

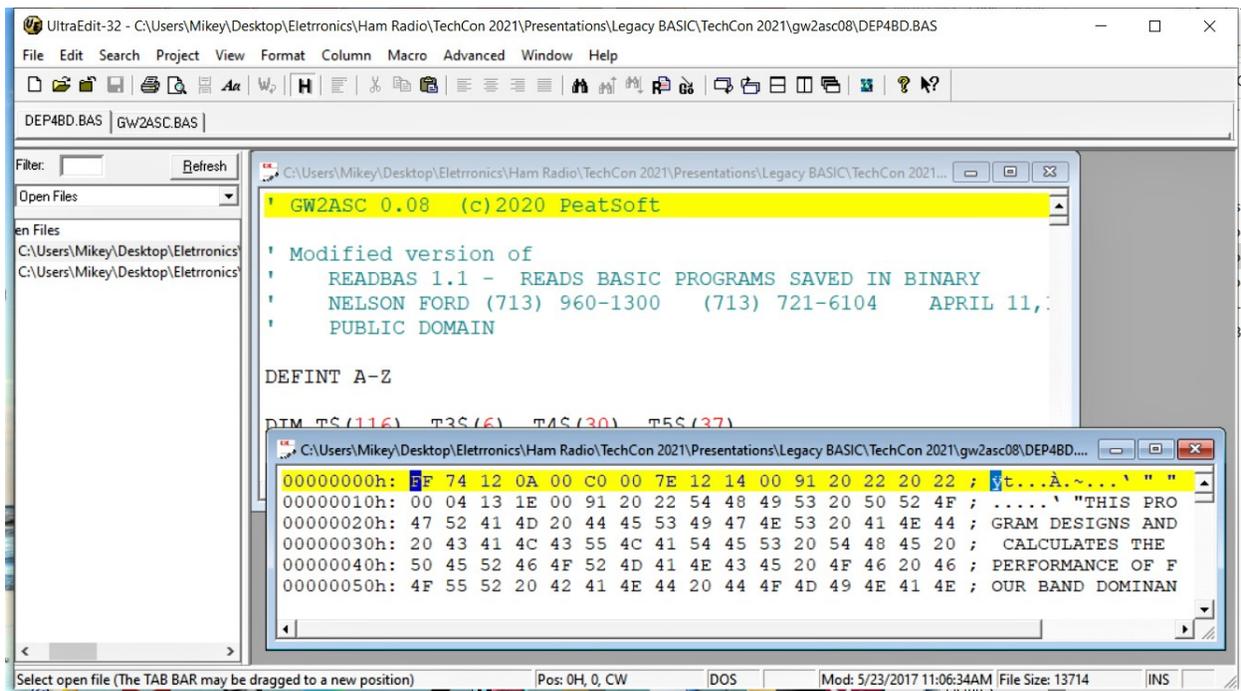
Demo 1

You cannot tell if a program is tokenized if the extension is **.BAS**; other extensions like TXT and ASC will be plain-text ASCII.

If you open tokenized and plain-text files in UltraEdit or a hex editor like HxD, you can see the difference.

A tokenized BASIC file will start with **FF** or **FE** as the first character, and you will see **00** representing the end of a program line. ASCII text will show **0D 0A** at the end of program lines.

Program lines with quoted text will be readable characters in either case.



Demo 2

PBCC command line compiler. Although it is a command line compiler, it also has a standard Windows GUI interface.

Open READBAS.BAS and test compile (no PBMAIN / WINMAIN)

```
10 DEFINIT A-Z: CLS : INPUT "FILE NAME"; FI$
20 INPUT "TO (1)SCREEN (2)PRINTER (3)DISK"; D
30 IF D=1 THEN F2$="SCRN:" ELSE IF D=2 THEN F2$="LPT1:" ELSE IF D=3 THEN INPUT
40 DIM X#(8): PRINT : PRINT "PRESS ANY KEY TO ABORT": PRINT
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: READ T$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: READ T3$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: READ T4$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
130 GET 1: X = ASC(X$): GET 1: IF X = 0 AND ASC(X$) = 0 THEN STOP
140 GET 1: N$ = STR$(ASC(X$)): GET 1: X = ASC(X$)
150 IF X > 0 THEN N$ = STR$(X * 256 + VAL(N$))
160 PRINT #2. RIGHT$(N$. LEN(N$) - 1): " "
```

Error 484 in C:\PBCC40\gw2asc08\Readbas.bas(8:017): Requires Procedure (Function/Sub Line 8: 10 DEFINIT A-Z: CLS : INPUT "FILE NAME"; FI\$

Compile failed at 9:21:45 PM on 2/26/2021

This error is because you are creating a Windows executable

Open READBAS-2.BAS and test compile (the READ statement format is wrong at line #60)

```
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: READ T$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: READ T3$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: READ T4$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
130 GET 1: X = ASC(X$): GET 1: IF X = 0 AND ASC(X$) = 0 THEN STOP
140 GET 1: N$ = STR$(ASC(X$)): GET 1: X = ASC(X$)
150 IF X > 0 THEN N$ = STR$(X * 256 + VAL(N$))
160 PRINT #2. RIGHT$(N$. LEN(N$) - 1): " "
```

Error 418 in C:\PBCC40\gw2asc08\Readbas-2.bas(16:024): Statement expected Line 16: 60 FOR T = 129 TO 243: READ T\$(T - 128): NEXT'tokens 129-243

Compile failed at 9:25:22 PM on 2/26/2021

Open READBAS-3.BAS and test compile (missed a READ statement edit, easy to fix)

```
40 DIM X#(8): PRINT : PRINT "PRESS ANY KEY TO ABORT": PRINT
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: T$ = READ$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: T3$ = READ$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: T4$ = READ$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
130 GET 1: X = ASC(X$): GET 1: IF X = 0 AND ASC(X$) = 0 THEN STOP
140 GET 1: N$ = STR$(ASC(X$)): GET 1: X = ASC(X$)
150 IF X > 0 THEN N$ = STR$(X * 256 + VAL(N$))
```

Error 418 in C:\PBCC40\gw2asc08\Readbas-3.bas(19:024): Statement expected
Line 19: 90 FOR T = 129 TO 165: READ T5\$(T - 128): NEXT'token 255 followed by 129-165
=====
Compile failed at 9:27:16 PM on 2/26/2021

Open READBAS-4.BAS and test compile. (the INPUT statement has no variable assignment. I have never seen this. The program may be a work in progress, or a funky dialect of BASIC was used.)

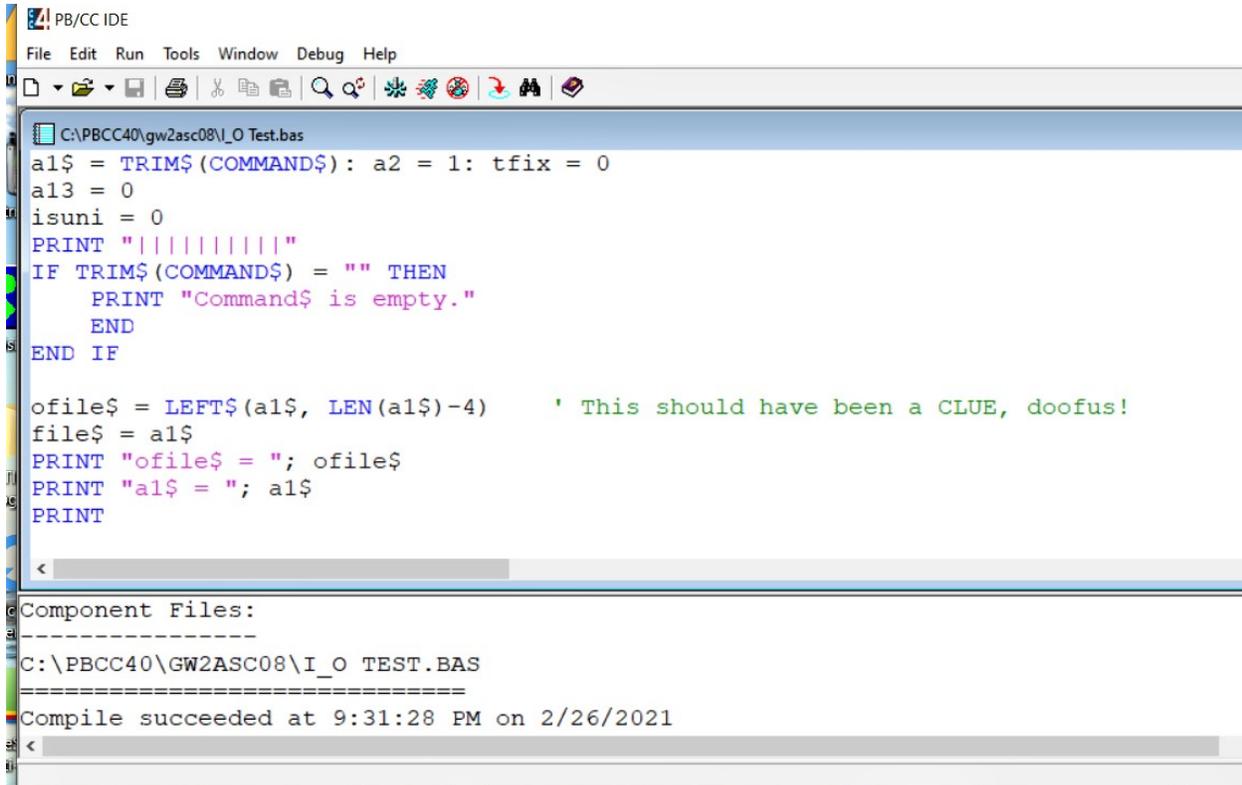
```
10 DEFINT A-Z: CLS : INPUT "FILE NAME"; FI$
20 INPUT "TO (1)SCREEN (2)PRINTER (3)DISK"; D
30 IF D=1 THEN F2$="SCRN:" ELSE IF D=2 THEN F2$="LPT1:" ELSE IF D=3 THEN INPUT
40 DIM X#(8): PRINT : PRINT "PRESS ANY KEY TO ABORT": PRINT
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: T$ = READ$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: T3$ = READ$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: T4$ = READ$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: T5$ = READ$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
<
```

Error 426 in C:\PBCC40\gw2asc08\Readbas-4.bas(13:079): Variable expected
Line 13: 30 IF D=1 THEN F2\$="SCRN:" ELSE IF D=2 THEN F2\$="LPT1:" ELSE IF D=3 THEN INPUT
=====
Compile failed at 9:29:21 PM on 2/26/2021

Demo-3

Open I_O Test.bas Scroll down to show **LEN(A1\$)-4**

I should have figured out why it needed “-4” to remove 3 letters from the right side of the filename. The closing quote mark passed by the COMMAND\$ command was the issue.



```

PB/CC IDE
File Edit Run Tools Window Debug Help
C:\PBCC40\gw2asc08\I_O Test.bas
a1$ = TRIM$(COMMAND$): a2 = 1: tfix = 0
a13 = 0
isuni = 0
PRINT "|||||||"
IF TRIM$(COMMAND$) = "" THEN
    PRINT "Command$ is empty."
END
END IF

ofile$ = LEFT$(a1$, LEN(a1$)-4)      ' This should have been a CLUE, doofus!
file$ = a1$
PRINT "ofile$ = "; ofile$
PRINT "a1$ = "; a1$
PRINT

Component Files:
-----
C:\PBCC40\GW2ASC08\I_O TEST.BAS
=====
Compile succeeded at 9:31:28 PM on 2/26/2021

```