

# At Last! Royalty-Free Application Licenses for M Developers

*by G. William Moore and Jules J. Berman*

## Abstract

InterSystems has announced that Open M for Windows comes with a license that permits developers to distribute programs, along with the Open M files needed to run their programs as stand-alone applications, without paying a royalty to InterSystems. The only restrictions on the developer are that the program must not operate on a network and must be priced under \$400. For the first time, individuals and small groups of M programmers can create their own applications to compete in a mass market where most software titles sell for below \$100. In effect, this brings the same opportunities to M programmers that have been enjoyed by C programmers for nearly two decades. Now M-programmers can create and distribute great software at a competitive price and can work in multilanguage development teams on Windows-based software projects using an M language component.

## Introduction

Imagine that you have just written an amazing software product with wide appeal that fits on a single floppy disk. If your program is written in C or Microsoft Visual Basic (VB), you can compile the program and sell your .EXE file in a competitive mass market.[1] As a C or VB programmer, you never have to pay the language vendor a cent. But if you are an M programmer, you would need to pay the M vendor a royalty fee for every disk you sell.[2] In many cases, this royalty fee would exceed the market cost of the competing application written

in C or VB. In fact, we have never seen an M distribution royalty below \$50, but the average price of a mass-market application is now about \$50. It is not surprising that there are no M language software applications sold on the mass market!

The de facto standard for development languages in 1994 is a royalty-free distribution for .EXE files. InterSystems Corporation (Cambridge, Massachusetts) is the first M language vendor that has met this standard. InterSystems has offered a new M development environment, Open M for Windows. For \$195, an M developer can purchase a licensed copy of Open M for Windows. Once an application is finished, the programmer can distribute the program files without paying a royalty to InterSystems, so long as the program is not networked and is priced under \$400. M code can be called from VB, a popular programming environment that operates under Windows and that permits the rapid development of an event-driven graphical user interface (GUI).

VB is an event-driven, object-oriented graphical environment that uses the BASIC language command structure. Fundamental M string-manipulation functions, such as \$A(), \$C(), \$E(), \$F(), \$P(), have near-equivalent functions in VB, or can be simulated by a few lines of VB code. VB graphical capabilities make use of the entire range of Microsoft Windows features, including either sequential execution or event-driven controls, input from the keyboard or mouse, programmable menus, icons and im-

ages, and multiple objects available simultaneously. These objects may contain their own data and programs, and may be represented by separate windows on the computer monitor. Perhaps VB's greatest advantage as a programming environment is its huge user base and numerous vendors who market VB add-ons that extend the power of VB for VB developers.

Unlike M, VB has only rudimentary capabilities for managing hierarchical arrays. The \$O() functionality in M, including the fast, reliable implicit sorting of large arrays, has no VB equivalent. Writing M code for VB takes advantage of the strengths of both environments. The final product is a Windows program, capable of dynamic data exchange (DDE) with other Windows programs, and ready to run on millions of PCs worldwide.

## How Open M Works: A Mini-Application Sample

Our laboratory has been working over the past several years with TRANSOFT, an M language program that translates between any two natural languages using the Roman alphabet.[3] An early version of M source code is published, and prototype translators have been constructed for sample texts in German, Dutch, Turkish, and romanized Japanese. The user supplies the source document, dictionary, and grammar through the File Manager user interface, which is able to store and index large files. File Manager is not widely known outside the M community, and does not have a user interface that

meets commercial standards. As our first Open M project, we attempted to attach TRANSOFT to a VB front end.

After an easy installation of Open M, one is faced with problems in achieving the VB-M interface. On page 5-14 of the DT Windows Guide, there is a half-page description of the interface, which is incomplete and misleading.[4] For example, it looks as though there are files containing a sample program with a VB-to-M interface, named VBXDTM.EXE, VBXDTM.MAK, VBXDTM.FRM, and VBXDTMG.BAS. In fact, VBXDTM.FRM and VBXDTMG.BAS are provided, but VBXDTM.EXE and VBXDTM.MAK must be made by the user. You can copy the sample code out of the PushToCall subroutine in the VBXDTM.FRM form and make your own VB interface.

The printed manual then directs the user to start up the DTM-Windows icon and type 'd fore^windll(50)<ENTER>'. We could not determine from the printed documents that we must previously have typed 'd ^%nspace <ENTER>' and selected OPENMDEMO. We then switch over to a VB program which contains this code:

```
A="S x="" D ^PROG": R
=ExecuteM(A,B,C,D)
```

where ^PROG is the name of the M program in the M environment, and VB variables ExecuteM, A, B, C, D, and R, are described on pages 5-13 and 5-14 of the manual. In addition to the printed documents, there are assorted ASCII text files and Windows help files. There is a 136K-byte ASCII file, VBFAQ.TXT, a gossipy text with frequently asked questions (FAQs) and hints about VB, downloaded from the Internet, but not a word about the VB-to-M interface. There are also DTWin 2.2 ReadMe, DTWin 2.2 Update, DTWin 6.1 ReadMe, and Visual M ReadMe icons.

We easily ported TRANSOFT into Open M with the ^%rload routine,

along with a complete copy of the Veterans Affairs File Manager and five megabytes of dictionary entries and grammar formulas. We constructed a simple VB front end to enter the source sentences and display the resulting translation, as shown in figure 1. On the right panel, the Open M Window shows the sequence of DTWINDLL.DLL calls received in the M environment, in this case, 'S x="" D ^PROG' sent over from the VB panel shown on the left. The source sentence, "bone diseases in animals" has been translated in the M environment into German, and has been ported back to the VB-environment under "TARGET SENTENCE:". All the real work of translation and file management is carried out in the M environment. Data are exchanged through MS-DOS-ASCII files. In this demonstration program, the DT-MAX window (right) and windows displayed by the VB .EXE file (left), must be operating simultaneously. We have not tested other styles of program design and data exchange described in the Open M documents yet.

## Obstacles

Open M for Windows is still in its infancy. Infants require a lot of attention and it is sometimes difficult to believe that infants are worth all the trouble they demand. In preliminary tests of Open M, we have encountered the kinds of problems associated with an infant product. First of all, the instruction manuals, though copious (five volumes of text) are inadequate for VB beginners. In fact, less than a page of text is devoted to the VB interface. Our many recent calls to InterSystems indicate that the company has not reconciled its marketing and legal strategies. For instance, our software came without an invoice, without a license contract (the assertion of royalty-free distribution is documented by a press release from the company), and without a list of the

distributable files. One InterSystems marketer sold us the package as the package that developers can use to distribute their Open M applications, while another InterSystems marketer insisted that the developer distribution package for Open M has yet to be officially released.

Also, we got the impression from InterSystems technical support people that they believe users will find the manuals and Windows Help modules sufficient to address any programmer's needs. In our opinion, this belief is hopelessly naive, and users will confront a myriad of problems that the manuals do not address. In addition, programmers will find product bugs. After all, this is a new product that must operate on a wide variety of PC configurations. We hope that InterSystems has the necessary commitment to give Open M programmers the support they will need.

## Conclusion

Until now, M vendors have targeted their efforts exclusively toward large corporate clients. Just as IBM and Digital Equipment Corporation have learned that individuals and small businesses represent a market force that cannot be ignored, M vendors are learning the same lesson. Once M code starts to appear in mass market applications, M vendors will want to attract private developers and small corporations, and will follow InterSystems's lead. There is every reason to expect that M will become a language with mass appeal, that M programmers will work along with C, C++, Basic, and Windows application developers in multilanguage applications, that a market for M instruction books will emerge in computer stores, software catalogs and book stores, and that M programmers will at last compete in the same open job market as C programmers. ■

# MSources '94

STILL Your Guide  
to the  
M Technology Association

It's the most complete reference for . . .

- . . . Software Sources
- . . . Applications
- . . . Services
- . . . M Implementors
- . . . Hardware Implementations

Use It Often!

## Advertiser Index

We appreciate these sponsors of the November issue, and all the companies who support the M community with experience, ability, and talent.

Ajnet .....	3
CoMed .....	60
Cue Data Services .....	46
CyberTools, Inc. ....	11
Data Methods, Inc. ....	24
Digital Equipment Corporation .....	cover 2
ESI Technology, Inc. ....	6
Geisinger Medical Center .....	63
Greystone Technology Corporation .....	8, cover 3
Henry Elliott & Company .....	5, 62
InterSystems Corporation .....	7
KB Systems, Inc. ....	17
Kennedy Hospitals .....	21
LCI .....	6
M-Tech .....	63
McIntyre Consulting .....	21
Micronetics Design Corporation .....	1
MUMPS AudioFax .....	61
Polylogics Consulting .....	62
Sentient Systems .....	63
Software Search .....	33
Sunquest .....	16

*This index appears as a service to our readers. The publisher does not assume any liability for errors or omissions.*

G. William Moore, M.D., Ph.D., is a staff pathologist and chief of the autopsy section at the Baltimore VA Medical Center, (Maryland), and holds appointments in the pathology departments at the University of Maryland and The Johns Hopkins University. He has been working with M Technology since the late 1970s. Contact him at Pathology and Laboratory Medicine Service (113), Baltimore Veterans Affairs Medical Center, 10 North Greene Street, Baltimore, MD 21201-1524. Phone: 1-410-605-7000 ext 5300; Fax: 1-410-605-7911; or through email: moore.george@baltimore.va.gov.

Jules J. Berman, Ph.D., M.D., is a staff pathologist and chief of the surgical pathology and cytology sections at the Baltimore VA Medical Center (Maryland) and holds an appointment in the pathology department at the University of Maryland. He has been working with M Technology since the late 1980s.

## Endnotes

1. T. Maxwell and B. Scott, *Visual Basic SuperBible* (Corte Madera, CA: Waite Group Press, 1992).
2. G.W. Moore and J.J. Berman, "Tyranny of M-vendor royalty: oppression of M-programmers," *M Professional* 11 (1994): 24.
3. W. Giere and G.W. Moore, "Translating English into German using VA File Manager," *M Computing* 1:4 (September 1993): 16-31.
4. *DT Windows Guide*, version 2.0 (Cambridge, MA: InterSystems, 1994).

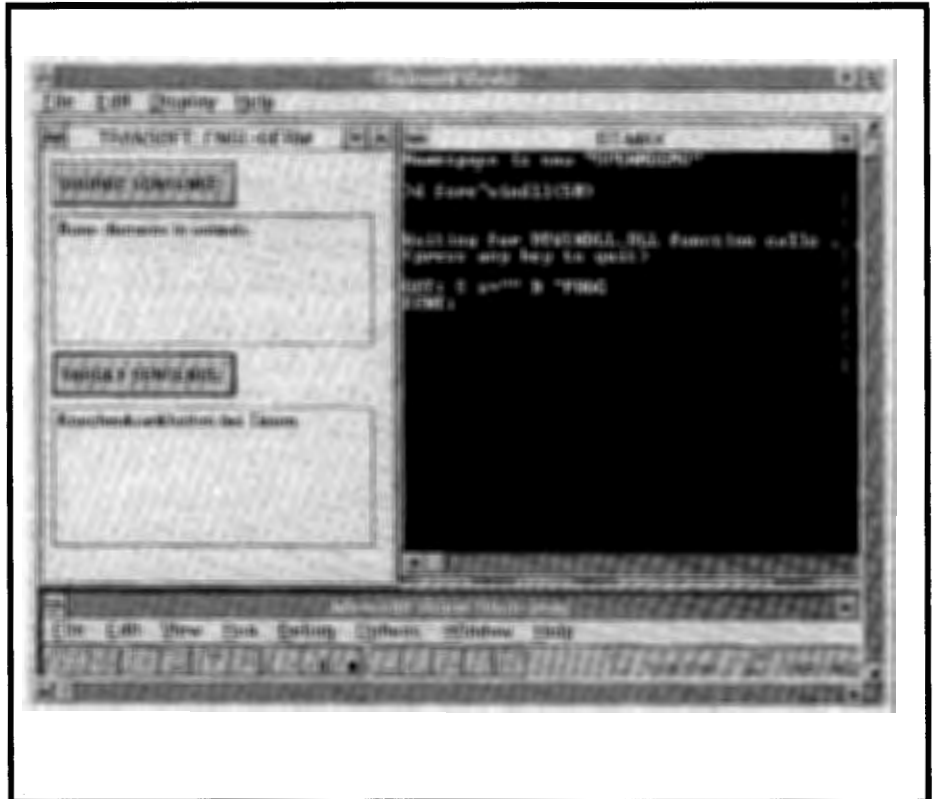


Figure 1. Sample sentence translated from English to German in a demonstration program in Visual Basic that interfaces to an M program, using Microsoft Windows and InterSystems Open M. On the right panel, the DT-MAX Window shows the sequence of DTWINDLL.DLL calls received in the M environment, namely, 'S x="" D ^PROG', sent over from the VB panel shown on the left. The source sentence, "bone diseases in animals" has been translated within the M environment into German, and the translation has been ported back to the VB environment under "TARGET SENTENCE:". All the real work of translation and file management is carried out in the M environment, whereas the user interface exploits the intuitive graphic features of Visual Basic.