

Action Source Code Formatter#

```
;
; FORMAT.ACT - Formats Action! sources
; with indented DO-OD, IF-FI pairs.
;
; by Harold Long
;
; Source should be in simple form, i.e.,
; one keyword per line, no "DO mumble OD"
; constructions, etc.
;
CHAR ARRAY SOURCE(255), ;Temporary
          DEST(255), ;String arrays
          KEYWORD ;Test word pointer
CARD ARRAY POS(6),NEG(6),RES(6),TEMP(6), LEAD(6) ;Keyword arrays
BYTE I,J,K ;Counters
BYTE CurPos=~[0], ;Current character pointer
      LastPos=~[0], ;Last Character position
      Spaces=~[0], ;Current indent value
      NextSpace=~[0], ;Next line indent value
      Indent=~[2] ;Number of spaces per indent
INT TempSpace=~[0] ;Back up this line only
;
; Setup keyword arrays with desired values
; To include additional words, add to list
; and modify Foo(0) to reflect total number
; of words in list.
;
PROC Setup()
  POS(0)=2          ;Number of words in list
  POS(1)="IF"      ;Roughly sorted by frequency
  POS(2)="DO"

  NEG(0)=3
  NEG(1)="FI"
  NEG(2)="OD"
  NEG(3)="RETURN"

  RES(0)=3
  RES(1)="MODULE"
  RES(2)="PROC"
  RES(3)="FUNC"

  TEMP(0)=2
  TEMP(1)="ELSE"
  TEMP(2)="ELSEIF"
  TEMP(3)="RETURN"

  LEAD(0)=4
  LEAD(1)="BYTE"
  LEAD(2)="CARD"
  LEAD(3)="INT"
  LEAD(4)="CHAR"

  TempSpace=-Indent
  Spaces=Indent
RETURN
```

```

;
; Strip out all leading spaces.
; Returns with stripped data in
; SOURCE
;
PROC Strip()
  FOR I=1 to SOURCE(0) ;Count spaces
  DO
    IF SOURCE(I)#32 THEN ;Exit on first non-space char
      EXIT
    FI
  OD
  IF SOURCE(I)=155 THEN
    SOURCE(0)=0
    SOURCE(1)=155
  FI
  IF SOURCE(0)#0 THEN
    ScopyS(DEST,SOURCE,I,SOURCE(0)) ;Move to delete spaces
    ScopyS(SOURCE,DEST,1,DEST(0)) ;Put back in source record
  FI
RETURN

```

```

;
; Extract substring: returns with
; start:(end-1) inclusive string in
; DEST
;

```

```

PROC SubStr(BYTE Start, BYTE End)
  IF End>Start THEN
    DEST(0)=(End-Start)
    FOR I=1 to DEST(0)
      DO
        DEST(I)=SOURCE(Start+I-1)
      OD
    ELSE
      DEST(0)=0
      DEST(1)=155
    FI
RETURN

```

```

;
; Find delimiter: returns next occurrence
; of space char in SOURCE
;

```

```

BYTE FUNC FindLim(BYTE Start, BYTE End)
  IF End>Start THEN
    FOR I=Start TO End
      DO
        IF SOURCE(I)=32 THEN
          EXIT
        FI
      OD
    ELSE
      I=0
    FI
RETURN(I)

```

```

;

```

```

; Test for lower case character
;
BYTE FUNC IsLower(BYTE c)
  IF (c>='a) AND (c<='z) THEN
    RETURN(1)
  FI
RETURN(0)

;
; Shift to upper case if lower
;
BYTE FUNC ToUpper(BYTE c)
  IF IsLower(c) THEN
    c==-$20
  FI
RETURN(c)

;
; Force substring to upper case just
; in case you forgot...
;
PROC SubUp()
  BYTE c
  FOR I=1 to DEST(0)
  DO
    c=DEST(I)
    DEST(I)=ToUpper(c)
  OD
RETURN

; Test Positive indent; examine DEST
; for match with positive keyword
;
BYTE FUNC TestPos()
  BYTE Match
  Match=0
  FOR I=1 TO POS(0)
  DO
    KEYWORD=POS(I)
    IF SCompare(DEST,KEYWORD)=0 THEN
      Match=Indent
    FI
  OD
RETURN(Match)

;
; Test Negative indent; examine DEST
; for match with negative keyword
;
BYTE FUNC TestNeg()
  BYTE Match
  Match=0
  FOR I=1 to NEG(0)
  DO
    KEYWORD=NEG(I)
    IF Scompare(DEST,KEYWORD)=0 THEN
      Match=Indent
    FI
  OD

```

```

RETURN(Match)

;
; Test for Reset; cancel any
; outstanding pos/neg indents
;
BYTE FUNC TestRes()
  BYTE Match
  Match=0
  FOR I=1 to RES(0)
  DO
    KEYWORD=RES(I)
    IF Scompare(DEST,KEYWORD)=0 THEN
      Match=Indent
    FI
  OD
RETURN(Match)

;
; Test for Temporary reset; back up
; line one space to emphasize word.
;
BYTE FUNC TestTemp()
  BYTE Match
  Match=0
  FOR I=1 to TEMP(0)
  DO
    KEYWORD=TEMP(I)
    IF Scompare(DEST,KEYWORD)=0 THEN
      Match=Indent
    FI
  OD
RETURN(Match)

;
; Test for 'leader' word, e.g., complex
; expression such that keyword may follow
;
BYTE FUNC TestLead()
  BYTE Match
  Match=0
  FOR I=1 to LEAD(0)
  DO
    KEYWORD=LEAD(I)
    IF Scompare(DEST,KEYWORD)=0 THEN
      Match=1
    FI
  OD
RETURN(Match)

;
; File handler;
;
; Opens Foo.ACT as input and
; Foo.FCT as output. Default
; filename is "TEST".
;
PROC FOpen(BYTE ARRAY FName)
  BYTE ARRAY INAME(16) ;Input file name

```

```

BYTE ARRAY ONAME(16) ;Output file
BYTE ARRAY IEXT=".ACT"
BYTE ARRAY OEXT=".FCT"
IF FName(0)=0 THEN
  Scopy(Fname,"D:TEST")
FI
FOR I=1 TO FName(0)
  DO
    INAME(I)=FName(I)
    ONAME(I)=FName(I)
  OD
FOR I=FName(0)+1 TO FName(0)+4
  DO
    INAME(I)=IEXT(I-FName(0))
    ONAME(I)=OEXT(I-FName(0))
  OD
INAME(0)=FNAME(0)+4
ONAME(0)=FNAME(0)+4
OPEN(2,INAME,4,0) ;Input is read only
OPEN(3,ONAME,8,0) ;Output is write only
RETURN

;
; Process Record; inputs a line from
; Foo.ACT, strips it, tests for leading
; keywords, adjusts indentation, and
; outputs to Foo.FCT.
PROC ProcRec()
  InputSD(2,SOURCE) ;Get record
  Strip() ;Delete leading spaces
  IF SOURCE(0)>0 THEN ;Skip blank lines
    CurPos=FindLim(1,SOURCE(0)) ;Find delimiter
    SubStr(1,CurPos) ;extract substring
    SubUp() ;Upper case
    IF TestLead() THEN ;Complex expression?
      LastPos=Curpos+1
      CurPos=FindLim(LastPos,Source(0)) ;Get next word
      SubStr(LastPos,Curpos) ;Extract
      SubUp() ;Upper case
    FI
    IF TestRes()#0 OR SOURCE(1)='; THEN
      Spaces=Indent
      TempSpace=-Indent
    FI
    Spaces== -TestNeg()
    NextSpace=TestPos()
    TempSpace== -TestTemp()
    CurPos=Spaces+TempSpace+1
    FOR I=1 TO 254 ;Blank target line
      DO
        DEST(I)=32
      OD
    DEST(0)=254
    DEST(255)=155
    SAssign(DEST,SOURCE,Curpos,SOURCE(0)+CurPos)
    ScopyS(SOURCE,DEST,1,SOURCE(0)+Curpos)
    TempSpace=0
  FI
  PrintDE(3,SOURCE) ;Write record

```

```

    Spaces==+NextSpace
RETURN

PROC Main()
BYTE ARRAY File(20)
CLOSE(2)
CLOSE(3)
GRAPHICS(0)          ;CLEAR SCREEN
POSITION(10,2)
PRINTE("Action! Formatter")
POSITION(2,4)
PRINTE("Formats Action! source files with")
POSITION(2,5)
PRINTE("indented DO-OD, IF-FI, etc. pairs.")
POSITION(2,7)
PRINTE("Specify input file as Dn:mumble")
POSITION(2,8)
PRINTE("Input extension of .ACT is assumed.")
POSITION(2,9)
PRINTE("Output file will be Dn:mumble.FCT")
Position(2,11)
PRINT("Input: ")
INPUTS(File)
FOpen(File)
Setup()
WHILE EOF(2)=0
DO
    ProcRec()
OD
CLOSE(2)
CLOSE(3)
POSITION(2,13)
PRINTE("DONE!")
RETURN

```