

The History of FreeM

Serena Willis

Coherent Logic Development

ABSTRACT

The history of M, as well as the specifics of FreeM—the implementation maintained by the author for the last decade—are extensive and complex topics. Here, we will present an overview of the M language and its history in general, proceeding thence specifically to the historical background and development of FreeM itself, as well as its current status and goals. Though publicly-available information is sparse, efforts have been made to cite as many sources* as possible.

1. M LANGUAGE OVERVIEW

MUMPS—now generally referred to as *M* to distance it from the infectious disease—is an imperative, general-purpose, command-oriented programming language supporting persistent, hierarchical, sparse arrays. It includes high-level support for multi-user, multi-processing applications with flexible concurrency control.

MUMPS, being an acronym for *Massachusetts General Hospital Utility Multi-Programming System*, was developed by Neil Pappalardo, Robert A. Greenes, and Curt Marble of Dr. Octo Barnett's lab at Massachusetts General Hospital in 1966-1967¹. Inspired by RAND Corporation's JOSS, and the TELCOMP and STRINGCOMP languages from Bolt, Beranek and Newman, MUMPS' earliest and most prominent use cases were in the field of health informatics^{2, 3}.

MUMPS code is divided into program units referred to as *routines*, which are roughly analagous to *modules* or *compilation units* in more mainstream languages. Routines are then further divided into labels (or *tags* in MUMPS parlance), which serve as entry points into the routine. Subroutines and functions (the latter known as *extrinsic functions*) are introduced as tags with an optional, parenthesized *formallist*, being a list of parameters expected by the subroutine or extrinsic function.

Perhaps the most unique feature of M, the *global*, is a persistent, hierarchical, sparse array allowing developers to easily implement database features directly in the application's primary development language:

```
ROUTINE ;
  SET ^PEOPLE("Willis, Serena", "DOB") = "1980-12-01"
  SET ^PEOPLE("Willis, Serena", "SEX") = "F"
  SET ^PEOPLE("Willis, Serena", "COUNTRY") = "US"
```

The above routine would be called with *DO ^ROUTINE*, resulting in a data structure in fixed storage where the elements (known in M parlance as *subscripts*), i.e., "*Willis, Serena*", are automatically sorted on insertion, allowing

* Note that all world wide web URLs have been shortened for readability. Following these URLs will automatically redirect the reader to the appropriate, original source. All such URLs were accessed by the author on 27 April 2025, unless otherwise specified.

¹ Greenes, R.A., Pappalardo, A.N., Marble, C.W., and Barnett, Dr. G. Octo, "Design and Implementation of a Clinical Data Management System," *Computers and Biomedical Research* 2(5), Laboratory of Computer Science, Massachusetts General Hospital, Boston (10 Mar 1969).

² Greenes, R.A., Barnett, Dr. G. Octo, Klein, Dr. Stuart W., Robbins, Dr. Anthony, and Prior, Roderick E., "Recording, Retrieval and Review of Medical Data by Physician-Computer Interaction," *New England Journal of Medicine* 282(6), Laboratory of Computer Science, Massachusetts General Hospital, Boston (5 Feb 1970).

³ Pendergrass, Henry P., Greenes, R.A., Barnett, Dr. G. Octo, Poltras, James W., Pappalardo, A. Neil, and Marble, Curtis W., *An On-Line Computer Facility for Systematized Input of Radiology Reports*, Massachusetts General Hospital, Boston (1 Mar 1969).

for rapid retrieval. Rough analogues can be found in the *PICK* operating system, as well as BBN *FILECOMP*, the latter of which was a direct influence in the early development of M.

Early versions of M included *MGH MUMPS* from Massachusetts General Hospital, and *Digital Standard MUMPS* from Digital Equipment Corporation. Notably, early commercial versions of M were commercial software, somewhat limiting the language's spread outside of its ecological niche of health informatics.

Vendor	Implementation
Mass. General Hospital	MGH MUMPS
Digital Equipment Corp.	Digital Standard MUMPS (DSM)
InterSystems Corp.	InterSystems Standard MUMPS (ISM), Open M, Cache, IRIS Data Platform
Data Tree	Data Tree MUMPS (DTM)
Micronetics	Micronetics Standard MUMPS (MSM)
Patterson & Gray	PSM-11, PSM-32, PSM-V
MGlobal International Inc.	CCSM, MacMUMPS, M Global MUMPS, M3 Lite
U.C. Davis	MicroMUMPS
Ray Newman	MUMPS V1
Fourth Watch Software LC	Reference Standard M (RSM)
Kevin O'Kane	Mumps-II, Open Mumps, GPL Mumps
M21 Ltd.	M21
Fidelity National Information Services	GT.M (Greystone Technology MUMPS)
YottaDB	YottaDB
Eugene Karataev	MiniM

Figure 1. Partial list of M implementations.

2. EARLY HISTORY OF FREEM

FreeM was developed in Germany in the mid-1990s by a developer who went by the pseudonym *Shalom ha-Ashkenaz*, whose actual identity remains unknown⁴, though it is thought by some that they are a dentist who learned C and developed FreeM on their own time. Shalom developed FreeM at a time when InterSystems Corporation (the company that developed the ISM implementation of M) was acquiring the majority of its competitors and absorbing their technology into their *Open M* product, which would later become *InterSystems Cache*⁵.

Implementation	Year
DataTree MUMPS (DTM)	1993
Digital Standard MUMPS (DSM)	1995 ⁶
Micronetics Standard MUMPS (MSM)	1998

Figure 2. M implementations absorbed by InterSystems Corp.

Shalom wished to provide a community-driven, open-source implementation of M as a bulwark against the growing threat of single-vendor hegemony over the M language. Its design—as well as some of the documentation included with the original sources—indicate that FreeM was originally targeted to the MS-DOS family of operating systems. It made use of a very limited subset of the C library, and included instructions for renaming the MS-DOS style 8.3 filenames in order to compile under UNIX⁷.

⁴ Walters, Dr. Richard F., “History and Continuing Evolution of FreeM: A Concept Whose Time Has Come (Again),” *M Computing* 7(2), p. 19 (May 1999). See <https://l.mumps.dev/4>.

⁵ *Ibid.* p. 18.

⁶ Goelz, Larry and Paladino, John, *Cover Letter re DSM*, Compaq Computer Corporation, Houston (31 May 1999). See <https://l.mumps.dev/r> [dead link].

⁷ ha-Ashkenaz, Shalom, “README,” *Generic Universal MUMPS Project* (18 Jun 1998). See <https://l.mumps.dev/7>.

At one point in FreeM’s early history, Shalom ported FreeM from MS-DOS to SCO UNIX, the UNIX System V Release III-derived descendant of Microsoft XENIX, now known as SCO OpenServer—a platform still supported by FreeM today⁸. This port brought support for the *scoansi* terminal type, including colors and ANSI X3.64 control mnemonics.

3. INITIAL INVOLVEMENT

The author’s mentor in computer programming and UNIX was Lawrence Landis, who involved himself heavily in the M programming language ca. 1991. Mr. Landis promoted the M language to the author from 1991 forward, and first demonstrated FreeM to her in August 1998. In 2010, the author incorporated her company, Coherent Logic Development, learned M, and began doing contract work in M through Mr. Landis’s company, Fourth Watch Software.

Mr. Landis was the owner of FreeM’s SourceForge repository⁹, which had not been touched in a number of years, following Fidelity National Information Services’ decision to release GT.M under a free software license. In August 2011, the author downloaded the source code for FreeM and did enough work on it to enable building and running under modern GNU/Linux systems and posted it to the *mumpster.org* forums¹⁰.

In 2014, Mr. Landis gave the author administrator access to the FreeM SourceForge repository and transferred maintainership of the project to her¹¹.

4. GENERIC UNIVERSAL M PROJECT

The *Generic Universal M Project* was conceived by Richard F. Walters, a professor from U.C. Davis. The GUMP, following the rising popularity of object-oriented programming in the 1990s, was intended to be a toolkit allowing M implementations to be built from discrete components with a well-defined and well-specified public interface among these components. These components included the global handler (supplying the functionality of persistent global storage), and the interpreter/compiler (responsible for implementing M language commands). The components would have been able to communicate over a network, or in-process on the same host, enabling distributed computing functionality¹².

Although the specification for the GUM interface to global handlers attained a reasonably well-specified level of completeness¹³, and Lawrence Landis and others developed a mostly-complete implementation of a GUM global handler¹⁴, none of the other envisioned components were ever completed, and specifically, the interpreter component was missing.

5. FREEM DONATED TO MUG DEUTSCHLAND

In July of 1998, Shalom ha-Ashkenaz donated the FreeM source code (then known as FreeMUMPS) to the M User’s Group-Deutschland (MUG-D), hoping the community would take the nascent implementation from its infancy through to a state of production-ready completeness and robustness. Shalom also placed a few conditions on his gift: a public release could not be made until a substantial set of milestones were reached. Per his conditions, the FreeMUMPS project must¹⁵:

- Implement the entirety of *ANSI X11.1-1995*
- Use Structured System Variables instead of *VIEW* commands and *\$VIEW* functions
- Raise the string size limits
- Implement MWAPI, OMI, X11 bindings, and GKS bindings

⁸ Willis, Serena, “Platform Notes (SCO OpenServer),” *FreeM Wiki* (3 Aug 2023). See <https://l.mumps.dev/8>.

⁹ Landis, Lawrence D., “Generic Universal MUMPS Project,” *SourceForge* (1999). See <https://l.mumps.dev/6>.

¹⁰ Willis, Serena, “FreeM 0.1.4,” *Mumpster* (2011). See <https://l.mumps.dev/2>.

¹¹ Landis, Lawrence D., “Generic Universal MUMPS Project,” *SourceForge* (1999). See <https://l.mumps.dev/6>.

¹² Landis, Lawrence D., “C2M,” *Generic Universal MUMPS Project* (1998). See <https://l.mumps.dev/9>.

¹³ Morris, Steve, “GUM API,” *www.rosecroft.net* (18 Nov 1996). See <https://l.mumps.dev/k>.

¹⁴ Walters, Dr. Richard F., Morris, Steve, and Landis, Lawrence D., “Generic Universal M Project,” *Coherent Logic Development GitLab* (12 Jun 1995). See <https://l.mumps.dev/l>.

¹⁵ ha-Ashkenaz, Shalom, “README,” *Generic Universal MUMPS Project* (18 Jun 1998). See <https://l.mumps.dev/7>.

- Be substantially free of major bugs

Although MUG-D readily accepted the contribution of FreeMUMPS, the organization itself lacked the manpower and expertise to complete the implementation. Just as it is now, the intersection of M community members who know enough of the M language and C language to work on a project this ambitious was quite small.

6. MERGING GUMP AND FREEM

Very shortly after the contribution of FreeMUMPS to MUG-D, Richard F. Walters and a small team of developers and administrative staff who had been working on the GUMP assumed maintainership of the FreeMUMPS source code, with Lawrence Landis managing the development efforts¹⁶. This included representatives from the *M Technology Association* (an M vendor association having several foreign branches), the *M Development Committee* (the M standards organization hosting the ANSI/ISO standards for the M language, then sponsored by the M Technology Association), and others¹⁷. The goals of this team were to:

- Meet Shalom's requirements for a public release of FreeMUMPS
- Convert FreeMUMPS into the first interpreter component of the GUMP

During this period, Ronald L. Fox of Diagnostic Laboratory Services in Honolulu, HI (who passed in 2010)¹⁸ ported FreeMUMPS from SCO UNIX to Red Hat 5 and glibc-6¹⁹. Steve "Saintly" Zeck of the U.C. Davis Veterinary Medical Teaching Hospital²⁰ also attempted to rewrite the symbol table code to lift string size limits²¹, David Whitten enhanced some of the implementation-specific extensions, and Lawrence Landis integrated Zeck's symbol table work.

In FreeM 0.1.0, the name of the implementation was changed from FreeMUMPS to Public Standard M, and again to Free Standard MUMPS and then FreeM when it was discovered leading up to the FreeM 0.2.0 release that the PSM acronym was already in use for Patterson & Gray's²² M implementation²³. Dr. Walters also received the implementation ID of 49 from then secretary of the M Development Committee, Don Piccone²⁴.

One of the contributors to FreeM at this stage—primarily in the area of M vendor routines—was Axel Trocha, who would later maintain a private fork of FreeM²⁵.

7. GT.M RELEASED AS FREE SOFTWARE

GT.M, an acronym for *Greystone Technology MUMPS*, is an M implementation that was released by Greystone Technology in 1986. Greystone was later acquired by Sanchez Computer Associates, which was in turn acquired by Fidelity National Information Services²⁶.

¹⁶ Walters, *History*, p. 19.

¹⁷ Landis, Lawrence D., "Contributors.txt [sic]," *Generic Universal MUMPS Project* (1998). See <https://l.mumps.dev/a>.

¹⁸ Robinson, Angie and Utley, Mark, *Ronald L. Fox (1951-2010)*, Find a Grave, Lehi. See <https://l.mumps.dev/e>.

¹⁹ Fox, Ronald L., "FreeMUMPS source and docs available in tgz format," *mumps-l mailing list*, Honolulu (19 Jan 1999). See <https://l.mumps.dev/d>.

²⁰ Zeck, Steven, *About Steve Zeck*, Davis (1998). See <https://l.mumps.dev/m>.

²¹ Zeck, Steven, "FreeM String Lengths," *Generic Universal M Project Mailing List* (21 May 1999). See <https://l.mumps.dev/i>.

²² Landis, Lawrence D., "Generic Universal MUMPS Project," *SourceForge* (1999). See <https://l.mumps.dev/6>.

²³ Landis, Lawrence D., "Changes.GUM," *Generic Universal MUMPS Project* (18 Feb 2000). See <https://l.mumps.dev/5>.

²⁴ *Ibid.*

²⁵ Trocha, Axel, *Free Mumps* (2 Sep 2004). See <https://l.mumps.dev/3>.

²⁶ Finextra, *Fidelity National Financial acquires Sanchez*, Finextra Research Limited, London (29 Jan 2004). See <https://l.mumps.dev/c>.

When GT.M was released under the GNU General Public License in 2000²⁷, it seemed to many to obviate the *raison d'être* for FreeM, as GT.M was a well-established, robust, and high-performance M implementation with which FreeM could not then compete. Unfortunately, at this time, the GUMP and FreeM projects lost all of their momentum, and new development along these lines rapidly ceased. The final GUMP team release of FreeM was 0.5.0²⁸. However, Axel Trocha's private port would continue to be developed for some years.

8. FORKED BY AXEL TROCHA

When the free software release of GT.M stalled the GUMP team's progress on the primary branch of development, Axel Trocha, an aforementioned contributor, continued development on the FreeM source code. Trocha added many interesting features to the FreeM codebase, including:

- A native port to Microsoft Windows
- Compiling FreeM as an Apache web server module, allowing FreeM to be used easily for web development
- The ability to output HTML code in a heredoc-style format, with any line of code beginning with a left angle bracket being interpreted as HTML with support for interpolated M locals and globals
- Extensions allowing FreeM to be used as a command-line shell, along the lines of UNIX *bash*, Windows *cmd.exe*, etc.

Trocha also maintains ownership of the *freem.net* Internet domain²⁹, and continued issuing public releases of his FreeM port on that site until sometime after 2004, at which point this fork was made entirely private. Currently, *freem.net* is a blank page. However, Trocha's fork of FreeM continues to the present³⁰ as the back-end storage engine and programming environment for the *Elven Runes* website³¹. The author has communicated with Mr. Trocha on occasion, and though he is supportive of the author's efforts, has chosen to remain in the background.

9. RESUMING PRIMARY DEVELOPMENT

In 2011, the author downloaded the FreeM source code from the GUM Project's SourceForge repository—dormant since 2000—and updated it just enough that it would compile and run on modern GNU/Linux systems. The author also quickly updated FreeM to support terminal sizes larger than 80x24.

10. TAKING MAINTAINERSHIP

In 2014, Lawrence Landis transferred administrative access of the GUMP repository, conferring maintainership of the primary branch of FreeM development to the author. Since then, many features have been added and many bugs corrected, including:

- Adding support for proper namespaces, configured through a *freem.conf* file, which standardizes routine and global storage locations
- Adding support for Structured System Variables
- Adding support for the asynchronous event specification from MDC Type A proposal *X11/1998-28*³²
- Adding support for constants via the *CONST* keyword
- Adding a *WITH* command allowing the specification of an implicit prefix to all subsequent variable references
- Adding a runtime *WATCH* command, tracking changes to specified local or global variables
- Adding an *ASSERT* command, which will fail with an error message if the following expression evaluates *false*

²⁷ Ryan, Greg and Bhaskar, K.S., *Sanchez Offers GT.M Database as Open Source Freeware to GNU/Linux Users*, Linux Today, Nashville (7 Nov 2000). See <https://l.mumps.dev/b>.

²⁸ Diamond, Jon, Dorman, Rod, Gerum, Winfried, Kreis, Greg, Landis, Lawrence D., Milligan, Lloyd, Morris, Steve, Murray, John, Pastors, Wilhelm, Schell, Kate, Schofield, Lyle, Trocha, Axel, Walters, Dr. Richard F., and Whitten, David, *FreeM 0.5.0*, Generic Universal M Project, Roseville (15 Apr 1999). See <https://l.mumps.dev/j>.

²⁹ Whois.net, *Whois freem.net (see Name Servers fields)* (8 Feb 2025). See <https://l.mumps.dev/n>.

³⁰ Trocha, Axel, *FreeM development has been restarted (E-mail)* (30 Mar 2020).

³¹ Elizalde and Trocha, Axel, *Elven Runes: A Tolkien/MUME Fan Site*. See <https://l.mumps.dev/q>.

³² Smith, Arthur B., "Event Processing," *X11/1998-28*, MUMPS Development Committee SC15/TG4 (28 Jun 1998). See <https://l.mumps.dev/f>.

- Removing the Steve Zeck symbol table implementation—which was unreliable—and reverting to the original implementation
- Adding support for the GNU *readline* library, with persistent command line history and editing
- Adding REPL-like functionality (in direct mode, any M expression beginning with a number will be prepended with an implicit *WRITE*)
- Adding transaction processing
- Adding *KVALUE* and *KSUBSCRIPTS*
- Preparing to support the *M Windowing API* (MWAPI)
- Adding the *fmadm* command-line utility, for system administration functions
- Adding support for after-image journaling and forward recovery of globals
- Writing a *texinfo* manual, from which the HTML manual is derived
- Porting to Solaris/SPARC, Solaris/x86, Linux/s390x, Linux/armv6l, Linux/armv7l, SCO OpenServer 5.0.7, Tru64 UNIX/alpha, AIX/ppc, Mac OS X/x86, GNU HURD, Cygwin, NetBSD, FreeBSD, OpenBSD, and WSL1/2

In addition, the FreeM web site³³ was created, to distribute downloads and documentation.

11. FUTURE

FreeM is envisioned as a client-oriented desktop M implementation, for developing graphical user interfaces that will run on mobile and desktop devices.

The author also intends to adopt the original vision of the GUMP team, dividing FreeM’s functionality into discrete components having a well-specified public interface, with the ability to run in distributed computing environments over a network.

FreeM’s mission is to advance the state-of-the-art in M implementations, and push the evolution of the language forward. Maintaining portability to as many vintage and modern UNIX systems as possible is held as a high priority, while portability of M routines and MDC standards compliance will be maintained through the use of the new *\$ZDIALECT* intrinsic special variable to ensure that such compliance does not conflict with the primary goal of elegantly advancing the state-of-the-art and finding new audiences for the concepts originated by Neil Pappalardo and Octo Barnett in 1966.

The FreeM project is also strongly committed to free software principles, and is firmly aligned with the goals of the GNU Project and the Free Software Foundation, believing that the ethical concerns surrounding proprietary software are at least as important as the practical concerns espoused by the Open Source movement³⁴.

12. CONCLUSION

FreeM has seen a colorful and turbulent history, touched by many capable hands. Though public development of any strain of the implementation was dormant for nearly a decade, fourteen years of almost continuous development have passed since the project resumed in 2011, and a decade since official maintainership passed in 2014, and great progress has been made.

FreeM—as is the case for all M projects—presses forward in a period where the future of the M programming language is uncertain. M Development Committee efforts concurrent with FreeM development have been sporadic and have missed many milestones, the community’s most prominent members are aging, and many organizations have migrated from large M applications to what is perceived as more "modern" replacements.

It is the opinion of the author that the tight integration of an expressive and dynamic language with a robust and performant persistent storage engine makes M a natural candidate for many new and general applications. Modern application development is plagued by overwhelming bars to entry, requiring mastery of many languages and database management systems and the bulky interfaces connecting them.

³³ Willis, Serena, *FreeM*, Coherent Logic Development, Las Cruces. See <https://l.mumps.dev/o>.

³⁴ Stallman, Richard M., “Free Software, Free Society: Selected Essays of Richard M. Stallman” (1-882114-98-1), p. 57, Free Software Foundation, Boston (2002).

Though not without significant warts accreted over the years, M has no such overhead. Thus, FreeM seeks to press its philosophical advantages by mitigating the language's accumulated cruft, and adding clean interfaces addressing the needs of today. With these goals in mind, development proceeds apace.

13. ACKNOWLEDGMENTS

Certainly, the intellects responsible for giving us M are too numerous to mention by name, but their importance cannot be overstated. The author would like to especially credit Lawrence Landis for his early and innumerable contributions to her development, both generally as a practitioner of software development, and specifically for introducing her to the M programming language, and inculcating in her a love of exploration and innovation.

To the author's wife, Miriam—a brilliant technologist in her own right—and daughters (without whose inestimable patience and understanding as the author buries herself in countless passion projects, such projects would not be possible) the deepest and most profound gratitude and love are due.

14. COPYRIGHT AND LICENSE

This document is Copyright © 2025 Serena Willis

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License³⁵, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover texts, and with no Back-Cover Texts.

\$Id: freem_history.ms,v 1.21 2025/04/28 14:42:52 snw Exp \$

³⁵ Free Software Foundation, *GNU Free Documentation License 1.3*, Boston. See <https://l.mumps.dev/p>.