

Internationalisation - Mission Impossible?

by Winfried Gerum



Winfried Gerum

In getting the new M language standard out of the think tanks, there has been a lot of hype. One of the claims is that "Internationalization made it into the language standard."

The MDC did an excellent job crafting the new standard. But internationalization requires something short of \$MIRACLE. The old language standard explicitly required the use of the ASCII character set and provided an ingenious way for pattern matching and collation. Now there is a framework to accommodate a wide array of additional character sets. Even without the new features, the M language did provide fairly good features to accomplish something like internationalization.

Like structured programming or object-oriented programming, internationalization of software can be facilitated by use of the proper programming language or by use of

some tools, but ultimately it can be achieved only by adjusting one's mindset accordingly.

The headline of this article is the first indication of what awaits you in internationalizing something: Europeans "-ise," and Americans "-ize." So what guidance do I have if the editor asks me to submit this column in the English language? And what does she mean if she tells me that the deadline for submitting manuscripts is 04/05/94? Americans read that as April 5th, 1994; the English interpret it as 4th of May 1994.

What do people want to achieve when they talk about internationalization? The idea is to adapt software that has been built for one cultural environment for use in a different one. If it can work at all, the capacity has to be designed into the software from the very beginning. The software needs an internal representation of data that is language- and culture-independent. This step is called internationalization. In order to interface these data to a particular user, one has to select a national environment. This step is called localization. It is not called nationalization, because cultural entities are quite different from political states.

In seemingly trivial cases, like transferring an accounting program devel-

oped in multilingual Belgium to neighboring France, a company discovered that in France one still uses fractions of the currency unit, the franc. The Belgians also call their currency a franc, but the fractions once called centimes are no longer used.

In translating text there is not just the issue of character sets: If you translate an English text to German, in most cases the resulting text is considerably longer. So a screen layout suitable for English words may be unacceptable if words are replaced by their usually longer German counterparts.

Formatting numbers may differ remarkably from country to country. Brazilians and Germans, for example, have a decimal comma instead of a decimal point, and they use a point to group thousands. Other countries may use still other formatting characters and may group four digits instead of three. Formatting monetary values adds additional complexity: The designators for the monetary units are of different lengths, and may or may not be separated with a space from the number. The monetary designator comes as a prefix or a suffix, or in countries with Portuguese influence it comes as an infix—there the monetary symbol replaces the decimal point in numeric values. And the length of monetary values for comparable goods may differ wildly with the underlying currencies.

The M language standard wisely speaks about characters, not about bytes. A character may be 7-bit ASCII in the United States, 8-bit in Western Europe, or 16-bit JIS in Japan. So don't make any assumptions about the number of bits per character when going international.

Western languages read left to right, top to bottom, but don't expect that to be universal.

The traditional M collating scheme did provide useful service even outside the United States. The new standard opens the door for additional implementor- or user-defined collation schemes. If you can't wait for them, refer to the September 1993 issue of *M Computing* (Tips 'n' Tricks, "How to Sort MacMumps and Other Strange Guys") for a method to accomplish your preferred collation scheme. Things may be really difficult there: One character may be viewed as a combination of other characters for collation purposes (e.g., German ä sorts like ae) or vice versa: A combination of characters may be treated for collation purposes as one single hypothetical character. While it is possible to write internationalized programs, it is quite a different issue if you want to present one single database at a given moment for various users according to different localizations.

For the representation of date and time values the traditional MUMPS special system variable \$HOROLOG has been remarkably culture-independent. The MDC recently discussed a feature called \$HOROLOG system function that retains a culture-independent format and adds time zone information, increased granularity (i.e., frac-

tions of a second), and clean collation properties.

Virtually all M sites have an algorithm translating between \$HOROLOG format and the common Gregorian calendar. That forms a basis for proper localization in many western places where various ways to write a Gregorian date are in use. The Gregorian calendar may be common, but it is by no means universal. There are a lot of other calendar systems.

Following are full details of two important and very interesting calendar systems. If you have Jewish or Muslim friends you might like to know which day is Yom Kippur or which month is Ramadan. All calendars can be translated from and to \$HOROLOG format, which is a convenient intermediate format when you want to convert a Gregorian date to a Hebrew date, for example. The functions presented here use an ISO style format for Hebrew and Muslim dates. In the absence of both the full details of the formats and the type fonts of their alphabets, this seems to be an appropriate approach.

The commonly used Gregorian calendar attempts to synchronize with the sun only. A full calendar cycle runs exactly 400 years. The length of a year is either 365 or 366 days with an average of 365.2425 days. A year has twelve months varying in length from 28 to 31 days, averaging 30.436875 days. In a previous column, "QUIT - The Ultimate Command" (*M Computing*, February 1993), I gave a function to transform \$H-format date to the Gregorian calendar, leaving the inverse transform as an exercise for the reader.

Armed with this function and the following ones you can convert \$H to Hebrew, Hebrew to Arabic, and Arabic to Gregorian format:

```
> SET JEWDATE=$$H2JEW(55977)
> WRITE JEWDATE
5754/07/24 (24th of Nisan 5754)
> SET H=$$JEW2H(+JEWDATE,+P(JEW
DATE,"/",2),+P(JEWDATE,"/",3)
)
> SET MOHDATE=$$H2MOH(H)
> WRITE MOHDATE
1414/10/23 (23d of Shawwal 1414)
> SET H=$$MOH2H(+MOHDATE,+P(MOH
DATE,"/",2),+P(MOHDATE,"/",3)
)
> SET GREGDATE=$$DATE(H)
> WRITE GREGDATE
1994/04/05 (5th of April 1994)
```

The Mohammedan (Arabic) calendar goes in sync with the moon only. A new moon indicates the beginning of a month. A year has twelve months of 29 or 30 days in length. The average length of a month is $29+(13752/25920)$ days. The odd months have 30 days, the even months usually 29 days. The last month has 29 days in common years and 30 days in leap years. If a particular year divided by 30 (=length of a full calendar cycle) gives a remainder of 2,5,7,10,13,16,18,21,24,26,29, then that is a leap year. As the faithful are forbidden to count a year with more than twelve months, the Mohammedan year (average length 354.3667 days) is about ten days short of a solar year. Therefore the Mohammedan years count faster and will finally end up with higher numbers than either the Christian or the Jewish calendar. The 1st of May 20874 is the 1st of Jumada'l-Ula 20874. Jewish and Mohammedan calendars almost coincide when 1st

Tishri 147099 is 8.Muharram 147099, but no day has the same date in both calendars.

The Mohammedan calendar is not in sync with the sun and the associated seasons. Nevertheless a faithful Muslim will have no doubt about the usefulness of that calendar. The 1st Muharram of the year 1 is 16-July-622 in the Christian calendar (-445031 in \$H format).

It is interesting to note that the M code doing the transformations performs its tricks without any iterations, but the division operators / \ and # are essential to these algorithms.

The Jewish calendar attempts to synchronize with the moon and the sun and with religious needs. A new moon indicates the beginning of a month. To bring the year on average in sync with the seasons, the Metonic cycle is used, which equates 235 months (revolutions of the moon around the earth) with 19 years (revolutions of the earth around the sun). A year has either twelve or thirteen months of 29 or 30 days. The average length of a month is $29 + (13753 / 25920)$ days, which is a bit longer than the month of the Mohammedan calendar. The difference adds up to one day in 2095 years. As an additional complication there are rules to avoid having certain holidays fall on a sabbath. Therefore a year does not begin on a Sunday, Wednesday or Friday. To achieve that, a common year has 353, 354, or 355 days, a leap year 383, 384, or 385 days. The rules are very complicated, so a full calendar cycle is a whopping 689,472 years! A year averages about 365.2468 ($=765433/25920 * 235/19$) days. The basic calculations are surprisingly short. Again the operators \ # are essential. The algorithm has in line TYPEOF+2 a maximum intermediate value in the range of 1E13,

which is no problem for MUMPS, but it may be a problem in other programming languages, as this needs integer arithmetic with 43 bits!

Both our Hebrew and Arabic algorithms extend to nonpositive years in astronomer style, i.e., they reckon with a year 0. If you prefer historian style, then do the following corrections on nonpositive years: add 1 to the year in \$\$MOH2H viz. \$\$JEW2H and subtract 1 from the year given by \$\$H2MOH viz. \$\$H2JEW.

Working on internationalization topics is very exciting, because one gains some insight into foreign cultures. It is an area full of surprises. Using M Technology is not a bad prerequisite for these adventures!

Winfried Gerum is president of Winner Software GmbH, Lohmühlweg 5, 91341 Röttenbach, Germany. His column appears regularly in *M Computing*.

```
;Mohammedan (Arabic) calendar
;Winfried Gerum
Q
TODAY(H) N D,M,Y S:$G(H)=" H=+$H
S D=$$H2MOH(H),Y=+D,M=$P(D,"/",2),D=+$P(D,"/",3)
Q D_ " _$$MOHMON(M) " _Y
;convert Mohammedan date to $H-Format
MOH2H(Y,M,D) Q: '$$CHKMOH(.Y,.M,.D) "' ;invalid value
N E,H,X
S H=Y-1,E=H,H=H#30,E=E-H\30
S H=H*354+$L($TR($E("0100101001001001001001001",
----->1,H),0))
Q E*10631+$J(M-1*29.5,0,0)+D+H-445032
;convert $H format to Mohammedan date (ISO-style format)
H2MOH(H) N D,E,M,Y
S D=H+445031,E=D,D=D#10631,E=E-D\10631
S Y=D+.5*30\10631
S D=D-(Y*354+$L($TR($E("0100101001001001001001001",
----->1,Y#30),0)))
S M=D\29.5 S:M<12 M=M+1
S D=D-$J(M-1*29.5,0,0)+1
S Y=E*30+Y+1
Q Y_" /" _$E(0,M<10)_M_" /" _$E(0,D<10)_D
;returns TRUE if Y,M,D is valid Mohammedan date
CHKMOH(Y,M,D) Q:$G(Y)'?1"-1.N 0 Q:$G(M)'?1.2N 0
Q:$G(D)'?1.2N 0 Q:M<1 0 Q:M>12 0 Q:D<1 0 Q:D>30 0 Q:D<30 1
Q:M#2 1 Q:M<12 0
Q +$E("0100101001001001001001001",Y#30)
;-converts number to name of month
MOHMON(M) Q $P("Muharram/Safar/Rabi al-Awwel/Rabi al-Achir/
----->Jumada'1-Ula/Jumada'1-Achira/Rajab/Shaban/Ramadan/
----->Shawwal/Zu'lkadah/Zu'lhijjah","/",M)
```

Continued on page 87

```

;Jewish (Hebrew) calendar
;Winfried Gerum
Q
;convert Jewish date to $H-Format
JEW2H(Y,M,D) Q: '$$CHKJEW(.Y,.M,.D) "" ;invalid date
N T S T=$$TYPEOF(Y)
Q $P(T,";","2")+$(P($T(@+T),";","2")," ,",M)+D-1
;convert $H format date to Jewish date (ISO-style format)
H2JEW(H) N D,L,M,T,Y
S Y=H+2045473\365.25-1
F Y=Y:1 S T=$$TYPEOF(Y),L=+T,T=$P(T,";","2) I H'<T,H-T<L Q
S D=H-T
S T=$P($T(@L)," ;",2)
F M=1:1 Q:$P(T," ,",M)>D
S D=D-$P(T," ,",M-1)+1
Q Y _"/" $E(0,M<10) _M _"/" $E(0,D<10) _D
;returns TRUE if Y,M,D is valid Jewish date
CHKJEW(Y,M,D) Q:$G(Y)'?1"-1.N 0 Q:$G(M)'?1.2N 0
Q:$G(D)'?1.2N 0 Q:M<1 0 Q:M>13 0 Q:D<1 0 Q:D>30 0
N TYPE S TYPE=+$$TYPEOF(Y) Q:TYPE<383&(M=13) 0
Q:D<30 1
Q:TYPE=353 ",1,5,7,9,11,"[" ,",M _","]
Q:TYPE=354 ",1,3,5,7,9,11,"[" ,",M _","]
Q:TYPE=355 ",1,2,3,5,7,9,11,"[" ,",M _","]
Q:TYPE=383 ",1,5,6,8,10,12,"[" ,",M _","]
Q:TYPE=384 ",1,3,5,6,8,10,12,"[" ,",M _","]
Q ",1,2,3,5,6,8,10,12,"[" ,",M _","]
;converts number to name of month
JEWMON(M,Y) Q $P("Tishri,Cheshvan,Kislev,Tebet,Shebat,Adar,
----->Adar,Sheni,Nisan,Iyar,Sivan,Tammuz,Av,Elul"," ,",
----->$S(M<7:M,1:$$TYPEOF(Y)<383+M))
; length day 25920 halaquim
;av.length month 765433 halaquim = 29+(13753/25920) days
;check the day of lunation of tishri => type of new year
;TYPEOF returns LengthOfYear;$H-ofFirstTishri
TYPEOF(YEAR) N D,E,H,Y
S H=689472,Y=YEAR#H,E=YEAR-Y\H S:Y=0 Y=H,E=E-1
S H=Y-1*235+1\19*765433+57444,D=H\25920
S H=$$AUX(H#25920,D#7,YEAR#19)
S $P(H,";","2)=E*251827457+D-2045475+$P(H,";","2)
Q H
;H=time (halaquim) of first new moon in the year
;D=day of first new moon in new year
;YR= year#19 (number in metonic cycle)
AUX(H,D,YR) G:" ,3,6,8,11,14,17,0,"[" ,",YR _","] LeapYear
Q:D=1 $S(H<9924:353,1:355) _":1"
Q:D=2 $S(H<16789:355," ,2,5,10,13,16,"[" ,",YR _","]
----->&(H<19440):355,1:"354;1")
Q:D=3 "354;" $S(H<9924:0,1:2)
Q:D=4 "354;1"
Q:D=5 $S(H<9924:354,H<19440:355,1:"353;2")
I D=6
I Q:" ,1,4,9,12,15,"[" ,",YR _","] $S(H<408:353,1:355) _":1"
I Q $S(H<9924:353,1:355) _":1"
Q $S(H<19440:355,1:"353;2")
LeapYear Q:D=1 $S(H<22091:383,1:385) _":1"
Q:D=2 $S(H<19440:385,1:"384;1")
Q:D=3 $S(H<19440:384,1:"383;2")
Q:D=4 $S(H<12575:383,1:385) _":1"
Q:D=5 $S(H<19440:385,1:"383;2")
Q:D=6 $S(H<22091:383,1:385) _":1"
Q $S(H<19440:385,1:"383;2")
;Table: cumulative length of months for each type of year
353 ; ,30,59,88,117,147,176,206,235,265,294,324,353
354 ; ,30,59,89,118,148,177,207,236,266,295,325,354
355 ; ,30,60,90,119,149,178,208,237,267,296,326,355
383 ; ,30,59,88,117,147,177,206,236,265,295,324,354,383
384 ; ,30,59,89,118,148,178,207,237,266,296,325,355,384
385 ; ,30,60,90,119,149,179,208,238,267,297,326,356,385

```

NOW SELLING!

MTA HAS VA KERNEL TOOLKIT VERSION 7.2

These are the features you'll enjoy:

- Multi-Term Look-Up for DHCP for better look-up capabilities
- Duplicate Resolution Utilities to merge duplicate data files
- Capacity Management Tools to measure resource consumption and identify/solve performance problems
- Tools for DHCP development and IRM -- developers, documenters, verifiers use these tools

AVAILABLE TO MEMBERS ONLY!

The TOOLKIT sells for \$80.00.

When placing your order, please indicate 3.5" or 5.25" disk, 1.4M or 1.2M tape.

**(You must have VA
Kernel 7.1 and MailMan 7.0)**

YOU DON'T HAVE VA KERNEL
VERSION 7.1?

SEE PAGE 99 FOR DETAILS ON
THIS NEW RELEASE.

ORDER THE KERNEL (INCLUDING
FILEMAN V. 20.0) AND TOOLKIT
FOR \$300.00.

See Page 100 To Order These And Other
MTA Software and Publications.

STOP IN THE RENO
MTA BOOKSTORE
SEE A FULL SELECTION
BUY YOUR COPIES TODAY !!!!