

<i>ZZZZZZZZZZZZZZZZ</i>	<i>22222222222222</i>
<i>ZZZZZZZZZZZZZZZZ</i>	<i>22222222222222</i>
<i>ZZZZ</i>	<i>2222 2222</i>
<i>ZZZZ</i>	<i>2222</i>
<i>ZZZZ</i>	<i>2222</i>
<i>ZZZZ</i>	<i>2222</i>
<i>ZZZZ</i>	<i>2222</i>
<i>ZZZZ</i>	<i>2222</i>
<i>ZZZZ</i>	<i>2222</i>
<i>ZZZZZZZZZZZZZZZZ</i>	<i>222 222222222222</i>
<i>ZZZZZZZZZZZZZZZZ</i>	<i>22222222222222</i>

EDITOR — ASSEMBLER

REVISION LEVEL 1.3

COPYRIGHT D.P.S.S 1980

```

*****
*                                     *
*          INTRODUCTION              *
*                                     *
*****

```

THE Z2 EDITOR—ASSEMBLER HAS BEEN WRITTEN SPECIALLY FOR THE NASCOM 2.
 ITS AIM IS TO ENABLE YOU TO EDIT AND ASSEMBLE SOURCE OR TEXT FILES WITH
 SPEED AND EASE. THE EDITOR PRESENTS YOUR SOURCE FILE AS IF YOU: HAD A
 VIEWFINDER ON A PRINTOUT OF IT.
 YOU MAY SCROLL UP OR DOWN THE FILE USING THE ARROW KEYS.
 STRING LOCATE AND FIND COMMANDS ENABLE YOU TO GET SWIFTLY TO THE POINT WITHIN
 A FILE THAT YOU ARE SEEKING.
 GLOBAL STRING REPLACEMENT IS PROVIDED TO TAKE THE TEDIOUS OUT OF MAKING MANY
 IDENTICAL FILE ALTERATIONS.
 FULL SCREEN EDITING IS PROVIDED TO ENABLE YOU TO AMEND A COMPLETE SCREEN IMAGE
 OF YOUR FILE IN ONE OPERATION.
 LINE NUMBERS WHICH WASTE 2 BYTES OF RAM FOR EVERY SOURCE LINE HAVE BEEN
 ELIMINATED AND A USER SELECTABLE TAB CHARACTER ALLOWS COMPRESSED FILES ALSO
 GIVING A NEATLY PRESENTED DISPLAY.
 THE Z2 ASSEMBLER HAS BEEN WRITTEN TO ENABLE YOU TO ASSEMBLE SOURCE FILES
 CREATED USING OTHER ASSEMBLERS, NORMALLY WITHOUT ALTERATION.
 WHERE MINOR ALTERATIONS ARE REQUIRED THE GLOBAL REPLACE COMMAND CAN BE USED.
 THE ASSEMBLER IS CURRENTLY BY FAR THE FASTEST FOR NASCOM WITH AVERAGE
 ASSEMBLY TIMES FOR A COMPLEX SOURCE FILE GENERATING 2K OF OBJECT CODE
 (E.G. NAS-SYS) IN THE ORDER OF 12 SECONDS.
 NUMEROUS ASSEMBLER OPTIONS ARE PROVIDED, SELECTABLE IN ANY COMBINATION.
 MEMORY MAPPED VDU OUTPUT ENSURES HIGH SPEED OUTPUT OF LISTINGS.
 LISTINGS OR FILES MAY BE OUTPUT TO :— VDU SCREEN, PARALLEL PRINTER , SERIAL
 PORT, ALL AT THE SAME TIME OR IN ANY COMBINATION.
 SPECIAL LIST CONTROL PSEUDO—OP ALLOWS PART LISTINGS.
 AN OPTIONAL FULLY SORTED SYMBOL TABLE IS PROVIDED.
 THE MERGE FEATURE ENABLES YOU TO ASSEMBLE MULTIPLE CROSS—REFERENCED PROGRAMS.
 THE ALTERNATE REGISTER SET IS NOT USED.
 A NEW ‘ LINK—FILE ’ PSEUDO—OP PERMITS CONTINUOUS ASSEMBLY OF MANY SEPARATE
 FILES SCATTERED THROUGHOUT RAM.
 BOTH EDITOR AND ASSEMBLER ARE SEPARATE MODULES THE ONLY COMMON ROUTINE BEING THE
 PRINTER/SERIAL/VDU DRIVER.
 Z2 FILES ARE SAVED/RESTORED ON CASSETTE USING NAS-SYS READ / WRITE ROUTINES ALL TAB
 CONTROLS , OPTIONS , FILE NAME ETC. ARE SAVED ALONG WITH THE FILE.
 FILES CAN BE MOVED/DUPPLICATED ANYWHERE IN RAM USING Z2’S FILE ‘COPY’ COMMAND.
 ASSEMBLED SEPARATELY OR OBJECT—CODE—MERGED USING THE MERGE OPTION.
 THIS FEATURE ALLOWS MANY SMALLER MODULES OF A LARGE SOURCE FILE TO BE
 DEBUGGED ETC. SEPARATELY.
 RAM USAGE FOR Z2 IS CURRENTLY :—

EDITOR PACKAGE	2K
ASSEMBLER PACKAGE	3K
Z2 ‘NEW’ FILE IMAGE	117 BYTES .
SYMBOL TABLE	8 BYTES PER ENTRY.
	(USER SPECIFIES SYMBOL TABLE
	WORK AREA.)
WORK AREA	0D00—0E00 (THIS SEPARATE WORK AREA IS
	NECESSARY AS Z2 WILL SHORTLY
	BE AVAILABLE IN EPROM.)

* Z2 COMMANDS *

FUNCTION	VERB	COMMENTS
EDIT	E FFF..F	<p>EDIT FILE NAMED FF..F (MAX 10 LETTERS) (IF FILE NAME IS OMITTED THE CURRENTLY ASSIGNED FILE NAME IS IMPLIED). A CHECK IS MADE ON THE FILE START ADDRESS RAM LOCATION ITS CONTENTS SHOULD BE THE LETTER 'F'. IF IT IS NOT THE LI ST COMMAND IS INVOKED AND THE EDIT COMMAND ABORTED.</p>
COPY	C AAAA	<p>COPY CURRENT FILE TO RAM ADDRESS AAAA. THE NEW END OF FILE POINTER IS CALCULATED AND STORED WITH THE FILE. THE NEW DUPLICATE FILE WILL AUTOMATICALLY BECOME THE CURRENT FILE. NEW AND OLD FILES MAY OVERLAP IF DESIRED AS THE COMMAND USES NAS—SYS 'ICOPY' RTN.</p>
LIST	L	<p>LIST INFO. FOR CURRENT FILE (NAME , START ADDRESS , END ADDRESS)</p>
OPTIONS	O	<p>SET FILE OPTIONS. THESE ARE LISTED UNDER THE 'ASSEMBLER OPTIONS' SECTION. EACH Z2 FILE HAS ITS OWN OPTION BYTE SAVED WITH THE FILE. ANY COMBINATION OF OPTIONS MAY BE SET.</p>
PRINT	P	<p>PRINT THE CURRENT FILE. (SEE OPTIONS P,V,X AND T) THE CURRENT FILE IS OUTPUT (ALONG WITH CONTROL INFO.) TO THE SELECTED DEVICE'S. IF THE ' T ' OPTION IS SET OUTPUT WILL CONTAIN ALL NECESSARY TABBING SPACES. THE 'V' OPTION DISPLAYS ONLY ONE LINE AT A TIME AT THE TOP OF THE VDU, ITS FUNCTION WHEN PRINTING FILES IS TO VISUALLY. MONITOR EACH LINE THAT IS BEING OUTPUT TO THE PIO/SERIAL PORTS. FULL VDU OUTPUT OF THE FILE CAN BE ACHIEVED USING THE EDITOR.</p>
MONITOR	M	<p>RETURN TO NAS-SYS MONITOR. THIS COMMAND EXECUTES A CONTROLLED RETURN TO NAS-SYS USING THE 'DF 5B' CALL.</p>
SETFILE ADDR. S AAAA		<p>SETS THE FILE POINTER TO ADDRESS AAAA ALL TABS , OPTIONS , INFO. ETC. FOR THIS FILE NOW BECOME ACTIVE.(A CHECK IS MADE ON ADDRESS AAAA ITS CONTENTS SHOULD START WITH THE LETTER 'F' ,IF IT DOES NOT A '?' WILL BE PUT IN THE STATUS DISPLAY AND THE FILE POINTER WILL REMAIN UNCHANGED)</p>

FUNCTION -----	VERB -----	
RESTORE	R	RESTORE A Z2 FILE FROM CASSETTE. NAS-SYS READ ROUTINES ARE USED • (THE FILE IS ONLY READ INTO RAM, YOU MAY NOW USE THE SETFILE ADDRESS COMMAND TO ACCESS THIS FILE.)
FILE	F	SAVE THE CURRENT FILE ON TO CASSETTE. THE FILE IS SENT TO THE SERIAL PORT WITH ALL INFO. (NAME , TABS , OPT IONS ETC.) IN NAS—SYS WRITE FORMAT.
ASSEMBLE	A	ASSEMBLE THE CURRENT FILE (REF ASSEMBLER SECTION)
SET WORK	W	ASSIGN SYMBOL TABLE WORK ADDRESS AAAA. (RAM LOCATION 0D60) .

@/W

NOTES:— AT ALL TIMES IN Z2 OR EDITOR MODE THE LF/CH KEY CAN BE USED TO
RECALL THE LAST LINE THAT WAS INPUT TO THE BOTTOM LINE OF THE SCREEN.
THE CURSOR IS SET TO THE END OF THE LINE. IF DESIRED THE LINE CAN
BE ALTERED AND ENTER PRESSED.

APART FROM THE STRING LOCATE/FIND COMMANDS 'ENTER' CAN BE PRESSED
WITH THE CURSOR AT ANY POSITION (OTHER THAN THE 1ST) OF THE LINE.
THE WHOLE LINE WILL BE TAKEN AS INPUT AND PROCESSED.

THE TOP R.H. CORNER OF THE SCREEN IS USED FOR STATUS DISPLAYS.
ANY BAD COMMAND , ADDRESS OR FILE POINTER WILL RESULT IN A '?'
BEING DISPLAYED HERE.

Z2 HAS AN INTEGRAL REPEAT KEY ROUTINE ACTIVE IN ALL MODES , THE
RATE OF REPEAT IS SELECTABLE. (SEE Z2 RAM LOCATIONS .)
WHEN USING THE UP AND DOWN ARROW KEYS IN EDIT MODE THE REPEAT
KEY RATE IS SET TO FAST BUT RESTORED TO ITS ORIGINAL SETTING
AFTER THE NEXT COMMAND IS ENTERED.

FOR 'W' , 'S' AND 'C' COMMANDS, ADDRESS 'AAAA' IS CHECKED FOR VALIDITY .
IF A '*' CHARACTER APPEARS IN THIS FIELD THE END OF FILE ADDRESS
FOR THE CURRENT FILE IS USED. E.G. ENTER 'W*' TO ASSIGN A WORK AREA
DIRECTLY AFTER THE CURRENT SOURCE FILE.

* *
* Z2 EDITOR *
* *

COPYRIGHT D.P.S.S. 1980

***SCREEN FORMAT ***

UPON ENTRY TO THE EDITOR THE CURRENT FILE WILL BE DISPLAYED.

THE TOP LINE OF THE SCREEN CONTAINS STATUS INFORMATION FROM TIDE EDITOR INFORMING YOU OF :—

FILE NAME
EDITOR MODE
COMMAND STATUS

STARTING AT THE SECOND LINE DOWN THE SCREEN THE FIRST 14 LINES OF FILE DATA WILL BE DISPLAYED. THROUGHOUT THE EDITING PROCESS THESE 14 VDU LINES WILL ALWAYS DISPLAY FILE DATA.

THE BOTTOM LINE OF THE SCREEN IS AN INPUT AREA WHERE YOU WILL TYPE COMMAND STATEMENTS AND NEW LINES OF SOURCE WHEN IN INPUT MODE.

THE 'TOP' AND 'END' OF YOUR CURRENT FILE ARE DENOTED BY MARKERS *TOF AND *EOF RESPECTIVELY. THESE ARE THE LIMITS OF YOUR FILE AND NO ALTERATIONS CAN BE MADE OUTSIDE THESE LIMITS.

WHEN EDITING A NEW FILE THESE MARKERS WILL OBVIOUSLY BE TOGETHER.

THE FIRST COLUMN OF THE WHOLE SCREEN SHOULD BE BLANK APART FROM A LINE IN THE CENTRE OF THE SCREEN HAVING A > .

THIS WILL BE REFERRED TO AS THE PROMPTED LINE.

* EDITOR COMMANDS *

FUNCTION	VERB	
INPUT	I	<p>GO TO 'INPUT MODE. LINES ENTERED ON THE BOTTOM LINE OF THE SCREEN (MAX 46 CHARS.] ARE INPUT INTO THE FILE BELOW THE PROMPTED LINE. INPUT STARTS FROM THE SECOND. COLUMN OF THE SCREEN TO KEEP DISPLAY ALIGNMENT CORRECT (THE CURSOR IS SET ACCORDINGLY). THE EDITOR REMAINS IN INPUT MODE UNTIL A NULL LINE IS ENTERED, OR CURSOR AT INITIAL POSITION.</p> <p>(** ANY CHARACTERS EXCEPT HEX A0 OR HEX C0 ARE VALID. A0 IS USED TO MARK TOP OF FILE AND C0 END OF FILE.)</p>
TOP	T	GO TO THE TOP OF FILE. (*TOF)
BOTTOM	B	GO TO THE BOTTOM OF FILE. (*EOF)
UP	"UP ARROW"	SCROLL FILE UP THE SCREEN. (HOLD KEY REPEAT) (RATE SET TO FAST
DOWN	"DOWN ARROW"	SCROLL FILE DOWN THE SCREEN. " " "
DELETE	DELXX	<p>DELETE LINES FROM THE FILE STARTING AT THE PROMPTED LINE.</p> <p>XX= DECIMAL NUMBER OF LINES TO DELETE. 1--99 IS VALID, IF XX IS OMITTED A DEFAULT OF 1 IS IMPLIED.</p>
	DEL*	<p>THE COMMAND 'DEL*' SHOULD BE USED CAUTION IT WILL DELETE ALL LINES FROM THE PROMPTED LINE TO THE END OF FILE.</p> <p>(ATTEMPTING TO DELETE WHILE THE PROMPTED LINE IS THE TOP OF FILE MARKER '*TOF' WILL HAVE NO EFFECT.)</p>
LOCATE	L/YY..Y	<p>LOCATE STRING YY..Y ANYWHERE IN THE FILE MAX STRING LENGTH IS 46 CHARACTERS.</p> <p>UNLIKE OTHER COMMANDS THE COMMAND STATEMENT IS TERMINATED AT THE CURSOR POSITION. THIS ENABLES SEARCHING OF STRINGS HAVING TRAILING BLANKS.</p> <p>THE FILE IS SEARCHED FROM THE PROMPTED LINE TO THE END OF FILE AND THEN CONTINUED FROM TOP OF FILE BACK TO THE CURRENT PROMPTED LINE.</p> <p>IF THE STRING WAS NOT FOUND A MESSAGE REFLECTING THIS WILL BE DISPLAYED, ALSO THE PROMPTED LINE WILL REMAIN AS IS.</p> <p>A DOWN ARROW IN THE EDITOR STATUS DISPLAY SAYS THAT THE STRING WAS FOUND BELOW THE LAST LINE WHERE THE COMMAND WAS ENTERED. SIMILARLY AN UP ARROW SAYS IT WAS ABOVE.</p> <p>A RIGHT ARROW SAYS THAT THE CURRENT PROMPTED LINE IS THE ONLY LINE CONTAINING THE STRING</p> <p>(USE THE LF/CH KEY THEM ENTER TO CONTINUE SEARCHING</p>

FUNCTION -----	VERB -----	COMMENTS -----
FIND	F/YY,,Y	<p>SAME AS LOCATE COMMAND EXCEPT THAT THE SPECIFIED STRING MUST BE AT THE START OF A LINE. (USEFUL FOR FINDING LABELS)</p>
REPLACE	R/XX..X/YY..Y/ZZ	<p>REPLACE STRING XX..X WITH STRING YY..Y ZZ NUMBER OF TIMES. THE CHARACTER STRING XX..X IS SEARCHED FOR FROM THE CURRENT LINE TO THE END OF FILE AND FOR EVERY OCCURENCE OF XX..X STRING YY..Y IS SUBSTITUTED. A COUNT OF THE NUMBER OF SUBSTITUTIONS IS KEPT AND DISPLAYED IN THE STATUS DISPLAY. THE NEW PROMPTED LINE WILL BECOME THE LINE WHERE THE LAST SUBSTITUTION WAS MADE, BUT IF STRING XX..X IS NOT FOUND OR COUNT ZZ IS EXHAUSTED THE PROMPTED' LINE WILL NOT CHANGE THE DELIMITING CHARACTER FOLLOWING THE 'R' IN THE COMMAND WILL BE USED AS A DELIMITER THROUGHOUT. E.G. R/MONDAY/TUESDAY/1 IS THE SAME AS :- R,MONDAY,TUESDAY,1 THE REPLACE COMMAND MUST CONTAIN THREE DELIMITERS. THE PARAMETER ZZ IS A DECIMAL COUNT 1—99 BUT IF AN ASTERISK * IS ENTERED THIS IMPLIES THE REPLACE FUNCTION WILL BE PERFORMED FROM THE PROMPTED LINE TO THE END OF FILE. (WITH THIS FORM OF THE REPLACE COMMAND THE PROMPTED LINE WILL ALWAYS REMAIN AS IS) IF ZZ IS OMITTED DEFAULT OF 1 IS IMPLIED</p>
CHANGE FULL SCREEN	C/	<p>GO TO CHANGE MODE AND PERFORM FULL SCREEN EDITING. WHEN THE SCREEN DISPLAY HAS BEEN CHANGED USING NAS-SYS CURSOR CONTROL PRESS ENTER KEY AND THE FILE IN RAM IS UPDATED. (BECAUSE TAB CHARACTERS ARE NOT DISPLAYED TABBING IS TEMPORARILY TURNED OFF WHILE THE SCREEN IS CHANGED. WHEN ENTER IS PRESSED TABBING IS RESTORED.) ENTER KEY MAY BE PRESSED WITH THE CURSOR IN ANY POSITION ON THE SCREEN. *** EXCEPT THE BOTTOM LINE *** NOTE:— THE FIRST COLUMN OF THE SCREEN IS VOID AND ANY CHARACTERS APPEARING HERE WILL BE LOST. ALSO ANY CHARACTERS SHIFTED OFF THE RIGHT SIDE OF THE SCREEN USING CURSOR CONTROL WILL BE LOST.</p>
CHANGE	C	<p>AS ABOVE BUT THE CHANGE FUNCTION IS ONLY APPLIED TO THE PROMPTED LINE. ALTERATIONS TO ANY OTHER LINES WILL BE IGNORED. (THE SINGLE LINE CHANGE IS MORE SUITABLE FOR LARGE FILES WHERE SMALL MODIFICATIONS ARE BEING MADE AS THE FULL SCREEN CHANGE COMMAND TAKES 14 TIMES LONGER TO PROCESS)</p>

FUNCTION -----	VERB -----	
SET-TABS	TAB	<p>SET FILE TABS.</p> <p>CURRENT TABS ARE DISPLAYED ON THE BOTTOM LINE OF THE SCREEN.</p> <p>THE CHARACTER DISPLAYED IN THE FIRST COLUMN IS NOT PART OF THE TAB SETTINGS BUT IS THE ACTUAL TAB CHARACTER TO BE USED.</p> <p>WHEREVER ANY CHARACTER OTHER THAN A SPACE APPEARS ON THE REST OF THIS LINE SIGNIFIES A TAB SETTING.</p> <p>YOU MAY NOW MODIFY THE TAB DISPLAY AND PRESS ENTER. THE 'T' OPTION WILL BE SET.</p> <p>THE CHARACTER THAT YOU PUT IN THE FIRST COLUMN ALWAYS BECOMES THE NEW TAB CHARACTER</p> <p>THE TAB CHARACTER YOU USE FOR ASSEMBLER SOURCE FILES SHOULD BE CHOSEN CAREFULLY, AS IT IS TREATED AS A DELIMITER. ANY CHAR. WITH A HEXADECIMAL VALUE OF LESS THAN 2F IS SUITABLE, BUT NOTE THAT IF YOU USE ONE OF THE ARITH. OPERATOR CHARACTERS ITS ARITHMETIC FUNCTION IS NO LONGER APPLIED.</p> <p>A '/' CHARACTER IS IDEAL IF THE DIVISION FUNCTION IS NOT REQUIRED.</p> <p>(LOWER-CASE KEY, WELL PLACED ON KEYBOARD)</p>
QUIT	Q	EXIT EDITOR AND GO TO Z2 COMMAND MODE.

```
*****
*                                     *
*      Z2 ASSEMBLER                  *
*                                     *
*****
```

COPYRIGHT D.P.S.S. 1980

THE Z2 ASSEMBLER IS A HIGH SPEED TWO PASS ASSEMBLER. ALL ASSEMBLY ERRORS ARE REPORTED TO THE VDU SCREEN IRRESPECTIVE OF ASSEMBLER OPTIONS. UPON COMPLETION A STATUS MESSAGE WILL BE OUTPUT SHOWING :-

‘ 1111 2222 3333 4444 5555 EE ’	1111 = SYMBOL TABLE START ADDRESS
	2222 = SYMBOL TABLE END ADDRESS
	3333 = LAST ORG ADDRESS
	4444 = FINAL PROG. COUNTER ADDRESS
	5555 = FINAL OBJECT CODE ADDRESS
	EE = ASSEMBLER ERROR COUNT.

Z2 REQUIRES LINES OF SOURCE TEXT TO BE FORMATTED ACCORDING TO THE SYNTAX DEFINED IN THE ZILOG Z80 ASSEMBLY LANGUAGE PROGRAMMING MANUAL. EACH LINE OF TEXT MAY CONTAIN UP TO FIVE FIELDS THUS :-

TEST1: ADD A,7 ; STEP COUNT ER

THE EXAMPLE SHOWS ALL FIVE FIELDS...

A LABEL....	TEST1
AN INSTRUCTION MNEMONIC....	ADD
OPERAND 1....	A
OPERAND 2....	7
COMMENTS....	STEP COUNTER

SOME INSTRUCTIONS ONLY REQUIRE 1 OPERAND AND SOME REQUIRE NONE AT ALL. THE FIELDS MAY BE SEPARATED BY ONE OR MORE DELIMITERS. (COMMAS , SPACES , ETC .)

* LABELS *

THE LABEL FIELD OF A SOURCE TEXT LINE IS A MEANS OF REFERENCING THAT LINE FROM ELSEWHERE IN THE FILE.

IT IS A SYMBOLIC TAG THAT WILL ALWAYS BE ASSOCIATED WITH ITS PARTICULAR LINE OF SOURCE TEXT AND AS SUCH MUST OBVIOUSLY BE UNIQUE.

A LABEL WILL ALWAYS BE A SYMBOL AND THEREFORE FOLLOWS THE SYMBOL PROTOCOL (SEE SYMBOLS SECTION)

FOR Z2 A LABEL CAN START ANYWHERE IN A LINE IF POSTFIXED BY A COLON.

THE COLON IS NOT NECESSARY IF LABEL STARTED IN COLUMN 1 BUT WILL BE ACCEPTED.

A LABEL IS NORMALLY ASSIGNED THE CURRENT VALUE OF THE PROGRAM COUNTER DURING ASSEMBLY EXCEPT IN THE CASE OF SOME PSUDEO OPS.

AN EXAMPLE OF THE USE OF A LABEL MIGHT BE :-

```
LD HL,1000
XOR A
LOOPA: ADC HL,DE
LD A,L
CP 6
JP Z,LOOPA
HALT
```

IN THIS EXAMPLE THE SYMBOLIC LABEL 'LOOPA' WILL BE ASSIGNED THE VALUE OF THE PROGRAM COUNTER AT THE 'ADC' INSTRUCTION.

* INSTRUCTION OP-CODE MNEMONICS *

THE Z80 INSTRUCTION SET HAS 76 UNIQUE MNEMONICS (ADD,HALT,DJNZ,ETC.)
IN ADDITION TO THESE THERE ARE SPECIAL ASSEMBLY LANGUAGE MNEMONICS
WHICH DO NOT RELATE TO ANY OF THE Z80 INSTRUCTIONS AND ARE CALLED PSEUDO
OPS, WHICH MAY OR MAY NOT GENERATE OBJECT CODE. (SEE PSEUDO-OPS SECTION.)

* OPERANDS *

AS MENTIONED PREVIOUSLY THE NUMBER OF OPERANDS REQUIRED DEPENDS SOLELY ON THE PARTICULAR INSTRUCTION OP-CODE MNEMONIC.

E.G.	LDIR...	0 OPERANDS
	PUSH...	1 OPERAND
	LD...	2 OPERANDS
	CALL...	1 OR 2 OPERANDS

OPERAND FIELDS IN A LINE OF SOURCE TEXT MAY BE :-

A REGISTER MNEMONIC	E.G. HL
A CONDITION CODE	E.G. NZ
A CONSTANT	E.G. 6 (2400H) NEWLINE

THE REGISTER MNEMONICS AND CONDITION CODES ARE FIXED AND ARE DEFINED IN THE Z80 ASSEMBLY LANGUAGE PROGRAMMING MANUAL.

THE CONSTANTS HOWEVER MAY TAKE ONE OF SEVERAL FORMATS :-

A NUMBER

BECAUSE OF THE INCOMPATIBILITY OF NASCOM EDITOR ASSEMBLERS CURRENTLY AVAILABLE THE NUMBER CONSTANTS IN Z2 ARE THE MOST VARIED. Z2 SUPPORTS THE 3 BASIC NUMBER BASES IN COMPUTING. DECIMAL, HEXADECIMAL AND OCTAL.

ANY NUMBER CONSTANT WHEN PLACED IN A LINE OF SOURCE MUST START WITH A DIGIT IN THE RANGE 0—9 OR A £ SIGN. [**the pound sign**]

THE £ SIGN IMMEDIATELY DENOTES THE FOLLOWING CHARACTERS ARE HEXADECIMAL THE CONSTANT MAY BE POSTFIXED WITH A NUMBER BASE IDENTIFIER...

‘ H ‘ ALSO DENOTES HEXADECIMAL.

‘ O ‘ DENOTES OCTAL.

‘ T ‘ DENOTES DECIMAL. (SEE OPTIONS.)

THE ABSENCE OF A POSTFIX NUMBER BASE I.D. DENOTES THE DEFAULT BASE WHICH MAY BE HEXADECIMAL OR DECIMAL. (SEE ASSEMBLER OPTIONS)

E.G.	CP 0CAH	FECA	(HEX)
	CP 120	FEOA	(OCTAL)
	CP £CE	FECE	(HEX)
	CP 12	EE0C	(DECIMAL)

A LITERAL

ASCII LITERAL VALUES CAN BE USED AND MAY BE PREFIXED BY, OR ENCLOSED IN SINGLE OR DOUBLE QUOTES.

E.G.	LD A,"?"
	CP 'T'
	LD A,"A"
	CP 'Z'

A SYMBOL

A SYMBOL CAN BE USED TO REPRESENT NUMBERS ,MEMORY ADDRESSES OR THE PROG COUNTER THE DOLLAR SIGN, WHEN USED AS AN OPERAND, DEMOTES THE CURRENT VALUE OF THE PROGRAM COUNTER AT ASSEMBLY TIME.

E.G. LD A, COMMA
JP \$ - 7

A MEMORY ADDRESS

AN OPERAND STATEMENT CONTAINING A PAIR OF PARANTHESES DENOTES AN IMPLIED MEMORY ADDRESS AND THAT OPERAND WILL ALWAYS REEVALUATED AS 16 BIT. (UNLESS THEY ENCLOSE A REGISTER MNEMONIC E.G. (HL))
THE PARENTHESSES MAY ENCOMPASS SYMBOL OR NUMBER OPERANDS.

E.G. LD HL , (COUNT)
• LD HL , (1000H)

ARITHMETIC OPERATORS , .

A COMPLEX OPERAND STATEMENT OF THE PREVIOUSLY MENTIONED CONSTANT TYPES CAN BE MADE USING Z2'S ARITHMETIC OPERATORS...

+	DENOTES ADDITION
-	“ SUBTRACTION
&	“ LOGICAL AND
.	“ LOGICAL OR
*	“ MULTIPLICATION
/	“ DIVISION

AN OPERAND STATEMENT MAY CONTAIN ANY NUMBER OF INDIVIDUAL CONSTANT TYPES LINKED BY ARITHMETIC OPERATORS. CALCULATION IS UNSIGNED 16 BIT, OVERFLOW IS IGNORED AND EXPRESSION EVALUATION IS FROM LEFT TO RIGHT.

E.G. LID A,COUNTER-6/2 EVALUATES AS “COUNTER” MINUS 6 DIVIDED BY 2

AN OPERAND STATEMENT MAY ONLY BE PRECEDED BY THE “-“ OPERATOR DENOTING NEGATIVE CONSTANTS.

E.G. CP -6 FEFA

* SYMBOLS *

A SYMBOL MAY BE USED ANYWHERE IN A SOURCE FILE AND IS USED AS AN IDENTIFYING TAG FOR A PARTICULAR FIXED VALUE.

SYMBOLS ARE IDEAL WHEN A PARTICULAR VALUE OCCURS MANY TIMES IN THE SOURCE. IF A SYMBOL WERE USED TO REPRESENT A VALUE, SHOULD THAT VALUE REQUIRE SUBSEQUENT ALTERATION, IT WOULD ONLY BE NECESSARY TO ALTER THAT SYMBOL ONCE RATHER THAN SCAN THROUGH THE WHOLE FILE ALTERING EACH ITERATION OF THE VALUE.

THE FOLLOWING RESTRICTIONS APPLY TO SYMBOLS YOU MAY USE:-

IT MUST BEGIN WITH AN ALPHABETIC LETTER BUT MAY BE FOLLOWED BY LETTERS OR NUMBERS AFTER THAT.

THE SYMBOL CHARACTER STRING MAY BE ANY SIZE BUT THE FIRST SIX MUST BE UNIQUE.

UPPER AND LOWER CASE LETTERS ARE VALID.

THE SYMBOL MAY NOT BE ONE OF THE ALREADY ASSIGNED REGISTER MNEMONICS, CONDITION CODES OR INSTRUCTION MNEMONICS.

*** PSEUDO—OPS ***

THESE ARE SPECIAL MNEMONICS FOR THE ASSEMBLER WHICH APPEAR IN THE INSTRUCTION MNEMONIC FIELD OF A SOURCE STATEMENT.

NOTE THAT Z2 USES AN ABBREVIATED FORM OF THE DEFINE PSEUDO-OPS E.G. DB INSTEAD OF 'DEFB'. USE THE GLOBAL STRING COMMAND 'R/DEFB/DB/*' WHEN EDITING NON-Z2 FILES.

- END** THIS OP SIGNIFIES THE END OF SOURCE FOR ASSEMBLING.
IT IS NOT NECESSARY TO INCLUDE THIS ON A Z2 SOURCE FILE
AS THE END OF FILE MARKER ALSO TERMINATES ASSEMBLY.
NO OPERAND! ARE REQUIRED.
- ORG XXXX.** SETS THE INTERNAL ASSEMBLER PROGRAM COUNTER TO THE VALUE XXXX.
THIS IS NOT NECESSARILY WHERE THE SOURCE CODE WILL BE LOADED
(SEE THE LOAD PSEUDO-OP) .
THE 'ORG' STATEMENT RESETS THE OBJECT LOADING PROCESS, TO
CONTINUE LOADING OBJECT CODE A FURTHER 'LOAD' STATEMENT IS
REQUIRED. * NOTE 1
- LOAD XXXX** SETS THE OBJECT CODE POINTER TO ADDRESS XXXX.
THIS OP DEFINES WHERE THE ACTUAL Z80 OBJECT CODE WILL BE LOADED
IN MEMORY. UNLESS THE PROGRAM YOU WRITE IS RELOCATABLE IT WILL
NOT EXECUTE PROPERLY AT THIS ADDRESS UNLESS THE MEMORY ADDRESS
OPERAND YOU HAVE SPECIFIED IS THE SAME AS THE PRECEDING ORG
STATEMENT. * NOTE 1
- EQU X** THIS OP PROVIDES A CONVENIENT MEANS OF ASSIGNING A CONSTANT
VALUE X TO A SYMBOL. THE STATEMENT MUST CONTAIN A LABEL
THIS BEING THE SYMBOL THAT IS EQUATED.
E.G. NEWLINE EQU 0DH
* NOTE 1
- DS XX** THE DEFINE STORAGE PSEUDO-OP IS USED TO RESERVE AREAS OF
STORAGE IN THE OBJECT CODE. THE CONTENTS OF THE STORAGE IS
UNSPECIFIED AND WILL USUALLY BE USED AS A WORK AREA. (NOTE 1)
ONE OPERAND SPECIFYING THE NO. OF BYTES IS REQUIRED.
- DB X,Y,Z** THE DEFINE BYTE PSEUDO—OP IS USED TO PLACE A ONE BYTE CONSTANT
OR A SERIES OF ONE BYTE CONSTANTS DIRECTLY INTO STORAGE.
WHEN DEFINING A SERIES OF CONSTANTS EACH BYTE OPERAND MUST
BE SEPARATED BY A COMMA (MAX NO. OF BYTES IS 48)
E.G. DEFB 0,SPACE,48H,'T'
EACH OPERAND MAY BE ANY OF THE PREVIOUSLY DEFINED CONSTANT TYPE
BUT MUST NOT EXCEED FF HEX. WHEN EVALUATED.
A SPECIAL FORMAT OF THE DEFINE BYTE PSEUDO-OP IS AVAILABLE
TO ENABLE YOU TO DIRECTLY GENERATE A VARIABLE LENGTH ASCII
LITERAL STRING.
E.G. DEFB 'REWIND CASSETTE',NEWLINE
THE STRING MUST NOT CONTAIN ANY (DELIMITING) TAB CHARACTERS.
NOTE FROM THE EXAMPLE THAT MORE THAN ONE OPERAND IS PERMITTED
BUT THE FIRST OPERAND SHOWN (THE ASCII LITERAL STRING) MUST
BE PREFIXED, OR ENCLOSED IN, SINGLE OR DOUBLE QUOTES.
(IF ENCLOSED IN QUOTES THE QUOTE S MUST BE MATCHED.)
E.G. DEFB "PROGRAM HALT
 DEFB 'ENTER TOTAL',0
 DEFB "17'S HIGHER"

DW XXXX THE DEFINE WORD PSEUDO-OP GENERATES 2 BYTES IN RAM BUT IN REVERSE ORDER. THIS IS THE REQUIREMENT FOR THE 16 BIT LOAD INSTRUCTIONS OF THE Z80 INSTRUCTION SET. THE SINGLE OPERAND MAY BE ANY OF THE PREVIOUSLY DEFINED CONSTANT TYPES.
E.G. DEFW 0C80H
JUMPTO:DEFW START

LIST X THIS PSEUDO-OP GENERATES NO OBJECT CODE BUT IS USED TO CONTROL LISTING OUTPUT. THE SINGLE OPERAND X MAY BE EITHER :-
LIST 0 TURN LISTING OFF
LIST 1 TURN LISTING ON
WITH LISTING OFF ALL OUTPUT TO PRINTER OR VDU IS INHIBITED. (ASSEMBLY ERRORS ARE ALWAYS OUTPUT TO VDU IRRESPECTIVE)
THIS PSEUDO-OP IS PARTICULARLY USEFUL FOR OBTAINING PART LISTINGS OF A LARGE SOURCE FILE. E.G. PLACE A 'LIST 0' STATEMENT AT THE FRONT OF YOUR SOURCE FILE. THEN PLACE A 'LIST 1' AT THE START OF A PARTICULAR SUBROUTINE YOU WANT LISTED, AND A 'LIST 0' AT THE END OF THE SUBROUTINE.
WHEN ASSEMBLY IS COMPLETE THE ONLY LISTING TO APPEAR ON VDU AND/OR PRINTER IS THE REQUIRED SUBROUTINE.

LF XXXX THE 'LINK-FILE' PSEUDO-OP IS USED TO CONTINUOUSLY ASSEMBLE A GROUP OF FILES SCATTERED IN RAM. THE ADDRESS OPERAND XXXX IS THE START ADDRESS OF ANOTHER Z2 SOURCE FILE MODULE. THE 'LF' STATEMENT SHOULD BE THE LAST LINE OF EACH FILE EXCEPT THE VERY LAST SOURCE FILE WHICH SHOULD HAVE AN 'END' STATEMENT. THE ADDRESS POINTED TO BY THE 'LF' STATEMENT SHOULD CONTAIN THE LETTER 'F', IF IT DOES NOT A '*OPRND' ERROR WILL BE OUTPUT. DURING ASSEMBLY, OPTIONS, TABS ETC FOR EACH FILE WILL BE LOADED AND BECOME ACTIVE. OPTIONS MAY VARY FROM FILE TO FILE, HOWEVER IF THE ERROR-HALT OPTION IS SET IN A FILE THAT HAS AN ERROR A RETURN IS MADE TO THE EDITOR AND THAT FILE WILL BECOME THE CURRENT FILE. * NOTE 1

* NOTE 1 :- THESE PSEUDO-OPS HAVE OPERANDS REQUIRING IMMEDIATE RESOLUTION. ALL SYMBOLS USED MUST HAVE ALREADY BEEN DEFINED TO THE ASSEMBLER PRIOR TO THE STATEMENT.

* ASSEMBLER ERROR MESSAGES *

PASS 1 ERROR MESSAGES.

‘*LABEL’ AN INVALID LABEL CHARACTER HAS BEEN USED.

‘*DBL LABEL’ A LABEL HAS BEEN USED TWICE IN THE SAME SOURCE FILE.

PASS 1 ERROR MESSAGES DO NOT APPEAR IN LISTINGS. THEY ARE
CLASSED AS REQUIRING IMMEDIATE CORRECTION BEFORE PROPER ASSEMBLY
CAN TAKE PLACE. THE ERROR HALT OPTION IS FORCED ON.

PASS 2 ERROR MESSAGES.

‘*SYNTAX’ THE FORMAT OF THE SOURCE LINE DOES NOT COMPLY WITH THE SYNTAX
DEFINED IN THE Z80 PROGRAMMING MANUAL.
E.G. LDIR HL
 LD (HL,B

‘*SIZE’ THE OPERAND WHEN EVALUATED IS TOO BIG.
E.G. LD A,1200H
 DEFB 324,0,’T’,O
 LD A,(IX+142)
 JR LOOPA (THE DISPLACEMENT OF LOOPA WHEN EVALUATED
 IS OUT OF RANGE)

‘*OPRND’ AN OPERAND STATEMENT IS INVALID OR CANNOT BE EVALUATED.
E.G. LD (SAVE),B
 ID HL,08R4 (INVALID DIGIT)

‘*UNDEF’ EITHER A SYMBOL OR AN INSTRUCTION OP-CODE MNEMONIC IS UNDEFINED.
E.G. BEGIN:CR 6

‘*RESVD’ A RESERVED LABEL OR SYMBOL HAS BEEN USED.
E.G. HALT:LD HL,2000H
 ADD A,BIT

* OTHER ERROR MESSAGES *

'TBL' THE SYMBOL TABLE WORK AREA POINTER (RAM 0D60, SEE 'M' COMMAND
HAS BEEN SET :- -
1) TO AN ADDRESS OUTSIDE ADDRESSABLE RAM.
2) TO AN ADDRESS LOWER THAN THE END OF FILE ADDRESS OF THE
CURRENT FILE BEING ASSEMBLED.
(FOR FILE SAFETY Z2 REQUIRES THE SYMBOL WORK AREA, WHICH
IS OF AN INDETERMINABLE SIZE, TO BE LOCATED HIGHER IN RAM
THAN THE CURRENTLY ASSEMBLED SOURCE FILE. WHEN LINKING
SOURCE FILES USING THE 'LF' PSEUDO-OP ENSURE THIS WORK
AREA IS SET HIGHER' THAN ALL THE FILES)
3) DURING ASSEMBLY THE SYMBOL WORK AREA HAS OVERFLOWED INTO
- ' . NON—ADDRESSABLE RAM.

ANY OF THESE CONDITIONS WILL FORCE AN IMMEDIATE RETURN TO
Z2 COMMAND MODE TO ENABLE A NEW WORK AREA TO BE SET USING
THE 'W' COMMAND.

* ASSEMBLER OPTIONS *

WHEN THE Z2 COMMAND 'A.' OR 'O' IS ENTERED A LIST OF OPTIONS IS DISPLAYED EVERY Z2 FILE HAS ITS OWN OPTION SETTINGS WHICH ARE SAVED WITH THE FILE TO CASSETTE ONCE SET THEY REMAIN ON UNTIL THE 'Z' OPTION INPUT IS USED TO CLEAR THEM.

OPTION -----	I.D. -----	COMMENTS. -----
VDU	V	SENDS LISTING/PRINTFILE OUTPUT TO VDU SCREEN.
PRINT	P	SENDS LISTING/PRINTFILE OUTPUT TO A PRINTER ON THE PARALLEL PORT (PIO). WHEN PRINTING FILES OUTPUT INCLUDES ALL FILE HEADER DATA. IF THE 'T' TAB OPTION IS SET OUTPUT WILL CONTAIN ALL NECESSARY TABBING SPACES.
SERIAL	X	SENDS PRINTFILE/LISTING OUTPUT TO THE SERIAL (UART) PORT. A SERIAL PRINTER OR CASSETTE MAY BE CONNECTED TO THIS PORT. FILE DATA OUTPUT IS A CONTINUOUS STRING OF CHARACTERS UNLIKE THE OUTPUT WHEN USING THE SAVE COMMAND WHICH PUTS DATA INTO 256 BYTE BLOCKS. THE DRIVE LED IS TURNED ON 1 SEC. BEFORE ANY OUTPUT TAKES PLACE AND IS ALSO TURNED OFF 3. SEC. AFTER OUTPUT IS COMPLETE. (THIS OPTION COULD BE USED WHEN EDITING BASIC FILES FOR SAVING THEM TO CASSETTE)
SYMLIST	S	SPECIFIES WHETHER A FULLY SORTED SYMBOL LIST IS TO BE OUTPUT UPON ASSEMBLY COMPLETION.
ERRSTOP	E	WHEN SET, SPECIFIES THAT UPON THE FIRST ASSEMBLY ERROR THE EDITOR IS TO BE INVOKED AND THE LINE IN ERROR WILL BECOME THE PROMPTED LINE. ('P' OR 'X' SHOULD NOT BE SET)
HEX	H	DEFAULT TO HEXADECIMAL NUMBERING SYSTEM. (DECIMAL NUMBERS TO HAVE A POSTFIXED LETTER 'T')
TAB	T	SET TABS ACTIVE. (THIS OPTION IS AUTOMATICALLY SET BY THE EDITOR TAB' COMMAND)
ZERO	Z	RESET ALL OPTIONS

MERGE M THIS OPTION ALLOWS YOU TO ASSEMBLE MULTIPLE SOURCE FILES WHICH CROSS REFERENCED SYMBOLS Z2 DOES NOT RESET THE INTERNAL PROGRAM COUNTER UNLESS AN ORG STATEMENT IS ENCOUNTERED. IT IS THEREFORE POSSIBLE TO KEEP ASSEMBLING SOURCE FILES WITH ONLY AN ORG STATEMENT IN THE FIRST FILE. OBJECT CODE WILL BE GENERATED FOR ASCENDING ADDRESSES AND THE SYMBOL TABLE WILL GROW WITH EACH ASSEMBLY. THIS FEATURE ENABLES YOU TO WORK VERY LARGE PROGRAMS WHILST ONLY HAVING TO ASSEMBLE AND EDIT ONE SUBROUTINE OF THAT PROGRAM.

WHEN ASSEMBLY IS COMPLETE SHOULD THERE HAVE BEEN ANY ERRORS DETECTED THE INTERNAL ASSEMBLER PROGRAM COUNTER WILL BE RESET TO ITS INITIAL VALUE UPON COMMENCEMENT OF ASSEMBLY.

*** NOTE *** WHILST THE MERGE OPTION IS ACTIVE CHECKING OF DOUBLE--SYMBOLS IS DISABLED. THIS IS THE ONLY WAY THIS FEATURE CAN WORK, HOWEVER WITH CAREFUL SELECTION OF SYMBOLS IT SHOULD NOT BE A PROBLEM.

FOR FASTEST ASSEMBLY RESET 'Z' ALL OPTIONS (TABS 'T' MAY BE SET)
ONLY ERRORS WILL BE LISTED ON THE VDU SCREEN.

* ASSEMBLER LISTINGS *

ASSEMBLER LISTING OUTPUT IS PASSED TO THE OUTPUT DRIVER ROUTINE WHICH IS PART OF THE 2K EDITOR. THE 'V' OPTION SPECIFIES THAT LISTING OUTPUT IS TO GO TO THE VDU SCREEN. (NOTE ERROR MESSAGES ARE ALWAYS OUTPUT TO THE VDU). VDU LISTING OUTPUT THAT EXCEEDS 48 BYTES IS TRUNCATED (COMMENTS FIELD) TO KEEP THE DISPLAY TIDY. IT IS POSSIBLE USING TABS TO TAB ALL COMMENTS COMPLETELY OFF THE YOU SCREEN.

WHEN EACH SCREEN OF LISTING IS FULL A MESSAGE 'MORE...' WILL APPEAR AT THE BOTTOM R.H. CORNER OF THE DISPLAY. THIS SAYS THAT MORE OUTPUT IS AVAILABLE AND TO CLEAR THE SCREEN YOU MAY PRESS ANY KEY (EXCEPT ENTER). IF ENTER. IS PRESSED AT ANY TIME TO THE 'MORE...' DISPLAY AN IMMEDIATE EXIT IS MADE FROM THE ASSEMBLER.

IF THE 'P' OPTION IS SET LISTING OUTPUT WILL BE SENT TO THE PIO PRINTER. ANY INFORMATION TABBED OFF THE SIDE OF THE VDU SCREEN WILL BE PRINTED CORRECTLY POSITIONED. (IF THE PRINTER HAS MORE THAN 48 PRINT COLUMNS) THE 'V' OPTION MAY BE SET WITH THE 'P' OPTION BUT THE VDU SCREEN WILL CLEAR AUTOMATICALLY WHEN FULL. THE SAME COMMENTS APPLY TO THE 'X' OPTION IF SET.

OBJECT CODE OF MORE THAN 4 BYTES, GENERATED FROM ONE LINE STATEMENT, WILL PRODUCE PRECEDING LINES OF LISTING DETAILING ALL HEXADECIMAL BYTES AND PROG COUNTER VALUES.

Z2 NORMALLY EXECUTES AT 1000 HEX.

DEPENDING ON THE WORK YOU ARE ABOUT TO PERFORM USING Z2, THE FOLLOWING TWO RAM LOCATIONS MAY REQUIRE INITIALIZATION :-

- D09 REPEAT KEY RATE (FF IS FASTEST 01 SLOWEST, D0 IS ABOUT THE AVERAGE RATE)
- D60 TWO BYTE SYMBOL TABLE WORK AREA POINTER (ADDRESS STORED WITH LO—ORDER BYTE FIRST)

CREATING NEW FILES :-

AFTER INITIAL EXECUTION OF Z2 (EITHER AFTER LOADING OR HAVING PRESSED) RESET THE FILE POINTER IS SET TO Z2'S INTERNAL 'NEW' FILE IMAGE.

ENTERING THE LIST COMMAND WILL SHOW WHERE THIS IS LOCATED IN RAM.

THE FIRST COMMAND YOU SHOULD ENTER TO Z2 IS 'COPY' SPECIFYING WHERE YOU WANT YOUR FIRST FILE TO BE CREATED IN RAM.

THIS COMMAND COPIES Z2'S 'NEW' FILE IMAGE TO THE ADDRESS YOU HAVE SPECIFIED AND THIS DUPLICATE FILE BECOMES THE CURRENT FILE.

(THIS COPY PROCEDURE MAY NOT BE NECESSARY IF Z2 IS NOT IN ROM. AS THE 'NEW' FILE IMAGE CAN BE USED TO CREATE A FILE)

NOW EDIT THE FILE USING THE EDIT COMMAND. E.G. 'E TEST1'

YOU CAN NOW CREATE/MODIFY THE FILE WHILST IN EDIT MODE.

AT A SUITABLE STAGE OF DEVELOPMENT YOU MAY USE THE 'FILE' COMMAND TO PUT A TEMPORARY COPY OF THE PROG ONTO TAPE.

ALTERNATIVELY IF YOU HAVE SUFFICIENT RAM YOU MAY USE THE COPY COMMAND TO MAKE A BACK—UP DUPLICATE OF THE FILE IN RAM WHILST YOU GO ON TO MAKE FURTHER ALTERATIONS TO THE FILE. (ALWAYS KEEP A LIST OF RAM ADDRESSES OF YOUR FILES AND ENSURE THAT ANY ADDITIONS TO THE CURRENT FILE DO NOT OVERLAP OTHER FILES) IDEALLY AFTER USING THE 'COPY' OR 'FILE' COMMAND YOU SHOULD NOW GIVE YOUR CURRENT FILE A NEW FILE A NAME AS YOU EDIT IT E.G. 'E TEST2'

Z2 IS DESIGNED TO EDIT ANY TEXT FILES NOT NECESSARILY ASSEMBLER SOURCE FILES (E.G. LETTERS , DOCUMENTS ETC.) HENCE THE MANY OUTPUT OPTIONS. THE OUTPUT DRIVE ROUTINE IS DESIGNED TO RUN EITHER :—

- 1 THE STANDARD NASCOM VDU SCREEN IN ITS CURRENT FORMAT.
- 2 A PRINTER OR DEVICE CONNECTED ON TO PIO PORTS A AND B IN THE FOLLOWING MANNER :—
 - PORT A - 8 BITS OF ASCII DATA ACTIVE—HIGH.
 - PORT B - BIT 7 IS A RETURN SIGNAL FROM THE PRINTER INDICATING PRINTER BUSY WHEN ACTIVE—HIGH.
 - BIT 6 IS A DATA STROBE SIGNAL ACTIVE—HIGH.
- 3 ANY DEVICE CONNECTED TO THE NASCOM SERIAL I/O PORT (UART)
(DATA RATE SELECTABLE USING NASCOM SWITCHES)

UPDATING OLD FILES :—

TO UPDATE OLD FILES ALL. THAT IS NECESSARY IS TO READ THE FILE INTO RAM FROM CASSETTE. YOU MAY COPY IT ELSEWHERE IN RAM OR LEAVE AS IS.

NOW USE THE SETFILE ADDRESS COMMAND TO SET FILE POINTER TO THE FIRST RAM LOCATION WHERE YOUR FILE IS LOCATED. JUST TYPE 'E' FOR EDIT' AND YOUR ORIGINAL FILENAME,TABS,OPTIONS ETC. WILL BE ACTIVATED.

CONVERTING OTHER FILES TO Z2 FILE MODULES.

TO CONVERT OTHER FILES TO Z2 FORMAT YOU MUST FIRSTLY REMOVE ANY HEX LINE NUMBERS FROM THE FRONT 2 BYTES OF EACH LINE OF SOURCE.
THE NEWLINE CHARACTER FOR Z2 IS 0D HEX AND CONSEQUENTLY ANY 1F HEX CHARACTERS WILL HAVE TO BE CONVERTED.

THERE SHOULD NOT BE ANY HEX. C0 , A0, OR 00 CHARACTERS IN THE FILE.

A SIMPLE PROG CAN BE WRITTEN USING Z2 TO CONVERT THE FILE.

ALL THAT REMAINS TO BE DONE NOW IS TO APPEND THE FILE CONTROL INFORMATION TO THIS FILE DATA.

WHEN Z2 IS INITIALLY EXECUTED THE FILE—POINTER IS SET TO Z2'S INTERNAL 'NEW' FILE IMAGE.

THE LIST COMMAND 'L' WILL SHOW WHERE IN RAM IT IS LOCATED.

YOU MUST NOW APPEND THE CRT IMAGES AND FILE CONTROL (TABS,NAME ETC.) TO YOUR DATA FILE.

E.G. IF YOUR NON—Z2 FILE IS LOCATED AT 3000 THRU 4800 HEX
Z2 NEW FILE IMAGE IS AT E.G. 2500 THRU 2575 HEX
CALCULATE THE FILE ADDRESS 3000—65 HEX = 2F9B
USING NAS-SYS COPY COMMAND ENTER 'C 2500 2F9B 65'
(APPENDED DATA LENGTH ON THE FRONT OF Z2 FILES IS 65 HEX BYTES)
CALCULATE THE STARTING ADDRESS OF EOF CRT IMAGE 2575—10 HEX = 2565
NOW COPY THE EOF CRT IMAGE USING NAS-SYS 'C 2565 4800 10'
(APPENDED DATA LENGTH ON THE END OF Z2 FILES IS 10 HEX BYTES)
YOU NOW HAVE A Z2 FILE MODULE LOCATED AT 2F9B THRU 4810 HEX
ALL THAT IS REQUIRED NOW IS TO SET THE CORRECT END OF FILE POINTER
ADDRESS. (CONTAINED WITHIN THE FILE MODULE)
THE END OF FILE ADDRESS IS 4810 HEX (SEE ABOVE)
THIS ADDRESS MUST BE STORED (LO-ORDER BYTE FIRST) AT 2F9B+1A HEX
= 2FB5.

IF YOU NOW SET THE FILE POINTER TO THIS FILE 'S 2F 9B' Z2 SHOULD
LIST THE FILE AS '2F9B 4810 FILE:— NEW'

YOU MAY NOW EDIT THE FILE AND NAME IT E.G. 'E STARTREK'

IT IS WISE TO SAVE THE NEWLY CREATED MODULE TO CASSETTE BEFORE
PROCEEDING ANY FURTHER.

THE 'COPY' COMMAND COULD BE USED TO RE-POSITION THE FILE ON A 1K
BOUNDARY, E.G. 'C 2600'

USING THE MERGE OPTION.

THE MERGE OPTION SHOULD NOT BE SET UNLESS YOUR PROGRAM IS SPLIT INTO SMALLER FILE MODULES.

TO USE THE MERGE OPTION CORRECTLY YOU MUST FIRST BUILD A COMPLETE SYMBOL TABLE FOR YOUR PROGRAM BY ASSEMBLING EACH INDIVIDUAL MODULE WITHOUT THE ERROR HALT OPTION. MODULES USING SYMBOLS WHICH ARE DEFINED IN SUBSEQUENTLY ASSEMBLED MODULES WILL OUTPUT THE '*UNDEF' ERROR MESSAGE.

WHEN ALL MODULES HAVE BEEN ASSEMBLED THE SYMBOL TABLE SHOULD NOW HOLD ALL SYMBOLS USED. RE-ASSEMBLY OF ALL MODULES SHOULD NO-LONGER PRODUCE ANY '*UNDEF' MESSAGES. YOU MAY NOW WORK ON ONE PARTICULAR MODULE AND IGNORE ALL OTHER FILE MODULES, AS LONG AS THE SYMBOL TABLE REMAINS INTACT.

IF YOU ARE NOT WORKING ON THE VERY LAST MODULE OF THE PROGRAM THEN REMEMBER THAT ANY ADDITIONS/ALTERATIONS TO THIS MODULE WILL AFFECT THE PROG-COUNTER VALUE OF LATER MODULES. CONSEQUENTLY WHEN YOU HAVE COMPLETED ALL CHANGES AND THE MODULE IS ASSEMBLING ERROR FREE YOU MUST RE-ASSEMBLE ANY MODULES THAT COME AFTER THE CHANGED MODULE TO GET THE NEW PROGRAM COUNTER VALUES PUT INTO THE SYMBOL TABLE.

NOW FINALLY ASSEMBLE ALL FILE MODULES TO GET 'THE NEW CORRECT OBJECT CODE. THIS PROCEDURE MAY SEEM COMPLICATED BUT THE BENEFITS ARE :-

1) SHOULD A USER ONLY HAVE 4K OF AVAILABLE FILE RAM HE MAY CREATE AS MANY SEPARATE 4K MODULES OF A LARGE PROGRAM AS HE DESIRES. ONCE THE ACCUMULATIVE SYMBOL TABLE HAS BEEN BUILT HE MAY WORK ON JUST ONE MODULE.

2) CONSIDERABLE TIME IS SAVED WHEN WRITING ONE PARTICULAR MODULE TO CASSETTE (IT IS RECOMMENDED THAT EACH MODULE BE ON A SEPARATE CASSETTE TAPE UNLESS THE CASSETTE DRIVE HAS A TAPE POSITION COUNTER).

3) VERY LARGE SOURCE FILES MAY BE BROKEN DOWN INTO MODULES .
E.G. 64K SOURCE INTO 8 OFF 8K MODULES.

NOTE THAT IF YOU WISH ALL FILE MODULES TO GENERATE OBJECT CODE FOR ASCENDING ADDRESSES THE FIRST MODULE ONLY SHOULD HAVE A NORMAL 'ORG' STATEMENT. ALL SUBSEQUENT FILE MODULES MUST HAVE 'ORG \$' STATEMENTS AS THE FIRST LINE. IT IS PERMITTED TO PUT NORMAL 'ORG' STATEMENTS IN SUBSEQUENT FILE MODULES IF CONTIGUOUS ADDRESSING IS NOT DESIRED BUT BE CAREFUL THAT OBJECT CODE ADDRESSES DO NOT OVERLAP.

NOMENCLATURE.

BYTE	8 DATA BITS
FILE	A BLOCK OF DATA. (ASCII—CHARACTERS, HEX CHARACTERS ,ETC.)
SOURCE FILE	BYTES OF HEXADECIMAL Z80 MACHINE CODE.
OBJECT FILE	FINAL OUTPUT AFTER ASSEMBLY OF A SOURCE FILE SHOWING BOTH
LISTING	HEXADECIMAL PROGRAM-COUNTER OBJECT CODE AND ALSO SOURCE LINES.
SYMBOL TABLE	TABLE OF ALL SYMBOLS USED IN A SOURCE FILE PLUS THEIR ASSOCIATED
	VALUES.

* NOTES ON USING Z2 *

Z2 FILE MODULES ARE LAID OUT IN RAM IN THE FOLLOW FORMAT :-

ADDRESS	+0	FILE NAME AND HEADER.	+00H
	+26	END OF FILE POINTER	+1AH
	+28	FILE FLAGS	+1CH
	+29	CURRENT FILE OPTIONS.	+1DH
	+30	FILE TAB CHARACTER	+1EH
	+31	FILE TAB SETTINGS	+1FH
	+79	END OF TAB SETTINGS	+4FH
	+80	TOP OF FILE CRT IMAGE	+50H
	+101	FILE DATA	+65H
	+XXX		

END OF FILE	-16	END OF FILE CRT IMAGE.	-10H
-------------	-----	------------------------	------

THE TOP-OF-FILE AND END-OF-FILE CRT IMAGES ARE NECESSARY FOR THE EDITOR TO WORK PROPERLY. ANY FILE NOT HAVING THESE CRT IMAGES WILL NOT BE SUITABLE. FOR EDITING USING Z2.

(SEE SECTION ON CONVERTING FILES TO Z2 FORMAT)