INTRODUCTION
-------------

R/C PERMITS A USER AT A REMOTE TELETYPewriter TO CREATE AND MAINTAIN SOURCE OR DATA FILES ON THE 8550 SYSTEM DISK. FILE TYPES CREATED AND MAINTAINED WITH R/C ARE: ALGOL, XALGOL, COBOL, FORTRAN, BASIC, AND DATA. THESE FILES REPRESENT NORMAL (80 CHARACTER PER CARD) PUNCHED CARD DECKS EXCEPT THAT THEY ARE STORED ON THE DISK. EACH RECORD CAN BE THOUGHT OF AS ONE CARD.

R/C ALLOWS THE USER TO:
- CREATE A VARIETY OF FILES
- RESEQUENCE FILES
- PRINT OR PUNCH FILES
- DELETE OR INSERT RECORDS INTO A FILE
- MODIFY RECORDS WITHIN A FILE
- SCAN A FILE FOR THE OCCURRENCE OF A STRING (OPTIONALLY REPLACING IT WITH ANOTHER STRING)
- REMOVE FILES
- COMPILe FILES
- PERFORM MANY OTHER FUNCTIONS

THE TELETYPewriter (IN CONJUNCTION WITH THE PROGRAM R/C) CAN BE CONSIDERED A KEYPUNCH EXTENSION WHICH ELIMINATES PUNCHED CARDS; IT OFFERS GREAT FLEXIBILITY IN FILE HANDLING.

R/C HAS TWO MAJOR RESTRICTIONS:

WHILE THE SEQUENCE NUMBER 99999999 IS PERMITTED FOR CONVENIENCE, THE MAXIMUM SEQUENCE NUMBER IS 2097151 (2^21-1).

THE MAXIMUM NUMBER OF RECORDS PERMITTED IN A FILE IS 8191.

IN THE DISCUSSION OF CERTAIN R/C VERBS, SOME OF THE ELEMENTS OF THE SYNTAX ARE GIVEN AS <M>, <N>, <I>, OR <J>. IN EACH CASE, THESE ELEMENTS REPRESENT INTEGER VALUES WHICH MUST BE PROVIDED BY THE USER. THE VALUE MAY BE A RECORD NUMBER FOR SOME VERBS, THE SEQUENCE NUMBER FOR OTHERS, OR AN INCREMENT AMOUNT. THE BRACKETED CONSTRUCT IS ONLY A FORM OF NOTATION USED TO REPRESENT AN INTEGER PARAMETER.
PROGRAM OPERATION AND RECORD SEQUENCING

DURING CREATION AND FILE MAINTENANCE, R/C AUTOMATICALLY ADVANCES (BY THE CURRENT VALUE OF THE "INCREMENT") THE SEQUENCE NUMBER OF EACH RECORD THAT IS INPUT. THE USER MAY SET THIS INCREMENT TO ANY DESIRED VALUE THROUGH USE OF THE "*INC" VERB (SEE BELOW), THE INCREMENT VALUE IS INITIALIZED TO 100 WHEN THE USER FIRST RUNS R/C.

THE INITIAL SEQUENCE NUMBER IS SET TO THE INCREMENT WHEN A NEW FILE IS OPENED BY A USER. FOR EXAMPLE, IF THE CURRENT INCREMENT WERE 100, THE FIRST SEQUENCE NUMBER IN THE FILE WOULD BE "1001". THIS INITIAL SEQUENCE NUMBER MAY BE CHANGED BY THE USER THROUGH THE USE OF THE CONSTRUCT "*<N>" (SEE BELOW).

AFTER EACH RECORD IS TYPED INTO THE FILE, THE SEQUENCE NUMBER OF THE NEXT RECORD IN THE FILE IS SET TO THE LAST SEQUENCE NUMBER PLUS THE INCREMENT. THIS SEQUENCE NUMBER IS NEXT TYPED ON THE TELETYPewriter.

IF THE FILE TYPE IS NOT "COBOL", THE SEQUENCE NUMBER IS FOLLOWED BY A COLON. IF A RECORD ALREADY EXISTS WITH THIS SEQUENCE NUMBER, LEADING ZEROS ARE TYPED AS A WARNING; OTHERWISE LEADING ZEROS ARE SUPPRESSED. AFTER THE SEQUENCE NUMBER IS TYPED OUT, THE USER MAY THEN ENTER THE DESIRED CONTENTS FOR THAT SEQUENCE NUMBER OR MAY ENTER A VERB TO PERFORM SOME OTHER FUNCTION.

ALTERNATIVELY, THE USER MAY SET THE SEQUENCE TO SOME OTHER VALUE THROUGH THE USE OF THE CONSTRUCT "*<N>" WHERE <N> IS THE DESIRED SEQUENCE NUMBER. NOTE THAT LEADING ZEROS ARE ACCEPTED BUT NOT NECESSARY WITH THIS CONSTRUCT.

BY USE OF THE "*<N>" CONSTRUCT AND THE "*INC" VERB TO SET THE SEQUENCE NUMBER INCREMENT, THE USER MAY SET UP HIS OWN NUMBERING SEQUENCE THROUGHOUT HIS FILE.

PROGRAM EXECUTION

INITIAL REMOTE TERMINAL OPERATIONS

FOR LOGGING IN TO A TELERTYPewriter, PRESS THE "DRG" BUTTON.
WAIT FOR A DIAL TONE FROM THE SPEAKER, AND DIAL THE COMPUTER
NUMBER. THE B5500 RESPONDS WITH THE MESSAGE:

BURROUGHS B=5500: <TT>/<BB>

(THE STATION NUMBER IS <TT>/<BB> WHERE <TT> IS THE
TERMINAL NUMBER AND <BB> IS THE BUFFER NUMBER.)

YOU MAY TYPE:

? LII <USERCODE>/<AUTHENTICATION CODE>

THIS LOG-IN MESSAGE MAY HAVE BEEN PRECEDED BY A "BG:" MESSAGE WHICH WOULD HAVE BLACKED OUT THE LINE ON WHICH THE
LOG-IN MESSAGE WAS TYPED.

THE B5500 VALIDATES THE <USERCODE> AND <AUTHENTICATION
CODE> AND RESPONDS BY TYPING OUT THE STATION NUMBER AND THE
TIME OF DAY OF THE LOG-IN.

TO CONNECT A REMOTE TERMINAL TO R/C, ENTER:

?? RUN R/CSEND.

THE B5500 RESPONDS BY EITHER TYPING OUT A "BG:" (BEGINNING
OF JOB) MESSAGE, A "SCHEDULED" MESSAGE, OR A "RUNNING" MESSAGE.
A "BG:" MESSAGE INDICATES THAT R/C WAS NOT PREVIOUSLY RUNNING BUT
HAS NOW BEEN ENTERED INTO THE MIX AND IS READY FOR USE. A
"SCHEDULED" MESSAGE INDICATES THAT R/C WAS NOT PREVIOUSLY RUNNING
AND IS SCHEDULED. IN THIS CASE, R/C IS NOT BROUGHT INTO THE MIX
UNTIL OTHER SYSTEM USERS COMPLETE THEIR WORK. THE "RUNNING"
MESSAGE INDICATES R/C IS ALREADY IN THE MIX.
WITH R/C IN THE MIX, IT AUTOMATICALLY SEARCHES OUT AND LOCKS
ONTO REMOTE TERMINALS WHICH HAVE REQUESTED CONNECTION (BY "RUN R/
C"). AS SOON AS YOUR TERMINAL IS LOCKED, R/C TYPES ONE OF THE
FOLLOWING MESSAGE SEQUENCES, ACCORDING TO THE MANNER IN WHICH R/C
ENDED DURING YOUR LAST R/C RUN (FIRST-TIME USERS ARE CONSIDERED
TO HAVE CAUSED NORMAL LAST ENTRIES):

INITIAL MESSAGE AFTER NORMAL TERMINATION OF LAST RUN:

<R/C VERSION NUMBER>
HELLO <USERCODE>

INITIAL MESSAGE AFTER "* END X" OR ABNORMAL LAST RUN
TERMINATION:

IF A FILE WAS OPEN
<R/C VERSION NUMBER>
<FILE NAME OF OPEN FILE>
HELLO <USERCODE>
<SEQUENCE NUMBER IN THE OPEN FILE>

OR IF NO FILE IS OPEN

<R/C VERSION NUMBER>
HELLO <USERCODE>

IF A MESSAGE HAS BEEN SENT FROM ANOTHER USER TO YOUR
USERCODE (SEE THE MAIL VERB), "MAIL X" IS TYPED INSTEAD OF
"HELLO".

EXAMPLES:

VERSION #
HELLO BLUM

THIS IS THE NORMAL INITIAL SEQUENCE FROM R/C.

VERSION #
HELLO* WILNER

This sequence indicates that R/C remembers the user's state from the previous run. The state includes the increment value, tab amount, save factor, percent on/off, and verb replacements.

Version #
File/Name
HELLO* SHARPE
00050600*

This sequence indicates that user has the file "FILE/NAME" open and is at the record with sequence number 50600. The leading zeros indicate that a record already exists with that number.

Final remote terminal operations

The user should terminate his use of R/C with the "* END" command (see below). R/C will respond with "GOOD BYE."

After he has ended R/C, if the user is done with the remote terminal he should log-out by entering: "?LO=

R/C Input

Input to R/C is either commands or records.

Commands are indicated by the character "*" in the first input position. If there is no open file, the "*" is optional. All other input is considered records to be placed in the open file. The format for a command is the "*" followed by an R/C verb and, if needed, its parameters. The verb and its parameters must be separated by a delimiter.

A delimiter is either a space or any special character except a ",", a "***", a ":", ".", or a "-". Multiple delimiters are treated as a single delimiter.
A command may be followed by another input (either a command
or a record) if it is terminated by a ";". An error in a command
of a multiple input inhibits the processing of the rest of that
input.

EXAMPLES:

1001* INC 50*

THIS IS AN EXAMPLE OF ONE OF THE INPUT COMMANDS.

1001BEGIN*

THIS IS AN EXAMPLE OF PLACING A RECORD AT SEQUENCE
NUMBER 100.

1001* INC 35* RESEQ*

THIS IS AN EXAMPLE OF MULTIPLE COMMANDS.

1001*35; THIS RECORD GOES AT 35*

THIS IS AN EXAMPLE OF A COMMAND FOLLOWED BY A RECORD OF
INPUT.

1001* OPEN A/B DATA* PRINT FOR HEJ*CLOSE*

THIS IS ANOTHER EXAMPLE OF MULTIPLE COMMANDS, NOTE
THAT THE "*" MUST APPEAR IN THE NEXT CHARACTER POSITION
FOLLOWING THE SEMICOLON OR THE REMAINDER OF THE RECORD IS
TREATED AS DATA.

AN INPUT LINE IS SENT TO THE COMPUTER BY TYPING THE
CHARACTER "*", TYPING ERRORS CAN BE CORRECTED, BY BACKspacing
AND LINE ERASING, BEFORE A MESSAGE IS SENT. THE BACKSPACE
CHARACTER IS THE APOSTROPHE (SHIFT 7) AND THE LINE ERASE
CHARACTER IS THE UP-ARRROW (SHIFT N). ALL THE FOLLOWING LINES OF
INPUT ARE EQUIVALENT (NOTE THE UNDERLINED CHARACTERS REPRESENT
USE OF THE SHIFT):

1001*THIS IS IT*

1001*THIS IS NOT, BUT THIS IS IT*
IF, AFTER BACKSPACING AND LINE ERASING, THE INPUT LINE CONTAINS MORE THAN 240 CHARACTERS, THE INPUT IS DISCARDED WITH AN "INPUT OVERFLOW" ERROR MESSAGE. DATA RECORDS ARE ALSO DISCARDED (WITH THE ERROR MESSAGE) IF THEY ARE TOO LARGE FOR THE FILE. (E. GTR 66 FOR COBOL FILES; GTR 80 FOR DATA FILES; AND GTR 72 FOR ALL OTHER FILES)

THERE ARE TWO CLASSES OF REQUESTS TO R/C: LONG AND SHORT. LONG OPERATIONS ARE THOSE THAT USUALLY ARE SLOW TO EXECUTE AND ARE CHARACTERIZED BY THE "WAIT..." MESSAGE. ALL OTHER REQUESTS ARE CLASSIFIED AS SHORT OPERATIONS. LONG OPERATIONS ARE SOMETIMES QUEUED BEFORE THE "WAIT..." MESSAGE, TO BE EXECUTED ONE AT A TIME. SHORT REQUESTS ARE DONE AS THEY ARE RECEIVED. THE USERS IN THE LONG REQUEST QUEUE (AND THE USER PERFORMING A LONG OPERATION IF IT IS NOT TYPING ON THE REMOTE) PERIODICALLY RECEIVE A FEW "HUBOUT" CHARACTERS OF REASSURANCE. R/C IGNORES ANY INPUT SENT BY USERS IN THE QUEUE OR BY THE USER WHOSE LONG OPERATION IS BEING PROCESSED.

IF A USER PRODUCES NO INPUT FOR FIVE MINUTES, HE IS SENT THE MESSAGE "LOOK ALIVE". IF HE DOES NOT RESPOND WITHIN ANOTHER FIVE MINUTE PERIOD, R/C PROCESSES A "+ END DS" FOR THAT USER.

R/C OUTPUT

OUTPUT TO THE TELETYPewriter OF THE SPECIAL CHARACTERS "", ", "", ", , "", , , AND 2 IS REPLACED BY A "$" CHARACTER IN ORDER THAT THEY DO NOT EVOKE TELETYPewriter CONTROL FUNCTIONS WITH WHICH THEY ARE ASSOCIATED. (THESE INCLUDE LINE=FEED, CARRiAGE=RETURN, MESSAGE=END, AND PAPER=TAPE=ON.)

WHEN THE "BREAK" KEY IS DEPRESSED DURING OUTPUT, THE OUTPUT IS TERMINATED WITH THE MESSAGE "BREAK".
R/C FILES

ALL FILES CREATED BY R/C ARE PERMANENT DISK FILES. THE SAVE FACTOR IS NORMALLY 7 DAYS, BUT IT MAY BE CHANGED BY THE SAVE VERB (SEE BELOW).

FILE TYPES

R/C ENABLES THE USER TO CREATE AND MAINTAIN ALGOL, COBOL, FORTRAN, XALGOL, BASIC, AND DATA FILES. THESE FILES HAVE 80 CHARACTER-LONG RECORDS (ONE CARD IMAGE).

XALGOL, BASIC, ALGOL AND FORTRAN FILES CONTAIN EIGHT DIGIT SEQUENCE NUMBERS LOCATED IN THE POSITIONS 73-80 OF THE CARD IMAGE.

COBOL FILES CONTAIN SIX-DIGIT SEQUENCE NUMBERS, PLACED IN POSITIONS 1-6 OF THE RECORD.

DATA FILES ARE NOT PHYSICALLY SEQUENCED ALTHOUGH R/C MAINTAINS AN INTERNAL, EIGHT-DIGIT NUMBER FOR EACH RECORD.

FILE NAMES

FILE NAMES MUST BE SUPPLIED TO R/C. THE FORM OF A NAME IS <FILE PREFIX> / <FILE SUFFIX>. THROUGHOUT THIS DOCUMENT, <FILE NAME> IS USED TO SPECIFY A FILE AND SHOULD BE IN THE FORM ABOVE. THE <FILE PREFIX> AND THE <FILE SUFFIX> MAY EACH BE NO LONGER THAN SEVEN CHARACTERS.

EXAMPLES:

A/B
GRIMY/GULCH
ZAP/1
16JAN/SUFFIX
0000000/DISK
RECORD REFERENCING

RECORDS IN THE OPEN FILE (SEE OPEN BELOW) ARE REFERRED TO BY THEIR SEQUENCE NUMBER. "DATA" FILES ARE IMPLICITLY SEQUENCED BY THE VALUE OF THE INCREMENT WHEN THEY ARE OPENED.

AN ALTERNATE METHOD OF REFERENCING RECORDS IN THE OPEN FILE IS RELATIVE SEQUENCE NUMBERS. A RELATIVE SEQUENCE NUMBER IS AN INTEGER PRECEDED BY A + OR - SIGN. IT MAY BE USED ANYPLACE A SEQUENCE NUMBER IS USED. IT IS TRANSLATED TO A SEQUENCE NUMBER BY MOVING FORWARD OR BACKWARD THE INDICATED NUMBER OF RECORDS AND USING THE SEQUENCE NUMBER OF THAT RECORD.

RECORDS IN A NON-OPEN FILE (EXTERNAL FILE) ARE REFERRED TO BY THEIR RELATIVE POSITION WITHIN THE FILE. THE FIRST RECORD IS 1, THE SECOND 2, ETC. ANY SEQUENCING THAT MAY BE ON THE RECORDS IS IGNORED.
FILE-HANDLING VERBS

This section describes verbs that handle files as a whole, rather than records within a file. However, a few verbs have options in R/C syntax that permit access to records within the file. The complete syntax is given for each verb as well as a discussion and examples of its use. An asterisk ('*') must always be the first character in the input string when a command is entered. If this is not followed, an existing record may be overwritten by the command itself.
FILE OPENING AND CREATION (OPEN)

* OPEN <FILE=NAME> <FILE=TYPE> NEW

* OPEN <FILE=NAME> <FILE=TYPE> OLD

* OPEN <FILE=NAME> <FILE=TYPE>

THE:"*OPEN" VERB ATTACHES THE USER TO THE DISK FILE <FILE>
NAME>. THE <FILE=TYPE> MUST BE EITHER "ALGOL", "COBOL",
"FORTRAN", "XALGOL", "BASIC", OR "DATA". IF THE <OPEN TYPE>
is
"NEW", A NEW DISK FILE IS CREATED. IF THE <OPEN TYPE> IS "OLD",
THE DISK FILE <FILE=NAME> IS OPENED AND RESEQUENCED BY THE
CURRENT VALUE OF THE INCREMENT. IF THE <OPEN TYPE> IS NEITHER
"NEW" NOR "OLD" THE DISK FILE <FILE=NAME> IS OPENED AND IT IS
READ TO DETERMINE ITS SEQUENCE NUMBERS. THIS LATTER FORM IS
SLOWER THAN THE OPEN "OLD".

EXAMPLES:

1* OPEN PROGRAM/SOURCE ALGOL NEW+
  1001
  THIS CREATES A NEW DISK FILE CALLED PROGRAM/SOURCE.

635001* OPEN ANOTHER/PROG DATA OLD+
  4501
  THIS OPENS THE FILE ANOTHER/PROG SEQUENCING IT BY THE
  CURRENT INCREMENT VALUE. NOTE THAT THE FILE THAT WAS OPEN
  IS FIRST CLOSED BEFORE THE NEXT FILE IS OPENED.

1* OPEN YET/ANOTHER COBOL+
  WAIT...
  READ ONLY FILE,
  7504
  THIS OPENS THE FILE YET/ANOTHER USING THE SEQUENCE
  NUMBERS WITHIN THE FILE. THE MESSAGE "READ ONLY FILE"
INDICATES THAT THE USER IS FORBIDDEN (BY THE FILE SECURITY SYSTEM) TO MODIFY THE FILE.

ERRORS:

DUP FILE: <FILE=NAME>

A FILE, <FILE=NAME>, ALREADY EXISTS AND THE USER IS TRYING TO CREATE A FILE WITH THAT NAME WITH AN "OPEN...
NEW".

NO FILE: <FILE=NAME>

THE USER IS TRYING TO OPEN A FILE, <FILE=NAME>, AND IT DOES NOT EXIST ON DISK.

BAD FILE: <FILE=NAME>

THE FILE <FILE=NAME> WHICH THE USER IS TRYING TO OPEN IS NOT BLOCKED CORRECTLY. THE CORRECT BLOCKING IS 10-WORD RECORDS WITH A MULTIPLE OF 3 RECORDS PER BLOCK.

INV USER: <FILE=NAME>

THE USER IS TRYING TO OPEN A FILE TO WHICH HE HAS NO ACCESS. IF THE USER HAS EITHER SECONDARY OR TERTIARY ACCESS, THE MESSAGE "READ ONLY FILE" IS TYPED.

FILE TOO LONG

THE USER IS TRYING TO OPEN A FILE WITH MORE THAN 8191 RECORDS.

SEQ OVERFLOW

THE FILE THE USER IS OPENING CAUSES THE SEQUENCE COUNTER TO EXCEED 2*2^31-151. THE FILE IS OPENED, BUT THE USER SHOULD RESEQUENCE IT.
SEQ ERR<NL>

THE FILE CONTAINS A RECORD WHOSE SEQUENCE NUMBER <N> IS LESS THAN THE SEQUENCE NUMBER OF THE PRECEEDING RECORD. THE FILE IS NOT OPENED.

BAD FILE TYPE: <FILE=TYPE>

<FILE=TYPE> IS NOT "ALGOL", "COBOL", "FORTRAN", "XALGOL", "BASIC", OR "DATA".
FILE CLOSING (CLOSE)

FILES ARE CLOSED BY USE OF THE FOLLOWING CONSTRUCT:

* CLOSE

THIS VERB DETACHES THE OPEN FILE, FROM R/C.

EXAMPLES:

56001* CLOSE*

THIS IS AN EXAMPLE OF CLOSING A FILE THAT IS IN THE CORRECT ORDER.

4501* RESEQ 100*
90001* CLOSE*
WAIT...

THIS IS AN EXAMPLE OF CLOSING A FILE THAT IS NOT IN ORDER.

ERROR:

NO FILE OPEN! CLOSE

THERE IS NO OPEN FILE TO CLOSE.
LISTINGS ON THE TELETYPewriter (LIST)

**TO LIST FILES OR ANY OF THEIR SEPARATE RECORDS, THE FOLLOWING CONSTRUCTS APPLY:**

- **LIST**
  - ----

- **LIST <FILE=NAME>**
  - ---- -----------

- **LIST <FILE=NAME> NO**
  - ---- ----------- --

- **LIST <FILE=NAME> <M>**
  - ---- ----------- ----

- **LIST <FILE=NAME> <M> <N>**
  - ---- ----------- ---- ----

- **LIST <M>**
  - ---- ----

- **LIST <M> <N>**
  - ---- ---- ----

THE "LIST" VERR CAUSES AN ENTIRE FILE OR PORTIONS OF A FILE TO BE LISTED ON THE TELETYPewriter. LISTING MAY BE DISCONTINUED BY PRESSING THE BREAK KEY ON THE TELETYPewriter.

THE FIRST FORM LISTS THE OPEN FILE.

THE SECOND FORM LISTS THE FILE <FILE=NAME>.

THE THIRD FORM LISTS THE FILE <FILE=NAME> WITHOUT THE RECORD NUMBERS.
THE FOURTH FORM LISTS <FILE-NAME> FROM THE <M>TH RECORD TO THE END.

THE FIFTH FORM LISTS <FILE-NAME> FROM THE <M>TH TO THE <N>TH RECORDS.

THE SIXTH FORM LISTS SEQUENCE NUMBER <M> OF THE OPEN FILE.

THE LAST FORM LISTS SEQUENCE NUMBERS <M> THROUGH <N> OF THE OPEN FILE.

EXAMPLES:

```plaintext
5001* LIST*
1001 BEGIN
2001 INTEGER I, J, K
3001 REAL X, Y, Z
4001 ARRAY A [0:9];
5001

55001* LIST 6900, 63*
89501 I = I + 5; J
91251 GO TO NEXT;
93001 HELP;
94001

300* LIST 60*
000060 MOVE A TO B;
000070

* LIST SOME/FILE*
11BEGIN
21 INTEGER I, J, K
31 REAL X, Y, Z
41 ARRAY A [0:9];
51 A [I] = X;
61END;

5001* LIST SOME/FILE NO*
BEGIN
INTEGER I, J, K;
REAL X, Y, Z;
ARRAY A [0:9];
A [I] = X;
END.
5001
```
* LIST LIBRARY/FILE 2
  21PROCEDURE READDATA 567,653
  31PROCEDURE WRITEDATA 654,789
  41PROCEDURE DATA 790,808

* LIST SOME/FILE 5
  51END.

* LIST SOME/FILE 200,500
  USE RECORD #5.

THE LAST EXAMPLE ILLUSTRATES THE COMMON ERROR OF
REFERENCING AN EXTERNAL FILE WITH SEQUENCE NUMBERS INSTEAD
OF RECORD NUMBERS.
COMPRRESSED FILE LISTINGS (QUICK)

A COMPRESSED FILE LISTING MAY BE OBTAINED FROM R/C BY USE OF THE CONSTRUCTS:

* QUICK
  = -----

* QUICK <FILE=NAME>
  = ------

* QUICK <FILE=NAME> NO
  = ------

* QUICK <FILE=NAME> <M>
  = ------

* QUICK <FILE=NAME> <M> <N>
  = ------

* QUICK <M>
  = ------

* QUICK <M> <N>
  = ------

THE "* QUICK" VERB LISTS ON THE TELETYPewriter DELETING ALL CONTIGUOUS BLANKS EXCEPT THE FIRST. THE FILE IS NOT AFFECTED BY THE VERB.

EXAMPLE:

45001* LIST 4301,44001
43001 FOR I = 1 TO 10 A STEP = 1 UNTIL 0 DO
44001 X[I] = SIN (Y)
45001* QUICK = 2, 1
43001 FOR I = 1 TO A STEP = 1 UNTIL 0 DO
44001 X[I] = SIN (Y)
45001
FILE REMOVAL (REMOVE)

TO REMOVE A FILE USE THE FOLLOWING CONSTRUCT:

* REMOVE <FILE-NAME>

THE REMOVE VERB REMOVES THE FILE <FILE-NAME> FROM DISK.

* REMOVE LISTING

REMOVES LINE/<USERCODE>, THE LISTING FILE FROM THE LAST COMPIRATION.

EXAMPLES:

32001* REMOVE A/B*
32001

5461* REMOVE ANOTHER/FILE*
NO FILE: ANOTHER/FILE
5461

** OPEN EXAMPLE/X COBOL OLD**
46500* REMOVE EXAMPLE/X*
1
LINE-PRINTER FILE REPRODUCTION (PRINT)

THE PRINT VERB:

*PRINT <A> <B>

PRINTS THE OPENED FILE ON THE LINE PRINTER (LABELED <A> <B>). IF "<B>" IS "DOUBLE" THEN DOUBLE SPACING IS USED.

* PRINT <A> <B> <M>

AS ABOVE, STARTING WITH SEQUENCE NUMBER <M>.

* PRINT <A> <B> <M> <N>

AS ABOVE, STOPPING WITH SEQUENCE NUMBER <N>.

EXAMPLES:

** OPEN TEST/CASE DATA** PRINT TC DOUBLE** CLOSE**

THIS EXAMPLE ILLUSTRATES AN INSTANCE WHERE A SEQUENCED FILE SHOULD BE TREATED AS DATA TO SHORTEN THE OPERATION. IF THE FILE WAS OPENED "ALGOL OLD" IT WOULD HAVE BEEN RESEQUENCED BY THE CURRENT VALUE OF THE INCREMENT AND THEN WHEN IT WAS CLOSED IT WOULD HAVE BEEN RECOPIED. IF IT WAS OPENED "ALGOL" IT WOULD HAVE BEEN READ TO DETERMINE ITS SEQUENCE NUMBERS, EITHER WAY WOULD HAVE MADE THE WHOLE OPERATION MUCH SLOWER THAN OPENING THE FILE "DATA".

87001* PRINT FOR USER*

WAIT...

87001
PUNCH-CARD FILE REPRODUCTION (PUNCH)

TO PUNCH A FILE:

* PUNCH <A> <B>

PUNCHES A CARD DECK (LABELED <A> <B>) OF THE OPENED FILE.

* PUNCH <A> <B> <M>

AS ABOVE, STARTING WITH SEQUENCE NUMBER <M>.

* PUNCH <A> <B> <M> <N>

AS ABOVE, STOPPING WITH SEQUENCE NUMBER <N>.

EXAMPLE:

76001* PUNCH A B 100+10+
WAIT...
76001
FILE COMPILATION (Compile)

Files may be compiled to the 85500 Library by the following construct:

* Compile <File-name>
  - -----------

This verb initiates the compilation of the open file to library using the compiler indicated in the open statement. The object code is named <file-name>. The listing output of the compilation is equated to "line/<userid>/<code>" on disk. The "* Listing" verb may be used to list the syntax errors.

* Compile <File-name> <Compiler>
  - -----------

Compiles the open file using the specified compiler.

Examples:

57001* Compile object/code
wait...

4701* Compile Test/Object eztran
queued(1).wait...

In the last example, the "eztran" compiler (eztran/disk) will be used. If the file is not in order, it will be reordered. Since this is a long operation the user gets a "wait" message. The "queued" message indicates that another user's long operation is being processed and that this long operation is queued until the other is done. The "1" indicates that this is the first request in the queue.
OUTPUT OF THE COMPILATION (LISTING)

---------------

THE LISTING FILE OF THE COMPILER IS EQUATED TO LINE/<USERCODE> ON DISK; THE FILE MAY BE ACCESSED BY THE USE OF THE LISTING VERB:

* LISTING <FILE-TYPE> <S> <L> <U>
  -------------  ---- --- ---

LISTS THE SEQUENCE NUMBERS RELATED TO SEGMENT <S> FROM RELATIVE ADDRESS <L> TO RELATIVE ADDRESS <U>. <FILE-TYPE> IS ALGOL, XLALGOL, BASIC, COBOL, OR FORTRAN AND INDICATES WHICH COMPILER CREATED THE LISTING FILE "LINE/<USERCODE>". (THIS FILE IS AUTOMATICALLY GENERATED BY THE COMPILATE VERB).

* LISTING <FILE-TYPE> ERRORS
  -------------

LISTS THE SYNTAX ERRORS OF YOUR LAST COMPILATION.

* LISTING
  -----

PRINTS THE LIST FILE OF YOUR LAST COMPILATION ON THE PRINTER.

EXAMPLES:

1* LISTING ALGOL 5*  25* 35*

WAIT,*SEGMENT = 51
  43001 REL. ADDR. = 26,
  44001 REL. ADDR. = 29,
  92001 REL. ADDR. = 32,
  93001 REL. ADDR. = 35,
1* LISTING ALGOL ERRORS*

WAIT,*
  78001ERROR 100 1.
FILE ZIPPED AS AN "EXECUTE" DECK (ZIP)

TO IMPLEMENT THE 45500 ZIP FUNCTION THROUGH R/C, USE THE FOLLOWING CONSTRUCT:

* ZIP
  - ---

THIS CONSTRUCT ZIPS THE OPENED FILE AFTER IT LINKS ALL THE CONTROL CARDS AS INDICATED BY "?". SEE THE ALGOL REFERENCE MANUAL FOR A DESCRIPTION OF THE "ZIP WITH FILE=ID" STATEMENT.

* ZIP <FILE=NAME>
  - ---

COPIES THE OPENED FILE CREATING <FILE=NAME> AND ZIPS <FILE=NAME>. NOTE THIS "ZIP" CONSTRUCT DOES NOT DESTROY THE OPEN FILE, AS DOES THE FIRST FORM.

EXAMPLES:

!COLUMN ON "*", COLUMN 73*
!OPEN MAKE/MANUAL DATA NEW, "EXECUTE XREF/JONES", *
  2000 % FILE DISK = TEACHER/0000094,*
  3000 % DATA CARD,*
  4000 % DISK SIX DOCONLY DOCUMENT FINAL,*
  5000 % 099999990,*
  6000 % END,*
  7000 % ZIP TEMP/NAME*
WAIT...
  7001

THIS EXAMPLE ILLUSTRATES HOW TO CREATE A CONTROL DECK AND INITIATE ITS EXECUTION. THE DECK WAS SAVED UNDER THE NAME "MAKE/MANUAL" SINCE THE ZIP CONSTRUCT INCLUDED THE DUMMY FILE "TEMP/NAME". (NOTE THAT THE ABOVE DECK WILL CREATE AN R/C USERS MANUAL.)

! * OPEN MAKE/MANUAL DATA; ZIP T/N; CLOSE*
WAIT...
RECORD COPYING (DITTO)

RECORDS MAY BE COPIED FROM ONE PLACE TO ANOTHER WITHIN A FILE BY THE CONSTRUCT:

* DITTO <M>
  - ----- --

COPIES CARD IMAGE <M> AS THE NEXT RECORD.

* DITTO <M> <N>
  - ----- --

COPIES THE CARD IMAGES <M> TO <N> AS THE NEXT RECORDS.

* DITTO OVERITE ON
  - ----- ------

* DITTO OVERITE OFF
  - ----- ------

* DITTO OVERITE
  - ----- ------

IF THE DITTO OVERITE IS OFF AND AN EXISTING RECORD IS ABOUT TO BE OVERWRITTEN, THE DITTO TERMINATES WITH AN "OVERITE OFF" MESSAGE. (IT IS INITIALLY OFF.) THE ABOVE COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT SETTING. ITS SETTING MAY BE REVERSED FOR ONE COMMAND BY PREFIXING THE DITTO WITH A **. (E.G. **DITTO 10,50.)

* DITTO <M> MOVE
  - ----- --

* DITTO <M>, <N> MOVE
  - ----- --

THE MOVE OPTION MOVES RECORDS <M> THRU <N> TO THE
CURRENT LOCATION. THIS OPTION IS VERY FAST, BUT OVERITING IS NOT ALLOWED.

EXAMPLES:

5001* DITTO 200,300*
7001* LIST*
1001ONE
2001TWO
3001THREE
4001FOUR
5001TWO
6001THREE
7001* DITTO 200,300 MOVE3+LIST*
1001ONE
4001FOUR
5001TWO
6001THREE
7001TWO
8001THREE
9001
COPYING EXTERNAL FILES (COPY)

WHOLE OR PARTIAL EXTERNAL FILES MAY BE COPIED INTO THE CURRENTLY OPENED FILE BY THE CONSTRUCTS:

* COPY <FILE=NAME>
  
* COPY <FILE=NAME> <M>
  
* COPY <FILE=NAME> <M>,<N>
  
* COPY <FILE=NAME> MERGE

THE COPY VERB COPIES RECORDS FROM ANOTHER FILE (<FILE=NAME>).


* COPY OVERITE ON
  
* COPY OVERITE OFF
  
* COPY OVERITE

IF THE COPY OVERITE IS OFF AND AN EXISTING RECORD IS ABOUT TO BE OVERWRITTEN, THE COPY TERMINATES WITH AN "OVERITE OFF" MESSAGE. (IT IS INITIALLY OFF.) THE ABOVE
COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT SETTING. ITS SETTING MAY BE REVERSED FOR ONE COMMAND BY PREFIXING THE COPY WITH A "-". (E.g., "-COPY A/B 10*50."

EXAMPLES:

1* OPEN SOURCE/MARK7 ALGOL NEW+
1001* COPY SOURCE/MARK6+
WAIT...
80001

5001* COPY LIBRARY/FILE 345, 368+
WAIT...
28001

1QUICK PATCH/FILE+
11A 00000050
21B 00000150
31C 00000200
41X 00000175
51Y 00000250
61Z 00000250

1OPEN MY/SOURCE ALGOL OLD+
4001* COPY OVERITE ONJ*COPY PATCH/FILE MERGE+
4001* LIST+
501A
10011
1501B
1751X
2001C
2501Z
30013
4001
INTRA-RECORD EDITING (INLINE)

Records may be edited by use of the "* INLINE" constructs described in the next paragraphs.

* INLINE <m>
  - ------- ----

* INLINE
  - ------- ----

* INLINE <m> <EDIT CHR>
  - ------- ---- ---- ----

* INLINE <EDIT CHR>
  - ------- ---- ---- ----

This sets up line <m> for inline editing. If the sequence number <m> is not included with the command, the previous record is used and the initial printing of it is suppressed.

* INLINE ECHO ON
  - ------- ----

* INLINE ECHO OFF
  - ------- ----

* INLINE ECHO
  - ------- ----

The modified record will be typed on the teletype if the inline echo is on. (It is initially on.) The above commands are used to set the option and to print its current setting. Its setting may be reversed for one command by prefixing the inline with a "=", (e.g., "INLINE =3").

To modify a portion of a record (card image) use the "* INLINE" verb. R/C prints the record number and the line, then gives a carriage return and line feed. It next spaces the print
BALL (ON THE NEW LINE) DIRECTLY BELOW THE FIRST CHARACTER
POSITION OF THE OLD LINE ABOVE. THE USER SPACES THE PRINT BALL
TO THE PROPER POSITION AND ACCOMPLISHES THE FOLLOWING ACTIONS:

TO INSERT A STRING, TYPE THE LETTER "I" FOLLOWED BY THE
STRING AND A "*".

TO DELETE A STRING, USE THE LETTER "D" FOLLOWED BY
SPACES UNDER THE CHARACTERS TO BE DELETED AND THEN A "*".

TO REPLACE A STRING, USE THE LETTER "R" FOLLOWED BY THE
NEW STRING OF THE SAME LENGTH AND A "*".

IF THE MODIFICATION IS TO BE DONE IN THE FIRST
CHARACTER, THEN INCLUDE THE "I", "R", OR "D" WITH THE INLINE
COMMAND.

EXAMPLES:

TO INSERT CHARACTERS INTO A RECORD:

```
6901*INLINE 5510
5510ABCDEFGHIJK
00005510: 12345
5510ABCDEFGHIJK
```

TO DELETE CHARACTERS FROM A RECORD:

```
77001*INLINE 67700
67700ABCDEFGHIJKLMNOP
00067700: D
97700ABCDEFGHIJKLMNOP
```

TO REPLACE CHARACTERS IN A RECORD:

```
6001*INLINE 551
5510ABCDEFGHIJKLMNOPQRST
000005510: 12345
5510ABCDEFGHIJKLMNOPQRST
```