Macintosh Allegro Common Lisp

System Requirements	To use Macintosh Allegro Common Lisp, you will need the following system:	· Any personal computer in the Apple® Macintosh family of systems (except the Macintosh 512K) · A second 800K disk drive	· Macintosh System Software Version 6.0 (or later) *Recommended Equipment* · 2 megabytes of RAM or more · A hard disk
Ordering Information	**Macintosh Allegro Common Lisp Version 1.2** APDA§ Order No. M0067LL/B	With your order, you'll receive: · Two disks containing the Macintosh Allegro Common Lisp compiler, and debugging and development tools	· *Macintosh Allegro Common Lisp User's Guide* · *Common Lisp: The Language* · *Common Lisp: The Index*
Auxiliary Products	**Portable Common Loops** APDA Order No. T0259LL/A	With your order, you'll receive: · Two disks containing the Portable Common Loops software	· A draft version of the Common Lisp Object System specifications
	Allegro Flavors	Allegro Flavors§ provides compatibility with Version 6.1 of the Flavors object-oriented programming system from Symbolics. Bodies of code written in Flavors can be run without translation. Allegro Flavors is available from:	Franz, Inc. 1995 University Avenue Berkeley, CA 94704 (415) 548-3600 Check with APDA for information about ordering this product directly from APDA.
Licensing Information	A license is required to distribute applications produced with Macintosh Allegro Common Lisp. For further information, please contact:	Apple Software Licensing Apple Computer, Inc. 20525 Mariani Avenue, M/S 38I Cupertino, CA 95014	**Apple Programmers and Developers Association (APDA)** Apple Computer, Inc. 20525 Mariani Avenue, M/S 33G Cupertino, CA 95014 1-800-282-APDA (1-800-282-2732) TLX: 171-576 AppleLink®: APDA Fax: (408) 562-3971 CompuServe: 766,2045 MCI Mail: Postrom GEnie: A.DEVELOPER3

Product Details

Object-Oriented Language Extensions

This release of Macintosh Allegro Common Lisp implements Object Lisp. Object Lisp is used by Macintosh Allegro CL to implement windows, menus, dialog boxes and streams.

Future versions of Macintosh Allegro CL will support an integrated Common Lisp Object System (CLOS) object-oriented environment. Users who are interested in CLOS now can purchase Portable Common Loops; see Ordering Information for details.

User Interface Tools

Macintosh Allegro Common Lisp provides error-checked access to Macintosh user interface features. User interface components--including windows, menus, and dialog boxes--can be modified both interactively and under program control.

· *Events.* Unlike most Macintosh languages, Macintosh Allegro Common Lisp does not require a main event loop. Instead, events are caught and dispatched by the Lisp runtime kernel. When an event occurs, program execution is interrupted and the event is handled. After event handling, normal program execution resumes.

Predefined event handlers are integrated into Macintosh Allegro CL's object-oriented user interface tools. New classes of objects can specialize their responses to events.

· *Windows.* Macintosh windows are accessible as high-level objects. Windows can be created and closed with simple Lisp functions. The size, title, and position of windows can be accessed and set through error-checked Lisp calls. In addition, all windows automatically move, resize, zoom, and close in response to appropriate mouse and keyboard events.

A window's response to any event is easily customizable; the programmer simply defines a Lisp function that is called when the event occurs.

· *Menus.* Menus are implemented as objects, with each one containing a list of menu items. A menu item consists of a title and a Lisp function that is called when the item is selected.

Menus and menu items can be changed both interactively and under program control.

· *Dialog boxes.* Dialog boxes and the items in them are implemented as objects. Like menu items, dialog items have a title and an action that will be performed when the item is selected.

Macintosh Allegro Common Lisp provides a number of dialog items as predefined objects, including buttons and radio buttons, editable and static text, scroll bars, check boxes, icons, and pop-up menus. You can also use the Macintosh List Manager to include one- and two-dimensional tables.

· *QuickDraw* routines. QuickDraw is the large set of well-defined graphics commands built into the Macintosh ROM that are used to create the Macintosh interface. Macintosh Allegro CL includes source code for a Lisp-level interface to these commands, with type-checking and argument coercion.

· *Direct system access.* Direct access to all Macintosh traps is available. Data can be directly read from or written to memory. Routines for efficient stack allocation of memory are provided.

Programming Environment

· *Editor.* Macintosh Allegro Common Lisp includes Fred, an integrated programmable editor. Fred combines the capabilities of EMACS (a popular UNIX® text editor) with the multiple-window, mouse-based editing style of the Macintosh. Matching delimiters--including parentheses, double quotation marks, and extended comment marks--automatically blink.

Fred ("Fred Resembles EMACS Deliberately") provides more than 60 keyboard commands tailored to Lisp programming needs, and is fully programmable from Lisp; any keystroke can be bound to any Lisp function. Macintosh Allegro CL provides more than 120 specialized Fred functions as a base for editor extensions.

· *Listener.* The Listener is a modified Fred window optimized for interactive Lisp programming. Text entered in the Listener by the user appears in bold type, and text printed by Lisp appears in plain type. A complete history of the user's interaction with Lisp is recorded. The Listener supports all standard Fred commands.

· *Foreign Function Interface.* The Foreign Function Interface lets you take advantage of preexisting libraries of code written in other languages, making it possible to use the optimum language for each section of a Macintosh Allegro Common Lisp program. Any compiler that produces Macintosh Programmer's Workshop (MPW) Version 2.0.2 object files can be used.

With Lisp running, a set of object files can be loaded and linked. Functions contained in these files can then be called from Lisp, and can themselves make calls to Lisp functions.

· *Stand-Alone Application Generator.* The Stand-Alone Application Generator produces ready-to-use Macintosh applications from programs written in Macintosh Allegro Common Lisp. To users, these programs are no different from Macintosh applications written in other languages; they can be run independently of the Macintosh Allegro CL development environment and launched simply by double-clicking on their icons.

Debugging Tools

· *Inspector.* A window-based Inspector allows interactive examination of all data in the operating environment.

· *Stepper.* The Stepper permits expression-by-expression evaluation of Lisp programs.

· *Stack backtrace.* The stack backtrace is available during error handling and break-loops. It displays a table of the functions on the stack at the time of the error, as well as the names and values of local slots in each frame. The functions and local slots can be examined by the Inspector and used in the Listener. The position of the program counter in each function is displayed, for use with the disassemble feature.

Features	*Benefits*
· Complete implementation of Common Lisp	· Offers more than 600 functions and supports a rich set of Lisp programming paradigms. · Provides a full Lisp programming environment on an affordable platform. · Ensures portability across different environments.
· Fast, compact compiler	· Compiles code quickly, yet runs in only 1 megabyte of memory.
· Stand-Alone Application Generator	· Turns your Lisp programs into ''double-clickable'' Macintosh applications so that anyone can run them without having Lisp itself.
· Foreign Function Interface	· Lets you use the most appropriate programming language (such as MPW C or MPW Pascal) for each section of a program.
· Integrated, mouse-based, programmable EMACS-style text editor	· Provides fast, powerful editing functions. · Can be personalized to meet your needs.
· Macintosh user interface	· Makes program development easier and more intuitive.
· User interface tools	· Lets you include standard features of the Macintosh user interface in your completed applications.
· Object-oriented development environment	· Makes it easier to program complex user interface features.
· Compatible with the MultiFinder® operating environment	· Allows Macintosh Allegro CL to compile in the background while you perform other tasks.
· Extensive on-line help	· Provides information on most built-in functions and variables (can be extended to include user-defined forms). · Offers a list of editor commands directly from a menu selection. · Lets you use Common Lisp functions to obtain information about the Lisp environment.
· Utilization of 6888x math coprocessors	· Speeds the execution of floating-point operations in your finished applications.

Macintosh Allegro Common Lisp
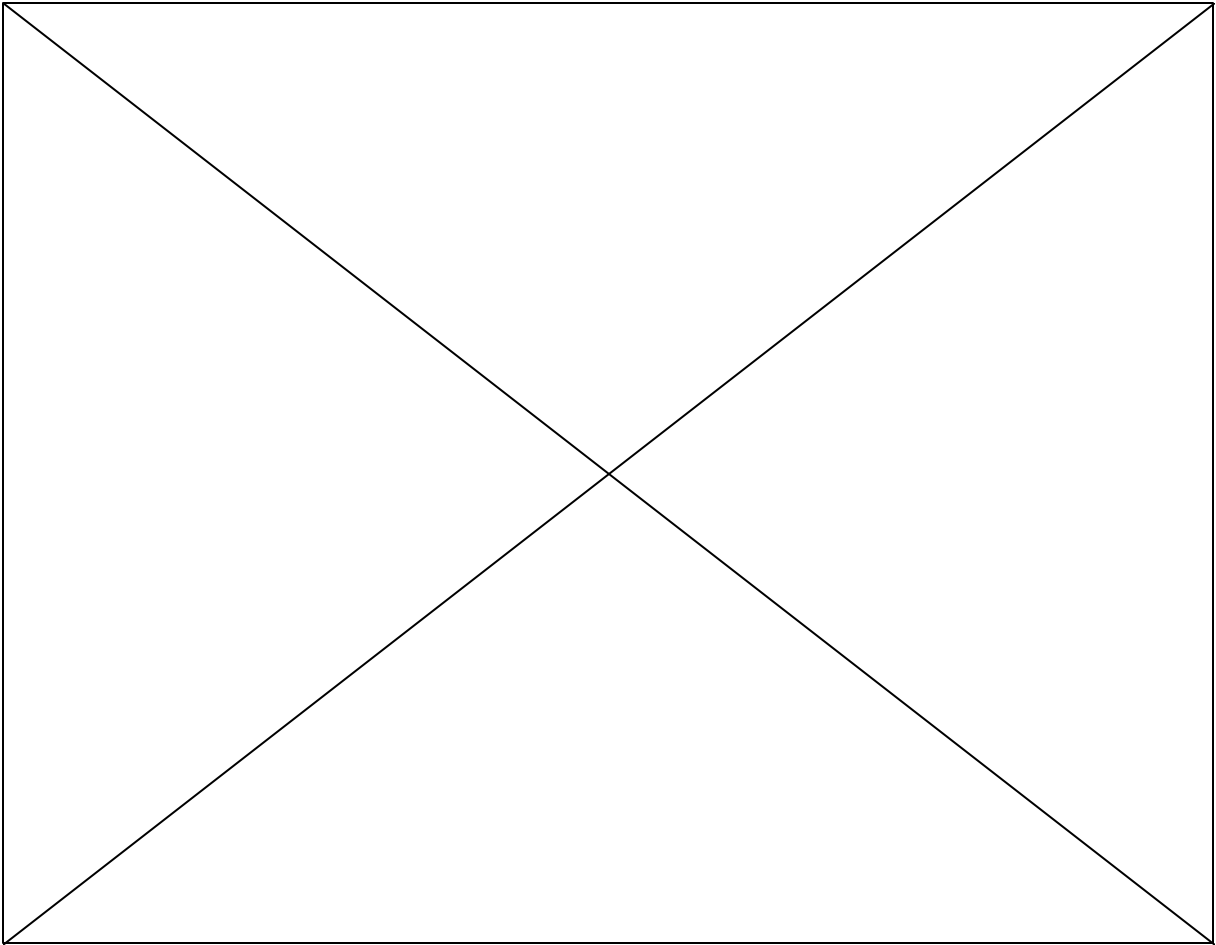
Overview

Common Lisp is the standard dialect of the Lisp programming language, which is widely used to develop artificial intelligence programs and other advanced applications and for rapid prototyping. Macintosh® Allegro Common Lisp§ is an extended implementation of Common Lisp; its versatility makes it a powerful system for developing all types of stand-alone Macintosh applications.

Macintosh Allegro Common Lisp (CL) is the first complete implementation of the dialect for any microcomputer, supporting all the features described in the definitive text Common Lisp: The Language, by Guy Steele. Because Common Lisp is available on many computers of various sizes,

applications developed on other machines can now readily be ported to the Macintosh family of systems.

The Macintosh Allegro CL compiler is fast, yet it can develop simple applications in only 1 megabyte of RAM. It produces efficient native 680x0 code that is optimized for the presence of a 6888x math coprocessor. File compilation and incremental compilation are both supported. An evaluator is provided to support expression-by-expression execution of programs, and a snapshot facility allows saving of complete Lisp environments for quick restarts.

The Stand-Alone Application Generator§ included with the package turns Macintosh Allegro

CL programs into ready-to-run Macintosh applications that users can launch with a double-click; a complete Lisp development environment is not required. The Foreign Function Interface§ lets you maximize your programming efficiency by using different programming languages, such as MPW§ C or MPW Pascal, for different sections of your program.

The package also contains high-level Macintosh interface tools, a programmable mouse-based editor with EMACS-style features, object-oriented programming extensions, and other development and debugging tools.