Century Systems Bar Code Printer

Century Eagle 4,
Century Eagle 5
Basic Interpreter White Paper

First Edition: July, 2004

Century Systems, Inc.

Copyright Century Systems, Inc. 2004
All Rights Reserved

Using the Basic Interpreter Program Loader

Explanation of uses for the Basic Interpreter

The basic interpreter is used when an incoming data stream is difficult, or impossible, to change to accommodate a new printer, or to make a change to an existing label design. The basic interpreter is useful for allowing the Century Eagle 4 to be a "drop in" replacement without having to change the data stream. This might occur if an end user wants to replace a Zebra 105SL with a Century Eagle 4 printer or an Eagle 4 printer is purchased to be added to replace one of the existing Zebra printers. In both cases the data that is currently being used by the printer does not have to be changed because the data is changed in the interpreter. In the second case, the interpreter allows large installations to replace one printer at a time without having to change the code globally for all printers.. If the basic program is written for the label formats, there would be no required change for the user. The interpreter is useful for any data stream that has a known start and stop character. In the attached Zebra example the start character is "^." If a TEC B-402 data stream were used, the start character would be "{."

The attached basic program takes the attached Zebra label and makes it print on the Century Eagle 4 printer "on the fly." When the data contained in the label changes the printer will still interpret the ZPL program. The attached program is customized to the Zebra label format. If the format were to change, i.e. add a text field, the interpreter will not function correctly unless the program is changed to accommodate the additional text field.

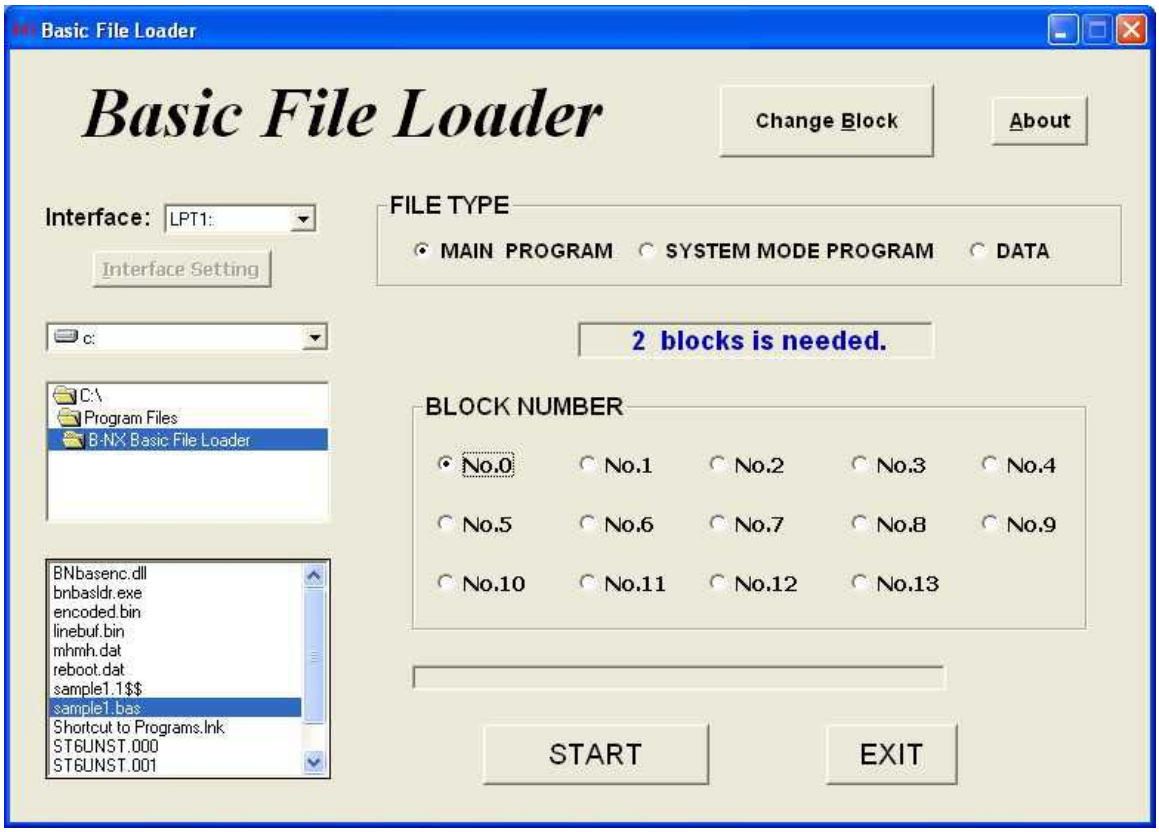
NOTE:

- All of the basic functions in the printer need to be OFF.
- All Basic programs need to have the ".bas" extension.
- DO NOT interrupt the Loader Program while it is loading the program to the printer.
- Program loader v1.0 or better is required.
- Flash the printer buffer before loading the basic program for at least as many blocks as are needed for the basic program.
- All programs go in Block 0.
- Labels for loops in the programming language (example *cmd_check) may not be greater then 12 characters long.
- Restart the program loader each time there is a change in the file to be loaded.

Copyright Century Systems, Inc. 2004
All Rights Reserved

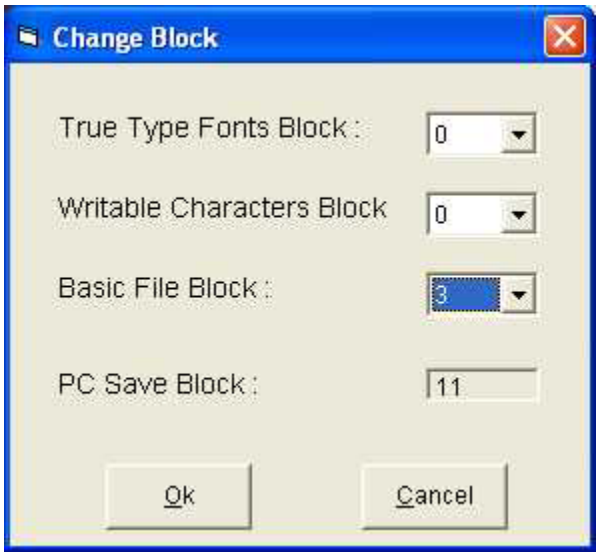
Loading Procedure for the Basic Interpreter

1. Launch basic file loader
2. Select the file to be loaded from the appropriate directory
3. Click the **Main Program** radio button
4. Click **Block Number No.0** radio button

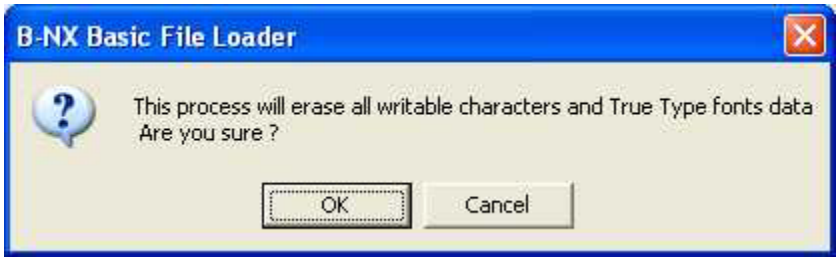


5. Click **Change Block**

Copyright Century Systems, Inc. 2004
All Rights Reserved



6. Make the **Basic File Block** at least 2 in this case 3 is selected

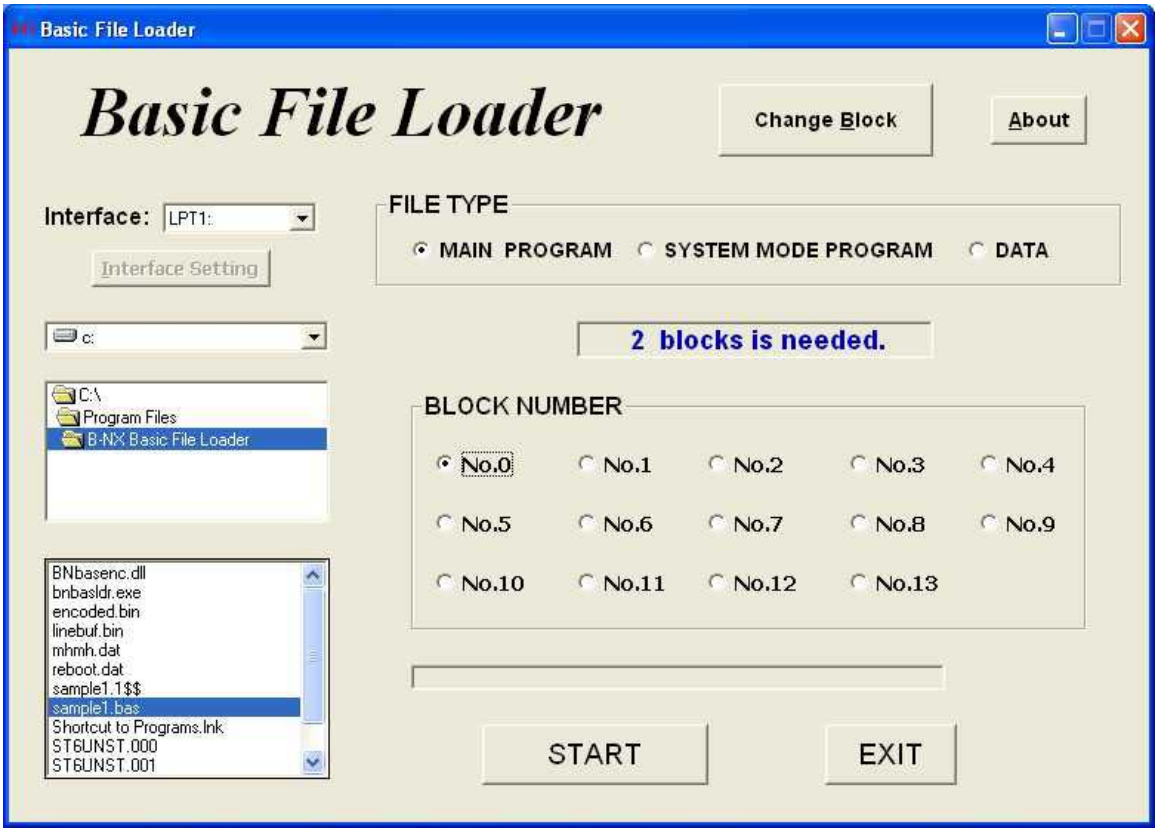


7. Click **OK**

   Printer will display SAVING 704 0

8. Click **OK**

Copyright Century Systems, Inc. 2004
All Rights Reserved



9. Click **START**

10. Click **OK**
    Printer will display NOW LOADING . . . .

11. Click **OK**

Copyright Century Systems, Inc. 2004
All Rights Reserved

12. Turn the interpreter on, on the printer
    a. Hold FEED and PAUSE keys while turning the printer on
        i. Printer will display <1>DIAG. Vx.xx
    b. Press the RESTART key twice
        i. Printer will display <8>BASIC
    c. Press PAUSE
        i. Printer will display BASIC ENABLE
    d. Press PAUSE
        i. Printer will display BASIC OFF`
    e. Press RESTART
        i. Printer will display BASIC ON
    f. Press PAUSE
        i. Printer will display FILE MAINTENANCE
    g. Turn the printer off and then on again
13. Send the data to be interpreted to the printer.

Copyright Century Systems, Inc. 2004
All Rights Reserved

Sample Program

```
REM ****************************************
REM * sample1.bas for a Century Eagle 4
REM *
REM * Written By: Tom Carner
REM * copyright Century Systems 2004
REM ****************************************
REM
REM ============================================
REM          Initialize Variables
REM ============================================
             LC = 0
             PROGNAME$= "lbl conv"
             VERNO = 1
REM ============================================
STARTCODE1$ = "^"
STARTCODE2$ = "~"
*MAIN_LOOP
     CLS
     LOCATE 1, 1
     PRINT PROGNAME$
     LOCATE 2, 1
     PRINT "v"
             LOCATE 2,2
             PRINT VERNO
             ON ERROR GOTO *ERROR_PROC
REM
     OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
     OPEN "COM2:9600,N,8,1" FOR RANDOM AS #2
     A$ = ""
     GOSUB *TL_HEADER
     DIM SDATA1$(0), SDATA2$(0)
*MAIN
             LC = LC+1
             SDATA1$(0) = ""
     WHILE SDATA1$(0) = ""
        CNT% = GETCMDZ(STARTCODE1$, STARTCODE2$)
     WEND
             CMDBUF$=SDATA1$(0)
     IF CMDBUF$ = "" GOTO *CMDBUFLOOP
     GOSUB *CMD_CHECK
             IF LC=36 THEN LC=0
*MAIN_END
     GOTO *MAIN
REM CMDBUF EERROR LOOP
REM ============================================
*CMDBUFLOOP
             GOTO *MAIN
REM LABEL HEADER
REM ============================================
*TL_HEADER
     PRINT #2, "{D0762,1016,0762|}"
     PRINT #2, "{AY;-02,0|}{AY;-02,1|}"
     PRINT #2, "{AX;-000,-000,-00|}"
```

```
        PRINT #2, "{C|}"
'
        PRINT #2, "{PV00;0183,0125,0040,0035,B,00,B|}"
                PRINT #2, "{PC000;0308,0599,1,1,N,+04,00,B|}"
                PRINT #2, "{XB00;0300,0170,9,1,04,0,0241,+0000000000,000,1,00|}"
'
'
        PRINT #2, "{PC010;0054,0741,1,1,N,+04,00,B|}"
                RETURN
REM ==================== END SUB ================
REM
REM SUB CMD_CHECK
REM ============================================
*CMD_CHECK
                IF LC=23 THEN *LC1
                IF LC=28 THEN *LC2
                IF LC=32 THEN *LC3
                IF LC=34 THEN *LC4
                RETURN
*LC1
                STR=LEN(CMDBUF$)-3
                DAT$=MID$(CMDBUF$,4,STR)
                PRINT #2, "{RV00;"+DAT$+"|}"
                RETURN
*LC2
                STR=LEN(CMDBUF$)-3
                DAT$=MID$(CMDBUF$,6,STR)
                PRINT #2, "{RB00;"+DAT$+"|}"
                RETURN
*LC3
                STR=LEN(CMDBUF$)-3
                DAT$=MID$(CMDBUF$,4,STR)
                PRINT #2, "{RC000;"+DAT$+"|}"
                RETURN
*LC4
                STR=LEN(CMDBUF$)-3
                DAT$=MID$(CMDBUF$,4,STR)
                IF MID$(DAT$,2,1)="," THEN *LQ1
                IF MID$(DAT$,3,1)="," THEN *LQ2
                IF MID$(DAT$,4,1)="," THEN *LQ3
                IF MID$(DAT$,5,1)="," THEN *LQ4 ELSE *LQ5
*LQ1
                PQ$=MID$(DAT$,1,1)
                PRINT #2, "{XS;I,000"+PQ$+",0002C5100|}}"
                RETURN
*LQ2
                PQ$=MID$(DAT$,1,2)
                PRINT #2, "{XS;I,00"+PQ$+",0002C5100|}}"
                RETURN
*LQ3
                PQ$=MID$(DAT$,1,3)
                PRINT #2, "{XS;I,0"+PQ$+",0002C5100|}}"
                RETURN
*LQ4
                PQ$=MID$(DAT$,1,4)
                PRINT #2, "{XS;I,"+PQ$+",0002C5100|}}"
```

Copyright Century Systems, Inc. 2004
All Rights Reserved

```
            RETURN
*LQ5
            PRINT #2, "{XS;I,9999,0002C5100|}}"
            RETURN
REM ==================== END SUB ================
REM SUB ERROR_PROC
REM ============================================
*ERROR_PROC
            BASERR=ERR
            IF ERR=52 THEN OPENFLAG% = 0 : RESUME NEXT
            IF ERR=53 THEN OPENFLAG% = 0 : RESUME NEXT
            IF ERR=62 THEN RESUME NEXT
            LED(2)=TRUE
            GOSUB *LED
            DISP.MSG$="ERROR("+RIGHT$(STR$(ERR),2)+"):"+RIGHT$(STR$(ERL),5)
            GOSUB *DISP
            A$=""
            WHILE A$=""
                                    A$=INKEY$
            WEND
            PPAUSE
            END
REM
REM
REM ============================================
*DISP
            CLS
            LOCATE 1,1
            PRINT LEFT$(DISP.MSG$+SPACE$(16),16)
            RETURN
REM
REM
REM ============================================
*LED
            IF LED(0)=TRUE THEN LED0 ON ELSE LED0 OFF   'POWER LED
            IF LED(1)=TRUE THEN LED1 ON ELSE LED1 OFF   'ONLINE LED
            IF LED(2)=TRUE THEN LED2 ON ELSE LED2 OFF   'ERROR LED
RETURN
REM
REM ============================================
REM
```

Copyright Century Systems, Inc. 2004
All Rights Reserved

Sample Zebra Label

```
^XA^TA0^JSO^LT0^MMT^MNW^MTD^PON^PMN^LH0,0^JMA^PR4,4^MD0^JUS^LRN^CI12^XZ
^EF
^XA^LL0609
^FT48,97^ACN,36,20
^FDLabel for a Zebra 105SL 203dpi^FS
^BY4,3,203^FT226,347^BCN,,Y,N
^FD>;8002283606^FS
^FT272,487^AFN,26,13
^FDData sent in ZPL^FS
^PQ1,0,1,Y^XZ
^EF
```

Label Output from a Zebra 105SL

(using the code above)

Copyright Century Systems, Inc. 2004
All Rights Reserved

Label Output from a Century Eagle 4

(using the sample Zebra code from above run through the basic interpreter program above)

Copyright Century Systems, Inc. 2004
All Rights Reserved