UnixTime or Epoch Time

Many computer systems store and process time as the elapsed seconds since 01/01/1970 00:00:00 and this method is commonly termed “unixtime” and it accepts the output of the Now() function directly. This function's counterpart is the HumanTime function - Humantime() and Unixtime() translate each-others outputs.

Unixtime removes much of the hassle in dealing with date formats as they are only converted back to “human readable” when needed and will use the date format of the locale. Any date numbers you generate can be checked against this web tool. Unixtime is not UTC implicitly. It can easily be made to be, but as it is calculated from local time, it will only be UTC if the time on the relevant system is UTC.

The following function for MMBasic returns the number of seconds since midnight on 1st January 1970 as an integer. UnixTime correctly takes account of leap years in its calculations but leap seconds are ignored - this conforms with the established method of calculating UnixTime as leap seconds are arbitrary and not generally predictable. Because of this, very small errors may exist when the output is compared to astronomical clocks.

There is little sanity checking performed on the strings you provide. If you try to crash it you'll probably succeed. If the argument string does not consist of 2 groups of 3 delimited values, -1 is returned to indicate an error. MMbasic uses +/-63bits to store integers, consequently, this function is immune to the Geek's Millennium until at least the year 292Billion.

The argument string is a concatenate of DATE, a space and the TIME e.g. “24-01-1999 17:01:58” (The associated Now() function produces correctly formatted strings). An option allows the date part to be interpreted as follows: DD-MM-YYYY when opt=0 or omitted (default action) YYYY-MM-DD when opt<>0 (ISO8601)

Syntax:

UnixTime(datestrDD-MM-YYYY HH:MM:SS[,opt])

Example Usage:

x%=UnixTime(“20-08-1995 06:00:00”)

Print UnixTime(Now())

x%=UnixTime(“20-08-1995 06:00:00”,0) x%=UnixTime(“1995-08-20 06:00:00”,1)

Dependencies:

- SPLIT Function (VB work_a_like)
- IsLeapYear Function
- IsDate and IsTime functions (VB Work_A_Like) (optional - large overhead for regexps)

```
  ' seconds since 01-01-1970. no checks on the argument format
  ' HT$ is DD-MM-YYYY HH:MM:SS or if opt=1 HT$ is YYYY-MM-DD HH:MM:SS
  ' Now() returns a suitable string
    Function UnixTime(HHT$,opt) As integer
        Local integer n,s,y,m
        Local String DD$,TT$,HT$
                HT$=HHT$
        m=Split(HT$," ")
        If m<>2 then UnixTime=-1:Exit Function
        DD$=SP$(1):TT$=SP$(2)
'following is error checking, remove if not using IsDate/IsTime due to
overhead of regexps
'[
        If Not IsDate(DD$,opt) Then UnixTime=-1:Exit Function' Not a proper
date
        If Not IsTime(TT$) Then UnixTime=-1:Exit Function' not a proper time
']
        m=Split(DD$,"-")
        If m<>3 then UnixTime=-1:Exit Function
        If opt=0 then
                    y=Val(SP$(3))
                else
                    y=Val(SP$(1))
                EndIf
                s=0
        For n=1970 To y-1
            s=s+365+IsLeapYear(n)
        Next
        For n=1 To Val(SP$(2))-1
            Select Case n
                Case 1,3,5,7,8,10
                    s=s+31
                Case 2
                    s=s+28+IsLeapYear(y)
                Case 4,6,9,11
                    s=s+30
            End Select
        Next
                If opt=0 Then
                    s=86400*(s+Val(SP$(1))-1)
                Else
                    s=86400*(s+Val(SP$(3))-1)
                EndIf
        m=Split(TT$,":")
        If m<>3 then UnixTime=-1:Exit Function
        s=s+(3600*Val(SP$(1))) + (60*Val(SP$(2))) + Val(SP$(3))
        UnixTime=s
    End Function
```

See Also: HumanTime()

Now()