MCL FAQ
(Frequently Asked Questions about Macintosh Common Lisp)
Updated: 22 June, 1994

• What is MCL?

Macintosh Common Lisp (MCL) is an object-oriented dynamic language (OODL) from Apple Computer, Inc. It implements the industry standard Common Lisp programming language and CLOS (as defined in *Common Lisp: The Language*, second edition), and is fully integrated with the Macintosh family of personal computers.

MCL is a completely integrated development environment, including a fast incremental compiler which produces efficient native 680x0 code, a window-based debugger, a sourcecode stepper, a dynamic object inspector, a stack backtrace inspector, a programmable Macintosh-style emacs-like editor, online documentation, and an interactive interface toolkit. MCL provides both high-level object-oriented user interface class library and complete low-level access to the Macintosh Toolbox.

Using MCL, you can create a standalone double-clickable Macintosh application using less than 2MB of disk space which can be run with 2MB or more of memory.

• What mailing lists are there for MCL users?

`info-mcl@cambridge.apple.com`
This is a mailing list for MCL users. Many MCL users find this list to be a valuable resource for sharing information with other MCL users. If you have any questions about MCL that aren't answered in this FAQ (or tips, suggestions, or code to share) send them to the address above and often another user will be able to help you.

You can read it in several ways.
1) Subscribe to info-mcl by sending mail to
`info-mcl-request@cambridge.apple.com`.
2) Read the "digest" version, `info-mcl-digest` (see below).
3) If you are on Applelink, join the `INFO.MCL$` group by sending a link to `ST.CLAIR`.
4) If you have access to netnews, join the `comp.lang.lisp.mcl` newsgroup.

`info-mcl-digest@cambridge.apple.com`
This is the same information as info-mcl, gathered into a single daily "digest" message.
To subscribe, please send mail to `info-mcl-digest-request@cambridge.apple.com`.

`bug-mcl@cambridge.apple.com`
Send mail to bug-mcl if you need to contact the MCL team at Apple. We are committed to helping you, and we will try to respond as quickly as possible.

If you encounter a problem which you think is MCL's fault, please let us know by sending us a bug report including a detailed description of your machine configuration, and a description of your problem. If you can send a *small* fragment of code which reproduces the problem, that would also be of great help.

• Is there an internet ftp site for MCL?
Yes, connect to `cambridge.apple.com` by anonymous ftp.

Apple offers the following directories as a service to MCL users:

`/pub/MACL/`	Code for MCL versions 1.3.2 and earlier.
`/pub/MCL2/`	Code for MCL versions 2.0 and later.
`/pub/clim/`	CLIM-related files
`/pub/comp.lang.lisp/`	Archives (rather out of date) of the comp.lang.lisp newsgroup.
`/pub/dylan/`	Information about Apple's other OODL, Dylan
`/pub/mail-archive/`	Archives of the info-mcl mailing list

The `/pub/MCL2/` directory has the following subdirectories:

`contrib/`	The MCL user-contributed software library
`docs/`	Additional documentation about MCL
`interfaces/`	Interfaces to the Macintosh Toolbox
`patches/`	Patches to MCL 2.0

The `/pub/MCL2/contrib/` directory is a rich resource of useful code and extensions to MCL. It exists thanks to the generosity and community spirit of other MCL users. You are free to download files that are useful to you, as long as you heed all copyright notices and restrictions that may accompany them. Apple does not attempt to control the contents of the contrib library, all files are made available on an as-is basis. We do not modify contributions, and in most cases we do not test them or make any guarantees of their functionality or usefulness.

If you have some MCL-related code that you'd like to share with other MCL users, Apple welcomes your contribution. Please upload it by anonymous ftp to the contrib directory, and send some mail to info-mcl announcing it. Apple cannot accept any code which is not authorized for public distribution by its author(s), so please make sure you include some notice with your code giving us permission to distribute it.

MCL versions 2.0 and greater include a CD-ROM containing portions of the contrib library as a service to our customers who do not have ftp access. As with our ftp site, Apple does not modify or verify such code, and all ownership and copyright remains with the author. If you contribute code to our ftp site, please indicate whether you prefer to grant or deny Apple permission to include your code on future CD-ROMs.

• What if I don't have ftp access?

You can receive files by email using our archive server. Send mail to `archive-server@cambridge.apple.com` for more details.

Most files are also made available on the MCL CD-ROM, which is included with the purchase of MCL.

• Where can I get MCL support on Applelink?

Applelink users can subscribe to `INFO.MCL$`, which is a copy of the info-mcl list (see above).

Applelink users can send mail to `"info-mcl@cambridge.apple.com@internet#"` (to reach users), or to `"bug-mcl@cambridge.apple.com@internet#"` (to reach the MCL team)

There are two areas on Applelink for discussions and files related to MCL:

Developer Support
  Developer Talk
    Macintosh Development Tool Discussions
      Macintosh Common Lisp Discussion

Developer Support
  Developer Services
    Development Platforms
      MCL related

• Where can I buy MCL?

MCL is available from the Apple Programmers and Developers Association (APDA).

Macintosh Common Lisp v. 2.0.1 Includes 3 Mac disks, a CD-ROM, and two manuals.	B0752LL/B -- $495.00
Macintosh Common Lisp v. 1.x to v. 2.0.1 Update Includes 3 Mac disks, a CD-ROM, and two manuals. Available to owners of pre-2.0 releases of MCL	B0753LL/B -- $200.00
Macintosh Common Lisp v. 2.0 to v. 2.0.1 Upgrade Includes 3 Mac disks and a CD-ROM, no manuals Available to owners of MCL 2.0	R0561LL/A -- $25.00

To order MCL, and to get a free catalog featuring over 300 other Apple programming products, contact:

APDA
Apple Computer, Inc
P.O. Box 319
Buffalo, NY 14207-0319
USA

800-282-APDA (800-282-2732), toll-free in the USA
800-637-0029, toll-free in Canada
716-871-6555, international
716-871-6511, fax

APDA	(Applelink)
APDA@applelink.apple.com	(internet)
A.DEVELOPER3	(GEnie)
76666,2405	(Compuserve)
APDA Sale	(America Online)

MCL is also available from international APDA programs in many countries. Contact APDA or your local Apple office for more information.

MCL is also available from SFA (Software Frameworks Association, formerly called MADA), an independent nonprofit organization dedicated to promoting the use of object technology in software development. Several levels of membership are available. SFA members may purchase MCL at a special discount. For more information, please contact SFA at:

Software Frameworks Association
Suite 202-B
10062 Miller Ave.
Cupertino, CA 95014

408-253-2765 (phone)
408-253-2767 (fax)
FRAMEWORKS@applelink.apple.com (email)

• Are site licenses and distribution licenses available?

Yes. For licensing information, please contact Apple Software Licensing at 408-974-4667, or email `sw.license@applelink.apple.com`.

You may distribute any quantity of commercial, double-clickable applications built with MCL for a one-time fee of $100 (or $10 for non-commercial use). For more details, please contact Apple Software Licensing.

• What Macintosh configurations will MCL run on?

MCL 2.0 will work on any Macintosh with at least 4 MB of RAM and 6MB of disk storage, Macintosh System Software v. 6.0.4 or later; or A/UX 3.0. A CD-ROM drive is recommended.

See the chart below to determine compatibility with specific Macintosh models.

• What's the difference between MCL 2.01 and MCL 2.0?

MCL 2.01 is essentially a maintenance update to MCL 2.0. If you get the 2.01 ptable init and install the following patches into your copy of MCL 2.0, you will have an equivalent system:

```
    MCL 2.0p1, MCL 2.0p2, cyclone-cursor-patch,
Centris-without-FPU, AUX-on-quadra
```

• Where can I get patches from?

On the Internet, you can get patches from the `/pub/MCL2/patches/` directory on the anonymous FTP server at `cambridge.apple.com`)

On Applelink, patches are available on the MCL forums (see above)

Patches for MCL are also distributed on the monthly Developer's CD-ROM, available by subscription from APDA (see above for ordering information).

• What's with the ptable init, anyways?

The ptable init initializes the memory management hardware (if present) in such a way that MCL can use it to make ephemeral garbage collection (EGC) more efficient. Since it deals at a very low level with both the hardware and the operating system, it is vulnerable to minor changes when a new Macintosh model or operating system version is released.

MCL will always run without the ptable init. In most configurations, just running the ptable init will slow down all memory accesses until you restart without it. The slowdown isn't very large, but you will have to decide whether or not the benefit of improved EGC performance is worth the cost, depending upon your exact usage of MCL and your Mac.

If your ptable init comes up with a big "X" through it while booting, that means it was not installed. This will not have any bad effect; ptable is simply being ignored. Usually this either because you have a machine with no PMMU, or you have virtual memory turned on.

You should consider using the ptable init if all of the following are true:
- Your Macintosh's CPU is an 040, 030, or 020 (with a PMMU installed), and
- You are not running with System 7's virtual memory turned on, and
- You prefer to run MCL with EGC enabled

Using ptable on an 040 machine will give you slightly better EGC performance, at the expense of slightly worse overall machine performance, so Quadra owners may prefer not to use ptable. The ptable init has a much larger effect on 68030 machines, greatly speeding up MCL when you use EGC, so it is recommended for 030 owners who use EGC.

For those of you who insist on even gorier details, here they are:
The ptable init changes the MMU page size to 4K. If MCL notices that the page size is 4K or 8K it will use the MMU as a write barrier so that ephemeral garbage collections (EGC's) will have less work to do in determining which pages of older generations need to be walked looking for pointers into the generation being GC'd. This speeds up EGC considerably. Thus, ptable is unnecessary if you don't use the EGC. 68040 macs default to an 8K page size, so you'll get MMU assistance even without ptable, though ptable will slightly speed up MCL's EGC at the expense of slightly slowing down the entire machine. 68030 Macs default to very large page sizes (1M or larger!). Thus, MCL's EGC gets no hardware support unless the ptable init is installed.

• Which version of MCL runs on what hardware?

Macintosh	**Recommended MCL version**
Mac Plus, SE, Portable	MCL 2.0 or later (ptable init will be ignored)
SE30, Classic, Classic II, Color Classic LC, LCII, LCIII, LC 520 II, IIx, IIci, IIcx, IIfx Quadra 700, 900, 950, 800	MCL 2.0 or later ptable init 2.0 or later
Powerbook Duo 210, 230 Powerbook 140, 145, 160, 165c, 180, 180c IIvi, IIvx Centris 650 (with FPU)	MCL 2.0 or later ptable init 2.01 or later
Centris 610, 650 (with no FPU) Quadra 660av, 840av	MCL 2.01 or later ptable init 2.01 or later

The 2.01 version of the ptable init is available for anonymous FTP from
`cambridge.apple.com:/pub/mcl2/patches/ptable-2.01.hqx.`

• Does MCL run on PowerPC?

MCL 2.01 does not run in native mode on PowerPC, nor does it generate native mode applications.

MCL 2.01 runs under emulation on PowerPC Macs, if you do two things:
1) You don’t use Apple’s Virtual Memory (VM is ok with the patch below), and
2) You turn off the “Modern Memory Manager” in the Memory control panel.

The PPC’s 68K emulator returns a hitherto-unheard-of MMU type which the released MCL doesn’t handle.

There is now a patch to make it run under VM. It’s available for anonymous ftp as
`cambridge.apple.com:/pub/mcl2/patches/MMU5-for-PowerMac`

To make it run under the modern memory manager will require some further investigation. Apple is committed to making sure MCL 2.0.1 runs smoothly on PowerPC, at least under the 68K emulation mode.

• Does MCL run on Macs with third-party accelerator cards?

Note that "works" or "does not work" below means "MCL runs with the board installed and enabled" or "MCL crashes with the board installed and enabled", respectively.

Daystar Turbo '040 33 MHz	works on IIci with MMU4-for-IIci patch. Should work on other machines, but this is untested.
Daystar Power Cache 50	Works on a IIci.
Radius Rocket 25	Should work on IIci with MMU4-for-IIci patch, but this is untested. Does not work with RocketShare on any machine, but will work with RocketWare.
Radius Rocket 25i	Centris-without-FPU patch will make it work, but this is untested. Also needs MMU4-for-IIci patch on a IIci. Again, will work with RocketWare, but not RocketShare.
Radius Rocket 33	Seems to work on IIci. Again, will work with RocketWare, but not RocketShare.
Tokamac 25Mhz	Doesn't work on IIci. Should work with MMU4-for-IIci patch, though this is not tested.
RasterOps 24XLTV	Does not work on Mac II.

• Is MCL code portable to other platforms?

MCL complies with the current industry standard for Common Lisp, as defined in "Common Lisp: The Language", second edition, by Guy Steele. This should guarantee a high degree of compatibility with Common Lisp implementations on many other platforms.

This specification changed somewhat between the first and second editions; please consult the second edition for descriptions of the changes.

• What is CLIM?

The Common Lisp Interface Manager (CLIM) is a cross-platform User Interface toolkit, which allows you to create user interfaces for Lisp applications that will run on Macintosh, Windows, Motif (X Windows), and Symbolics Lisp Machines. For more information about CLIM, please send mail to `clim-request@bbn.com`, or contact Lucid at:

Lucid, Inc.
707 Laurel Street,
Menlo Park, CA 94025 U.S.A
800-843-4204, or 415-329-8400
415-329-8480 (fax)
`sales@lucid.com`

• What do CCL and MACL stand for?

MCL was originally developed by a small company in Cambridge, Mass. called Coral Software, under the name “Coral Common Lisp” (CCL). Later, Coral entered a marketing agreement with Franz, Inc., a major vendor of unix-based Lisps, to sell the Coral product under the name “Allegro Common Lisp”.

In 1988, Apple Computer purchased Coral and its assets, and the product was renamed to be “Macintosh Allegro Common Lisp” (MACL). In 1992, the 2.0 version of the product was renamed to Macintosh Common Lisp, or MCL.

• Where can I get more information about Common Lisp?

Appendix F of the MCL 2.0 manual ("*For more information*") contains many useful references, including a bibliography of popular Common Lisp textbooks.

The `comp.lang.lisp` newsgroup is another excellent source of information about Common Lisp and other lisp dialects.

A comprehensive "*Lisp FAQ*" document is periodically posted to comp.lang.lisp. This contains answers to questions programmers often ask, including questions of style, usage, and common bugs. It also includes a comprehensive guide to lisp vendors and implementations on various platforms. A copy of this is also included on the MCL cd-rom in the folder "`Mail Archives & Other Docs:Answers to Freq Asked Questions`"

The latest version is always available by anonymous ftp from CMU and Thinking Machines. To get the CMU library, ftp to `cs.cmu.edu`, and type

```
cd /afs/cs.cmu.edu/user/mkant/Public/Lisp-Utilities/
```

Note: you must cd to this directory in one command, as intermediate directories are protected from anonymous ftp access. This directory contains the FAQ document, as well as a large collection of other Common Lisp code, utilities, and documentation.

The Thinking Machines site is at `ftp.think.com:/public/think/lisp/`

• What's the future direction for MCL?

Apple is continuing to enhance and support MCL. A maintenance release, MCL 2.01, is now available, primarily to extend MCL support to the latest Macintosh machines, such as the Centris 610, the Centris 660av, and the Quadra 840av.

We are currently planning future releases of MCL to provide a variety of enhancements, including PowerPC support. We have not announced future products at this time.

Programming questions

Many programming questions about Common Lisp are answered in the Lisp FAQ (see above). This section is specifically for MCL.

• Even though my screen is set to 256/thousands/millions of colors, I only seem to get 8 different colors in my MCL windows. Why ?

This is a very frequently-asked question. Windows created by MCL default to the original Macintosh Quickdraw routines, which only support 8 colors (6 plus black and white). In order to get the full spectrum you must:

- Be running on a machine with 32bit quickdraw. (System 7 provides this, or get a special system extension for System 6)
- Specify `:color-p t` when creating windows. One way to do this is

```
(make-instance 'window :color-p t)
```

Another approach is to make this a default for your custom window class:

```
(defclass my-window (window)
   (...)
   (:default-initargs :color-p t ...))
```

• I've created an application, but it's not handling high-level events (and thus Apple events). What should I do?

It is likely that your toplevel-function neglected to call `(startup-finished)`, a new "feature" of MCL 2.0.1 that is documented in the release notes. This feature makes it possible to print from the Finder when printing requires launching the MCL app.

Here is the relevant section of the release notes:

New startup housekeeping

In order to make printing from the Finder work correctly, high level events are disabled until MCL's startup code enables them. If you call `SAVE-APPLICATION` and specify a value for the `:TOPLEVEL-FUNCTION` parameter, your toplevel function must evaluate the following form (either directly, or indirectly via a call to `CCL::STARTUP-CCL`):

```
(startup-finished)
```

You should do this after processing the `(FINDER-PARAMETERS)`. If you neglect the call to `STARTUP-FINISHED`, your application will not receive any high level events (e.g. AppleEvents).

• I'm having a lot of troubles writing TCP/IP code.

Try turning off the `:async` bit in the `#_control` trap in `%tcp-control` and see if that helps.

• `encode-universal-time` returns bizarre (too large) values when called with a time zone supplied.

This is a known bug. MCL neglected to initialize the `era` field in the `longDateRec` record passed to `#_LongDate2Secs`. The `leap-year-patch` from our ftp site fixes this bug and another obscure bug with `encode-universal-time` near leap years.