

Le présent fichier PDF **Sommaire & introduction** concerne le document

ExperLisp, For the MacIntosh

Il s'agit d'un Manuel de 220 pages pour un logiciel créé en 1985-1986 par la société Californienne **ExperTelligence, Inc.**

Le directeur de cette société est Denison Bollay (voir 3^{ème} page de couverture) et l'Auteur du manuel est Dean A. Ritz

A l'intérieur de ce Manuel, deux disquettes de 3" pour la version 1.5 du logiciel ExperLisp, un langage de programmation LISP.

Cet ouvrage est disponible sous le numéro **AC 18215** dans l'inventaire de la collection de l'ACONIT, Association pour un conservatoire de l'informatique et de la télématique, Grenoble.

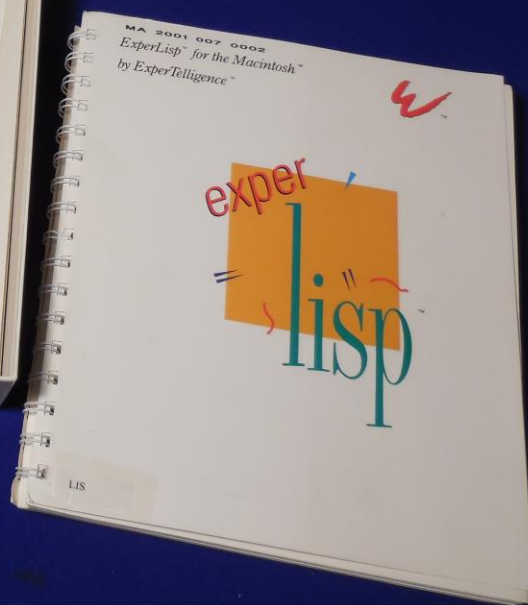
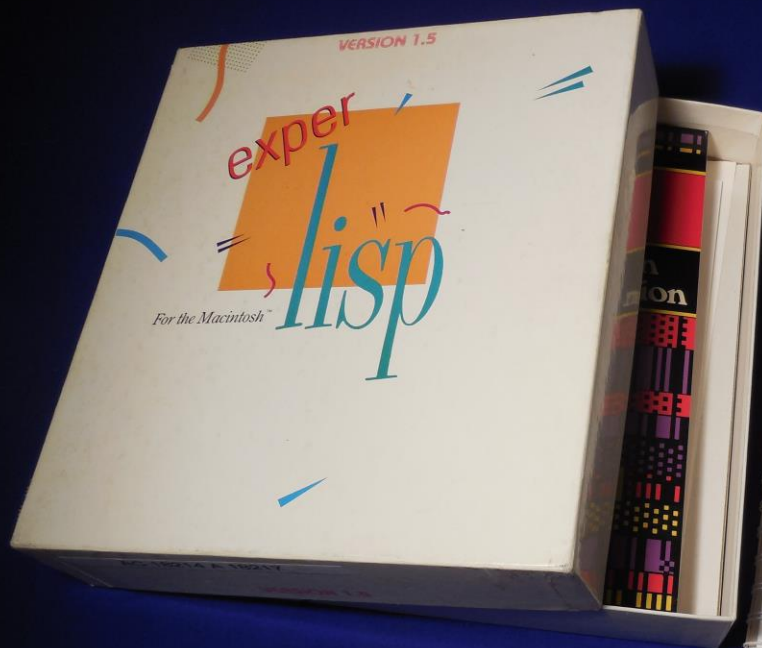
Auparavant il appartenait aux collections de la Bibliothèque Municipale de Grenoble, d'où la référence MA 2001 007 0003 qui apparaît sur la page de couverture.

Le présent PDF fournit un scan d'extraits de cet ouvrage, avec copie de :

- une photo du Manuel à coté du coffret le contenant ainsi que plusieurs autres documents la page de couverture du Manuel,
- la 3^{ème} page de couverture avec les deux disquettes 3" du logiciel,
- la 2^{ème} page de couverture avec par Denison Bollay la présentation du logiciel,
- la 4^{ème} page de couverture mentionnant la société ExperTelligence,
- la page d'en-tête,
- les 2 pages du Sommaire,
- les pages 7 à 12 d'introduction « in the Beginning ».

Pour utiliser ce document, mentionnez l'auteur Dean A. Ritz, la société ExperTelligence, ainsi que la Base de Données ACONIT n° 18215.

Il est rappelé que la loi sur les droits d'auteur n'autorise que les reproductions partielles de documents à titre individuel et pour des motifs de recherche.



ExperLisp™ for the Macintosh™
by *ExperTelligence™*



ExpertLisp. At Expertelligence we
are products that introduce and
Artificial Intelligence.

use a serious LISP environment for
every line. It is the first compiled
for a microcomputer, offering speed
and implementations. Because of this,
development environment for Expert
indicated applications.

just the beginning for ExpertLisp.
to extend ExpertLisp to make it an
development environment. We encourage
submitting your ideas.

your owner's registration card. This
receive all information concerning

VERSION 1.5

expert
lisp

LO 2001 007 0001

© Copyright Expertelligence™ 1985



VERSION 1.5

expert
lisp

FILE COMPILER SAMPLE PROGRAMS

© Copyright Expertelligence™ 1985



IMPORTANT READ BEFORE OPENING PRODUCT

If you open the sealed plastic envelope containing the diskette, it signifies your unconditional acceptance of, and agreement to comply with, the terms and conditions of sale stated in the licensing agreement on the inside of the back cover of this manual. Therefore, do not open this envelope until you have read and agreed to the license. If you cannot accept and comply, please return the product(s) unopened to the original product retailer within 14 days from date of purchase. You should also take a minute now to complete the owner's registration certificate and return it to Expertelligence. Thank you for your cooperation.

Expertelligence
559 San Ysidro Road
Santa Barbara, CA 93108
(805) 969-7874

W

March 27, 1985

Dear ExpertLisp Owner:

Thank you for purchasing ExpertLisp. At Expertelligence we develop and market software products that introduce and apply the concepts of Artificial Intelligence.

We are excited to introduce a serious LISP environment for the Apple Macintosh computer line. It is the first compiled implementation of LISP for a microcomputer, offering speed that rivals minicomputer implementations. Because of this, ExpertLisp is a superior development environment for Expert Systems and other sophisticated applications.

Release 1.0 represents just the beginning for ExpertLisp. We are already working to extend ExpertLisp to make it an even more powerful development environment. We encourage you to work with us by submitting your ideas.

Be sure to send in your owner's registration card. This will insure that you receive all information concerning future releases.

Sincerely,

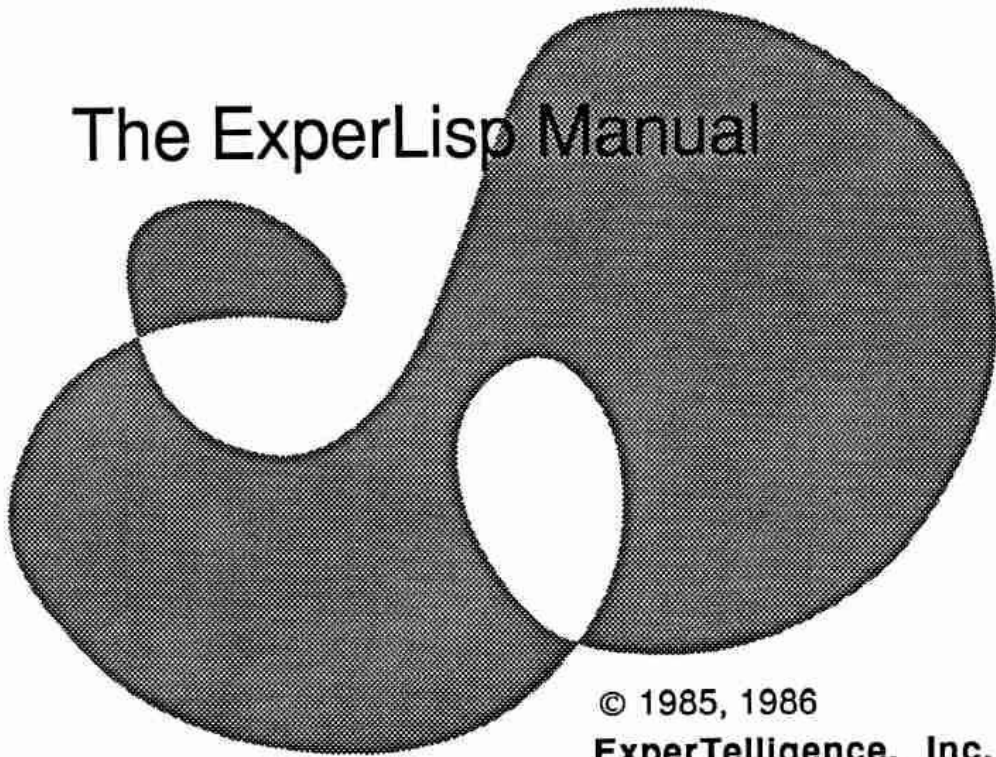
D. Bollay

Denison Bollay
President
Expertelligence, Inc.



ExperTelligence, Inc.
559 San Ysidro Road
Santa Barbara, CA 93108
(805) 969-7874

The ExperLisp Manual



© 1985, 1986
ExperTelligence, Inc.

Created by Dean A. Ritz
Assisted by Emily K. Celmins
& Jon R. Richter

Table Of Contents

Straight Talk	5
In the Beginning	7
In This Box you Should Have Found:	7
Hardware Requirements	7
The Purpose of the ExperLisp Manual	7
Lisp - A Gentle Introduction to Symbolic Computation	8
In Case You Haven't Heard: ExperLisp is...	8
Origins of ExperLisp	10
Other Influences	10
Future Directions	11
Currently Available ExperLisp Enhancements	12
About the ExperLisp Environment	13
The Pull-Down Menus	14
Writing, Editing, and Compiling ExperLisp Programs	17
Cursor Movement	18
Writing an ExperLisp Program	19
Saving Files...	21
Retrieving Files...	23
Printing Text Windows	24
ExperLisp Data Types	27
Integer Numbers	27
Floating Point Numbers	28
Symbols	28
Strings	29
Lists	30
Pointers	30
Handles	30
Arrays	31
Streams	31
Functions	32
ExperLisp Graphics	33
Window Fun	34
2-Dimensional Bunny Graphics	35
3-Dimensional Bunny Graphics	38
Spherical Bunny Graphics	39
Hutch Management	41

Quickdraw Graphics	43
Quickdraw Regions and Polygons	45
Menu Management	51
The Reference Section	59
Reference Section Overview	61
Categorical Listing	63
Alphabetical Listing	75
Appendices	187
Files on Your ExperLisp Disk	189
More On Graphics	191
Graphics Field	191
Home and Origin	192
Manipulating Graphics Windows	193
Manipulating the Listener Window	194
Creating New Graphics Windows	195
Sending Graphics to the Window of Your Choice	196
Saving and Printing Graphics	196
Integer Graphics	196
Coordinate System	197
Simple Animation Using Regions	197
Escaping From an Infinite Loop	198
Saving the Source Code of Functions	199
Memory Management	201
Using Handles	201
Garbage Collection	206
Tell The Driver	207
Creating Patterns	211
Changing an Existing Pattern	211
The Pattern of Patterns	212
Calling an Assembly Language Routine from Lisp	217
Argument Passing to a Builtin	217
Responsibilities	217
The Dirty Work	218
Sample Program	218

In the Beginning

In This Box You Should Have Found:

- (1) Two copies of the ExperLisp system disk. Please note there is no System Folder on these disks. Use a system diskette to boot the Macintosh.
- (2) A copy of the ExperLisp Manual from which you are reading.
- (3) A copy of *Lisp - A Gentle Introduction to Symbolic Computation* by David S. Touretzky, which is supplied as a Lisp tutorial.

Hardware Requirements

ExperLisp requires a Macintosh computer or Mac XL with at least 512k of memory and two disk drives or a hard drive. ExperLisp operates under revision C or later of Macworks. ExperLisp will take advantage of additional memory up to 4 megabytes.

The Purpose of the ExperLisp Manual

The purpose of this manual is to provide a brief overview of the ExperLisp implementation of Lisp. It assumes knowledge of Lisp and its uses. It also assumes knowledge of the Macintosh features. This manual works to tell you "about" rather than "why".

This manual is tutorial in nature when it covers features unique to ExperLisp; it relies on the Reference Section for additional detail. Most of the manual is devoted to the Reference Section. This section begins with a grouping of primitives by category, followed by a detailed description of each arranged alphabetically.

The descriptions are by no means exhaustive explanations of the primitives, but they still provide much valuable information. It is recommended that you refer to additional texts such as *Common Lisp: The Language* by Guy L. Steele Jr., and *Inside Macintosh* by Apple Computer Inc., for more information.

Lisp - A Gentle Introduction to Symbolic Computation

David S. Touretzky's book has been included to aid those who are unfamiliar with Lisp. It is an excellent introductory book covering relevant Lisp topics which apply to all dialects of the language.

There are some differences between the implementations Touretzky describes and ExperLisp. To bridge these differences, a special file named TOURETZKY has been included. When compiled, the contents of the file slightly alters the ExperLisp dialect to more closely adhere to his text. The changes made by compiling the file TOURETZKY are not permanent. To revert back to normal ExperLisp simply exit and re-open the application. The TOURETZKY file can be removed from your copy of ExperLisp when this option is no longer desired.

In Case You Haven't Heard: ExperLisp is...

Compiled - ExperLisp is the first personal computer implementation which is compiled. It performs at speeds up to 10 faster than interpreted versions. No provision has been made to save your compiled code in the initial release. A file compiler will be available in the near future, as well as the ability to save a "snapshot" of memory.

Graphics - ExperLisp provides a flexible graphics package. Graphics objects called 'bunnies' enable the creation of 2-dimensional, 3-dimensional, and spherical graphics. Access to the Quickdraw routines of the Macintosh toolbox enables complex manipulation of defined and arbitrary shapes.

Toolbox Access - ExperLisp programs can incorporate windows, menus, the mouse, memory management, and other features of the Macintosh toolbox.

Lexically Scoped - ExperLisp is lexically scoped. This means symbols are bound within the scope of their containing body. References to a symbol will refer to its current binding. Lexical scoping prevents conflicts between different variables with the same names. For example, the variables in the lambda-list of a function are lexically scoped.

```
(SETQ X 5)
-> 5
(DEFUN FOO (X) (PRINT X) (GLUE) (PRINT X))
-> FOO
```

```

(DEFUN GLUE () (PRINT X) (SETQ X (ADD1 X)))
-> GLUE
(FOO 10)
    10
    5
    10
-> 10
X
-> 6

```

We will be implementing full closures and imbedded lambdas in a future compiler. You can declare variables SPECIAL to impose dynamic scoping of them. See SPECIAL in the Reference Section for more information.

First Class Functions - ExperLisp functions are first class. It simplifies some customization of your environment. For example, if you wish to rename LISTP to LIST? {for Scheme users} it is accomplished by:

```

(SETQ LIST? LISTP)
-> FUNCTION #6801362A

```

 This example, as others in this manual, uses an arrow '->' to indicate the result that is returned by executing the form which lies immediately above (used to increase the clarity and readability of this document). The ExperLisp application, though, prints a semi-colon ';' rather than an arrow. The above example would actually be printed by ExperLisp as:

```

(SETQ LIST? LISTP)
;FUNCTION #6801362A

```

One may take advantage of the first classness of functions to:

```

(DEFUN NUMS-TO-TEN (FUNCTION)
  (FUNCTION 1 2 3 4 5 6 7 8 9 10))
-> NUMS-TO-TEN
(NUMS-TO-TEN +)
-> 55

```

```
(NUMS-TO-TEN  *)  
-> 3628800.
```

Origins of ExperLisp

Once upon a time, ExperTelligence had been experimenting with expert systems on a Symbolics 3600 Lisp Machine, and was looking for a way to develop and market expert systems on an affordable but very powerful computer. But alas![†]...no suitable machine existed.

Then came the introduction of the Apple Macintosh. The Macintosh provided both the features and the muscle necessary for expert systems. It still lacked, however, a suitable environment for AI applications. Thus, ExperTelligence embarked upon the development of a Lisp development environment.

ExperTelligence developed a compiler, customized for the unique qualities of the Macintosh and its toolbox, and for the power of the Motorola 68000 microprocessor. The development environment was be customized for the Macintosh, with primitives written in 68000 assembly language and Lisp rather than C or Pascal.

To obtain the maximum speed, the builtin functions {i.e. CAR, CDR, CONS...there are over 450 of them} were written in hand crafted assembly language. Every trick was used to optimize their performance. For example, LISTP is a BMI instruction, and NULL is a BEQ. Thusly, at every possible point, ExperTelligence worked to take advantage of the powerful capabilities of the Macintosh and in doing so, developed a user oriented language, ExperLisp.

Other Influences

ZetaLisp - The development of ExperLisp has some close ties with ZetaLisp. The ExperLisp compiler was written in ZetaLisp. ExperTelligence also created the cross compiler used on the Symbolics 3600 Lisp Machine to generate 68000 code for the Macintosh. Contact ExperTelligence regarding a license for this product.

[†] "Alas" is an English expression which has its roots in the Greek phrase "ailinon," meaning "Woe for Linus." In mythology Linus was the son of Apollo who was torn to death by dogs before he reached maturity. "Alas" was used to express lament over something which perished before it had the chance to bear fruit. "Alas" is now used to express lament over anything, such as the lack of a suitable machine on which to produce and deliver AI (Artificial Intelligence) applications.

Common Lisp - Common Lisp appears to be the upcoming standard for Lisp, at least for new implementations. In order to broaden its appeal, as well as increase its portability, ExperLisp will continue moving towards Common Lisp. With the upcoming "snapshot" method for saving environments, and the ease with which this system is extensible, most users are able to redefine primitives to closely adhere to the Lisp dialect of their choice.

The Macintosh - ExperLisp release 1.0 offers extensive access to toolbox routines of the Macintosh. We retained the names given these routines by Apple, opening a wider source of support for ExperLisp users. The *Inside Macintosh* documentation is highly recommended because it provides detailed insights of many of our primitives and of the Macintosh.

Future Directions

 Please be sure to send in the completed registration form to insure your receipt of update materials and relevant information. We cannot send any information if we don't know who you are.

ExperLisp continues to be under active development. A future release is to be named ExperCommonLisp. Some features include:

Stand-alone Development - The goal of this developer's version is to allow a more powerful development environment for the Macintosh.

Snapper saves a snapshot of the contents of Macintosh memory. This is similar to "worlds" on the Symbolics. The system can be loaded from any snapshot, restoring the exact state at which the environment was saved. A developer could set up an application and then take a snapshot. When this application is booted by opening the icon, a user will not even know that it is implemented in Lisp.

Additionally, a file compiler will allow your compiled code to be saved and loaded from disk. ExperTelligence will use this feature to keep you abreast of the latest advancements of ExperCommonLisp.

Object Orientation - ExperCommonLisp includes full facilities for object oriented programming with inheritance. For maximum speed, all of its class system primitives are written in 68000 assembly language

A widely recognized purely object oriented programming environment is Xerox's Smalltalk-80. An object oriented system allows the creation of black-box programs (methods) which can be used to modify objects (instances) of a particular structure (class). These advantage are integrated into ExperCommonLisp.

Currently Available ExperLisp Enhancements

ExperLisp-Talk, an add-on product available only from Expertelligence, gives your Macintosh the freedom of speech. It enables ExperLisp programs to integrate real-time synthesis of human speech from unrestricted English text.

Expert Systems - ExperLisp was initially created to ease the development of expert systems. Expertelligence currently markets two expert system shells: ExperOPS5, implemented at Science Applications International Corp. (SAIC), ExperFACTS developed at MC-Systems. Both of these products operate in the ExperLisp environment. Several other expert systems are currently under development in ExperLisp.

ExperLink - ExperLink is a communications package which links an Apple Macintosh running ExperLisp, to a Symbolics Lisp Machine. Users can transfer files between the two machines, as well as create a Symbolics Lisp Listener in ExperLisp. Programs can be written and edited in ExperLisp, and remotely compiled on the Symbolics. The special forms MAC-EVAL and SYMBO-EVAL enable ExperLisp programs to mix forms for evaluation on the Macintosh and on the Symbolics.