

MASTER TABLE OF CONTENTS

1. Introduction	1.1
1.1. Interlisp as a Programming Language	1.1
1.2. Interlisp as an Interactive Environment	1.3
1.3. Interlisp Philosophy	1.5
1.4. How to Use this Manual	1.7
1.5. References	1.8
2. Litatoms	2.1
2.1. Using Litatoms as Variables	2.2
2.2. Function Definition Cells	2.5
2.3. Property Lists	2.5
2.4. Print Names	2.7
2.5. Characters and Character Codes	2.12
3. Lists	3.1
3.1. Creating Lists	3.4
3.2. Building Lists From Left to Right	3.6
3.3. Copying Lists	3.8
3.4. Extracting Tails of Lists	3.9
3.5. Counting List Cells	3.10
3.6. Logical Operations	3.11
3.7. Searching Lists	3.12
3.8. Substitution Functions	3.13
3.9. Association Lists and Property Lists	3.15
3.10. Sorting Lists	3.17
3.11. Other List Functions	3.19
4. Strings	4.1
5. Arrays	5.1
6. Hash Arrays	6.1
6.1. Hash Overflow	6.3

6.2. User-Specified Hashing Functions	6.4
7. Numbers and Arithmetic Functions	7.1
7.1. Generic Arithmetic	7.3
7.2. Integer Arithmetic	7.4
7.3. Logical Arithmetic Functions	7.8
7.4. Floating Point Arithmetic	7.11
7.5. Other Arithmetic Functions	7.13
8. Record Package	8.1
8.1. FETCH and REPLACE	8.2
8.2. CREATE	8.3
8.3. TYPE?	8.5
8.4. WITH	8.5
8.5. Record Declarations	8.6
8.5.1. Record Types	8.7
8.5.2. Optional Record Specifications	8.14
8.6. Defining New Record Types	8.15
8.7. Record Manipulation Functions	8.16
8.8. Changetran	8.17
8.9. Built-In and User Data Types	8.20
9. Conditionals and Iterative Statements	9.1
9.1. Data Type Predicates	9.1
9.2. Equality Predicates	9.2
9.3. Logical Predicates	9.3
9.4. The COND Conditional Function	9.4
9.5. The IF Statement	9.5
9.6. Selection Functions	9.6
9.7. PROG and Associated Control Functions	9.7
9.8. The Iterative Statement	9.9
9.8.1. I.s.types	9.10
9.8.2. Iteration Variable I.s.ops	9.12
9.8.3. Condition I.s.ops	9.15
9.8.4. Other I.s.ops	9.16
9.8.5. Miscellaneous Hints on I.S.Oprs	9.17
9.8.6. Errors in Iterative Statements	9.19

	9.8.7. Defining New Iterative Statement Operators	9.20
10. Function Definition, Manipulation, and Evaluation		10.1
10.1. Function Types		10.2
10.1.1. Lambda-Spread Functions		10.3
10.1.2. Nlambda-Spread Functions		10.4
10.1.3. Lambda-Nospread Functions		10.5
10.1.4. Nlambda-Nospread Functions		10.6
10.1.5. Compiled Functions		10.6
10.1.6. Function Type Functions		10.6
10.2. Defining Functions		10.9
10.3. Function Evaluation		10.11
10.4. Iterating and Mapping Functions		10.14
10.5. Functional Arguments		10.18
10.6. Macros		10.21
10.6.1. DEFMACRO		10.24
10.6.2. Interpreting Macros		10.28
11. Variable Bindings and the Interlisp Stack		11.1
11.1. The Spaghetti Stack		11.2
11.2. Stack Functions		11.4
11.2.1. Searching the Stack		11.5
11.2.2. Variable Bindings in Stack Frames		11.6
11.2.3. Evaluating Expressions in Stack Frames		11.7
11.2.4. Altering Flow of Control		11.8
11.2.5. Releasing and Reusing Stack Pointers		11.9
11.2.6. Backtrace Functions		11.11
11.2.7. Other Stack Functions		11.13
11.3. The Stack and the Interpreter		11.14
11.4. Generators		11.16
11.5. Coroutines		11.18
11.6. Possibilities Lists		11.20
12. Miscellaneous		12.1
12.1. Greeting and Initialization Files		12.1
12.2. Idle Mode		12.4
12.3. Saving Virtual Memory State		12.6

12.4. System Version Information	12.11
12.5. Date And Time Functions	12.13
12.6. Timers and Duration Functions	12.16
12.7. Resources	12.19
12.7.1. A Simple Example	12.20
12.7.2. Trade-offs in More Complicated Cases	12.22
12.7.3. Macros for Accessing Resources	12.23
12.7.4. Saving Resources in a File	12.23
12.8. Pattern Matching	12.24
12.8.1. Pattern Elements	12.25
12.8.2. Element Patterns	12.25
12.8.3. Segment Patterns	12.27
12.8.4. Assignments	12.28
12.8.5. Place-Markers	12.29
12.8.6. Replacements	12.29
12.8.7. Reconstruction	12.30
12.8.8. Examples	12.31
13. Interlisp Executive	13.1
13.1. Input Formats	13.3
13.2. Programmer's Assistant Commands	13.5
13.2.1. Event Specification	13.6
13.2.2. Commands	13.8
13.2.3. P.A. Commands Applied to P.A. Commands	13.20
13.3. Changing The Programmer's Assistant	13.21
13.4. Undoing	13.26
13.4.1. Undoing Out of Order	13.27
13.4.2. SAVESET	13.28
13.4.3. UNDONLSETQ and RESETUNDO	13.29
13.5. Format and Use of the History List	13.31
13.6. Programmer's Assistant Functions	13.35
13.7. The Editor and the Programmer's Assistant	13.43
14. Errors and Breaks	14.1
14.1. Breaks	14.1
14.2. Break Windows	14.3

14.3. Break Commands	14.5
14.4. Controlling When to Break	14.13
14.5. Break Window Variables	14.14
14.6. Creating Breaks with BREAK1	14.16
14.7. Signalling Errors	14.19
14.8. Catching Errors	14.21
14.9. Changing and Restoring System State	14.24
14.10. Error List	14.27
15. Breaking, Tracing, and Advising	15.1
15.1. Breaking Functions and Debugging	15.1
15.2. Advising	15.9
15.2.1. Implementation of Advising	15.10
15.2.2. Advise Functions	15.10
16. List Structure Editor	16.1
16.1. DEdit	16.1
16.1.1. Calling DEdit	16.2
16.1.2. Selecting Objects and Lists	16.4
16.1.3. Typing Characters to DEdit	16.5
16.1.4. Copy-Selection	16.5
16.1.5. DEdit Commands	16.6
16.1.6. Multiple Commands	16.10
16.1.7. DEdit Idioms	16.10
16.1.8. DEdit Parameters	16.12
16.2. Local Attention-Changing Commands	16.13
16.3. Commands That Search	16.18
16.3.1. Search Algorithm	16.20
16.3.2. Search Commands	16.21
16.3.3. Location Specification	16.23
16.4. Commands That Save and Restore the Edit Chain	16.27
16.5. Commands That Modify Structure	16.29
16.5.1. Implementation	16.30
16.5.2. The A, B, and : Commands	16.31
16.5.3. Form Oriented Editing and the Role of UP	16.34
16.5.4. Extract and Embed	16.35

16.5.5. The MOVE Command	16.37
16.5.6. Commands That Move Parentheses	16.40
16.5.7. TO and THRU	16.42
16.5.8. The R Command	16.45
16.6. Commands That Print	16.47
16.7. Commands for Leaving the Editor	16.49
16.8. Nested Calls to Editor	16.51
16.9. Manipulating the Characters of an Atom or String	16.52
16.10. Manipulating Predicates and Conditional Expressions	16.53
16.11. History commands in the editor	16.54
16.12. Miscellaneous Commands	16.55
16.13. Commands That Evaluate	16.57
16.14. Commands That Test	16.60
16.15. Edit Macros	16.62
16.16. Undo	16.64
16.17. EDITDEFAULT	16.66
16.18. Editor Functions	16.68
16.19. Time Stamps	16.76
17. File Package	17.1
17.1. Loading Files	17.5
17.2. Storing Files	17.10
17.3. Remaking a Symbolic File	17.15
17.4. Loading Files in a Distributed Environment	17.16
17.5. Marking Changes	17.17
17.6. Noticing Files	17.19
17.7. Distributing Change Information	17.21
17.8. File Package Types	17.21
17.8.1. Functions for Manipulating Typed Definitions	17.24
17.8.2. Defining New File Package Types	17.29
17.9. File Package Commands	17.32
17.9.1. Functions and Macros	17.34
17.9.2. Variables	17.35
17.9.3. Litatom Properties	17.37
17.9.4. Miscellaneous File Package Commands	17.38

17.9.5. DECLARE:	17.40
17.9.6. Exporting Definitions	17.42
17.9.7. FileVars	17.44
17.9.8. Defining New File Package Commands	17.45
17.10. Functions for Manipulating File Command Lists	17.48
17.11. Symbolic File Format	17.50
17.11.1. Copyright Notices	17.52
17.11.2. Functions Used Within Source Files	17.54
17.11.3. File Maps	17.55
18. Compiler	18.1
18.1. Compiler Printout	18.3
18.2. Global Variables	18.4
18.3. Local Variables and Special Variables	18.5
18.4. Constants	18.7
18.5. Compiling Function Calls	18.8
18.6. FUNCTION and Functional Arguments	18.10
18.7. Open Functions	18.11
18.8. COMPILETYPELIST	18.11
18.9. Compiling CLISP	18.11
18.10. Compiler Functions	18.13
18.11. Block Compiling	18.17
18.11.1. Block Declarations	18.17
18.11.2. Block Compiling Functions	18.20
18.12. Compiler Error Messages	18.22
19. Masterscope	19.1
19.1. Command Language	19.3
19.1.1. Commands	19.4
19.1.2. Relations	19.7
19.1.3. Set Specifications	19.10
19.1.4. Set Determiners	19.13
19.1.5. Set Types	19.13
19.1.6. Conjunctions of Sets	19.14
19.2. SHOW PATHS	19.15
19.3. Error Messages	19.17

19.4. Macro Expansion	19.17
19.5. Affecting Masterscope Analysis	19.18
19.6. Data Base Updating	19.22
19.7. Masterscope Entries	19.22
19.8. Noticing Changes that Require Recompiling	19.25
19.9. Implementation Notes	19.25
20. DWIM	20.1
20.1. Spelling Correction Protocol	20.4
20.2. Parentheses Errors Protocol	20.5
20.3. Undefined Function T Errors	20.6
20.4. DWIM Operation	20.7
20.4.1. DWIM Correction: Unbound Atoms	20.8
20.4.2. Undefined CAR of Form	20.9
20.4.3. Undefined Function in APPLY	20.10
20.5. DWIMUSERFORMS	20.11
20.6. DWIM Functions and Variables	20.13
20.7. Spelling Correction	20.15
20.7.1. Synonyms	20.16
20.7.2. Spelling Lists	20.16
20.7.3. Generators for Spelling Correction	20.19
20.7.4. Spelling Corrector Algorithm	20.19
20.7.5. Spelling Corrector Functions and Variables	20.21
21. CLISP	21.1
21.1. CLISP Interaction with User	21.6
21.2. CLISP Character Operators	21.7
21.3. Declarations	21.12
21.4. CLISP Operation	21.14
21.5. CLISP Translations	21.17
21.6. DWIMIFY	21.18
21.7. CLISPIFY	21.22
21.8. Miscellaneous Functions and Variables	21.25
21.9. CLISP Internal Conventions	21.27
22. Performance Issues	22.1
22.1. Storage Allocation and Garbage Collection	22.1

22.2. Variable Bindings	22.5
22.3. Performance Measuring	22.7
22.3.1. BREAKDOWN	22.9
22.4. GAINSPACE	22.11
22.5. Using Data Types Instead of Records	22.13
22.6. Using Incomplete File Names	22.13
22.7. Using "Fast" and "Destructive" Functions	22.14
23. Processes	23.1
23.1. Creating and Destroying Processes	23.2
23.2. Process Control Constructs	23.5
23.3. Events	23.7
23.4. Monitors	23.8
23.5. Global Resources	23.10
23.6. Typein and the TTY Process	23.11
23.6.1. Switching the TTY Process	23.12
23.6.2. Handling of Interrupts	23.14
23.7. Keeping the Mouse Alive	23.15
23.8. Process Status Window	23.16
23.9. Non-Process Compatibility	23.17
24. Streams and Files	24.1
24.1. Opening and Closing File Streams	24.2
24.2. File Names	24.5
24.3. Incomplete File Names	24.9
24.4. Version Recognition	24.11
24.5. Using File Names Instead of Streams	24.13
24.5.1. File Name Efficiency Considerations	24.14
24.5.2. Obsolete File Opening Functions	24.14
24.5.3. Converting Old Programs	24.15
24.6. Using Files with Processes	24.16
24.7. File Attributes	24.17
24.8. Closing and Reopening Files	24.20
24.9. Local Hard Disk Device	24.21
24.10. Floppy Disk Device	24.24
24.11. I/O Operations to and from Strings	24.28

24.12. Temporary Files and the CORE Device	24.29
24.13. NULL Device	24.30
24.15. Deleting, Copying, and Renaming Files	24.31
24.16. Searching File Directories	24.31
24.17. Listing File Directories	24.33
24.18. File Servers	24.36
24.18.1. Pup File Server Protocols	24.36
24.18.2. Xerox NS File Server Protocols	24.37
24.18.3. Operating System Designations	24.38
24.18.4. Logging In	24.39
24.18.5. Abnormal Conditions	24.41
25. Input/Output Functions	25.1
25.1. Specifying Streams for Input/Output Functions	25.1
25.2. Input Functions	25.2
25.3. Output Functions	25.7
25.3.1. PRINTLEVEL	25.11
25.3.2. Printing numbers	25.13
25.3.3. User Defined Printing	25.16
25.3.4. Printing Unusual Data Structures	25.17
25.4. Random Access File Operations	25.18
25.5. Input/Output Operations with Characters and Bytes	25.22
25.6. PRINTOUT	25.23
25.6.1. Horizontal Spacing Commands	25.25
25.6.2. Vertical Spacing Commands	25.26
25.6.3. Special Formatting Controls	25.27
25.6.4. Printing Specifications	25.27
25.6.4.1. Paragraph Format	25.28
25.6.4.2. Right-Flushing	25.29
25.6.4.3. Centering	25.29
25.6.4.4. Numbering	25.29
25.6.5. Escaping to Lisp	25.30
25.6.6. User-Defined Commands	25.31
25.6.7. Special Printing Functions	25.32
25.7. READFILE and WRITEFILE	25.33

25.8. Read Tables	25.33
25.8.1. Read Table Functions	25.34
25.8.2. Syntax Classes	25.35
25.8.3. Read Macros	25.39
26. User Input/Output Packages	26.1
26.1. Inspector	26.1
26.1.1. Calling the Inspector	26.2
26.1.2. Multiple Ways of Inspecting	26.2
26.1.3. Inspect Windows	26.3
26.1.4. Inspect Window Commands	26.4
26.1.5. Interaction With Break Windows	26.5
26.1.6. Controlling the Amount Displayed During Inspection	26.5
26.1.7. Inspect Macros	26.6
26.1.8. INSPECTWS	26.6
26.2. PROMPTFORWORD	26.9
26.3. ASKUSER	26.12
26.3.1. Format of KEYLST	26.13
26.3.2. Options	26.15
26.3.3. Operation	26.17
26.3.4. Completing a Key	26.18
26.3.5. Special Keys	26.19
26.3.6. Startup Protocol and Typeahead	26.20
26.4. TTYIN Display Typein Editor	26.22
26.4.1. Entering Input With TTYIN	26.22
26.4.2. Mouse Commands [Interlisp-D Only]	26.24
26.4.3. Display Editing Commands	26.25
26.4.4. Using TTYIN for Lisp Input	26.28
26.4.5. Useful Macros	26.29
26.4.6. Programming With TTYIN	26.29
26.4.7. Using TTYIN as a General Editor	26.32
26.4.8. ? = Handler	26.33
26.4.9. Read Macros	26.34
26.4.10. Assorted Flags	26.36

26.4.11. Special Responses	26.38
26.4.12. Display Types	26.38
26.5. Prettyprint	26.39
26.5.1. Comment Feature	26.42
26.5.2. Comment Pointers	26.44
26.5.3. Converting Comments to Lower Case	26.46
26.5.4. Special Prettyprint Controls	26.47
27. Graphics Output Operations	27.1
27.1. Primitive Graphics Concepts	27.1
27.1.1. Positions	27.1
27.1.2. Regions	27.1
27.1.3. Bitmaps	27.3
27.1.4. Textures	27.6
27.2. Opening Image Streams	27.8
27.3. Accessing Image Stream Fields	27.10
27.4. Current Position of an Image Stream	27.13
27.5. Moving Bits Between Bitmaps With BITBLT	27.14
27.6. Drawing Lines	27.17
27.7. Drawing Curves	27.18
27.8. Miscellaneous Drawing and Printing Operations	27.20
27.9. Drawing and Shading Grids	27.22
27.10. Display Streams	27.23
27.12. Fonts	27.25
27.13. Font Files and Font Directories	27.31
27.15. Font Profiles	27.32
27.16. Image Objects	27.35
27.16.1. IMAGEFNS Methods	27.36
27.16.2. Registering Image Objects	27.39
27.16.3. Reading and Writing Image Objects on Files	27.40
27.16.4. Copying Image Objects Between Windows	27.41
27.17. Implementation of Image Streams	27.42
28. Windows and Menus	28.1
28.1. Using The Window System	28.2
28.2. Changing Window Command Menus	28.7

28.3. Interactive Display Functions	28.9
<hr/>	
28.4. Windows	28.12
<hr/>	
28.4.1. Window Properties	28.13
<hr/>	
28.4.2. Creating Windows	28.13
<hr/>	
28.4.3. Opening and Closing Windows	28.15
<hr/>	
28.4.4. Redisplaying Windows	28.16
<hr/>	
28.4.5. Reshaping Windows	28.16
<hr/>	
28.4.6. Moving Windows	28.19
<hr/>	
28.4.7. Exposing and Burying Windows	28.20
<hr/>	
28.4.8. Shrinking Windows Into Icons	28.21
<hr/>	
28.4.9. Coordinate Systems, Extents, And Scrolling	28.23
<hr/>	
28.4.10. Mouse Activity in Windows	28.27
<hr/>	
28.4.11. Terminal I/O and Page Holding	28.29
<hr/>	
28.4.12. The TTY Process and the Caret	28.30
<hr/>	
28.4.13. Miscellaneous Window Functions	28.31
<hr/>	
28.4.14. Miscellaneous Window Properties	28.33
<hr/>	
28.4.15. Example: A Scrollable Window	28.34
<hr/>	
28.5. Menus	28.37
<hr/>	
28.5.1. Menu Fields	28.38
<hr/>	
28.5.2. Miscellaneous Menu Functions	28.42
<hr/>	
28.5.3. Examples of Menu Use	28.43
<hr/>	
28.6. Attached Windows	28.45
<hr/>	
28.6.1. Attaching Menus To Windows	28.48
<hr/>	
28.6.2. Attached Prompt Windows	28.50
<hr/>	
28.6.3. Window Operations And Attached Windows	28.50
<hr/>	
28.6.4. Window Properties Of Attached Windows	28.53
<hr/>	
29. Hardcopy Facilities	29.1
<hr/>	
29.1. Low-level Hardcopy Variables	29.5
<hr/>	
30. Terminal Input/Output	30.1
<hr/>	
30.1. Interrupt Characters	30.1
<hr/>	
30.2. Terminal Tables	30.4
<hr/>	
30.2.1. Terminal Syntax Classes	30.5
<hr/>	
30.2.2. Terminal Control Functions	30.6
<hr/>	
30.2.3. Line-Buffering	30.9

30.3. Dribble Files	30.12
30.4. Cursor and Mouse	30.13
30.4.1. Changing the Cursor Image	30.13
30.4.2. Flashing Bars on the Cursor	30.16
30.4.3. Cursor Position	30.17
30.4.4. Mouse Button Testing	30.17
30.4.5. Low Level Mouse Functions	30.18
30.5. Keyboard Interpretation	30.19
30.6. Display Screen	30.22
30.7. Miscellaneous Terminal I/O	30.24
31. Ethernet	31.1
31.1. Ethernet Protocols	31.1
31.1.1. Protocol Layering	31.1
31.1.2. Level Zero Protocols	31.2
31.1.3. Level One Protocols	31.3
31.1.4. Higher Level Protocols	31.4
31.1.5. Connecting Networks: Routers and Gateways	31.4
31.1.6. Addressing Conflicts with Level Zero Mediums	31.5
31.1.7. References	31.5
31.2. Higher-level PUP Protocol Functions	31.6
31.3. Higher-level NS Protocol Functions	31.7
31.3.1. Name and Address Conventions	31.7
31.3.2. Clearinghouse Functions	31.9
31.3.3. NS Printing	31.12
31.3.4. SPP Stream Interface	31.12
31.3.5. Courier Remote Procedure Call Protocol	31.15
31.3.5.1. Defining Courier Programs	31.15
31.3.5.2. Courier Type Definitions	31.17
31.3.5.2.1. Pre-defined Types	31.17
31.3.5.2.2. Constructed Types	31.18
31.3.5.2.3. User Extensions to the Type Language	31.19
31.3.5.3. Performing Courier Transactions	31.20
31.3.5.3.1. Expedited Procedure Call	31.22
31.3.5.3.2. Expanding Ring Broadcast	31.23

<u>31.3.5.3.3. Using Bulk Data Transfer</u>	31.24
<u>31.3.5.3.4. Courier Subfunctions for Data Transfer</u>	31.25
<u>31.4. Level One Ether Packet Format</u>	31.26
<u>31.5. PUP Level One Functions</u>	31.28
<u>31.5.1. Creating and Managing Pups</u>	31.28
<u>31.5.2. Sockets</u>	31.28
<u>31.5.3. Sending and Receiving Pups</u>	31.29
<u>31.5.4. Pup Routing Information</u>	31.30
<u>31.5.5. Miscellaneous PUP Utilities</u>	31.31
<u>31.5.6. PUP Debugging Aids</u>	31.32
<u>31.6. NS Level One Functions</u>	31.36
<u>31.6.1. Creating and Managing XIPs</u>	31.36
<u>31.6.2. NS Sockets</u>	31.37
<u>31.6.3. Sending and Receiving XIPs</u>	31.37
<u>31.6.4. NS Debugging Aids</u>	31.38
<u>31.7. Support for Other Level One Protocols</u>	31.38
<u>31.8. The SYSQUEUE mechanism</u>	31.41
