

NAME

send – submit RJE job

SYNOPSIS

send argument ...

DESCRIPTION

Send is a command-level interface to the RJE subsystems *hasp*(VIII) and *uvac*(VIII). It allows the user to collect input from various sources in order to create a run stream consisting of card images. *Send* creates a temporary file, with a special format, to contain the collected run stream, and then queues the file for transmission by invoking *haspqr* or *uvacqr*, as appropriate. Further processing of the job is controlled by the appropriate PWB/UNIX RJE subsystem and the host computer to which the job is submitted.

Possible sources of input to *send* are: ordinary files, standard input, the terminal, and the output of a command or shell file. Each source of input is treated as a virtual file, and no distinction is made based upon its origin. Typical input is an ASCII text file of the sort that is created by the editor *ed*(I). An optional format specification appearing in the first line of a file (see *fspec*(V)) determines the settings according to which tabs are expanded into spaces. In addition, lines that begin with “~” are normally interpreted as commands controlling the execution of *send*. They may be used to set or reset flags, to define keyword substitutions, and to open new sources of input in the midst of the current source. Other text lines are translated one-for-one into card images of the run stream.

The run stream that results from this collection is treated as one job by the RJE subsystems. *Send* provides a card count for the run stream, and the queuer that is invoked announces the position that the job has been assigned in the queue of jobs waiting to be transmitted. The initial card of a job submitted to an IBM system must have a “/” in the first column. The initial card of a job submitted to a UNIVAC system must begin with a “@RUN” or “`run”, etc. Any cards preceding these will be excised. If a host computer is not specified before the first card of the runstream is ready to be sent, *send* will select a reasonable default. In the case of an IBM job, all cards beginning “/*\$” will be excised from the runstream, because they are HASP command cards.

The arguments that *send* accepts are described below. An argument is interpreted according to the first pattern that it matches. Preceding a character with “\” causes it to lose any special meaning it might otherwise have when matching against an argument pattern.

.	Close the current source.
–	Open standard input as a new source.
+	Open the terminal as a new source.
: <i>spec</i> :	Establish a default format specification for included sources, e.g., :m6t-12:.
: <i>message</i>	Print message on the terminal.
–: <i>prompt</i>	Open standard input and, if it is a terminal, print <i>prompt</i> .
+: <i>prompt</i>	Open the terminal and print <i>prompt</i> .
– <i>flags</i>	Set the specified flags, which are described below.
+ <i>flags</i>	Reset the specified flags.
= <i>flags</i>	Restore the specified flags to their state at the previous level.
! <i>command</i>	Execute the specified PWB/UNIX <i>command</i> via the one-line Shell, with input redirected to /dev/null as a default. Open the standard output of the command as a new source.

<i>\$line</i>	Collect contiguous arguments of this form and write them as consecutive lines to a temporary file; then have the file executed by the Shell. Open the standard output of the Shell as a new source.
<i>~comment</i>	Ignore this argument.
<i>=:keyword</i>	Prompt for a definition of <i>keyword</i> from the terminal.
<i>keyword=^xx</i>	Define <i>keyword</i> as a two-digit hexadecimal character code.
<i>keyword=string</i>	Define <i>keyword</i> in terms of a replacement string.
<i>host</i>	Job is to be submitted to: A , B , 1110 . The pseudonyms A and B are built into RJE to represent any IBM host connection. Their actual destinations are immaterial to RJE. The pseudonym 1110 is built into RJE to represent any UNIVAC host.
<i>filename</i>	Open the specified file as a new source of input.

Arguments of the form “*!chdir directory*” will be trapped so that the *send* process can execute the specified *chdir* itself. The original directory will be restored at the end of any source that contains a *chdir*.

The flags recognized by *send* are described in terms of the special processing that occurs when they are set:

- l List card images on standard output. EBCDIC characters are translated back to ASCII.
- q Do not output card images.
- f Do not fold lower case to upper.
- t Trace progress on diagnostic output, by announcing the opening of input sources.
- k Ignore the keywords that are active at the previous level and erase any keyword definitions that have been made at the current level.
- r Process included sources in raw mode; pack arbitrary 8-bit bytes one per column (80 columns per card) until an end-of-file.
- i Do not interpret control lines in included sources; treat them as text.
- s Make keyword substitutions before detecting and interpreting control lines.
- y Suppress error diagnostics and submit job anyway.
- g Gather mode, qualifying –l flag; list text lines before converting them to card images.
- h Write listing with standard tabs.
- p Prompt with “*” when taking input from the terminal.
- m When input returns to the terminal from a lower level, repeat the prompt, if any.
- a Make –k flag propagate to included sources, thereby protecting them from keyword substitutions.
- c List control lines on diagnostic output.
- d Extend the current set of keyword definitions by adding those active at the end of included sources.

Control lines are input lines that begin with “~”. In the default mode +ir, they are interpreted as commands to *send*. Normally they are detected immediately and read literally. The –s flag forces keyword substitutions to be made before control lines are intercepted and interpreted. Arguments appearing in control lines are handled exactly like the command arguments to *send*, except that they are processed at a nested level of input.

The two possible formats for a control line are: “~argument” and “~ argument ...”. In the first case, where the “~” is not followed by a space, the remainder of the line is taken as a single argument to *send*. In the second case, the line is parsed to obtain a sequence of arguments delimited by spaces. In this case the quotes

“`” and “”” may be employed to pass embedded spaces.

The interpretation of the argument “.” is chosen so that an input line consisting of “~.” is treated as a logical end-of-file. The following example illustrates some of the above conventions:

```
send -
~ argument ...
~.
```

This sequence of three lines is equivalent to the command synopsis at the beginning of this description. In fact, the “-” is not even required. By convention, the *send* command reads standard input if no other input source is specified. *Send* may therefore be employed as a filter with side-effects.

The execution of the *send* command is controlled at each instant by a current environment, which includes the format specification for the input source, a default format specification for included sources, the settings of the mode flags, and the active set of keyword definitions. This environment can be altered dynamically. When a control line opens a new source of input, the current environment is pushed onto a stack, to be restored when input resumes from the old source. The initial format specification for the new source is taken from the first line of the file. If none is provided, the established default is used or, in its absence, standard tabs. The initial mode settings and active keywords are copied from the old environment. Changes made while processing the new source will not affect the environment of the old source, with one exception: if **-d** mode is set in the old environment, the old keyword context will be augmented by those definitions that are active at the end of the new source. When *send* first begins execution, all mode flags are reset, and no keywords are defined.

The initial, reset state for all mode flags is the “+” state. In general, special processing associated with a mode *x* is invoked by flag **-x** and is revoked by flag **+x**. Most mode settings have an immediate effect on the processing of the current source. Exceptions to this are the **-r** and **-i** flags, which apply only to included source, causing it to be processed in an uninterpreted manner.

A keyword is an arbitrary ASCII string for which a replacement has been defined. The replacement may be another string, or (for IBM RJE only) the hexadecimal code for a single 8-bit byte. At any instant, a given set of keyword definitions is active. Input text lines are scanned, in one pass from left to right, and longest matches are attempted between substrings of the line and the active set of keywords. Characters that do not match are output, subject to folding and the standard translation. Keywords are replaced by the specified hexadecimal code or replacement string, which is then output character by character. The expansion of tabs and length checking, according to the format specification of an input source, are delayed until substitutions have been made in a line.

All of the keywords definitions made in the current source may be deleted by setting the **-k** flag. It then becomes possible to reuse them, although this is not recommended. Setting the **-k** flag also causes keyword definitions active at the previous source level to be ignored. Setting the **+k** flag causes keywords at the previous level to be ignored but does not delete the definitions made at the current level. The **=k** argument reactivates the definitions of the previous level.

A keyword may not be redefined, except redundantly, if it is active at some level of source input and its replacement is not null. Prompts for keywords that have already been defined at some higher level will simply cause the definitions to be copied down to the current level; new definitions will not be solicited. Only in the case where a keyword is defined by a null replacement, **A=**, is a redefinition allowed, **A=a**. Prompts for the keyword, **=:A**, will be satisfied by either definition.

Keyword substitution is an elementary macro facility that is easily explained and that appears useful enough to warrant its inclusion in the *send* command. More complex replacements are the function of a general macro processor(*m4*(I)), perhaps. To reduce the overhead of string comparison, it is recommended that keywords be chosen so that their initial characters are unusual. For example, let them all be upper case.

Send performs two types of error checking on input text lines. Firstly, only ASCII graphics and tabs are permitted in input text. Secondly, the length of a text line, after substitutions have been made, may not exceed 80 bytes for IBM, or 132 bytes for UNIVAC. The length of each line may be additionally constrained by a

size parameter in the format specification for an input source. Diagnostic output provides the location of each erroneous line, by line number and input source, a description of the error, and the card image that results. Other routine errors that are announced are the inability to open or write files, and abnormal exits from the Shell. Normally, the occurrence of any error causes *send*, before invoking the queuer, to prompt for positive affirmation that the suspect run stream should be submitted.

The *hasp* subsystem, which supports IBM RJE, operates in EBCDIC code. The *send* command is therefore required to translate ASCII characters into their EBCDIC equivalents. The standard conversion is based on the character set described in "Appendix H" of *IBM System/370 Principles of Operation* (IBM SRL GA22-7000). Each ASCII character in the octal range 040-176 possesses an EBCDIC graphic equivalent into which it is mapped, with four exceptions: broken vertical bar into ":", "^^" into "¬", "[" into "ç", "]" into broken vertical bar. In listings requested from *send* and in printed output returned by *hasp*, the reverse translation is made from EBCDIC to ASCII, with the qualification that EBCDIC codes that do not have ASCII equivalents are translated into "^^". The *uvac* subsystem, on the other hand, operates in ASCII code, and any translations between ASCII and field-data are made, in accordance with the UNIVAC standard, by the host computer.

Additional control over the translation process is afforded by the **-f** flag and hexadecimal character codes. As a default, *send* folds lower-case letters into upper case. For UNIVAC RJE it does more: the entire ASCII range 140-176 is folded into 100-136, so that "^^", for example, becomes "@". In either case, setting the **-f** flag inhibits any folding. Non-standard character codes are obtained as a special case of keyword substitution.

When invoked under the name *gath*, the *send* command establishes initial flag settings **-lgq** and suppresses announcement of a zero card count. While in **-gq** mode, long lines that are detected elicit a diagnostic but are not truncated. Also, in this mode, it is potentially useful to convey non-graphics to standard output. To prevent *gath* from deleting non-printing characters, each may be declared as a single character keyword whose replacement is itself. To retain backspaces, for example, supply the argument "BS=BS", where BS denotes the ASCII character whose octal code is 010.

The UNIVAC 1110 capability is only supported at the BTL Piscataway location.

FILES

/bin/sh	Shell
/tmp/sh*	Shell temporary
/usr/rje/sys	PWB/UNIX system name, e.g., "A"
/usr/rje/lines	RJE configuration table

And, where *xxxx* is either *hasp* or *uvac*:

/usr/xxxx/pool/stm*	temporary
/usr/xxxx/xmit???	queued output
/usr/xxxx/xxxxqer	queueing program
/usr/xxxx/xxxxlock	null file for lockout
/usr/xxxx/xxxxstat	queue status record

SEE ALSO

help(I), m4(I), sh(I), ascii(V), ebcidic(V), fspec(V), hasp(VIII)
Guide to IBM Remote Job Entry for PWB/UNIX Users by A. L. Sabsevit.

DIAGNOSTICS

"non-graphic deleted", "undefined tab deleted", "long line detected", "long line truncated", "illegal card excised" – followed by the resulting card image.

"Errors detected" – type "y" to submit anyway.

Use *help*(I) for explanations of error messages.

BUGS

Standard input is read in blocks, and unused bytes are returned via *seek*(II). If standard input is a pipe, multiple arguments of the form “-” and “-:*prompt*” should not be used, nor should the logical end-of-file “~”.