

```

cStyle$ MACRO arg:=<>
LOCAL bs, PP, quot, c$, parseLen, P$,Esc,QuotState
P$ equ <arg>
cDest$ textequ <arg>
bs INSTR P$,<\>
Esc = 0      ; 1 = escape sequence
QuotState = 0 ; 1 = currently writing to quoted text

if bs ne 0
  quot SUBSTR P$,1,1
  ifidn quot,<!">
  elseifidn quot,<!'>
  else
    .err <invalid string>
  endif
  cDest$ textequ <>
  parseLen SIZESTR P$
  PP=1
  REPEAT parseLen
    c$ SUBSTR P$, PP, 1
    ifdif c$,quot
      if Esc eq 0
        ifidn c$,<\>
          Esc = 1
      else
        if QuotState
          cDest$ CATSTR cDest$,c$
        else
          cDest$ CATSTR cDest$,<,>,quot,c$
          QuotState = 1
        endif
      endif
    endif
  else
    ifidn c$,<n>
      if QuotState
        cDest$ CATSTR cDest$,quot,<,13,10>
        QuotState = 0
      else
        cDest$ CATSTR cDest$,<,13,10>
      endif
    elseifidn c$,<t>
      if QuotState
        cDest$ CATSTR cDest$,quot,<,9>
        QuotState = 0
      else
        cDest$ CATSTR cDest$,<,9>
      endif
    elseifidn c$,<0>
      if QuotState

```

```

        cDest$ CATSTR cDest$,quot,<,0>
        QuotState = 0
    else
        cDest$ CATSTR cDest$,<,0>
    endif
elseifidn c$, <\>
    if QuotState
        cDest$ CATSTR cDest$,c$
    else
        cDest$ CATSTR cDest$,<,>,quot,c$
        QuotState = 1
    endif
else
    .err <invalid escape sequence>
endif
Esc = 0
endif
endif
PP=PP+1
ENDM
if QuotState
    cDest$ CATSTR cDest$,quot
endif
parseLen SIZESTR cDest$
if parseLen GT 1                                ; remove leading comma
    ifidn @SubStr(<%cDest$>,1,1),<,>            ;
        cDest$ SUBSTR cDest$,2                ;
    endif                                        ;
endif                                          ;
endif
endif

ifidn cDest$,<">    ; empty quote
    cDest$ TEXTEQU <0> ;
elseifidn cDest$,<'> ;
    cDest$ TEXTEQU <0> ;
endif ;

EXITM cDest$
ENDM

```