

```
' Time Zone: Extreme Coding Challenge
' entry of Harmon V. -- MainWin version -- Sep 2008
' released as public domain
' LB\timezone1.bas
' Program will run for 60 seconds before stopping.
```

```
' The 27 megacities with populations of 10 million or more.
' daylight savings/summer start and stop times from www.dateandtime.co
m
```

```
Rem
```

#	City, Country	Pop.	ST	DT start & End
Data 1,	"Tokyo, Japan",	"33,600,000",	"UTC+9",	"NONE"
' 35*41'N x 139*46'E				
Data 2,	"Seoul, South Korea",	"23,400,000",	"UTC+9",	"NONE"
Data 3,	"Mexico City, Mexico",	"22,400,000",	"UTC-6",	
"Sun 06-APR-08 02:00", "Sun 26-OCT-08 02:00"				
Data 4,	"New York City, USA",	"21,962,000",	"UTC-5",	
"Sun 09-MAR-08 02:00", "Sun 02-NOV-08 02:00"				
Data 5,	"Mumbai, India",	"21,600,000",	"UTC+5.5",	"NONE"
Data 6,	"Delhi, India",	"21,500,000",	"UTC+5.5",	"NONE"
Data 7,	"São Paulo, Brazil",	"20,600,000",	"UTC-3",	
"Sun 19-OCT-08 00:00", "Sun 17-FEB-09 00:00"				
Data 8,	"Los Angeles, USA",	"18,000,000",	"UTC-8",	
"Sun 09-MAR-08 02:00", "Sun 02-NOV-08 02:00"				
Data 9,	"Shanghai, China",	"17,500,000",	"UTC+8",	"NONE"
Data 10,	"Osaka, Japan",	"16,700,000",	"UTC+9",	"NONE"
Data 11,	"Cairo, Egypt",	"16,100,100",	"UTC+2",	
"Fri 25-APR-08 00:00", "Fri 29-AUG-08 00:00"				
Data 12,	"Buenos Aires, Argentina",	"16,000,000",	"UTC-3",	
"Sun 19-OCT-08 00:00", "Sun 16-MAR-09 00:00"				
Data 13,	"Kolkata, India",	"15,700,000",	"UTC+5.5",	"NONE"
Data 14,	"Manila, Philippines",	"15,600,000",	"UTC+8",	"NONE"
Data 15,	"Jakarta, Indonesia",	"15,100,000",	"UTC+7",	"NONE"
Data 16,	"Karachi, Pakistan",	"15,100,000",	"UTC+5",	
"Sun 01-Jun-08 00:00", "Sat 01-NOV-08 00:00"				
Data 17,	"Tehran, Iran",	"13,500,000",	"UTC+3.5",	
"Fri 21-MAR-08 00:00", "Sun 21-SEP-08 00:00"				
Data 18,	"Beijing, China",	"12,800,000",	"UTC+8",	"NONE"
Data 19,	"Dhaka, Bangladesh",	"12,750,000",	"UTC+6",	"NONE"
Data 20,	"Lahore, Pakistan",	"12,700,000",	"UTC+5",	
"Sun 01-JUN-08 00:00", "Sat 01-NOV-08 00:00"				
Data 21,	"London, England",	"12,500,000",	"UTC+0",	
"Sun 30-MAR-08 01:00", "Sun 26-OCT-08 02:00"				
Data 22,	"Paris, France",	"12,000,000",	"UTC+1",	
"Sun 30-MAR-08 02:00", "Sun 26-OCT-08 03:00"				
Data 23,	"Istanbul, Turkey",	"11,800,000",	"UTC+2",	

```
"Sun 30-MAR-08 03:00", "Sun 26-OCT-08 04:00"
Data 24, "Rio de Janeiro, Brazil", "11,500,000", "UTC-3",
"Sun 19-OCT-08 00:00", "Sun 17-FEB-09 00:00"
Data 25, "Lagos, Nigeria", "10,200,000", "UTC+1", "NONE"
Data 26, "Moscow, Russia", "10,100,000", "UTC+3",
"Sun 30-MAR-08 02:00", "Sun 26-OCT-08 03:00"
Data 27, "Bangkok, Thailand", "10,100,000", "UTC+7", "NONE"
```

```
Mainwin 90 36
```

```
Cls
```

```
Print Tab(20);"Time Zone -- Extreme Coding Challenge"
```

```
Print
```

```
Struct timeZoneData, _
  Bias As Long, _
  StandardName As Char[64], _
  StandardDateYear As Short, _
  StandardDateMonth As Short, _
  StandardDateDayOfWeek As Short, _
  StandardDateDay As Short, _
  StandardDateHour As Short, _
  StandardDateMinute As Short, _
  StandardDateSecond As Short, _
  StandardDateMilliseconds As Short, _
  StandardBias As Long, _
  DaylightName As Char[64], _
  DaylightDateYear As Short, _
  DaylightDateMonth As Short, _
  DaylightDateDayOfWeek As Short, _
  DaylightDateDay As Short, _
  DaylightDateHour As Short, _
  DaylightDateMinute As Short, _
  DaylightDateSecond As Short, _
  DaylightDateMilliseconds As Short, _
  DaylightBias As Long

  CallDll #kernel32, "GetTimeZoneInformation"
, timeZoneData As Struct, ret As Long
  Global myUTCOffset
  myUTCOffset = timeZoneData.Bias.struct + timeZoneData.
DaylightBias.struct

  ncities = 27
  Dim city$(ncities), pop$(ncities), st$(ncities), dtstart$(
ncities), dtstop$(ncities)
```

```
' load arrays
For a = 1 To ncities
    Read n, a$, b$, c$, d$
    If d$<>"NONE" Then Read e$ Else e$ = ""
    city$(a) = a$
    pop$(a) = b$
    st$(a) = c$
    dtstart$(a) = d$
    dtstop$(a) = e$
'    Print n;" "; a$; tab(28);" "; b$;" "; c$;" "; d$;" "; e$
Next a

[again]
Timer 0
' * Program must first show the current date and time based upon
'    Liberty BASIC's native Date$() and Time$() functions.
dateNow$ = Date$()
timeNow$ = Time$()
Locate 1,4
Print "Current local Date & Time    : "; dateNow$; " "; timeNow$

' * Program must calculate GMT
' convert local time to UTC/GMT time
dateTimeUTC = Date$("days") + Time$("seconds")/86400 +
myUTCOffset/1440
Print "Current UTC/GMT Date & Time : "; days2DT$(dateTimeUTC)

' * Program must then allow user to select at least 2 other time zones
'
' * Program must finally display the GMT as well as the selected date(
s) and time(s).
Print
Print "## City & Country           Population      Date      Time"

For a = 1 To ncities
    dateTimeUTC = Date$("days") + Time$("seconds")/86400 +
myUTCOffset/1440
    UTCOffset = Val(Mid$(st$(a),4))/24
' convert hours to decimal days
    cityDateTime = dateTimeUTC + UTCOffset

    t1$ = dtstart$(a)
    t2$ = dtstop$(a)
    If onDST(cityDateTime, t1$, t2$) Then cityDateTime =
cityDateTime + 1/24
```

```
    Print Using("###",a);" "; city$(a); Tab(28);" "; pop$(a);" "
; days2DT$(cityDateTime)
'    Print
Next a
loops = loops + 1
If loops>60 Then [xitprog]
Timer 1000, [again]
Wait

[xitprog]
Print "Done."
End
```

```
' returns 1 if we're on daylight savings/summer time, else returns 0
' t1$ & t2$ format = "Fri 21-MAR-08 00:00"
Function onDST(nday, t1$, t2$)
    If t1$="NONE" Then
        onDST = 0
    Else
        starttime$ = Right$(t1$, 5)
        sec1 = 3600*Val(Left$(starttime$,2)) + 60*Val(Right$(starttime$,2))
        dt1 = Date$(Mid$(t1$, 5, 9)) + sec1/86400

        stoptime$ = Right$(t2$, 5)
        sec2 = 3600*Val(Left$(stoptime$,2)) + 60*Val(Right$(stoptime$,2))
        dt2 = Date$(Mid$(t2$, 5, 9)) + sec2/86400

        ' is our local time "nday" between our start & stop dates?
        If dt1<nday And nday<dt2 Then onDST = 1 Else onDST = 0
    End If
End Function ' onDST

' Convert decimal days format to standard "MM/DD/YYYY HH:MM:SS" format
Function days2DT$(DyTm)
    xdate$ = Date$(Int(DyTm))
    xtime = (DyTm Mod 1)*86400 ' in seconds since midnight
    xtime$ = sec2hms$(xtime)
    days2DT$ = xdate$; " "; xtime$
End Function ' days2DT$

' Convert n seconds into standard HH:MM:SS time string
Function sec2hms$(n)
    hh = Int(n/3600)
    mm = Int((n Mod 3600) / 60)
```

```
ss = Int((n Mod 60) + 0.5)
sec2hms$ = Right$(Str$(hh+100),2);":"; _
           Right$(Str$(mm+100),2);":"; _
           Right$(Str$(ss+100),2)
End Function ' sec2hms$
```