REZ.COM Quick Reference

R Read file. Entered as Rxxx.COM, Rxxx.SYM or Rxxx.CTL. This must be done after starting REZ.com itself. Don't make command-line specifications. Start up REZ First and then tailor your session.

S Save file. Entered as Sxxx.ASM, Rxxx.SYM or Rxxx.CTL. The response to entering the Save command for the CTL or SYM file will occur immediately, but the "saving" of the ASM file is actually a generating process.
When you enter "Sxxx.ASM", you will open up the ASM file, but you must then execute the List command, in the form of Ls,e to actually put the generated assembler code to the file. When the List command completes, enter "Z" at the command prompt to close the ASM file.

X (Y/N)Purge SYM & CTL tables ?
This command is useful in CTL data building sessions where it's important to get rid of bunch of symbols, but you want to keep the Ctl formatting. Save CTL, Purge all, Read CTL and then re-Build SYM

T Toggle TRIM On/Off (On = Labels appear on instruction, as opposed to being on their own line.

O Set Offset to where code will actually be loaded, and subsequently how much space you've allocated to the CTL table. Better a bit too much offset than to overwrite the front end of your code. The default offset is 4000H, which is good, but for most code disassemblies, I find o6000h to be a better bet.

? Dump stats

B Build code (Also entered as Bs,e where s = starting addr and e = End addr, in hex digit (0,1,2,3…,A,B,C,D,E,F).

L List code (Also entered as Ls,e where s = starting addr and e = end addr, in hex digit (0,1,2,3…,A,B,C,D,E,F).
Also used after entering the Save command for the .ASM file to cause rez to build data to the file.

A Assemble code. Similar to the List command. (Also entered as As,e where s = starting addr and e = end addr, in hex digit (0,1,2,3…,A,B,C,D,E,F).

D Dump storage to console (Also entered as ds,e where s = starting addr and e = end addr, in hex digit (0,1,2,3…,A,B,C,D,E,F).

C CTL table list or entry. Using just the C char on the command line will cause the CTL table stats to be displayed.
You can toggle the definition of an area my entering your own CTL entry by entering Ca,f where a=addr in code and f=function. Functions supported are…
I – Start instruction string
B – start of DB, single byte format
W – start of word format data.
S – Begining at addr, make data DS (existing data ignored)
E – make end at addr
K – kill CTL entry for addr

SAMPLE SESSION

```
A0>rez      ←=========================Use this to start REZ
RESOURCE by Ward Christensen
Z-80 - Zilog Version 4.1
Restart 103h!!
19.03.85
Memory open to ECFF

<*>o6000   ←========================== don't use default of 4000, set to 6000
<*>?     ←========================== ask for status display
SYMTBL=2800 2800
PC    =0100
OFFSET=6000
MEMTOP=ECFF
CTLTBL=2100 2100
<*>rrez.com  ←======================== now load code,
7F80 1F80   ←======================== code actually at 6000-7F80,
<*>D0100    ←======================== show me dump, start at 0100H
0100: C3 06 01 C3 B6 01 AF 32 FB 1F 32 FD 1F CD 6F 01 .......2..2...o.
0110: 52 45 53 4F 55 52 43 45 20 62 79 20 57 61 72 64 RESOURCE by Ward
0120: 20 43 68 72 69 73 74 65 6E 73 65 6E 0D 0A 5A 2D  Christensen..Z-
0130: 38 30 20 2D 20 5A 69 6C 6F 67 20 56 65 72 73 69 80 - Zilog Versi
0140: 6F 6E 20 34 2E 31 20 0D 0A 52 65 73 74 61 72 74 on 4.1 ..Restart
0150: 20 31 30 33 68 21 21 20 0D 0A 31 39 2E 30 33 2E  103h!! ..19.03.
0160: 38 35 24 FD E5 DD E5 CD 05 00 DD E1 FD E1 C9 D1 85$.............
0170: 0E 09 CD 63 01 21 00 00 39 22 FE 20 31 FE 20 CD ...c.!..9". 1. .
0180: A3 1E 0D 0A 4D 65 6D 6F 72 79 20 6F 70 65 6E 20 ....Memory open
0190: 74 6F 20 00 2A 06 00 2E 00 2B 22 75 1F CD C4 1C to .*....+"u....
01A0: CD 04 1E CD 04 1E 21 00 00 22 79 1F CD B7 1E AF ......!.."y.....
01B0: 32 66 1F 32 68 1F 31 FE 20 CD 5F 1E 11 7D 1F 1A 2f.2h.1. ._..}..
01C0: FE 0D 20 0F 3A 38 1F CD F2 01 20 E3 F5 3E 0D 32 .. .:8.... ..>.2
01D0: 7E 1F F1 21 FE 01 01 13 00 ED B1 20 5A F5 CD F2 ~..!....... Z...
01E0: 01 28 02 3E 2A 32 38 1F F1 09 09 09 4E 23 46 EB .(.>*28.....N#F.
01F0: C5 C9 FE 4C C8 FE 44 C8 FE 42 C8 FE 41 C9 3B 41 ...L..D..B..A.;A
<D>L100,112  ←==========list code, instr format starting at 0100h
0100         JP     0106H
0103         JP     01B6H
```

```
0106          XOR    A
0107          LD     (1FFBH),A
010A          LD     (1FFDH),A
010D          CALL   016FH
0110          LD     D,D  ←====from dump, start of constant-in-storage
0111          LD     B,L
<L>C100,I  ←=============== we do this to be unambiguous
<*>c110,b  ←=============== flip over to DB generation
<*>L100    ←=============== now let's see what we got
0100          JP     0106H
0103          JP     01B6H
0106          XOR    A
0107          LD     (1FFBH),A
010A          LD     (1FFDH),A
010D          CALL   016FH      ←===== looks like addr where code resumes
0110          DB     'RESOURCE by Ward Christensen'
012C          db     0DH,0DH
012E          DB     'Z-80 - Zilog Version 4.1 '
0147          db     ODH,OAH
0149          DB     'Restart 103h!!'
0158          db     ODH,OAH
015A          DB     '19.03.85$'
0163          db     0FDH,0E5H,0DDH,0E5H,0CDH,0DDH
016B          db     0E1H,0FDH,0E1H,0C9H
016F          db     0D1H,0EH,9,0CDH
<*>C016F,I  ←=====set a flip back to instruction-stream mode
<*>L100  ←=======let's look again and see what we got
0100          JP     0106H
0103          JP     01B6H
0106          XOR    A
0107          LD     (1FFBH),A
010A          LD     (1FFDH),A
010D          CALL   016FH
0110          DB     'RESOURCE by Ward Christensen'
012C          db     0DH,0DH
012E          DB     'Z-80 - Zilog Version 4.1 '
0147          db     ODH,OAH
0149          DB     'Restart 103h!!'
0158          db     ODH,OAH
015A          DB     '19.03.85$'
0163          db     0FDH,0E5H,0DDH,0E5H,0CDH,0DDH
016B          db     0E1H,0FDH,0E1H,0C9H
016F          POP    DE  ←========this looks good, but…
0170          LD     C,9
0172          CALL   0163H   ←=====looks like code starts at 0163H
0175          LD     HL,0
0178          ADD    HL,SP
<L>c163,I ←==================Say that the code starts earlier than 016F
<*>L100  ←=======show product
0100          JP     0106H
0103          JP     01B6H
0106          XOR    A
0107          LD     (1FFBH),A
010A          LD     (1FFDH),A
010D          CALL   016FH
0110          DB     'RESOURCE by Ward Christensen'
012C          db     0DH,0DH
012E          DB     'Z-80 - Zilog Version 4.1 '
0147          db     ODH,OAH
0149          DB     'Restart 103h!!'
0158          db     ODH,OAH
015A          DB     '19.03.85$'
0163          PUSH   IY
```

```
0165          PUSH   IX
0167          CALL   5
016A          POP    IX
016C          POP    IY
016E          RET
016F          POP    DE
<L>
:
:
<*>B100,2100  ←= code and constants mapped, now B to "build" symbols table
                  but extend upper address, here beyond where the program
                  code ended at 1F80H. This allows REZ to account for
                  areas beyond the actual end of the code where lousy
                  programming practices allow for this type of mischief.
:
a flurry of screen activity ensues…
:
<*>L100  ←=======let's look again and see what we got
0100  t0100: ;0100  ←== this and many other symbolics gen'd…
              JP     0106H
0103          JP     01B6H
0106  j0106: ;0106
                     XOR    A
0107          LD     (1FFBH),A
010A          LD     (1FFDH),A
010D          CALL   016FH
0110          DB     'RESOURCE by Ward Christensen'
012C          db     0DH,0DH
012E          DB     'Z-80 - Zilog Version 4.1 '
0147          db     ODH,OAH
0149          DB     'Restart 103h!!'
0158          db     ODH,OAH
015A          DB     '19.03.85$'
0163  c0163:        ;0163
                     PUSH   IY
0165          PUSH   IX
0167          CALL   c0005  ;5
016A          POP    IX
016C          POP    IY
<*>T  ←==================toggle trim ON
Trim On ←===this means put symbolics in same line as assembly code
<*>L100
0100  t0100: JP     0106H   ←===== Name-on-instr-line, from TRIM
0103          JP     01B6H
0106  j0106: XOR    A  ←==== same here
0107          LD     (1FFBH),A
010A          LD     (1FFDH),A
010D          CALL   016FH
0110          DB     'RESOURCE by Ward Christensen'
012C          db     0DH,0DH
012E          DB     'Z-80 - Zilog Version 4.1 '
0147          db     ODH,OAH
0149          DB     'Restart 103h!!'
0158          db     ODH,OAH
015A          DB     '19.03.85$'
0163  c0163: PUSH   IY
0165          PUSH   IX
0167          CALL   c0005  ;5
016A          POP    IX
016C          POP    IY
016F  c016F: POP    DE
<L>?  ←==================just to see how the stats changed
SYMTBL=2800 3428
```

```
PC    =0170
OFFSET=6000
MEMTOP=ECFF
CTLTBL=2100 2220
<*>srez.ctl ←===== save our CTL table. It's worth the effort.
<*>srez.asm ←===== we sure want to do this !
++WRITING .ASM ENABLED
USE Z COMMAND OR E CONTROL TO CLOSE FILE++
<*>L100,2100
:
another flurry of screen activity as the code is "listed" to the .ASM file.
:
<*>Z
2100                END
++ ASM FILE CLOSED++
<*>^C  ←==========bail out here
A0>REN REZ.MAC=REZ.ASM  ←====rename output so Macro-80 can digest
:
:
; modify code to add ".z80" header to enable z80 compiles
; and an org statement, starting at 0100h to make the comp[iled
; output match up in comparison to the symbolic names
;
A0>M80  ←===start up m80 first, otherwise we might see "invalid machine
             opcode" abend under MyZ80. I suspect it's a parity setting
             discrepancy between the 8080 code and the z80 engine. This
              seems to bypass the issue.

*=REZ/M/L  ←====this will generate listing to .PRN file

*^C  ←=== bail out of Macro-80
A0>
```

Enjoy !

Mark E Sharafinski
MarkSrph@aol.com