bscrje/HOSTX file1
   ```

   This command would result in **file1** being processed and sent by the **bscrje** process named by the pipe **/usr/bscrje/HOSTX**.

2. If the user does not specify the pipe name, the **rje** process searches the user's environment list looking for the **BR** variable. If **BR** is set, its value is used as the name of the *rjepipe*. (This is the easiest and the best way of furnishing the **rje** process with the pipe name. See "Configuring Your Environment" in "Basic Concepts" above.)

3. If the **BR** variable is not set, the **rje** process looks for a pipe named HOSTA in the following directories, in order:

   - the user's current working directory
   - **/usr/bscrje**
   - **/tmp**

If the **rje** process is unable to obtain the name of the *rjepipe* by any of these means, the **rje** command aborts and the message `HOST NOT FOUND` is displayed.

The RJE Application Program Interface

The send and cancel commands can also be invoked from a C program written by the user that employs the Application Program Interface (API) functions. In this case the C program calls the API external function, which then executes the command. The list of options and arguments available to the API functions is the same as it is for **rje** commands invoked from the shell.

The RJE API functions are found in the library **/usr/lib/librje.a**, so this library must be linked to the user's object module(s). One way to do this is to enter:

```
cc usrprog1.c usrprog2.c −lrje
```

where **usrprog1.c** and **usrprog2.c** are source files containing C language programs. (See the **cc**(1) and **ld**(1) manual pages in the *UNIX System V Programmer's Reference Manual* for further information about compiling and loading programs.)

Figure 2 lists the **rje** commands and API functions available to the user.

Function	rje Command	API Function
Send Job Files	rje −s	srje( )
Cancel Jobs in Queue	rje −c	crje( )
Display Job Queue	rje −x	—
List RJE Statistics	rje −l	—
Console Command	rje −o	—

Figure 2: **rje** Commands and Functions for the User

Sending Job Files

You can send a file (or files) to a remote host by executing the send command, **rje −s**, from the UNIX system shell, or by calling the **srje( )** function from a C program.

In the first case, the send command is invoked just like any other regular task from the command line. In the second case, the C program calls the API external function **srje( )**, which executes the send command. In both cases, the list of options supplied to the send command is the same.

Send Command Options

The following options can be specified with the **rje −s** command or with **srje()** when submitting jobs:

−h *rjepipe* Specifies the pipe to the BSC/RJE Emulator+ process, **bscrje**. (See "The **−h** *rjepipe* Option" in "Using the AT&T BSC/RJE Emulator+" above.)

−x Specifies that the file(s) be sent in transparent mode. This is necessary if binary (machine readable) data are being transmitted (for example, an executable program).

−tl Used when transmitting in transparent mode (the **−x** option). The file immediately following the **−tl** option is converted to EBCDIC. This is useful when sending a combination of text and data files.

−tb Specifies tab expansion. (Valid only in 2780/3780 modes.) Any tab characters found in the text will be expanded to eight blanks. (You should not use this option if the **−x** option is also specified.)

−k *'text'* Specifies that *text* is a console command for the host. *text* must be enclosed in single quotes if the string contains UNIX system shell metacharacters (e.g., $ and *). For example:

```
rje −s −k '/*$DJ 3421'
```

(If single quotes are included in the text they must be preceded by a backslash.)

NOTE For HASP JES commands, you should not include the /* in the message. The **−k** option should always be followed by text enclosed in single quotes.

−nc For 3780 mode only. Specifies that blank compression not be performed. The default is to perform blank compression.

−sof *path* Specifies that you are sending a sign-off card file. *path* is the pathname of the file containing the sign-off card(s). This causes the RJE session with the host to end.

−son *path* Specifies that you are sending a sign-on card file. *path* is the pathname of the file containing the sign-on card(s). This option *must* be specified when signing on to a remote host as a HASP workstation.

−em Specifies that the trailing blanks in a 2780 record be truncated and replaced by a single EM character. This option provides greater throughput when the AT&T BSC/RJE Emulator+ is operating as a 2780 workstation.

2 Specifies that control not return to the user until two messages have been returned by the AT&T BSC/RJE Emulator+. This implies that control will not be returned until the file or files have been sent. (Normally the first message that is returned indicates that the command is queued; the second message indicates that the file has been sent.) If this option is not specified, control returns immediately. Note that this option is not preceded by a hyphen.

−w Same as option 2.

Invoking the Send Command from the Shell

The format to execute the send command from the shell is:

rje −s [**−h** *rjepipe*] [*option...*] *file...*

where **−h** *rjepipe* has its standard meaning. (See "The **rje** Command" in "Using the AT&T BSC/RJE Emulator+" above.) Up to fifteen filenames may be specified on one command line. Multiple files are concatenated and sent as one job stream to the host.

Simple filenames are expanded during processing to their full pathnames. For example, **rje –s myjob** would be expanded to **rje –s /usr/mylogin/myjob**. The total length of the command line, including all the expanded pathnames, cannot exceed 350 characters.

Examples

1. **rje –s file1**

   This command will cause the contents of **file1** to be transmitted.

2. **rje –s –h /usr/bscrje/HOSTX file1 file2**

   The **bscrje** process will send the contents of **file1** and **file2** using the named pipe **/usr/bscrje/HOSTX**.

3. **rje –s –x –tl jclhead bindat –tl jcltail**

   This command will transmit **jclhead, bindat** and **jcltail** in transparent mode; **jclhead** and **jcltail** will be converted to EBCDIC.

4. **rje –s –k '$DJ1234'**

   This command will submit the string **$DJ1234** to a host running JES2 as a console command.

5. **rje –s –k "\$D 'ISRWT'"**

   This command will submit the string **$D 'ISRWT'** to a host running JES2 as a console command.

Invoking the Send Command from a C Program

The send command can be executed from a C program by using the API function **srje( )** found in the library **/usr/lib/librje.a.** (This library must be linked to the user's object modules. See "The RJE Application Program Interface" in "Using the AT&T BSC/RJE Emulator+" above.)

The synopsis of **srje( )** is:

```
int srje (tok_no,ibuf,rtmsgs)
int tok_no;
char *ibuf;
int rtmsgs;
extern int srje_err;
extern int srje_jobno;
```

where:

tok_no	is the number of tokens in **ibuf**. A token is defined as a group of one or more characters terminated by blanks.
ibuf	is a string containing all the options and filenames to be processed. In other words, it is the send command line without **rje** **–s**. The string must be null-terminated.
rtmsgs	is an integer (1 or 2). 2 is the same as specifying the send command option **2**. 1 is the same as not specifying it.
srje_err	is an integer error code defined in the API library.
srje_jobno	is an integer containing the local job number of the submitted job.

Example

```
main()
{
    extern int srje_err;
    extern int srje_jobno;
    int rc;
    char ibuf[80];

    rc=srje(0,"-h /usr/bscrje/HOSTA file.a",1); /* LINE EX1 */
    if (rc==-1)
        printf("Send file.a failed, error code %d", srje_err);
    else
        printf("Send file.a queued, job number %d", rc);

    strcpy(ibuf,"-x file.b -tl file.c");
    rc=srje(0,ibuf,2); /* LINE EX2 */
    if (rc==-1)
        printf("Send file.b file.c failed, error code %d, job number %d",
                      srje_err, srje_jobno);
    else
        printf("Send file.b file.c succeeded, job number %d", srje_err);
}
```

In the preceding C program, **srje()** is invoked twice, once in LINE EX1 and again in LINE EX2. In LINE EX1, the send command is issued to the **bscrje** process using the *rjepipe* named **/usr/bscrje/HOSTA**. The **bscrje** process will send **file.a**, and control will return to the program as soon as the job is queued. In LINE EX2, two files will be sent transparently: **file.b** will not be translated to EBCDIC, while **file.c** is an ASCII file that will be translated to EBCDIC before being sent. The **−h** *rjepipe* option is not specified because the name of the *rjepipe* has been furnished using the environment variable **BR**. Control will return to the program after the files have been sent to the host or an error has occurred.

srje( ) Return Values

When **srje()** is invoked from a C program, the response is indicated by the return value of the **srje()** function call and the external variable **srje_err** as follows:

- If the call is successful
  - With **rtmsgs = 1**, the **srje()** function returns the local job number assigned to the request, and **srje_err** is undefined.
  - With **rtmsgs = 2**, **srje()** returns a value of 0, and **srje_err** contains the local job number.
- If the call is unsuccessful
  - **srje()** returns a value of −1, and **srje_err** contains the error code indicating the cause of the failure. (See Figure 3 in "Responses to the Send Command" below for a list of the possible values for **srje_err**.) For **srje()** calls with **rtmsgs = 2**, the job may be successfully queued but not transmitted. In this case, **srje()** returns −1, **srje_err** contains the error code, and **srje_jobno** contains the local job number.

NOTE You can determine the cause of an unsuccessful call by declaring the external variable **srje_err** in your program and then processing the contents of **srje_err** if the **srje()** function returns a value of −1.

Responses to the Send Command

Local Messages

If the send command is invoked from the shell, responses from the BSC/RJE Emulator+ are displayed on the screen as English language messages. If the send command is invoked from a C program, there are no English language messages, but the **srje()** return value and the external variable **srje_err** are set to indicate the response. The possible BSC/RJE Emulator+ messages and the corresponding **srje()** and **srje_err** values are as follows:

- SRJE COMMAND QUEUED as Rnnnn

  This indicates that the file has been queued internally for transmission, and its local job number is R*nnnn* where *nnnn* are decimal digits. (Note that *nnnn* is also the job number to use if the job must be cancelled.) This message is displayed only if the **2** option is not specified.

  Correspondingly, with **rtmsg = 1**, the **srje( )** function returns the job number (*nnnn*) and **srje_err** is 0.

- RDR SUCCESSFULLY SENT FILE R*nnnn*

  This indicates that the file has been transmitted to the remote host. This message is displayed only if the **2** option is specified. (The **2** option specifies that the command is to wait until the job has been sent before returning.)

  Correspondingly, with **rtmsg = 2**, the **srje( )** function returns 0 and **srje_err** is set to the job number, *nnnn*.

- ERROR: *message*

  This indicates that the file could not be transmitted for the reason given in *message*. If this message appears, the user should expect no further responses.

  Correspondingly, the **srje( )** function returns −1 and **srje_err** is set to indicate the error.

  Figure 3 lists each possible error message and the corresponding **srje_err** error code.

srje_err Code	Error Message
1	ERROR: OPTIONS NOT SPECIFIED
2	ERROR: HOST NOT FOUND
3	ERROR: HOST NOT ACTIVE
4	ERROR: HOST DOES NOT REPLY
5	ERROR: AN INVALID OPTION IS PRESENT
6	ERROR: OPTION REPEATED
7	ERROR: INVALID PARAMETER FOR OPTION
8	ERROR: REQUIRED OPTION IS MISSING
9	ERROR: OPTION IS INVALID FOR CURRENT MODE
10	ERROR: OPTION IS INVALID FOR COMMAND
11	ERROR: FILE NAMES EXCEED MAXIMUM CHARACTER LENGTH
12	ERROR: RDR FAILED TO SEND - CANNOT OPEN FILE
13	ERROR: RDR FAILED TO SEND - CANNOT READ FILE
15	SRJE COMMAND REJECTED - QUEUE OVERFLOW
16	SRJE COMMAND QUEUED AS Rnnnn
17	SUCCESSFULLY SENT Rnnnn
19	ERROR: COMMAND FAILED- CHECK LOGFILE/CONSOLE FOR REASON

Figure 3: Error Codes and Error Messages for the Send Command

If you get the message ERROR: HOST NOT FOUND (**srje_err** code 2), it could be because an invalid pathname has been specified for the **−h** *rjepipe* command option. (Check the setting of your **BR** environment variable.)

Host Messages

When the remote host receives a job, it normally responds with a message conveying the job number that has been assigned to the user's job. This message is displayed at the BSC/RJE system console (see the "The Console Operator Command" below) and written to the log file specified by the System Administrator at system start-up time.

Receiving Job Files

Job output and other files coming from the remote host are initially written to the output directory for the **bscrje** process, normally **/usr/bscrje/output**.

The system uses the following naming convention to generate a unique default name for every file received into the **output** directory:

- Files received on a printer are named P*MMDD.999*, where;
  - P indicates a printer file
  - *MM* represents the month of the year
  - *DD* represents the day of the month
  - *999* is a sequence number
- Files received on a punch are named N*MMDD.999*, where;
  - N indicates a punch file
  - *MM* represents the month of the year
  - *DD* represents the day of the month
  - *999* is a sequence number

Automatic Routing of Job Output

If the dispatch option is specified by the System Administrator when the emulator is started every file placed in the output directory is examined for a PATH comment card. When a PATH comment card is found the output file is automatically routed to the specified directory or file. The format of the PATH comment card is:

```
//*#(PATH=pathname)
```

The output file is moved to the directory or file specified by *pathname*. If *pathname* specifies a directory, the output will be placed in that directory with the system-generated filename. If a filename is specified, the output will be moved and renamed accordingly. The moved file will have the same owner as the directory it is placed in and will not be writable by 'other.' The following restrictions apply to the specification of *pathname*:

- It must be a full pathname.
- The length cannot exceed 31 characters (including slashes).
- If *pathname* specifies a directory, that directory must be writable by 'other.'
- If the specified file already exists and is writable by 'other' that file will be overwritten.

It is recommended that all output files be routed this way to make user identification of output files easier.

Every file received from the host by the emulator is placed in the output directory. If the dispatcher is active it will check the first 15,000 characters of every file placed in the output directory for a PATH statement. The PATH statement can be placed in any file. If it is placed in the JCL after the job card, all standard printer output associated with that job will be routed to the specified destination. If it is placed in a data file directed to standard punch, that punch file will be moved to the desired directory or file.

The following examples use the standard IBM Utility IEBGENER. See the IBM Utilities manual for additional details on the use of IEBGENER. In these examples SYSOUT=A specifies that the output will be sent to a printer and SYSOUT=B specifies that it will be sent to a card punch. Your host may use other classes; check with your System Administrator if you are not sure what the SYSOUT classes are for the host you are using.

In the following example the data in FILE1 is being sent to the printer (SYSOUT=A). Since the JCL and job-related information are also sent to the printer, all the output from this job will be routed to one virtual printer file. After the file is placed in the output directory the dispatcher will move it to **/usr/***xxx***/jclout**, the file specified in the PATH statement.

```
//.. JOB ...
//*#(PATH=/usr/xxx/jclout)
//STEP    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=FILE1,DISP=SHR
//
```

In the following example the data in FILE1 is sent to a card punch (SYSOUT=B) and the JCL and job related-information is sent to the printer. Therefore the output from this job will be sent to two virtual files. The virtual printer file is moved to **/usr/*xxx*/jclout** based on the first PATH statement that is part of the JCL file. The virtual punch file is moved to **/usr/*xxx*/punout** based on the second PATH statement that is part of the punch file received from the host. (See an IBM JCL manual for a description of concatenated DD statements.)

```
//.. JOB ...
//*#(PATH=/usr/xxx/jclout)
//STEP     EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD SYSOUT=B
//SYSIN    DD DUMMY
//SYSUT1   DD DATA
//*#(PATH=/usr/xxx/punout)
/*
//         DD DSN=FILE1,DISP=SHR
//
```

Note that the PATH statement will remain in the file **/usr/*xxx*/punout**. The user could delete it with the following statement:

fgrep -v '//*#(PATH=' /usr/*xxx*/punout > /usr/*xxx*/punout2

When routing output files, note that:

- When the dispatcher is running it maintains a log file (**DISPTCH.LOG**) in the RJE home directory. This file contains a record of all automatic routing requests and their status.
- If the PATH statement is used in the JCL, the MSGLEVEL on the JCL JOB statement must be non-zero. When the MSGLEVEL is set to zero the JCL is not printed and the PATH statement would not be part of the file placed in the output directory. The message in DISPTCH.LOG would indicate that automatic routing had not been requested.
- When the PATH statement is part of the JCL some host systems will translate the *pathname* on the PATH statement to upper case. In this occurs the user must either create an upper case directory structure or ask the System Administrator to set the JES2 initialization parameter &PRTRANS to "no" so translation to upper case does not take place.

- Routing of job output is possible only if your System Administrator has started the emulator process with the **−dy** option, which invokes the dispatcher process. (See the *System Administrator's Guide* for details.)

User Notification of Job Complete

If the dispatch option is specified by the System Administrator when the emulator is started, users can receive a notification via UNIX mail of when a job is completed by having a USER field on a comment card in the JCL of the jobfile. For example,

```
//*#(USER=xxx)
```

where *xxx* is the UNIX logon id of the user who is to receive the notification. If both the PATH and USER options are to be used, they must both be present on the same comment card. For example,

```
//*#(PATH=/usr/sample/output,USER=logon)
```

When routing output files, note that:

- If the USER statement is used in JCL, the MSGLEVEL on the JCL JOB statement must be non-zero. When the MSGLEVEL is set to zero, the JCL is not printed and the USER statement would not be part of the file placed in the output directory.
- When the USER statement is part of the JCL, some host systems will translate the logon name on the USER statement to upper case. In this situation the user must either have an upper case logon name or have the System Administrator arrange to have the JES2 initialization parameter PRTRANS set to "no" so translation to upper case does not take place.
- User notification of job completion is possible only if your System Administrator has started the emulator process with the **−dy** option, which invokes the dispatcher process. (See the *System Administrator's Guide* for details.)

Tag Files

When your System Administrator starts the BSC/RJE Emulator+ with the **−tg** option, a tag file (identified by the .TAG suffix) will be created for every file received. (See question 10 in "What the **startrje** Questions Mean" in the *System Administrator's Guide.*) The tag file contains the following information about its associated file:

TIME RECEIVED:	MM DD YY
RECEIVED ON:	PRT or PUN
TYPE:	TRANSPARENT or NON-TRANSPARENT

The tag file is moved into the appropriate destination directory if automatic routing is in effect and an automatic routing jobcard was included in the original jobfile sent to the host. The default destination is the **output** directory of the **bscrje** process (normally **/usr/bscrje/output**).

Cancelling Jobs

You can cancel a job queued locally for transmission by executing the cancel command, **rje −c**, from the UNIX system shell, or by calling the **crje( )** function from a C program. Unless you have logged in as root, you may only cancel jobs you have sent.

A job can be cancelled only if it is waiting on the queue. If it is currently being transmitted, or if it has already been sent, the cancel command will fail and an appropriate message will be returned.

Invoking the Cancel Command from the Shell

The format to execute the cancel command from the shell is:

rje −c [−h *rjepipe*] *jobno*

where **−h** *rjepipe* has its standard meaning. (See "The **rje** Command" in "Using the AT&T BSC/RJE Emulator+" above.) *jobno* specifies the number of the internally queued file and is returned to the user by the send command (see "Responses to the Send Command" above). The *jobno* consists of four digits preceded by R.

Example

- **rje −c R1234**

  This cancels job R1234.

Invoking the Cancel Command from a C Program

The cancel command can be executed from a C program by using the API function **crje( )** found in the library **/usr/lib/librje.a.** (This library must be linked to the user's object modules. See "The RJE Application Program Interface" in "Using the AT&T BSC/RJE Emulator+" above.)

The synopsis of **crje( )** is:

```
int crje (rjepipe, jobno)
char *host;
int jobno;
extern int srje_err;
```

where:

`rjepipe` is a character pointer to a buffer containing the rje pipe name for the **bscrje** process.

`jobno` is an integer that was returned from the **srje()** function.

`srje_err` is an integer error code defined in the API library.

Example

```
main()
{
    extern int srje_err;
    int jobno, rc;
    char *rjepipe="/usr/bscrje/HOSTA";

    jobno=srje(0,"file.a",1);
    if (jobno==-1)
        printf("Send file.a failed, error code %d", srje_err);
    else
    {
        printf("Send file.a succeeded, job number %d", jobno);
        rc=crje(rjepipe, jobno); /* LINE EX1 */
        if (rc==-1)
            printf("Cancel file.a failed, error code %d", srje_err);
        else
            printf("Cancel file.a succeeded.");
    }

    jobno=srje(0,"file.b",2); /* LINE EX2 */
    if (jobno==-1)
        printf("Send file.b failed, error code %d", srje_err);
    else
    {
        rc=crje(rjepipe, jobno); /* LINE EX3 */
        if (rc==0)
            printf("Crje should have failed");
        else
            printf("Cancel file.b failed as expected");
    }
}
```

This example demonstrates the connection between **srje()** and **crje()**. On successful completion **srje()** returns the job number, which can then be used as an argument to **crje()** if the file is not sent immediately. This use of the job number is illustrated in LINE EX1. The invocation of **srje()** in LINE EX2 specifies that it should wait for two return messages, which ensures that

srje() will return after the file has been sent. Therefore, the invocation of **crje()** in LINE EX3 will fail, because the job will not be on the queue when **crje()** is invoked.

crje( ) Return Values

When **crje()** is invoked from a C program, the response is indicated by the return value of the **crje()** function call and the external variable **srje_err** as follows:

- If the call is successful
  - **crje()** returns a value of 0, and **srje_err** is undefined.
- If the call is unsuccessful
  - **crje()** returns a value of −1, and **srje_err** contains the error code indicating the cause of the failure. (See Figure 4 in "Responses to the Cancel Command" below for a list of possible values for **crje_err.)**

NOTE

You can determine the cause of an unsuccessful call by declaring the external variable **srje_err** in your program and, if **crje()** returns a value of −1, processing the contents of **srje_err**.

Responses to the Cancel Command

Local Messages

If the cancel command is invoked from the shell, responses from the BSC/RJE Emulator+ are displayed on the screen as English language messages. If the cancel command is invoked from a C program, there are no English language messages, but the **crje()** return value and the external variable **srje_err** indicate the response. (Note that **srje_err** is used for both the **srje()** and **crje()** functions.)

The possible BSC/RJE Emulator+ messages and the corresponding **crje()** and **srje_err** values are as follows:

- SUCCESSFULLY CANCELLED Rnnnn

  This indicates that the job has been successfully cancelled from the queue and will not be sent to the host.

  Correspondingly, the **crje( )** function returns 0.

- ERROR: *message*

  This indicates that the cancel command could not be completed for the reason given in *message.*

  Correspondingly, the **crje( )** function returns −1, and **srje_err** is set to indicate the error. Figure 4 lists each possible error message and the corresponding **srje_err** error code.

srje_err Code	Error Message
1	ERROR: OPTIONS NOT SPECIFIED
2	ERROR: HOST NOT FOUND
5	ERROR: AN INVALID OPTION IS PRESENT
6	ERROR: OPTION REPEATED
8	ERROR: REQUIRED OPTION IS MISSING
10	ERROR: OPTION IS INVALID FOR COMMAND
14	CRJE COMMAND REJECTED - FILE NOT QUEUED

Figure 4: Error Codes and Error Messages for the Cancel Command

If you get the message ERROR: HOST NOT FOUND (**srje_err** code 2), it could be because an invalid pathname has been specified for the **−h** *rjepipe* command option. (Check the setting of your **BR** environment variable.)

Displaying the Local Job Queue

To see if a job is on the local job queue waiting to be sent to the host, enter:

```
rje −x [−h rjepipe] [−e]
```

where **−h** *rjepipe* has its standard meaning. (See "The **rje** Command" in "Using the AT&T BSC/RJE Emulator+" above.)

This command will display information about all the jobs on the queue. (If the **−e** option is omitted, only the jobs that you have submitted are displayed.) For information about cancelling jobs on the queue, see "Cancelling Jobs" above.

The BSC/RJE Emulator+ Statistics Command

To display a list of the current statistics maintained by the **bscrje** processes, enter:

rje −l [**−h** *rjepipe*]

where **−h** *rjepipe* has its standard meaning. (See "The **rje** Command" in "Using the AT&T BSC/RJE Emulator+" above.)

The Console Operator Command

The console operator command, **rje −o**, is used to attach or detach the terminal that the command is issued from as the BSC/RJE Emulator+ system console. Only one terminal can be assigned as the BSC/RJE Emulator+ console for each **bscrje** process.

In addition to local BSC/RJE Emulator+ messages, all console operator messages sent from the remote host will be displayed at the terminal designated as the console. Console operator messages are also written to the console log file specified by the System Administrator at start-up time. Thus, by examining the tail of the log file one can always get an up-to-date status.

The format of the console operator command is:

```
rje −o [−h rjepipe] [−st]
```

where **−h** *rjepipe* has its standard meaning. (See "The **rje** Command" in "Using the AT&T BSC/RJE Emulator+" above.) The other option is:

−st specifies that the console should be detached from the terminal from which the command is issued.

Examples

1. To designate your terminal as the console, enter

   ```
   rje −o
   ```

   After the command has been executed, the following message will be displayed:

   ```
   CONSOLE CHANGED TO /dev/tty## by USER userid
   ```

   where **/dev/tty** is the device id of your terminal, and **userid** is your login.

2. To detach your terminal as the system console, enter

   ```
   rje −o −st
   ```

Table of Contents

Running the AT&T BSC/RJE Emulator+

Introduction

NOTE

The runtime directory that is referred to in this document is the directory that contains the executable files for the AT&T BSC/RJE Emulator+. This is the directory from which the Emulator+ must be run. The default name for the directory is **/usr/bscrje**, but the System Administrator may choose any name for it.

Before starting the AT&T BSC/RJE Emulator+, you must:

Step 1. Log in as root.

Step 2. Change directory to the runtime directory (normally **/usr/bscrje**).

Step 3. Set and export the **PCD** environment variable.

- On the 3B2 Computer, it should be set to the ISC slot number to be used, for example:

```
PCD=2
export PCD
```

- On the 3B5/15 Computer, it should be set to the IOA device number to be used, for example:

```
PCD=201
export PCD
```

- On the 3B4000 Computer, it should be set to a combination of the pe number and the ISC slot number. For example, if installing on pe120, slot 6:

```
PCD=120.6
export PCD
```

Step 4. Execute **startrje** to create a command file. It is the execution of this command file that actually starts the AT&T BSC/RJE Emulator+.

System Start-up

startrje creates a command file containing the **bscrje** start-up command and the code to download **lmrje** to the communications processor. The **startrje** script generates the **bscrje** command with appropriate options based on the user's responses to a series of questions. When you use the **startrje** script, you can either save the command file for execution at a later time or execute it immediately.

As the System Administrator, you may designate one terminal as the console terminal for the BSC/RJE Emulator+. The shell that runs on that terminal must have super-user privileges to start the **bscrje** process. All status and error messages from the **bscrje** process will be displayed on this terminal.

NOTE

See Appendix A for a complete list of the **bscrje** command options.

The startrje Shell Script

startrje is a shell script that helps the System Administrator who is new to BSC/RJE 2780/3780/HASP emulation to start up the AT&T BSC/RJE Emulator+. **startrje** walks the System Administrator through the various AT&T BSC/RJE Emulator+ starting options, skipping those options that have been made inapplicable by answers to previous questions. Afterwards, the **bscrje** start-up command is formatted according to the user's selections. This command, along with the code to download the communications board, is written to a file for execution. It is the execution of this file that starts the AT&T BSC/RJE Emulator+. **startrje** allows the System Administrator either to save this command file for execution later, or to execute the command file immediately. In the latter case the command file is not saved.

Since the AT&T BSC/RJE Emulator+ may be started in many different ways, there are several mutually exclusive start-up options. For example, if a 2780 workstation is specified, the **−p** option may *not* be used, since it is only applicable to a HASP workstation. **startrje** will prevent such conflicting specifications by skipping questions that do not apply. For example, once a 2780 workstation has been chosen, **startrje** will not ask questions about the **−p** option.

Responding to the startrje Questions

Following most of the questions that **startrje** asks is a list of possible responses. For example, after the question **emulator**? the list **(2780), 3780, HASP** appears. This tells you that 2780, 3780, and HASP are all valid responses. Any one of these options may be chosen by typing **2780**, **3780**, or **HASP**. Alternatively, you may press RETURN to select the default value, which is enclosed in parentheses. In this example pressing RETURN would select a 2780 workstation. You can also select one of the options that follow a **startrje** question by typing only as many characters as are needed to identify the beginning of the option name. In this example, typing 3 would select a 3780 workstation since only the 3780 option begins with a 3.

Some questions, such as those that ask for a file name, are not followed by a list of possible responses. You can skip these questions by pressing RETURN.

What the startrje Questions Mean

Since some questions are mutually exclusive, you will not see each question every time you run **startrje**. What follows is a list of all the possible questions asked by **startrje** and what they mean:

1. **Do you want instructions? y or n**

   If you answer y, a short description of **startrje** and simple instructions follow.

2. **Do you want to use this script? (n) or y**

   If you answer **n**, you will exit from **startrje**. Enter y to continue.

3. **Name of the rje pipe? (default is $sbsc/HOSTA)**

   $sbsc represents the home directory of the **bscrje** login. The **rje** user interface process communicates with the **bscrje** controller process through a special file called an rje pipe. This pipe is created when **bscrje** is invoked. Unless you provide a different path name, the default rje pipe name will be determined when the **bscrje** command file is executed as follows:

   - The UNIX system password file is searched to obtain the **bscrje** login directory name (normally **/usr/bscrje**).
   - Then the file is created in the **bscrje** login directory and named HOSTA.

4. **Emulator? (2780), 3780, HASP**

   Which IBM workstation, 2780, 3780, or HASP, is the AT&T BSC/RJE Emulator+ to emulate.

5. **Line control characters are to be in? (EBCDIC), ASCII**

   The AT&T BSC/RJE Emulator+ can understand EBCDIC and ASCII line control character sets. Note that the character set you choose should coincide with the line definition in the IBM communication controller that the AT&T BSC/RJE Emulator+ is to be connected to. This question sets the **−as** option.

6. **Is the workstation to have a console? (n), y**

   If the AT&T BSC/RJE Emulator+ is to have a console to which console messages will be sent, you should answer y. The 3B Computer console terminal is initially set to be the AT&T BSC/RJE Emulator+ console. You can make another terminal the console terminal after you have started the AT&T BSC/RJE Emulator+ by using the **rje −o** command. (See "The Console Operator Command" in the *User's Guide.*) This question sets the **−c** option.

7. **Pathname of the console output file--no quotes are necessary?**

   When the AT&T BSC/RJE Emulator+ is running, the terminal that receives the console information may be changed. To keep a record of all the console information, a console log file may be created in the runtime directory. This question asks you for a name for that file. When the AT&T BSC/RJE Emulator+ is started the log file will be overwritten if it already exists. If you do not want to have a log file, press RETURN. This question sets the –l *path* option.

8. **Should a signon card be sent after system initialization: (n), y**

   (You may want to use this option if your host requires a sign-on card.) Once the BSC/RJE Emulator+ has been started and the link to the host established, the host may require a sign-on card. If you answer **y**, a sign-on card is sent automatically. If you prefer to send the sign-on card manually, answer **n** and send the sign-on card using the **rje –s** send command before you send other files to the host. (See "Sending Job Files" in the *User's Guide.*)

   □ **Name of file containing the signon card--no quotes needed?**

   Enter the file name or pathname of the file containing the sign-on card that is to be sent to the host. Note that **startrje** checks to see if this sign-on file exists. If no file name is entered, the user will have to send a sign-on card manually. This question sets the **–son** *path* option.

9. **Should automatic routing be performed? (n), y**

   If this option is chosen, any job sent to the host that contains a line of the form **//*#(PATH=filename)** will have its output copied to *filename* after it arrives in the output directory of the **bscrje** process, normally **/usr/bscrje/output**. Also, the **DISPTCH.LOG** file will be created in the **/usr/bscrje** directory and will contain a record indicating where the file was routed. Jobs that do not include a line of the form **//*#(PATH=filename)** will remain in **/usr/bscrje/output** after they are received.

10. **Should .TAG files be created? (n), y**

    When this option is chosen, for every **file** received from the host a corresponding **file.TAG** will be created. **file.TAG** will contain information about the time **file** was received, what device it was received on, and whether the data were sent transparently. This question sets the **–tg** option.

11. **Should the emulator dynamically route to the printer? (n), y**

    If chosen, this option will cause the AT&T BSC/RJE Emulator+ to route any files it receives to a physical printer (as opposed to the virtual printer of the AT&T BSC/RJE Emulator+). This question sets the **−dp** option.

12. **Should the emulator mode be transparent for receiving files? (n), y**

    When the emulator is in this mode, the ability to translate data from EBCDIC to ASCII is disabled. This question sets the **−xp** option.

13. **How many active readers? (1), 2, 3, 4, 5, 6, 7**

    This question only applies to HASP workstations. Specify the number of active readers that the HASP workstation should maintain. This question sets the **−r** *n* option.

14. **How many active punches? (1), 2, 3, 4, 5, 6, 7**

    This question only applies to HASP workstations. Specify the number of active punches that the HASP workstation should maintain. This question sets the **−n** *n* option.

15. **How many active printers? (1), 2, 3, 4, 5, 6, 7**

    This question only applies to HASP workstations. Specify the number of active printers that the HASP workstation should maintain. This question sets the **−p** *n* option.

16. **Should the emulator ignore Block Sequence Numbers? (n), y**

    This question only applies to HASP workstations. Some hosts occasionally send out data blocks with incorrect Block Check Bytes (BCB's) when running in HASP mode. In some cases such bad BCB's can be ignored without ill effects. THIS OPTION SHOULD BE USED WITH CAUTION. This question sets the **−pv** option.

17. **This system is to be what kind of a station in point to point operation mode? (primary), secondary**

    When the workstation is in point to point communication with another workstation, one workstation must be set up as the primary workstation and the other as the secondary workstation. When the workstation is communicating with the host, the host is configured as a secondary workstation so this workstation should be configured as a primary workstation. This question sets the **−sc** option.

18. **Terminal identification (1-32 characters long)?**

    For a 2780 or 3780 workstation, this option defines a terminal identification string of 1 to 32 characters. If you do not type any characters, the option will be skipped. This question sets the **−id** *name* option.

19. **Should the multirecord feature be disabled? (n), y**

    If you answer y, the multirecord feature for 2780 and 3780 transmission blocks will be disabled. This question sets the **−nm** option.

20. **Should the emulator have the self halting feature? (n), y**

    This question applies only to 2780/3780 workstations. If you answer y, the emulator will automatically terminate itself if there has been no meaningful exchange of data for some period of time. This question sets the **−hs** *n,t* option.

    □ **Specify the number of idle ENQs (between 1 and 255, inclusive)?**

    This allows you to specify the number of idle ENQs the emulator will initiate or accept from the host before it terminates.

    □ **Specify the time interval (between 1 and 255, inclusive)?**

    This allows you to specify in seconds the time the emulator will wait before sending out each ENQ.

21. **Should the emulator have extended timeout feature? (n), y**

    This question allows you to specify the **−to** option. The **−to** option specifies how many seconds the emulator is to wait after acknowledging a block from the host before timing out, and it is valid only for 2780/3780 modes. If you answer y, you will be asked to

    □ Specify the timeout in seconds

    Enter a value from 1 to 25.

22. **Should the emulator be configured to accommodate DOS/POWER 2.2.0? (n), y**

    DOS/POWER version 2.2.0 requires a special Inter-Record Separator (IRS) character in its data blocks. The usual IRS character hex 1E must be changed to hex 15. If you answer **y**, the emulator selects the IRS character hex 15. This question sets the **−nl** option. It applies only to 3780 workstations. This question sets the **−nl** option.

23. Enter a file name in which the system start-up command generated by this interactive session is to be stored. Then to start the AT&T BSC/RJE Emulator+ type the name of that file. If no file name is entered the Emulator+ will be started immediately and the command will **not** be saved. Remember, to start the BSC/RJE Emulator+ successfully you must be the super-user or root.

    If you enter a file name, the command you have created will be stored in that file. If you press RETURN, the BSC/RJE Emulator+ will be started with the options you have specified.

NOTE

See **Appendix C** for sample AT&T BSC/RJE Emulator+ configurations for 2780, 3780, and HASP workstations and **Appendix D** for sample IBM Host (JES and POWER) configurations for 2780, 3780, and HASP workstations.

Using the AT&T BSC/RJE Emulator+

After you have successfully started the AT&T BSC/RJE Emulator+, you should receive a message that the RJE Controller has been initialized, and the AT&T BSC/RJE Emulator+ banner page will be displayed. You must then press RETURN to display your system prompt. If connection to the host is via a dial-up line, you should dial in at this point.

NOTE

On the 3B2 Computer, if the RJE CONTROLLER INITIALIZED message does not appear, it is possible that the ISC board slot number, which is used in the **startrje** script, needs to be changed.

When the AT&T BSC/RJE Emulator+ is first started, the message

```
read on SRJEQ failed
```

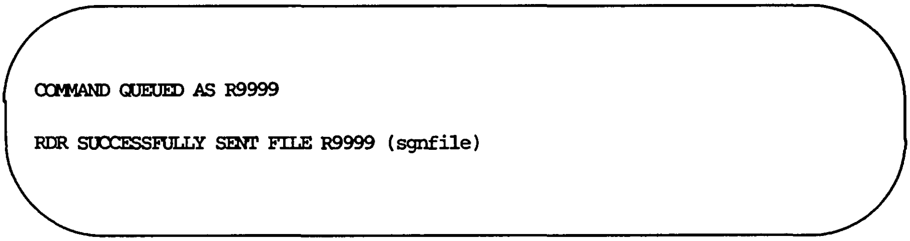
may be displayed. This does not indicate an error. The **SRJEQ** file is created when the first job file is queued for transmission; it holds the names of all the job files currently in queue. After the controller has been initialized, the AT&T BSC/RJE Emulator+ automatically reads **SRJEQ** to start job processing. But when the AT&T BSC/RJE Emulator+ is first brought up, there is nothing in the queue; therefore the read on **SRJEQ** fails.

Signing On to a Host

When you run the **startrje** script, you are given the option of having a sign-on card sent automatically when the AT&T BSC/RJE Emulator+ is started. If you do not choose this option, and a sign-on card is required, you may send it manually using the **rje –s** send command after the AT&T BSC/RJE Emulator+ has been started.

Automatic Sign-on

If you decide to have a sign-on card sent automatically, it will be sent after the message CONTROLLER INITIALIZED is displayed. After the sign-on card has been sent, the following messages are displayed:



```
COMMAND QUEUED AS R9999

RDR SUCCESSFULLY SENT FILE R9999 (sgnfile)
```

where **sgnfile** is a file that contains the proper sign-on card for the remote host, for example

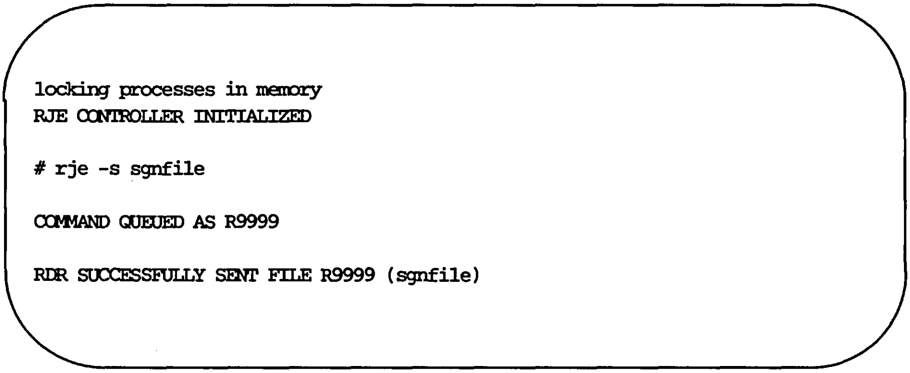
```
/*SIGNON    RMT60
```

where **RMT60** starts in column 16.

This automatic sign-on option is most helpful if the AT&T BSC/RJE Emulator+ must be restarted (for example, after a system crash) with jobs left in the queue. When the AT&T BSC/RJE Emulator+ is restarted, the sign-on card is automatically sent as the first job, and the remaining jobs in the queue are processed as they would have been before the AT&T BSC/RJE Emulator+ was restarted.

Manual Sign-on

If you decide to send a sign-on card manually, you must clear the queue of all jobs by removing the **SRJEQ** file in the AT&T BSC/RJE Emulator+ runtime directory (normally **/usr/bscrje**). This file can be removed before the AT&T BSC/RJE Emulator+ is started. If the queue has not been cleared before the sign-on card is sent, the sign-on card job file will be placed at the bottom of the queue and, consequently, will not be processed as the first job. (Note that any jobs that were previously in the queue must be resubmitted after signing on to the host.) Once you have cleared the queue (i.e., **SRJEQ** has been removed) and started the AT&T BSC/RJE Emulator+, and after the message `CONTROLLER INITIALIZED` has been displayed, you should send the sign-on card using the **rje –s** send command. Messages similar to those above will be displayed. For example:

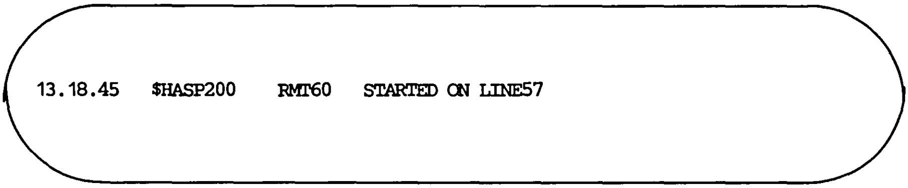


```
locking processes in memory
RJE CONTROLLER INITIALIZED

# rje -s sgnfile

COMMAND QUEUED AS R9999

RDR SUCCESSFULLY SENT FILE R9999 (sgnfile)
```

NOTE

When the AT&T BSC/RJE Emulator+ is started in HASP mode, the **−son** option must be specified with the **rje −s** command when sending a sign-on card. In 2780 or 3780 mode, the **−son** option is not required.

When the sign-on card has been successfully received, the host responds with a message indicating that the remote has been started. For example:



```
13.18.45   $HASP200   RMT60   STARTED ON LINE57
```

The user may now send files to and receive files from the remote host.

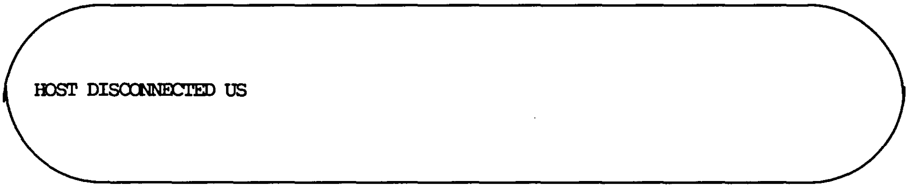
Signing Off from the Host

To sign off from the remote host, a sign-off card should be sent using **rje −s** with the **−sof** option specified. For example,

rje −s −sof *sgnoff*

where *sgnoff* is the pathname of a file containing the sign-off card.

When sign-off from the host is complete, the following message should be displayed at the system console:



```
HOST DISCONNECTED US
```

Running Multiple BSC/RJE Emulators

To run multiple RJE's, perform the following steps:

Step 1. Execute the **clonerje** script in the runtime directory of the BSC/RJE Emulator+. The script will ask you for the full path name of the new runtime directory.

Step 2. Set **BR** to the default pipe name for the **clonerje** script by entering:

BR="*full path name of new runtime directory*"/HOSTA

Step 3. Change directory to the new runtime directory.

Step 4. Set and export the PCD environment variable.

- On the 3B2 Computer, it should be set to the ISC slot number to be used by the new emulator, for example:

```
PCD=2
export PCD
```

- On the 3B5/15 Computer, it should be set to the IOA device number to be used by the new emulator, for example:

```
PCD=201
export PCD
```

- On the 3B4000 Computer, it should be set to a combination of the pe number and the ISC slot number. For example, if installing on pe120, slot 6:

```
PCD=120.6
export PCD
```

Step 5. Follow the standard start-up procedure, i.e. execute **startrje** to create a **bscrje** command file, then execute the command file to start another AT&T BSC/RJE Emulator+.

Step 6. Advise users which *rjepipe* they should use.

System Shutdown

Like any other process running under the UNIX operating system, the BSC/RJE Emulator+ **bscrje** process may be killed explicitly from the shell. However, it is better to invoke the **stoprje** shell script from the runtime directory of the **bscrje** process (usually **/usr/bscrje**).

The following steps will stop the AT&T BSC/RJE Emulator+:

Step 1. Log in as root.

Step 2. Change directory to the runtime directory of the BSC/RJE Emulator+ **bscrje** process (normally **/usr/bscrje**).

Step 3. Set and export the **PCD** environment variable.

- On the 3B2 Computer, it should be set to the ISC slot number to be used, for example:

  PCD=2
  export PCD

- On the 3B5/15 Computer, it should be set to the IOA device number to be used, for example:

  PCD=201
  export PCD

- On the 3B4000 Computer, it should be set to a combination of the pe number and the ISC slot number. For example, if installing on pe120, slot 6:

  PCD=120.6
  export PCD

Step 4. Stop the AT&T BSC/RJE Emulator+ by executing the **stoprje** shell script.

NOTE

If more than one **bscrje** process is running, be sure that you are in the correct directory and have the **PCD** environment variable properly set.

Appendices

bscrje Command Options

bscrje *rjepipe mode port* [**−hx**] [**−as**] [**−pr**] [**−c**] [**−l** *path*] [**−son** *path*] [**−dy**]
[**−tg**] [**−dp**] [**−xp**] [**−pv**] [**−sc**] [**−r** *n*] [**−n** *n*] [**−p** *n*] [**−id** *name*]
[**−nm**] [**−hs** *n,t*] [**−to** *n*] [**−nl**]

The first three arguments are positional, that is, they should always be specified in the above order.

rjepipe Specifies the named pipe that uniquely identifies the remote host. Must be the first argument to the command.

mode Specifies the mode (2780/3780/HASP) of the current emulation session. Must be the second argument to the command.

Example: **3780**

port Specifies the name of the communication port that is being used. Must be the third argument to the command.

Example: **/dev/isc00**

−hx Specifies that the communication line is in half-duplex operation (such as a dial-up line). The default is the full-duplex line.

−as Specifies that the line control characters are in ASCII code. The default is EBCDIC line control characters.

−pr Specifies even parity for ASCII line. The default is odd parity for ASCII line and no parity for EBCDIC.

−c Specifies the 2780/3780 console. Will cause all single block messages sent by the host to be displayed on the console.

−l *path* Specifies the pathname *path* of the file that is to receive all system console output. All console messages received during the emulation session will be written to this read-only disk file. This file actually records most of the activity during the session.

Example: **−1 LOGFILE**

−son *path* Will cause signon card(s) to be sent automatically after initialization. This option must be followed by the pathname *path* of the file containing the signon card(s) to be sent. The signon file will be sent first, even if there are old jobs waiting in the queue.

Example: **−son SGNR111**

−dy Will implement the automatic routing feature for receiving job files.

−tg Will cause **.TAG** files to be created for every file sent by the host.

−dp Will cause the AT&T BSC/RJE Emulator+ to route a received file to a physical printer automatically.

−xp Specifies that the emulator should operate in transparent mode. The ability to translate data from EBCDIC to ASCII is disabled.

−pv Specifies that the emulator should not perform any Block Check Byte (BCB) sequence checks. Data blocks with invalid BCB's will be accepted. THIS OPTION SHOULD BE USED WITH CAUTION.

−sc Specifies that the system is a secondary station in point to point operation. The default is primary.

−r *n* Specifies the number of active readers that the session should maintain. This option must be followed by an integer value *n* that is within the range 1 to 7. The default is 1 reader. This option is only valid for HASP mode.

Example: **−r 4**

−n *n* Specifies the number of active punches that the session should maintain. This option must be followed by an integer value *n* that is within the range 1 to 7. The default is 1 punch. This option is only valid for HASP mode.

Example: **−n 2**

−p *n* Specifies the number of active printers that the session maintains. This option must be followed by an integer value *n* that is within the range 1 to 7. The default is 1 printer. This option is only valid for HASP mode.

Example: **−p 5**

−id *name* Specifies the 1-5 character terminal identification string *name.* This option is valid for 2780/3780 modes only.

Example: **−id TY732**

−nm Specifies that the multirecord feature for 2780/3780 transmission blocks should be disabled. The default is multirecord transmission; the number of records transmitted per block is defined in the IBM 2780/3780 manuals. This option is valid only for 2780/3780 emulation modes.

−hs *n,t* Specifies that the emulator should terminate itself if there has not been any data other than idle ENQ's for some period of time. (If this occurs, the **bscrje** process exits with a return code of 255.) The first parameter, *n*, specifies the number of idle ENQ's that the emulator will initiate (if the emulator is sending out idle ENQ's) or accept (if it is the host that is sending out idle ENQ's) before it terminates itself. The second parameter, *t*, sets the idle period timer for the emulator, that is, it determines how much time should elapse between ENQ's. This option is valid only for 2780/3780 emulation modes.

−to *n* Specifies how many seconds the emulator will wait after acknowledging a block from the host before timing out. If not specified, the default is 3 seconds. Valid values are 1 to 25. This option is valid only for 2780/3780 emulation modes.

−nl DOS POWER version 2.2.0 requires a special Inter-Record Separator (IRS) character in its data blocks. The normal IRS character, hex 1E, must be changed to hex 15. This option specifies that the IRS character be hex 15.

AT&T BSC/RJE Emulator+ Files

The files listed below are installed on the 3B Computer during the installation procedure and removed from the 3B Computer during the removal procedure. The files are installed in the home directory of the **bscrje** login. The default name of the **bscrje** login directory is **/usr/bscrje**; however, the System Administrator may choose any name for this directory.

File Name	Description
/usr/bscrje/bscrje	the **bscrje** controller program
/usr/bscrje/clonerje	the shell script to clone another **bscrje** process
/usr/bscrje/disptch	the received file postprocessor
/usr/bscrje/lmrje	the **bscrje** line monitor program
/usr/bscrje/rje	the **bscrje** user interface program
/usr/bscrje/sgn025	a sample IBM signon
/usr/bscrje/sgnoff	a sample IBM signoff
/usr/bscrje/startrje	the shell script to start the **bscrje** process
/usr/bscrje/stoprje	the shell script to stop the **bscrje** process
/usr/lib/librje.a	the **bscrje** user interface object library
/usr/options/bscrje.name	the name of the AT&T BSC/RJE Emulator+

NOTE

Additional hardware-specific files for the communications board are also installed.

Sample AT&T BSC/RJE Emulator+ Configurations

3B Computer 2780 Emulation Configuration

This configuration represents a sample 3B Computer configuration of the 2780 workstation when the bi-synchronous communication emulation is started for remote job entry.

Summary: This configuration is compatible with the 2780 JES configuration in Appendix D. The line control is EBCDIC (by default) and half-duplex. Multi-record transmission is expected (by default), pathname is specified for console logging, and automatic signon is set. The automatic routing/execution feature is enabled and files sent by the host will cause .TAG files to be created.

Detail:

rjepipe – HOSTA
mode – 2780
port – /dev/iscl0
–hx ---> half-duplex operation
–sc ---> secondary station
–c---> specifies system console
–l LOGFILE --> console logfile
–son SIGNON --> file where signon resides
–dy ----> automatic routing/execution
–tg ----> .TAG files are to be created

3B Computer 3780 Emulation Configuration

This configuration represents a sample 3B Computer configuration of the 3780 workstation when the bi-synchronous communication emulation is started for remote job entry.

Summary: This configuration is compatible with the 3780 JES configuration in Appendix D. The line control is EBCDIC (by default) and half-duplex. Multi-record transmission is expected (by default), pathname is specified for console logging, and automatic signon is set. The automatic routing/execution feature is enabled and files sent by the host will cause .TAG files to be created.

Detail:

rjepipe – HOSTA
mode – 3780
port – /dev/iscl0
–hx ---> half-duplex operation
–sc ---> secondary station
–c---> specifies system console
–l LOGFILE --> console logfile
–son SIGNON --> file where signon resides
–dy ----> automatic routing/execution
–tg ----> .TAG files are to be created

3B Computer HASP Emulation Configuration

This configuration represents a sample 3B Computer configuration of the HASP workstation when the bi-synchronous communication emulation is started for remote job entry.

Summary: This configuration is compatible with the HASP JES configuration in Appendix D. The line control is EBCDIC (by default) and half-duplex. Multi-record transmission is expected (by default), pathname is specified for console logging, and automatic signon is set. The automatic routing/execution feature is enabled and files sent by the host will cause .TAG files to be created. Two active readers and printers are specified.

Detail:

rjepipe – HOSTA
mode – HASP
port – /dev/isc10
–r 2 ---> specifies two active readers
–p 2 ---> specifies two active printers
–hx ---> half-duplex operation
–l LOGFILE --> console logfile
–son SIGNON --> file where signon resides
–dy ----> automatic routing/execution
–tg ----> .TAG files are to be created

Sample IBM Host (JES and POWER) Configurations

A Sample Host Definition for a Remote 2780 RJE Station

Summary: This defines to JES no compression, variable length records, blocked record format, multiple records supported, and no horizontal tab support. This workstation will have a console and the mode of transmission will be transparent.

System defaults will be taken on all printers, punch, and reader definitions unless specifically stated below.

Detail:

RMT60	Defines remote 60
2780	Terminal type
NOABUFEX	Specifies that this is not a 2770 with additional buffers
NOBUFEX	Specifies that this is not a 2770 with 512 byte buffers
BUFSIZE=400	Defines the largest buffer size to be sent or received
NOCOMP	Specifies that this device does not have compression facility
CONDEST=60	Specifies route code
CONSOLE	Specifies that this device has an operator console
DISCINTV=0000	Specifies terminal disconnect time in seconds if no successful text transmission occurred
VARIABLE	Specifies a variable length data record for terminal
MRF	Specifies that this is a 2780 with multiple-record support

HARDWARE	Specifies no multileaving support
NUMPR=1	Specifies one (1) remote printer
NUMPU=1	Specifies one (1) remote punch
NUMRD=1	Specifies one (1) remote reader
PASSWORD=""	Specifies no password for this terminal
ROUTECDE=60	Specifies that 60 is the remote id for this device
SETUPINF	Specifies that messages issued will not require response
NOTABS	Specifies no horizontal tab support
TRANS	Specifies text transparency
BLOCKED	Specifies blocked data record format
WAITIME=0001	Specifies the time in seconds to wait at completion of processing an inbound data stream

A Sample Host Definition for a Remote 3780 RJE Station

Summary: This defines to JES compression, variable length records, blocked record format, multiple records not supported, and no horizontal tab support. This workstation will have a console and the mode of transmission will be transparent.

System defaults will be taken on all printer, punch, and reader definitions unless specifically stated below.

Detail:

RMT62	Defines remote 62
3780	Terminal type

NOABUFEX	Specifies that this is not a 2770 with additional buffers
NOBUFEX	Specifies that this is not a 2770 with 512 byte buffers
BUFSIZE=510	Defines the largest buffer size to be sent or received
COMP	Specifies that this device does have compression facility
CONDEST=62	Specifies route code
CONSOLE	Specifies that this device has an operator console
DISCINTV=0000	Specifies terminal disconnect time in seconds if no successful text transmission occurred
VARIABLE	Specifies a variable length data record for terminal
NOMRFS	Specifies that this terminal does not have the multiple record feature
HARDWARE	Specifies no multileaving support
NUMPR=1	Specifies one (1) remote printer
NUMPU=1	Specifies one (1) remote punch
NUMRD=1	Specifies one (1) remote reader
PASSWORD=""	Specifies no password for this terminal
ROUTECDE=62	Where 62 is the remote id for this device
SETUPINF	Specifies that messages issued will not require response
NOTABS	Specifies that this 3780 has no horizontal tab support
TRANS	Specifies text transparency
BLOCKED	Specifies blocked data record format
WAITIME=0001	Specifies the time in seconds to wait at completion of processing an inbound data stream

Printer Definitions:

PRWIDTH=132 ----> PR1 - PR5

A Sample Host Definition for a Remote HASP RJE Station

Summary: This defines to JES no compression, variable length records, blocked record format, multiple records not supported and no horizontal tab support. This workstation will have a console and the mode of transmission will be transparent.

System defaults will be taken on all printer, punch, and reader definitions unless specifically stated below.

Detail:

RMT64	Defines remote 64
S/360	Terminal type
NOABUFEX	Specifies that this is not a 2770 with additional buffers
NOBUFEX	Specifies that this is not a 2770 with 512 byte buffers
BUFSIZE=400	Defines the largest buffer size to be sent or received
NOCOMP	Specifies that this device does not have compression facility
CONDEST=64	Specifies route code
CONSOLE	Specifies that this device has an operator console
DISCINTV=0000	Specifies terminal disconnect time in seconds if no successful text transmission occurred
VARIABLE	Specifies a variable length data record for terminal
NOMRF	No multiple-record support

MULTI	Specifies that there is multileaving support
NUMPR=2	Specifies two (2) remote printers
NUMPU=1	Specifies one (1) remote punch
NUMRD=2	Specifies two (2) remote readers
PASSWORD=""	Specifies no password for this terminal
ROUTECDE=64	Specifies that 64 is the remote id for this device
SETUPINF	Specifies that messages issued will not require response
NOTABS	Specifies no horizontal tab support
TRANS	Specifies text transparency
BLOCKED	Specifies blocked data record format
WAITIME=0001	Specifies the time in seconds to wait at completion of processing an inbound data stream

Printer Definitions:

PRWIDTH=132 ----> PR1 - PR5

Error Messages

In addition to the error messages listed in the sections "Responses to the Send Command" and "Responses to the Cancel Command" of the *User's Guide,* the following error messages may appear.

ATTN: UNABLE TO SEND - LINE BID NOT ACKNOWLEDGED...RETRYING

Communication line with the host has not been established. Check line and modem connections.

ERROR: COMMUNICATION LINE/MODEM MALFUNCTION DETECTED (LOST CTS)

The communication line connection to the host has been broken. Check phone line and modem connections and sign back on.

ERROR: CONTROLLER NOT ACTIVE

Attempting to issue an **rje** command when the AT&T BSC/RJE Emulator+ is not running. The AT&T BSC/RJE Emulator+ must be started.

ERROR: HOST NOT FOUND

An invalid path was specified for HOST (check that the **BR** environment variable is set to the correct full pathname for the HOST special file; invalid permissions have been set on the HOST special file (the user must have read/write permissions).

ERROR: RDR FAILED TO SEND FILE (filename) - **cannot read file**

Permissions on file are set incorrectly, or the file does not exist.

ERROR: PRT FAILED TO RECEIVE -- **unable to open './output/PMMYY.NNN'**

The **/usr/bscrje/output** output directory has not been created. The system administrator should create the output directory with read and write permissions for all non-super users.

HOST DISCONNECTED US

The host has disconnected the remote. (Sent in response to the signoff card.)

CONSOLE ALREADY SET TO /dev/tty

The terminal the **rje –o** command was issued from was already set as a console.

ERROR: ARGUMENT INVALID FOR MODE

An invalid argument was specified for the mode the AT&T BSC/RJE Emulator+ was started in. Example: specifying **–em** (a 2780 option) with the send command when the AT&T BSC/RJE Emulator+ is operating in 3780 mode.

ERROR: FAILED TO RECEIVE - PROTOCOL EXCEPTION

This indicates that a protocol error was received during transmission. The user should sign back on to the host.

ERROR: KEYBOARD FAILED TO SEND MESSAGE - LINE BID RETRIES EXHAUSTED.

This message indicates that a command sent from the console (most likely the signon card) has not been sent, because the communication line to the host was never established. Check line and modem connections and restart the AT&T BSC/RJE Emulator+.

ERROR: FAILED TO RECEIVE - INVALID BLOCK/RECORD FORMAT

The format of the record sent is invalid for the mode the AT&T BSC/RJE Emulator+ is operating in. The host and AT&T BSC/RJE Emulator+ configurations should be checked for compatibility.

HOST NOT RESPONDING - CANCELLING ALL ACTIVE READERS

Indicates a problem at the host site. Call the appropriate host operations personnel.

rje not found

Check path to include **/usr/bscrje** (the directory in which the AT&T BSC/RJE Emulator+ is started).

can't open pipe /usrdir/P9999

(where **usrdir** is the directory from which the command was issued)

The **/usr/bscrje/tmp** directory has not been created or does not have write permissions set for all users. The system administrator should create the **tmp** directory with proper permissions.

Error 16 Can't open /dev/isc##

This is in response to trying to start the AT&T BSC/RJE Emulator+ when it (or another AT&T product) is already running on that device. If the AT&T BSC/RJE Emulator+ needs to be restarted, the system administrator should execute the **stoprje** script, then restart the AT&T BSC/RJE Emulator+.

Can't open lmrje for reading!

This message results from trying to start the AT&T BSC/RJE Emulator+ in any directory other than the runtime directory.

ERROR: RDR FAILED TO SEND FILE (filename) PROTOCOL EXCEPTION

The named file was not successfully transmitted because a protocol error was detected. The user should sign back on to the host.

ATTN: RDRx REPEATING REQUEST TO INITIATE TRANSMISSION

The host has failed to acknowledge a request to transmit on Hasp reader x.

This occurs when the number of readers assigned through the **startrje** command does not match the number of readers defined by the host for this remote station.

ATTN: UNABLE TO SEND - LINE BID NOT ACKNOWLEDGED

The host has ignored repeated attempts to bid for the line. The host may be down or there may be a problem with the communications line.

ATTN: RECEIVED BAD HASP RCB - REQUESTING RETRANSMISSION

The host has transmitted an invalid RCB. This may indicate that the host gen does not agree with the 3B computer gen parameters. Otherwise, the user should sign back on to the host.

ATTN: RECEIVED HASP PERMISSION DENIAL FOR TRANSMISSION FROM RDRx

The host refused a request to transmit an RDRx. This occurs when the number of readers assigned through the **startrje** command does not match the number of readers assigned by the host.

ERROR: HOST REPEATEDLY REJECTING OUR DATA TRYING TO SYNC

The host has rejected our transmission. The user should try to sign back on to the host. If the problem persists, there may be a problem with the modem or communications line.

FAILED TO SEND <filename> - FORMATTING ERROR

This message indicates that the file the user wants to transmit contains binary sequences that will be interpreted as BSC control information. Request retransmission in transparent mode (the −x option).

NO DATA EXCHANGE WITH HOST -- EMULATOR BEING SHUT DOWN

The Emulator+ did not receive responses from the host within the expected time. Increase the **−to** timer value and restart the Emulator+.

Protocol Profile

Half-duplex

The AT&T BSC/RJE Emulator+ will send and receive in both directions, but not simultaneously.

Stop and Wait Transmission

After transmitting, the AT&T BSC/RJE Emulator+ will always wait for some kind of response or until the response timeout is reached.

Recognition of Text Blocks

The AT&T BSC/RJE Emulator+ recognizes information that begins with SOH or STX and ends with ETB or ETX as valid text blocks.

Recognition of Transparent Text Blocks

The AT&T BSC/RJE Emulator+ recognizes information beginning with DLE STX and ending with DLE ETB or DLE ETX as a valid transparent text block.

Recognition of Message Acknowledgments

After transmitting a text block, the AT&T BSC/RJE Emulator+ recognizes the following control sequences as message acknowledgments:

- ACK0 (positive acknowledgment 0)
- ACK1 (positive acknowledgment 1)
- WACK (wait acknowledgment)
- NAK (negative acknowledgment)

- EOT (end of transmission)
- text block (conversational mode)

Alternating Acknowledgments (2780/3780)

- The 2780/3780 BSC protocols use two positive acknowledgments, ACK0 and ACK1. The first text block received is acknowledged with ACK1. Subsequent text blocks are acknowledged by alternating ACK0 and ACK1.

Error Correction

- If the AT&T BSC/RJE Emulator+ detects a bad block check character in a received text block, it requests a retransmission of that text block by sending a NAK.
- If the AT&T BSC/RJE Emulator+ receives a NAK, it retransmits the last block of text.

ACK Query

After the AT&T BSC/RJE Emulator+ has sent a block of text, it waits several seconds for a response. The wait parameter is set by the response timer. If the AT&T BSC/RJE Emulator+ does not receive an acknowledgment within the specified timeout, or if it receives the wrong ACK, it sends an ACK query which is an ENQ.

The AT&T BSC/RJE Emulator+ will continue to send ENQ until it receives a response or it reaches the retry limit.

Recognition and Generation of Line Bids (2780/3780 Point-to-Point Operation)

The AT&T BSC/RJE Emulator+ recognizes line bids. It also generates line bids when it is necessary to gain control of the line.

TTD (Temporary Text Delay) (2780/3780)

The AT&T BSC/RJE Emulator+ responds to a TTD by sending a NAK and waiting for transmission to begin. The AT&T BSC/RJE Emulator+ generates TTD when it wishes to retain control of the line but does not have a text block ready to transmit.

EBCDIC/ASCII Operation

The AT&T BSC/RJE Emulator+ can use both the EBCDIC and ASCII character sets.

Space Compression/Expansion (3780)

This special feature can only be used when the 3780 is operating in the non-transparent mode. Each group of two or more consecutive spaces is replaced by an IGS(EBCDIC) or GS(ASCII) character followed by a character (space count character) defining the number of spaces removed. For a group of over 64 space characters, an additional IGS or GS character and space count character are inserted.

Conversational Mode

This standard feature improves line efficiency in applications where an immediate processor message to the 2780/3780 is required in response to a message from the 2780/3780 terminal.

The response message from the processor must begin with STX(DLE STX if transmitted in transparent mode).

The 2780/3780 printer must be in the ready state.

HASP Controller Functions

Generating the Initial Sequence

HASP workstations establish communications by transmitting the initial sequence SOH ENQ.

Handling Interleaving

The BSC HASP protocol provides for interleaving, which is a fully conversational form of transmission. A station gains control of the line whenever it receives a text block or a positive message acknowledgment (ACK0). At this point, the station can transmit either a text block or an ACK0. Consequently, two stations can exchange data blocks indefinitely with no intervening ACK0s.

Maintaining Sequence Counts

Each HASP text block contains a block control byte (BCB). Four bits within this byte are used for a modulo 16 block sequence count. The controller maintains both a send and receive count.

Handling Multileaving

HASP protocol allows records from several components to be combined into a single data block for transmission. A record can be any one of the traditional record types, such as a card image or a print line.

The controller builds data records for transmission to the host by combining records from active reader components. The controller examines the multileaving information within received data blocks and routes the various records to the proper printer and punch components.

HASP Control Sequences

- **ACK0 (positive acknowledgment)** is the positive acknowledgment for a correctly received textblock. (Note: Unlike 2780/3780, HASP does not use alternating acknowledgments). ACK0 is also exchanged in order to maintain line activity when neither station has text blocks to send.
- **DLE (data link escape)** signals the start of transparent text when it precedes STX. It signals the end of transparent text when it precedes ETB.
- **ETB (end of block)** is the end of message delimiter.
- **NAK (negative acknowledgment)** is used to request retransmission of text block that is received with a bad block check character(BCC).
- **SOH (start of header)** is the start of message delimiter for non-transparent text block.
- **SOH ENQ (initial sequence)** is transmitted before sign-on. The first station to initiate transmission sends SOH ENQ.
- **STX (start of text)** is the start of text delimiter. It immediately follows the initial SOH in non-transparent text blocks and the initial DLE in transparent text blocks.

Timeouts

One-Second Timeout

Used when operating on a point-to-point line on a contention basis. It is the period of time the primary station allows a secondary terminal to respond to an enquiry (ENQ). The primary station automatically retransmits the ENQ character after one second.

Two-Second Timeout

A receiving terminal must respond to a block check within two seconds. If unable to do so, the receiving terminal remains in receive mode and waits for the transmitting station to send an ENQ to solicit the response.

Three-Second Timeout

This timeout is used on a point-to-point line on a contention basis. It is the period of time that a secondary terminal allows a primary terminal to reply to an ENQ character. At the end of the timeout, the ENQ character is automatically encoded again and sent to the primary terminal.

A transmitting terminal will wait three seconds for a response to a block check sequence. If a reply is not received, an ENQ is automatically encoded and sent to the receiving terminal to solicit a response.

A receiving terminal initiates a three-second timeout upon receiving a sync pattern (SYN SYN). It must receive an STX character or another sync pattern within this time. If none is received, the terminal abandons synchronization and waits for another sync pattern. Upon receiving an STX character, another three-second timeout is initiated.

Component Selection

3780:	DC 1 printer
	DC 2 or DC 3 Card reader
2780:	ESC 4 punch
	ESC (note)

NOTE: The character following the ESC for printer selection can be any printer control character, such as a / (print and single space), or A (print and skip to 1). If an invalid two-character sequence is received, it is either ignored or an erroneous output component function control is initiated, depending on the specific character following the ESC character.

Carriage Control – Skip to Channel

If the standard commands for single space, double space, triple space, space suppress, and skip to channel 1 (e.g. space to top of form) are issued, the emulator will provide the correct newlines, carriage returns, and formfeeds.

If any of the other commands (skip to channels 2 through 12) is issued, the printer file will contain the ASCII translations of the EBCDIC ESC sequences shown in the following table. A user-written postprocessor can then use this printer file to create an output file containing the appropriate formfeeds and linefeeds needed to format the printed output.

ASCII	EBCDIC	Carriage Operation
ESC Q	ESC/	Single space
ESC R	ESC S	Double space
ESC S	ESC T	Triple space
ESC M	ESC M	Space suppress
ESC A	ESC A	Skip to channel 1
ESC B	ESC B	Skip to channel 2
ESC C	ESC C	Skip to channel 3
ESC D	ESC D	Skip to channel 4
ESC E	ESC E	Skip to channel 5
ESC F	ESC F	Skip to channel 6
ESC G	ESC G	Skip to channel 7
ESC H	ESC H	Skip to channel 8
ESC I	ESC I	Skip to channel 9
ESC J	ESC J	Skip to channel 10
ESC K	ESC K	Skip to channel 11
ESC L	ESC L	Skip to channel 12

BSC Definitions

Code	Code Meaning	Control State	Message Transfer State
ENQ	Inquiry	Can you accept transmission	Between blocks: Please respond or repeat the last response. Terminating a block: Discard this block and respond with NAK.
ACK0	Acknowledgment	I can accept transmission	Even block received and validated.
ACK1	Acknowledgment	None	Odd block received and validated.
STX	Start of text	Change to the message transfer state and start computing a CRC.	Clear check circuitry and start computing new CRC.
NAK	Negative ack	I cannot accept transmission	CRC not valid, can accept retransmission.
TTD	Temporary text delay	Transmission will begin, respond NAK and wait.	Transmission will continue, respond NAK and wait.
WACK	Wait before transmit	Inquire again later and delay transmission.	Inquire again !ater, delay transmission.
ETB	End of block	None	Check value follows, another text block to follow.
ETX	End of text	None	Check value follows. This completes the text transmission.

Code	Code Meaning	Control State	Message Transfer State
RVI	Reverse interrupt	None	Affirmative ack. and signal that the slave station wants master to relinquish the line.
EOT	End of transmission	End of message transfer state.	Not valid in text.
PAD	Leading	Establish bit sync.	Establish bit sync.
	Trailing	Turn-around time.	Turn-around time.
SYN	Synchronous idle.	Establish or assure character synchronism, or time-fill.	Establish or assure character synchronism, or time-fill.
EM	End of media	None	Used to indicate the end of a record when transmitting records of variable length (2780).
IRS	Interchange separator	None	Used in a multi-record transmission (3780) to separate records within blocks of text.
ITB	Intermediate text block		Encoded for a record of text (2780). This causes a CRC to be calculated, but does not cause line turnaround. 3780 can receive but cannot transmit an ITB.

iscconfig Command Options

The format of the **iscconfig** command is as follows:

iscconfig [**−P** *pe number*] *driver* [**−a**] [**−r**] [**−d**]

iscconfig manages the ISC card configuration on a 3B2 Computer or on an ACP attached to a 3B4000 Computer. Using this command, you can add new ISC cards to the configuration, remove cards from the configuration, or change the driver assignments on existing cards.

iscconfig also manages the device special files for the ISC cards in the configuration. It creates device entries when new cards are added to the configuration and removes them when cards are deleted from the configuration. The following is a description of the available options:

−P *pe number* is used only when configuring an ACP on a 3B4000 Computer. The *pe number* is the processor ID of the ACP to be configured.

driver is the name of the software driver to be configured.

The following options are mutually exclusive:

−a allows the user to add new cards or modify the driver assignment on existing cards.

−r allows the user to remove cards from the existing configuration.

−d displays the current configuration.

iscconfig also manages the creation of device entries in **/dev** and in **/adj/pe??/dev** for ACP's.

On a 3B2 Computer, device file names are of the form

type slot minor

where *type* is the lower case representation of the *driver* specified on the command line, *slot* is the slot number the ISC card is installed in, and *minor* is either 0 or 1 (for port 0 or port 1 on the ISC card). For example, a card in slot 4 assigned to the SNABSC driver will have device file names **/dev/snabsc40** and **/dev/snabsc41**. The major node for both devices is 4, and the minor numbers are 0 and 1 respectively.

On an ACP, device file names also incorporate the PE number of the ACP. They are of the form

type pe number slot minor

For example, a card in slot 4 of PE 120 assigned to the SNABSC driver will have device file names **/adj/pe120/dev/snabsc120.40** and **/adj/pe120/dev/snabsc120.41**. Redirect driver devices will also be created in **/dev** with the same names.

Index

Items in the index may appear in more than one document. Where this occurs, page numbers following an item are grouped by document. One of the following keys precedes each group of page numbers and tells you which guide to look in.

AG	*Administrator's Guide*
UG	*User's Guide*