



What Time Is It?

Martin Phillips
Ladybridge Systems



Last Will

∞ and ∞

Testament

∞ of ∞

Defining a Moment in Time

An **Epoch Value** counts the interval between the start of the **Era** and the moment in time being defined.

Both the Era reference point and the interval increment are open to choice.

Pick databases use 31 December 1967...
...but in local time.

Defining a Moment in Time

Linux/Unix use 1 January 1970 GMT

Windows uses various reference points:

1 January 1980 (FAT32 file system)

1 January 1900 (Excel)

1 January 1753 (Microsoft SQL Server)

1 January 1601 (NTFS file system)

Defining a Moment in Time

Most time management schemes use seconds as the counted interval.

Fractional values can be used for greater accuracy.

Negative values represent times before the era reference point.

Windows uses 100nS increments for NTFS file times.

Defining a Moment in Time

QM adopts the Linux reference point (1 Jan 1970) for epoch values.

The counter increment is seconds.

Fractional values can be used if required.

Negative values represent times before 1 Jan 1970.

Epoch Values and Time Zones

An epoch value is a time zone independent way to represent a moment in time.

Epoch values in QM, Linux and Unix are counted from 1 January 1970 UTC.

UTC (Coordinated Universal Time) is essentially the same as GMT.

Epoch Values and Time Zones

Translation of a time to a local time zone requires the time zone offset to be added to the epoch value.

This needs to be transparent to the user.

Different users of the same system may need to see the same epoch value as different local time values.

Epoch Values and Time Zones

A QM process is assumed to operate in the time zone of the underlying system.

The TZ environment variable can be used to set an alternative time zone.

Using Local Time Values in QM

An application can get the current local time using the `TIME()` function.

This is seconds since midnight.

The time as seconds since 31 December 1967 can be obtained using `SYSTEM(1005)`.

Users in different time zones will see different values from these functions.

Using Epoch Values in QM

An application can get the current time as an epoch value with the EPOCH() function.

```
TIME = EPOCH()
```

This will return the same value regardless of where the user is.

Using Epoch Values in QM

The E conversion code is used to translate an epoch value to an external form date/time in the user's local time zone.

This code has a combination of the elements of the D and MT conversions.

Users in different time zones see their own local representation of the epoch value.

Using Epoch Values in QM

Using a conversion code of EDMYTS, epoch value 1234567890 can be converted to:

GMT: 13 02 2009 23:31:30

EST: 13 02 2009 18:31:30

Pacific/Wake: 14 02 2009 11:31:30

Using Epoch Values in QM

Special E conversion codes include:

EA Fri Feb 13 23:31:30 2009

EISO8601W 2009W075

EISO8601T 20090213T233130

Using Epoch Values in QM

Translate an epoch value to a local Pick date:

$DT = MVDATE(EP)$

Translate an epoch value to a local Pick time:

$DT = MVTIME(EP)$

Using Epoch Values in QM

Translate an epoch value to a local Pick date and time:

```
DT = MVDATE.TIME(EP)
```

The date and time values in DT are separated by an underscore.

Using Epoch Values in QM

Translate a local Pick date and time to an epoch value:

$$EP = MVEPOCH(DT)$$

The date and time values in DT are separated by an underscore.

or

$$EP = MVEPOCH(DATE, TIME)$$

Using Multiple Time Zones

The QMBasic SET.TIMEZONE statement can be used to change the local time zone for the QM session:

```
SET.TIMEZONE 'EST'
```

Subsequent use of any of the epoch conversion functions will appear to be in this new zone.

Other operations are not affected (e.g. sh date)

Time Zone Names

Linux/Mac:

Defined in the Olson database within the operating system and is automatically updated.

AIX/Windows:

The TZ environment variable includes the offset from GMT (e.g. "PST8")

Another Millennium Crisis?

Many Linux/Unix systems will fail on 19 January 2038.

The time value will exceed 32 bits.

The epoch time range is 13 December 1901 to 19 January 2038.

Some systems have already addressed this by providing 64 bit library functions...

...Others have not.

Another Millennium Crisis?

Windows poses different problems.

The standard C programming libraries use 32 bit time values.

They also don't allow negative numbers.

Epoch values are therefore limited to a range from 1 January 1970 to 19 January 2038.

Another Millennium Crisis?

Windows has time handling APIs that use 64 bit values.

These appear to solve the problem but assume that all users of the system are in the same time zone.

Examples of Epoch Use Within QM

Login times are stored as epoch values. LISTU will show this in the local time zone of the user executing the command.

The SLEEP command.

Phantom log file timestamps.

Error log file shows all times in the time zone of the server.

Compile times shown by the MAP command.

The Lost Hour

When the clocks go forward, there is a one hour period in the local representation of the time that never exists.

Half an hour after 0145 is 0315.

With epoch values, there is no confusion.

The Ambiguous Hour

When the clocks go back, there is a one hour period that appears to exist twice in the local representation of the time.

Half an hour after 0145 is 0115 (again).

Again, with epoch values, there is no confusion.

Is this a Problem?

A German steel works was damaged by failure to handle a clock adjustment correctly.

Automated drug administration systems can harm patient health.

Tasks can apparently finish before they began or take an hour longer than they really did.

Other QM Time Functions

SYSTEM(12)

Same as TIME(). Returns seconds since midnight in local time zone.

Other QM Time Functions

SYSTEM(19)

For Pick compatibility, returns a unique value

dddddtttta

formed from local time zone date and time with possible alphabetic suffix.

This function is unreliable in multi-timezone applications.

Other QM Time Functions

SYSTEM(1005)

Returns combined date and time as

$DATE() * 86400 + TIME()$

in local timezone based on Pick day numbering.

Other QM Time Functions

SYSTEM(1019)

Returns offset in seconds of local timezone from UTC.

This will be negative for zones to the west of longitude zero.

Other QM Time Functions

SYSTEM(1020)

Returns milliseconds since midnight UTC.

Other QM Time Functions

SYSTEM(1030)

Returns login time as

$\text{DATE()} * 86400 + \text{TIME()}$

in local timezone based on Pick day numbering.

Other QM Time Functions

SYSTEM(1035)

Same as EPOCH(). Returns seconds since 1 January 1970 UTC.

Other QM Time Functions

SYSTEM(1038)

Returns login time as an epoch value.

Other QM Time Functions

STATUS

Field 13	Local time of last access
Field 14	Local date of last access
Field 15	Local time of last modification
Field 16	Local date of last modification
Field 34	Epoch value of last access
Field 35	Epoch value of last modification

Application Uses

Real time control systems

Time interval calculations

Multi-time zone applications

Unambiguous time references – e.g. legal

Questions

