

```

1      !Utilities.F95
2      !2021.04.06.0535cdt JMS- Traveler2/Athlon64/WinXPPro/APF9.0/Ogl1.2.1
3
4      !Table of Contents:
5      ! Subroutine DateAndTime22(DaTime)
6      ! Subroutine ElapsedTime(DaTime1,DaTime2,TElapsed,Minutes,iP)
7      ! Subroutine TimeStamp(cL,Chrono,iW)
8      ! Subroutine WSF(cL,iW)
9      ! Subroutine MyLenTrim(cL,nCu,iP)
10     ! Subroutine Beamer(N,Nmax,iP)
11     !-----7-9
12     Subroutine DateAndTime22(DaTime)
13     !2021.03.21.1015cdt JMS- Present date & time string -> DaTimeLabel*22
14     !Elapsed is updated first,
15     !     so using the same argument twice results in returning the DaTime value.
16     !--Arguments
17     implicit none
18     character::DaTime*22
19     !--Internals
20     character::cDate*8,cTime*10,cZone*5
21     integer*4::iYMDZHMSM(8)
22     !--EndDefs-----
23     call DATE_AND_TIME(cDate,cTime,cZone,iYMDZHMSM)
24     write(DaTime, &
25          "( i4, '.',i2.2, '.',i2.2, '.',i2.2,i2.2, '.',i2.2, '.',i3.3)") &
26          iYMDZHMSM(1:3),iYMDZHMSM(5:8)
27
28     return
29     !-----7 9
30     Subroutine ElapsedTime(DaTime1,DaTime2,TElapsed,Minutes,iP)
31     !2021.03.22.1015cdt JMS- Present date & time string -> DaTimeLabel*22
32     !Elapsed is updated first,
33     !     so using the same argument twice results in returning the DaTime value.
34     !--Arguments
35     implicit none
36     character::DaTime1*22
37     character::DaTime2*22
38     character::TElapsed*22
39     real*16   ::Minutes
40     integer*4::iP           !Write enable>5: write(iP,...)
41     !--Internals
42     integer*4::iYMDZHMSM(8,2)
43     integer*4::i,iTimer(3)
44     character::cL*80
45     !--EndDefs-----
46     iYMDZHMSM = 0
47     do i=1,2           !2021.03.21.0945.52.765
48         if(i==1) &
49             read(Datetime1,"(i4,1x,i2,1x,i2,1x,2i2,1x,i2,1x,i3)") &
50                 iYMDZHMSM(1:3,i),iYMDZHMSM(5:8,i)
51         if(i==2) &
52             read(Datetime2,"(i4,1x,i2,1x,i2,1x,2i2,1x,i2,1x,i3)") &
53                 iYMDZHMSM(1:3,i),iYMDZHMSM(5:8,i)
54         iTimer(i) = ( ( ( iYMDZHMSM(3,i)      * 24 & ! hours/day
55                       +iYMDZHMSM(5,i) )    * 60 & ! minutes/hour
56                       + iYMDZHMSM(6,i) )    * 60 & ! seconds/min
57                       + iYMDZHMSM(7,i) )    *1000 & !mseconds/sec

```

```

58      +      iYMDZHMSM(8,i)                                ;enddo!i
59      iTimer(3)=iTimer(2)-iTimer(1); if(iTimer(3)<0) itimer(3)=0
60      iYMDZHMSM(1:8,2) = 0
61      iYMDZHMSM( 8 ,2) = mod(iTimer(3), 1000)                !milliseconds
62      iYMDZHMSM( 7 ,2) = mod(iTimer(3), 60000)/ 1000        !seconds
63      iYMDZHMSM( 6 ,2) = mod(iTimer(3),3600000)/ 60000      !minutes
64      iYMDZHMSM( 5 ,2) = mod(iTimer(3),3600000)/ 3600000    !hours
65      iYMDZHMSM( 3 ,2) =      iTimer(3)                    /86400000 !days
66      write(TElapsed,"( 8x,i2.2,'.',i2.2,i2.2,'.',i2.2'.',i3.3)") &
67      iYMDZHMSM(3,2),iYMDZHMSM(5:8,2)
68      minutes = iTimer(3)/60000._16
69      ! if(1>0) return !//Diagnostic
70      if(iP<6)                                               return
71      write(cL,"(38x,'Elapsed time:')") ; call TimeStamp(cL,TElapsed,iP)
72      ! 2021.03.21.0945.52.765
73      write(cL,"(57x,'year ^m ^d ^h^m ^s ^ms')"); call WSF(cL,iP)
74                                               return
75 End Subroutine ElapsedTime
76 !-----7 9
77 Subroutine TimeStamp(cL,Chrono,iW) !2021.03.25.0815
78 !2021.03.25.0815cdt JMS- Turning leading zeros to spaces.
79 !2021.03.21.0925cdt JMS- Expanded timestamp to 22 characters.
80 ! Hence 52 characters are available for your use.
81 !2013.11.04.1340cst JMS- Writes a labeled timestamp to the screen (& #iW if>12)
82 ! 79 characters wide with a trailing char(0).
83 ! - Useful in timing alogrithms & also in chronologically
84 ! sorting output hardcopies days after-the-fact.
85 ! - Traveler2/Athlon64/WinXPPro/APF9.0/Ogl1.2.1
86 implicit none !arguments
87 character*(*)::cL
88 character ::Chrono*22
89 integer*4 ::iW
90 !--- !internals
91 character ::ChronoL*22
92 character :: c80*80
93 integer*4 :: i,n,NumCharUsed
94 !-----
95 if(iW.lt.5) return
96 ! --Replace Chrono leading zeros with spaces:
97 ChronoL = Chrono
98 do n=1,22; i=iChar(ChronoL(n:n))
99 if((i>=49).and.(i<=57)) exit
100 if( i==48 ) ChronoL(n:n) = char(32) ;enddo!n
101
102 ! --Leave label room for Chrono, right justified: e.g. 2021.03.25.0723.48.890
103 c80 = char(32)
104 c80( 1: 5) = "!--- "
105 call MyLenTrim(cL,NumCharUsed,0)
106 !NumCharUsed=len_trim(cL)
107 if(NumCharUsed.gt.52) NumCharUsed=52
108 c80( 6:NumCharUsed+ 5) = cL(1:NumCharUsed)
109 c80(58: 79) = ChronoL
110 c80(80: 80) = char(0)
111 call WSF(c80,iW)
112 return
113 End Subroutine TimeStamp
114 !-----7 9

```

```

115 Subroutine WSF(cL,iW)                                !2013.11.04.0545
116 !2013.11.04.0545cst JMS- Writes nowhere, to screen, or to the screen & a file.
117 !                                     - Label length is arbitrary.
118 !                                     - Traveler2/Athlon64/WinXPPro/APF9.0/Ogl1.2.1
119 implicit none                                       !arguments
120 character*(*)::cL
121 !character      ::cL*80
122 integer*4 ::iW
123 !---                                               !internals
124 integer*4 :: i,nCU
125 !-----
126 call MyLenTrim(cL,nCU,0)
127 select case(iW)
128   case( : 5); return
129   case(6 :12); do i=1,nCU ;write( 6,"(a1,\)") cL(i:i) ;enddo!i
130                 write( 6,"(')')")
131   case(13: ); do i=1,nCU ;write( 6,"(a1,\)") cL(i:i) ;enddo!i
132                 write( 6,"(')')")
133                 do i=1,nCU ;write(iW,"(a1,\)") cL(i:i) ;enddo!i
134                 write(iW,"(')')")
135   end select !iW
136   return
137 End Subroutine WSF
138 !-----7 9
139 Subroutine MyLenTrim(cL,nCu,iP)                       !2013.11.07.0510
140 !2013.11.07.0510cst JMS- My version of len_trim.
141 !                                     Trailing spaces, C/R's, & LF's are char(0)'d.
142 !                                     Interior char(0)'s are converted to spaces.
143 !                                     - Traveler2/Athlon64/WinXPPro/APF9.0
144 implicit none
145 character*(*)::cL                                       !arguments
146 integer*4 ::nCU
147 integer*4 ::iP
148
149 integer*4 ::nSize                                       !internals
150 integer*4 ::i
151 character ::A0*1,A32*1
152 !-----
153 nSize=len(cL)
154 if(iP.gt.5) then; write(iP,*) 'cL=',cL
155                 write(iP,*) 'cL=',(iChar(cL(i:i)),i=1,80)
156                 write(iP,*) 'nSize =',nSize
157 endif !iP>5
158
159 A0 =char(0) ;A32=char(32) ;nCU=0
160 do i=nSize,1,-1
161   select case(iChar(cL(i:i)))
162     case( 0) ;goto 10
163     case(10) ;goto 10
164     case(13) ;goto 10
165     case(32) ;goto 10
166     case default; if(nCU.gt.0) cycle ;nCU=i ;cycle
167   end select !iChar(cL(i:i))
168 10  if(nCU.eq.0) cL(i:i)=A0
169     if(nCU.gt.0) cL(i:i)=A32
170   enddo!i
171

```

```
172     if(iP.gt.5) then
173         write(iP,"('nCU=',i6)") nCU
174         do i=1,nSize ;write(iP,"(i4,\)") ichar(cL(i:i));enddo
175             write(iP,"(' ')")
176     endif!iP>5
177     return
178 End Subroutine MyLenTrim
179 !-----7 9
180 Subroutine Beamer(N,Nmax,iP)                                !2009.01.22.0800
181 !2009.01.22.0800cst JMS- Derived from Beamer (above)
182 !2007.10.10.0830cdt JMS- Nstat=(N*78)/Nmax
183 !2006.08.02.1700cdt JMS- HPPavilion/WinXPPro/APF9.0/Ogl1.2.1
184 !1990.11.19.1240cst JMS- 386/DOS/FORTRAN
185 implicit none
186 !--Arguments
187 integer*4 ::N,nMax
188 integer*4 ::iP          !Write enable>5: write(iP,...)
189 !--Internals
190 integer*4 :: kStat,nStat
191 !--EndDefs-----
192                                     if(iP<6) return
193     if(Nmax .lt. 1) stop 'BeamerL: Nmax < 1'
194     if(N .le. 1) then; Kstat=0 ;write(unit=iP,"(1x \)") ;endif
195                                     Nstat=(N*78)/Nmax
196 10 if(Nstat.gt.Kstat) then; Kstat=Kstat+1
197                                     write(unit=iP,"('*'\)") ;goto 10 ;endif
198     if(N .ge.Nmax )
199                                     write(unit=iP,"('* ')")
200     return
201 End Subroutine Beamer
202 !-----7 9
```