

R/C  
---  
REFERENCE  
-----  
MANUAL  
-----

BY

RON BRODY

12/22/71 VERSION 94

INTRODUCTION  
-----

R/C PERMITS A USER AT A REMOTE TELETYPEWRITER TO CREATE AND MAINTAIN SOURCE OR DATA FILES ON THE B5500 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 "100:". 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.

THE EXAMPLES ILLUSTRATE THE SEQUENCE NUMBER OF THE RECORD TYPED ON THE LEFT SIDE OF THE PAGE AS IT APPEARS ON THE TELETYPEWRITER AND THE MANNER IN WHICH THESE RECORDS ARE INCREMENTED. ALTHOUGH ALGOL, XALGOL, BASIC AND FORTRAN FILES ACTUALLY CARRY THE SEQUENCE NUMBER IN CHARACTER POSITIONS 73-80 OF THE RECORD, R/C TYPES THE NUMBER ON THE LEFT MARGIN OF THE TELETYPEWRITER.

PROGRAM EXECUTION  
-----INITIAL REMOTE TERMINAL OPERATIONS  
-----

FOR LOGGING-IN TO A TELETYPEWRITER, PRESS THE "ORIG" 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:

? LI: <USERCODE>/<AUTHENTICATION CODE>+  
-----

THIS LOG-IN MESSAGE MAY HAVE BEEN PRECEDED BY A "?BO+"  
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/C)END,+  
-----

THE B5500 RESPONDS BY EITHER TYPING OUT A "BOJ" (BEGINNING  
OF JOB) MESSAGE, A "SCHEDULED" MESSAGE, OR A "RUNNING" MESSAGE.  
A "BOJ" 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 %" 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 USERS 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 ".", 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:

100:\* INC 50\*

THIS IS AN EXAMPLE OF ONE OF THE INPUT COMMANDS.

100:BEGIN\*

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

100:\* INC 3;\* RESEQ\*

THIS IS AN EXAMPLE OF MULTIPLE COMMANDS.

100:\*35; THIS RECORD GOES AT 35\*

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

100:\* OPEN A/B DATA;\* PRINT FOR ME;\*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-ARROW (SHIFT N). ALL THE FOLLOWING LINES OF INPUT ARE EQUIVALENT (NOTE THE UNDERLINED CHARACTERS REPRESENT USE OF THE SHIFT):

100:THIS IS IT\*

100:THIS IS NOT, BUTNTHIS IS IT\*

100:THE7IS IS IT←

100:THESE777IS IS IT←

IF, AFTER BACKSPACING AND LINE ERASING, THE INPUT LINE CONTAINS MORE THAN 240 CHARACTERS, THE INPUT IS DISCARDED WITH AN "INPUT OVERFLW" ERROR MESSAGE. DATA RECORDS ARE ALSO DISCARDED (WITH THE ERROR MESSAGE) IF THEY ARE TOO LARGE FOR THE FILE. (I. 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 "RUBOUT" 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 ≥ IS REPLACED BY A "s" 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:

```
!* OPEN PROGRAM/SOURCE ALGOL NEW+
100!
```

THIS CREATES A NEW DISK FILE CALLED PROGRAM/SOURCE.

```
63500!* OPEN ANOTHER/PROG DATA OLD+
4500!
```

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.

```
!* 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 2,097,151. THE FILE IS OPENED, BUT THE USER SHOULD RESEQUENCE IT.

SEQ ERR=<N>

THE FILE CONTAINS A RECORD WHOSE SEQUENCE NUMBER <N> IS  
LESS THAN THE SEQUENCE NUMBER OF THE PRECEDING 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 DETATCHES THE OPEN FILE, FROM R/C.

EXAMPLES:

```
      5600!* CLOSE*  
      ;
```

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

```
      450!* RESEQ 100*  
      9000!* 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" VERB 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:

```
500:* LIST+
100:BEGIN
200: INTEGER I, J, K ;
300: REAL X, Y, Z ;
400: ARRAY A [0 : 9] ;
500:
```

```
5500:* LIST 8900,+3+
8950: I := I + 5 ;
9125: GO TO NEXT ;
9300: HELP;
9400:
```

```
300* LIST 60+
000060 MOVE A TO B.
000070
```

```
!* LIST SOME/FILE+
1:BEGIN
2: INTEGER I, J, K ;
3: REAL X, Y, Z ;
4: ARRAY A [0 : 9] ;
5: A [I] := X ;
6:END.
```

```
500:* LIST SOME/FILE NO+
BEGIN
INTEGER I, J, K ;
REAL X, Y, Z ;
ARRAY A [0 : 9] ;
A [I] := X ;
END.
500:
```

```
1* LIST LIBRARY/FILE 2,4+
2:PROCEDURE READDATA 567,653
3:PROCEDURE WRITEDATA 654,789
4:PROCEDURE DATA 790,808
;
```

```
1* LIST SOME/FILE 5+
5: A [I] := X ;
6:END.
;
```

```
1* LIST SOME/FILE 200,500+
USE RECORD #S.
;
```

THE LAST EXAMPLE ILLUSTRATES THE COMMON ERROR OF  
REFERENCING AN EXTERNAL FILE WITH SEQUENCE NUMBERS INSTEAD  
OF RECORD NUMBERS.

COMPRESSED 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:

```
4500:* LIST 4300,4400+
4300:      FOR I :=      A STEP -1 UNTIL 0  DO
4400:      X [I] := SIN (Y) ;
4500:* QUICK -2, + 1+
4300: FOR I := A STEP -1 UNTIL 0 DO
4400: X [I] := SIN (Y) ;
4500:
```

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  
COMPILATION.

EXAMPLES:

3200:\* REMOVE A/B+  
3200:

546:\* REMOVE ANOTHER/FILE+  
NO FILE: ANOTHER/FILE  
546:

!\* OPEN EXAMPLE/X COBOL OLD+  
46500\* REMOVE EXAMPLE/X+  
;

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".

```
8700!* PRINT FOR USER+
WAIT...
8700!
```

PUNCHED-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:

7600:\* PUNCH A B 100,+10+  
WAIT...  
7600:

FILE COMPILATION (COMPILE)  
-----

FILES MAY BE COMPILED TO THE B5500 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/<USERCODE>" 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:

```
5700:* COMPILE OBJECT/CODE*  
WAIT...  
!
```

```
479:* 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  
USERS 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, XALGOL, BASIC, COBOL, OR FORTRAN AND INDICATES  
 WHICH COMPILER CREATED THE LISTING FILE "LINE/<USERCODE>".  
 (THIS FILE IS AUTOMATICALLY GENERATED BY THE COMPILE VERB).

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

LISTS THE SYNTAX ERRORS OF YOUR LAST COMPILATION.

\* LISTING  
 -----

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

EXAMPLES:

```

!* LISTING ALGOL 5, 25, 35+
WAIT... SEGMENT = 5:
  4300: REL. ADDR. = 26.
  4400: REL. ADDR. = 29.
  9200: REL. ADDR. = 32.
  9300: REL. ADDR. = 35.

```

```

!* LISTING ALGOL ERRORS+

```

```

WAIT...
  7800:ERROR 100 I.
:

```

```

8900:* LISTING+
8900:

```



FILE ZIPPED AS AN "EXECUTE" DECK (ZIP)  
 -----

TO IMPLEMENT THE B5500 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.+
  200 : %% FILE DISK = TEACHER/0000094.+
  300 : %% DATA CARD.+
  400 : $ DISK SIX DOONLY DOCUMENT FINAL+
  500 : @999999999+
  600 : %% END.+
  700 : * ZIP TEMP/NAME+
WAIT...
  700 :
```

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 HANDLING VERBS

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:

```
500:* DITTO 200,300+
700:* LIST+
100:ONE
200:TWO
300:THREE
400:FOUR
500:TWO
600:THREE
700:* DITTO 200,300 MOVE;*LIST+
100:ONE
400:FOUR
500:TWO
600:THREE
700:TWO
800:THREE
900:
```

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>).

THE FIRST FORM COPIES THE WHOLE FILE. THE SECOND FORM  
 COPIES THE <M>-TH RECORD (WHERE THE FIRST RECORD OF <FILE-  
 NAME> IS 1, THE SECOND RECORD IS 2, THE THIRD IS 3, ETC.).  
 THE THIRD FORM COPIES THE <M>-TH THROUGH THE <N>-TH RECORDS.  
 THE LAST FORM USES THE SEQUENCE NUMBERS OF THE RECORDS OF  
 <FILE-NAME> TO DETERMINE THEIR POSITION IN THE OPEN FILE.

\* 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:

```
!* OPEN SOURCE/MARK7 ALGOL NEW+
  100!* COPY SOURCE/MARK6+
WAIT...
  8900:

  500!* COPY LIBRARY/FILE 345, 368+
WAIT...
  2800:

!QUICK PATCH/FILE+
  1!A 00000050
  2!B 00000150
  3!C 00000200
  4!X 00000175
  5!Y 00000250
  6!Z 00000250
!OPEN MY/SOURCE ALGOL OLD+
  400!* COPY OVERITE ON;*COPY PATCH/FILE MERGE+
  400!* LIST+
  50!A
  100!1
  150!B
  175!X
  200!C
  250!Z
  300!3
  400!
```

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:  
-- -----

```
300:*INLINE 30500+
30500:ABCDEFGHIJK
00030500: I12345+
30500:ABC12345DEFGHIJK
```

TO DELETE CHARACTERS FROM A RECORD:  
-- -----

```
7700:*INLINE 67700+
67700:ABCDEFGHIJKLMNQP
00067700: D +
67700:ABHIJKLMNQP
```

TO REPLACE CHARACTERS IN A RECORD:  
-- -----

```
600:* INLINE 551+
551:ABCDEFGHIJKLMNQPQRST
00000551:R12345+
551:A12345GHIJKLMNQPQRST
```