F.G.H.
4/25/85

LispCourse #22: Overview of Lisp Packages; Using the Documentation

Lisp Packages

Recall: Lisp packages are sets of related functions contained on a single file (or small set of files) that are designed to carry out some particular task in the Interlisp environment, e.g., text editing, file transfer, etc.

There are by convention two types of public packages in the Interlisp world, *LispUsers* packages and *Lisp Library* packages. The former are random packages submitted by random hackers. The latter are packages maintained by the AISBU Lisp development group.

A third kind of "package" are the large application systems build on top of Interlisp such as LFG and NoteCards. We will not talk about these types of large applications here.

To get the functionality of a package (if it is not already contained in the default sysout), you need to find and then load the file(s) containing the package.

LISP LIBRARY Packages of Interest to the Non-Programmer

At PARC, the following packages are stored on {eris}<lisp>*release*>library> [where *release* is Hormony or Intermezzo, or ...]. They are packages designed and implemented and maintained by the AISBU Lisp group. Thus their reliability and usability is extremely high.

The Big Ones

FILEBROWSER ž Edit, Delete, Load, Compile, Copy, Rename, See, Hardcopy, & Info on files on any file device through a common graphic interface. Call: (FILEBROWSER FILEPATTERN). *Documentation = ???*

SKETCH ž Sketch is a drawing program that enables you to place text and graphics to achieve desired images. The figures can be copy-selected into TEdit documents to allow a mixture of text and graphics in the same document. *Documentation = SKETCH.TEDIT*

TEDIT ž The Interlisp-D text editor. *Documentation = IRM*

GRAPHER ž contains a collection of functions and an editor for laying out, displaying, and editing graphs (i.e., networks of nodes and links). *Documentation = GRAPHER.TEDIT*

Mail Handling

LAFITE ž Interlisp mail program, a la Laurel/Hardy. *Documentation = LAFITE.TED*

MAINTAIN ž Lisp implementation of Grapevine MAINTAIN program for adding or deleting names from mail distribution lists. *Documentation = ???*

NSMAIL ž Add on to Lafite for handling NS Mail. *Documentation = NSMAIL.TEDIT*

MAILSCAVENGE ž The Lisp Library package MAILSCAVENGE is used to rebuild the internal pointers in a mail file that has been damaged. Lafite generally reports ‘‘Can’t parse file’’ and terminates its Browse command when it detects damage in a file. The simplest remedy is to call MAILSCAVENGE, then browse the file again. *Documentation = MAILSCAVENGE.TEDIT*

Networks & Files

COPYFILES ž Functions for copying sets of files from one directory to another. *Documentation = COPYFILES.TEDIT*

FTPSERVER ž Lisp implementation of PUP FTP (File Transfer Protocol) server. Lets others FTP from you while you’re running lisp. *Documentation = FTPSERVER.TEDIT*

RS232, RS232CHAT, RS232EXEC, RS232FTP, RS232LOGIN ž Basic software drivers for serving the RS232 serial interface including CHAT and FTP service. *Documentation = RS232.TTY*

TCP, TCPCHAT, TCPFTP ž Basic software drivers for communicating with TCP/IP (Arpanet protocols) hosts (e.g., Sun workstations, Vaxes, etc.) including CHAT and FTP service. *Documentation = TCP.TEDIT*

TELERAID Interlisp-D has facilities for looking at sysout files and machines across the network, used by hackers to debug when your DLion goes into a 9XXX error in the maintenance panel. *Documentation =TELERAID.TEDIT*

Special Printers

FX80STREAM ž a library of routines used for driving an Epson FX-80 dot-matrix printer. With FX80STREAM you can use the full set of Interlisp-D device-independent graphical operations to compose pages on your FX-80, including printing TEdit documents. *FX80STREAM.TEDIT*

FXPRINTER, FXPARALLELPRINTER ž allows a user to print Lisp files to an Epson FX-80 printer. *Documentation = FXPRINTER.TED & FXPARALLELPRINTER.TED*

PRINTER ž package to do hardcopy of bitmaps and listfiles with multiple fonts for C.ITOH (Cheap!!!) printer connected to Dolphin parallel port. *Documentation = PRINTER.TTY*

Fonts & Bitmaps

READAIS ž Read, write, transform AIS (color/grey scale) files (common files containing pretty images). *Documentation = READAIS.TXT*

BIG ž Function NEWFONT sets up default fonts to be "size", where size is one of the atoms BIG, MEDIUM, STANDARD, SMALL. Changes the prettyprint fonts, the default font for the break window, tyipin, etc. Does this by resetting FONTPROFILE and then setting the fonts for all of the known windows. *Documentation = BIG.TTY*

BITMAPFNS ž Miscellaneous functions for manipulating BITMAPs. Reading and writing bitmaps, reading certain press files, creating window with image of bitmap. *Documentation = BITMAPFNS.TTY*

EDITBITMAP ž provides an interface (EDIT.BITMAP) for manipulating bitmaps. It puts up a menu of bitmap manipulation commands, one of which is HAND.EDIT which accesses EDITBM, the Interlisp-D bitmap editor. Other commands include shifting (in four directions), rotation (left and right 90 degree), inverting (horizontally, vertically, about diagonals), interchanging black and white, adding a border. *Documentation = EDITBITMAP.TTY*

Misc Tools

PAGEHOLD ž Redefines the default PAGEFULLFN, and provides hooks for making individual non-TTYDISPLAYSTREAM windows scrollable. Scrolling is "held" for up to PAGE.WAIT.SECS seconds, during which time an attached

"button" on the window softly flashes, and then the hold is "released". Holding down either SHIFT key will continue the "hold" (i.e., prevent "release"); letting up on either SHIFT key will "release" the "hold". *Documentation = PAGEHOLD.TEDIT*

SAMEDIR ž Advises MAKEFILE so that user can't inadvertently write out a file onto a directory other than the one it came from. Checks FILEDATES property against connected directory. *Documentation = SAMEDIR.TTY*

SINGLEFILEINDEX ž Package for giving user an alphabetical function index on the front of any lisp file listed thru lisp. Index number for a function indicates function's linear occurence within file. Within the lisp source, each function is preceded by it's index number right justified on the page. *Documentation = SINGLEFILEINDEX.TEDIT*

Games & Demos

HANOI ž Displays and solves famous Towers of Hanoi problem. Can run as a background process. HANOIWINDOW can be reshaped. Call: (HANOI NRINGS WINDOW FONT ONCE). *Documentation = HANOI.TTY*

KINETIC ž Graphics demo. Fast random BITBLTs on a window. Call: (KINETICDEMO). *Documentation = ???*

UTILPROC ž Simple utility processes, including hall-of-mirrors demo. *Documentation = ???*

WINK ž Movie of Marilyn Monroe winking. Call: (SHOWMOVIE). Needs BITMAPFNS. *Documentation = ???*

LISPUSERS Packages of Interest to the Non-Programmer

The following packages are stored on {eris}<lispusers> at PARC. They are packages designed and implemented by random Lisp hackers and are not supported by the AISBU Lisp group. Their reliability and usability various greatly.

Tools ž Editors, Printing functions, Graphics, etc.

AREDIT ž Tool for submitting, viewing, & editing Lisp ARs (action requests/Interlisp bug reports) from within Interlisp. Type (AR.FORM) to create

a form for Getting, Putting, & Submitting ARs. *Documentation = AREDIT.TEDIT*

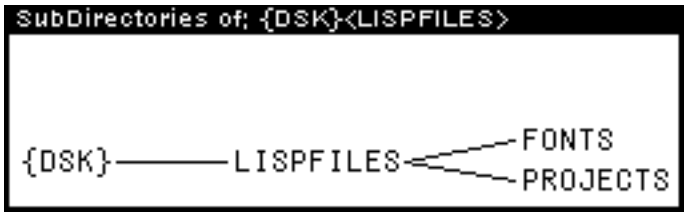
ARCHIVETOOL ž An Interlisp-D interafce to the PARC Cedar Archive system. Based on the FileBrowser and Lafite. It automatically composes and sends archive messages to the Archivist. It also parses the returned messages and deletes the archived files. Retrieve requests are not yet supported. *Documentation = ARCHIVETOOL.TED*

BIZGRAFIX ž Pie & bar chart, line graph creation. *Documentation = BIZGRAFIX.TEDIT*

CALENDAR ž A calendar/appointment-reminder program. Displays any year, month or day. Messages alert you at predetermined time, or can optionally be mailed (via Lafite) to any desired recipients. CALL: (CALENDAR). *Documentation = CALENDAR.TED*

COMAPRETEXT ž COMPARETEXT is a rather non-standard text file comparison program which tries to address two problems: (1) the problem of detecting certain types of changes, such as detecting when a paragraph is moved to a different part of a document; and (2) the problem of showing the user what changes have been made in a document. *Documentation = COMPARETEXT.TXT*

DIRECTORYTOOLS ž DirectoryTools contains one user function: GRAPH.DIRECTORY which is used for graphing the subdirectory structure under a file pattern. e.g:



Documentation = DIRECTORYTOOLS.TEDIT

EDITFONT ž Gives the user functions for creating and editting DISPLAY fonts which can be read and written as STRIKE font files. Includes: EDITFONT, BLANKFONTCREATE, READSTRIKEFONTFILE, and WRITESTRIKEFONTFILE. *Documentation = EDITFONT.TTY*

EDITHIST ž An extension to Interlisp's file package that permanently preserves the history of new versions of files. Every time a file is remade, a new entry

containing DATE, AUTHOR, FILE, CHANGES, & COMMENTS info is added. *Documentation = EDITHIST.DOC*

EXEC ž This small package allows the user to create extra EXEC windows in which to do EVALQTing. A new EXEC window may be created in either of two ways. The user may either do (EXEC) or button the EXEC menu item added to the background menu. *Documentation = EXEC.TTY*

FINGER ž Finger is a facility for determining and displaying information about other users running Interlisp-D. It displays the user's name, the Etherhostname (or the octal net address when no nameserver is available) and the user's idle time (time since last keystroke or mouseaction). Only other users who have the finger server loaded will be displayed. *Documentation = FINGER.TEDIT*

HEADLINE ž Three functions for manipulating windows containing headlines: HEADLINE, BILLBOARD, and CLOSEHEADLINES. Useful for titling a screen image or leaving message on screen while away. *Documentation = HEADLINE.TTY*

HISTMENU ž Provides simple way to access Interlisp history list using menu. REDO, UNDO, FIX, and ?? selected items. Call: (HistoryMenu histMenuLength histMenuPosition). *Documentation = HISTMENU.TED*

LANDPRESS ž Allows landscape printing of ASCII text files on press printers (e.g., Dovers). Function LANDPRESS produces same product as MAKEPRESS, only printed sideways on the page allowing for wider output. *Documentation = LANDPRESS.TTY*

MAILOPS ž Functions for poking into and "scavenging" Laurel/Lafite/... mail files. *Documentation = MAILOPS.TEDIT*

NOTEPAD ž Allows user to do artwork at bitmap level in NOTEPAD windows. Trajectories: sketch, line, circle, ellipse, open curve, closed curve. Objects/editting: text, are of screen, shade rectangle, fill, edit area. Style: brush, use mask, mask, use grid, grid, use symmetry, point of symmetry, text font, shade. *Documentation = NOTEPAD.TTY*

PRESSTOIP ž PRESS TO InterPress. Converts PRESS files to INTERPRESS files. Call: (PRESS.TO.IP PRESSFILE IPFILE). *Documentation = PRESTOIP.TTY*

PROMPTREMINDERS ž User can be periodically reminded of important things by messages which are aggressively winked and flashed in the PROMPTWINDOW. *Documentation = PROMPTREMINDERS.TTY*

REMIND ž Facility for scheduling LISP events to take place at a specified later time. Reminders are stored on REMINDERS.LISP on user's directory and an entry on AFTERSYSOUTFORMS causes them to be loaded via LOADREMINDERS which may also be put in user's init file. *Documentation = REMIND.TTY*

SPACEWINDOW ž Puts a small "Space Allocation" window on screen. Shows a bar-chart of the amounts of the four types of memory space that have been allocated (fixed data, variable data, atoms, pnames). Display is updated every 60 seconds. *Documentation = SPACEWINDOW.TXT*

Hardcopying Screen Images

COPYIMAGE ž a small utility which facilitates making hardcopy or PRESS output of either ANY window on a screen, or else the entire screen. To use load COPYIMAGE.DCOM which updates the background menu, and then button the CopyImage item of the background menu. Hardcopy output goes to the FULLPRESS printer of your DEFAULTPRINTINGHOST. *Documentation = COPYIMAGE.TTY*

FULLSCREEN ž This package allows an entire screen's image to be printed on an Interpress printer. *Documentation = FULLSCREEN.TXT*

Changes/Additions to the Interlisp-D Interface

ANIMATE ž This small package contains functions for moving a non-rectangular bitmap smoothly around the screen, ways of using these to get big cursors, and bitmaps for a large arrow and a hand to be used as large cursors. *Documentation = ANIMATE.TTY*

AUXMENU ž This package creates a middle button background menu containing a number of convenient functions. These functions include a menu-driven CNDIR, LOGOUT, and other functions that require little user-interaction. The purpose of the package is to minimize typeing. *Documentation = AUXMENU.TEDIT*

DEDITK ž Adds single button method for combining the most frequently combined pairs of BI/BO and BEFORE/AFTER/DELETE/REPLACE in DEDIT - Load and call (DEDITK). *Documentation = DEDITK.TED*

MACWINDOW ž Advises SHRINKW and EXPANDW to produce a zooming effect by showing the outline of their window arguments as they shrink or grow. *Documentation = MACWINDOW.TXT*

MOVE-WINDOWS ž MOVE-WINDOWS is a tool to help you re-arrange your screen quickly. Once loaded, buttoning in the background will shift you into window-moving mode. Buttoning again in the background will get you out of that mode. In window-moving mode, buttoning in a window with either the LEFT or MIDDLE button will either reshape or move the window: if you button down near a corner, the corner is moved; near a side, that side is moved; in the center, the whole window is moved. Buttoning in a window with the RIGHT button will call the usually window menu (DOWINDOWCOM). This is the best way to close windows, for example. *Documentation = MOVE-WINDOWS.TEDIT*

SNAPSCROLL ž Loading SNAPSCROLL advises (SNAPW) so that snapshot windows are henceforth shapeable and scrollable. This enables the user to, for example, snap a long list off the screen, and then reshape it to a reasonable size and scan it at will. *Documentation = SNAPSCROLL.TED*

TEDITKEY ž TEditKey is a package which provides a keyboard interface to TEdit. On a Dandelion, the interface takes advantage of the non-Alto keys. On Dorados and Dolphins, a window mimicking the Dlion function keys provides the same abilities. *Documentation = TEDITKEY.TEDIT*

TINYTIDY ž TINYTIDY takes the icons on your screen and lines them up along the edge. *Documentation = TINYTIDY.TEDIT*

Clocks

CROCK ž Function for creating and manipulating an analog face clock. Menu allows user to change style of clock. Call: (CROCK REGION). *Documentation = CROCK.TEDIT*

LCROCK ž Puts a digital/analog clock in the unused area of the LOGOW. (START.LCROCK <myLogo> <Position>) starts it up. <myLogo>, if non-NIL, will replace the "Interlisp-D" logo; and <Position>, if non-NIL, will be the lower-left corner (NIL means to use the position of the existing logo window). The

glovalvar CROCKUPDATERATE.MS is the number of milliseconds between automatic updates. *Documentation = LCROCK.DOC*

Games & Demos

BLTDEMO ž Implements Smalltalk graphics demo in Interlisp. Spinning star, bouncing ring & box. Call: (BOUNCE X Y) where X and Y are velocities defaulting to 3. Box shows whatever is near cursor. Interseting recursive effects can be seen if you move the cursor near the box. *Documentation = BLTDEMO.TXT*

FACEINVADER ž A game. The object of the game is to shoot the bouncing 'face' before it overruns your base. Call: (FI INSTRUCTIONS?). *Documentation = ???*

JARGON ž N random broken definitions from the imfamous hacker's dictionary, snarfed from MIT-AI. (The globalvar JARGON.FILE.LOCATION points to the database file.) Starts up on load, or type (JARGON.READ) *Documentation = ???*

KAL ž Kaleidoscope demo. Call: (KAL). Control with middle button menu. *Documentation = KAL.TED*

LIFE ž This Life program is a translation of the SmallTalk version in the book Goldberg, Robson: The Language and its Implementation. *Documentation = LIFE.TXT*

LINEDEMO ž This package contains a couple of random demonstration programs having to do with drawing random lines or polygons. *Documentation = LINEDEMO.TED*

NQUEENS ž Solves N Queens problem. How to place N queens on a chess board so that they don't attack each other. Graphics demo. Call: (NQUEENS N). *Documentation = ???*

PACMAN ž Game. Runs in b/w or color. CALL: (PACMAN). *Documentation = ???*

PEANO ž Peano curves graphics demo. CALL: (PEANODEMO LEVEL SCALE). *Documentation = ???*

PLAY ž Offers Interlisp-D users a disciplined way to play simple musical melodies on Xerox 1108 machines. (PLAY.DEMO) demos the PLAY package. Main functions: PLAY.NOTES, PLAY.MELODY, PLAY.KEYBOARD. *Documentation = PLAY.TTY*

QIX ž QIX is a small graphic demo modelled after the videogame of the same name. *Documentation = QIX.TEDIT*

SOLITAIRE ž The card game Solitaire (graphics demo). Call: (SOLO). *Documentation = ???*

TRAJECTORY-FOLLOWER ž Provides a function which causes a snake to crawl along a trajectory. Trajectory is specified by a set of KNOTS and a CLOSED flag. *Documentation = TRAJECTORY-FOLLOWER.TTY*

DMT Equivalents (i.e., run after machine has been idle for a bit)

BLACKOUT ž (Blackout text interval) makes a back window the size of the screen and bounces a square around on it, like DMT, etc. Good for servers. Text and interval default to "Type Key" and NIL (= forever) if not given. *Documentation = BLACKOUT.TEDIT*

BOUNCE ž Is another dmt variant. Blacks out the screen & draws patterns on it until you hit a mouse button or type any character. Can be started 3 ways. *Documentation = ???*

FRACTAL ž An eyewash DMT program that draws fractals. *Documentation = FRACTAL.TEDIT*

Alternative Page Hold Schemes

NOWAITPRINT ž a function which will temporarily diddle a window's pagehold characteristics so that a print-without-holding may be performed. *Documentation = NOWAITPRINT.TTY*

YAPFF ž is Yet Another Page Full Function. I actually don't like this one much better than any of the others around, but its another point in the space of possible actions on end-of-page. *Documentation = YAPFF.TEDIT*

Using the Interlisp-D Documentation

The State of Interlisp-D w.r.t. Documentation

Sad, sad, sad, sad.

Interlisp-D documentation is:

1. *Incomplete* ÿ much of Interlisp is simply undocumented

2. *Out of date* ÿ last major manual revision was October, 1983

3. *Dispersed* ÿ documentation is spread across various binders, files, file servers, etc.

4. *For programmer's only* ÿ There is almost no user level documentation for the Interlisp-D environment; its all oriented toward the programmer and the system implementor. Interlisp-D the language is not properly distinguished from Interlisp-D the computer environment.

Bottom line is that documentation is THE major flaw of Interlisp-D, especially for the non-programming user of the Interlisp-D environment.

How to Use the Available Documentation

Isolate the user information from among the programmer/implementor information.

How? Beats me!!!!!!

Documentation Sources

General

Interlisp Reference Manual
(October 1983 version, plus updates available on {eris}<lispmanual>)

Release Notes (Chorus, Fugue 1 to 6, Carol, Harmony, Intermezzo)

Documentation files for LispUsers and LispLibrary Packages

Installation and use on a Dlion

1108 Users Guide

Mesa Users Guide (Chapter 2, Getting Started & Chapter 35, Othello)

{eris}<lisp>harmony>doc>Hello.tedit (PARC only)

{eris}<lisp>release>doc>GettingStarted.tedit (PARC only)

{eris}<lisp>harmony>doc>LocalFile.TEDIT

Installation and use on a Dolphin/Dorado

{eris}<lisp>release>doc>GettingStarted.tedit (PARC only)

{indigo}<altodocs> (PARC only)

Introductions (?)

Friendly DLion primer from LRDC

Sysdoc stuff?

Overview of the relevant sections of the IRM (October, 1983 version)

Sections 6.1, 18.16, & 18.17 ÿ files from lisp's point of view

Chapter 8 ÿ the P.A. including the history list

Chapter 9 ÿ Error handling and Breaks

Chapter 11 ÿ the File package including INIT file maintenance

Section 14.1 ÿ Sysouts

Section 14.2 ÿ GREET and INIT files

Section 14.3 ÿ Directories

Section 14.7 ÿ GAINSPACE when arrays full

Section 18.14 ÿ the keyboard

Section 18.18 ÿ Hardcopies

Section 18.20 ÿ Processes and the PSW

Section 19.20 ÿ Windows

Chapter 20 ÿ DEdit, TEdit, CHAT, Break windows, EDITBM, TTYIN

Basically, ignore the rest if you don't know how to program well.

Release Notes

Since the manual is constantly being made obsolete by new releases of the system, you should learn to use the release notes.

For each new release, study the release notes carefully trying to remember the things that have changed; just have to wade through looking for things that make sense from user's point of view. The index at front is a rough guide.

It is often best to consult the release notes before going to the IRM, since much of the information contained in the IRM will be out-of-date compared to the Release Notes.

Online help - the APROPOS function

In general, there is no online help facilty in Interlisp-D.

There is one handy function, however, called APROPOS. APROPOS takes a single argument which is an aribitrary literal atom. (APROPOS LitAtom) will search through your virtual memory, looking for all atoms whose name contains the atom LitAtom. (APROPOS 'FLG) prints the names, values, and function definitions for all atoms in the current virtual memory that have FLG in their name. (APROPOS 'FONT) prints the names, values, and function arguments for all atoms in the current virtual memory that have FONT in their name.

Example, find the name of the variable that tells the name of the release (e.g., Harmony or Intermezzo) being used:

```
96_(APROPOS (QUOTE RELEASE]
RELEASE.MONITORLOCK
                - Function arglist: (LOCK EVENIFNOTMINE)
RELEASE.PUP     - Function arglist: (EPKT)
RELEASERESOURCE
                - Property list: (MACRO (ARGS (& &) (SUBPAIR
&
ARGS --)))
RELEASEBREAKWINDOW
                - Function arglist: (BRKDS PREVIOUSDS)
RELEASE.XIP     - Function arglist: (EPKT)
NIL
97_(APROPOS 'NAME]
FILENAME        - Function arglist: (NAME)
FILENAMEFIELD   - Function arglist: (FILE FIELDNAME)
PACKFILENAME    - Function arglist: U
NAMEFIELD       - Function arglist: (FILE SUFFIXFLG DIRFLG)
FULLNAME        - Function arglist: (X RECOG)
DIRECTORYNAME   - Function arglist: (DIRNAME STRPTR CREATE?)
PACKFILENAME.STRING
                - Function arglist: U
HOSTNAME        - Function arglist: U
                - Variable value: NIL
DIRECTORYNAMEP  - Function arglist: (DIRNAME HOSTNAME)
HOSTNAMEP       - Function arglist: (NAME)
TYPENAME        - Function arglist: (DATUM)
```

```
STKNTHNAME      - Function arglist: (N POS)
STKNAME         - Function arglist: (POS)
SETSTKNAME      - Function arglist: (POS NAME)
STKARGNAME      - Function arglist: (N POS)
SETSTKARGNAME   - Function arglist: (N POS NAME)
COMPILEDTYPENAMEP
                - Function arglist: (X)
RENAMEFILE      - Function arglist: (OLDFILE NEWFILE)
UNPACKFILENAME  - Function arglist: (FILE ONEFIELDFLG DIRFLG
STRING)
UNPACKFILENAME.STRING
                - Function arglist: (FILE ONEFIELDFLG
DIRFLG)
USERNAME        - Function arglist: (FLG STRPTR
PRESERVECASE)
                - Variable value: HALASZ
SETUSERNAME     - Function arglist: (NAME)
ALTOFILENAME    - Function arglist: (X)
TYPENAMEP       - Function arglist: (DATUM TYPE)
                - Property list: (DMACRO (X
(COMPILEDTYPENAMEP X
)))
CHANGENAME1     - Function arglist: (DEF X Y)
CHANGENAME1A    - Function arglist: (DEF OLD NEW MAP)
PROPNAMES       - Function arglist: (ATM)
ROOTFILENAME    - Function arglist: (NAME COMPFLG)
PROCESS.NAME    - Function arglist: (PROC NAME)
ETHERHOSTNAME   - Function arglist: (PORT USE.OCTAL.DEFAULT)
CANONICAL.HOSTNAME
                - Function arglist: (HOSTNAME)
GREETFILENAME   - Function arglist: (USER)
CHANGENAME      - Function arglist: (FN FROM TO)
FSTKNAME        - Function arglist: (POS)
FIRSTNAME       - Variable value: FRANK
FONTNAME        - Function arglist: (NAME)
                - Variable value: PARC
DIRFILENAME     - Function arglist: (FILEGROUP)
DIRPRINTNAME    - Function arglist: (FILEGROUP FLG)
UPPERCASEFILENAMES
                - Variable value: T
RESTORENAMES    - Function arglist: (FN)
MSHASHFILENAME  - Variable value: NIL
DEFAULTRENAMEMETHOD
                - Variable value: NIL
RECORDFIELDNAMES
                - Function arglist: (RECORDNAME FLG)
RENAME          - Function arglist: (OLD NEW TYPES FILES
METHOD)
RESOURCENAME    - Property list: (CLISPWORD (FORWORD .
resourceName))
COMP.NAMEDLET   - Function arglist: (ARGS)
NAMEDLET        - Property list: (DMACRO COMP.NAMEDLET)
MSWORDNAME      - Function arglist: (X)
MOUSESTATE-NAME
                - Function arglist: (KEYNAME MOUSEONLYFLG)
NONSYSPROPNAMES
                - Function arglist: (ATM)
INSPECTABLEFIELDNAMES
                - Function arglist: (DECL TOPONLYFLG)
NAMEOFEDITW     - Function arglist: (NAME TYPE)
PARSE.NSNAME    - Function arglist: (NAME #PARTS
DEFAULTDOMAIN)
NSNAMETYPE#     - Variable value: 89
                - Property list: (GLOBALVAR T)
```

```
JOBNAME          - Variable value: {LPT}LISPPRINT:PARC.;1
NSNAME           - Property list: (COURIERDEF (
COURIER.READ.NSNAME COURIER.WRITE.NSNAME
COURIER.NSNAME.LENGTH))
NSNAME.TO.STRING
                 - Function arglist: (NSNAME FULLNAMEFLG)
COURIER.READ.NSNAME
                 - Function arglist: (STREAM PROGRAM TYPE)
COURIER.WRITE.NSNAME
                 - Function arglist: (STREAM NAME PROGRAM
TYPE)
COURIER.NSNAME.LENGTH
                 - Function arglist: (NSNAME PROGRAM TYPE)
NSNAME2          - Property list: (COURIERDEF (
COURIER.READ.NSNAME COURIER.WRITE.NSNAME))
NS.SERVER.NAMES.TO.ADDRESSES
                 - Variable value: NIL
EQUAL.CH.NAMES   - Function arglist: (NAME1 NAME2)
CH.NAME.TO.STRING
                 - Function arglist: (NSNAME FULLNAMEFLG)
CANONICAL.CH.NAME
                 - Function arglist: (NAME)
CH.CANONICAL.NAME
                 - Function arglist: (NAME)
FONTNAME.IP      - Function arglist: (FONTDESC)
FLOPPY.NAME      - Function arglist: (NAME)
FLOPPY.SET.NAME
                 - Function arglist: (NAME)
FLOPPY.GET.NAME
                 - Function arglist: NIL
MAKESYSNAME      - Variable value: INTERMEZZO
MBUTTON.CHANGENAME
                 - Function arglist: (TEXTOBJ OBJ NEWNAME)
GV.PORTFROMNAME
                 - Function arglist: (SERVERNAME)
FULLUSERNAME     - Function arglist: (UNPACKEDFLG)
GV.NEWNAME       - Function arglist: (NAME GV.NEWNAME
IDENTIFYUSER PASSWORD)
GVNAMETYPE       - Variable value: 1
REGROOTNLSNAME   - Variable value: "GrapevineRServer"
LA.SHORTFILENAME
                 - Function arglist: (FILE EXT
KEEPVERSIONFLG)
DEFAULTMAILFOLDERNAME
                 - Variable value: {DSK2}ACTIVE.MAIL
LA.LONGFILENAME
                 - Function arglist: (FILENAME EXT)

PROFILEFILENAME
                 - Function arglist: NIL
TOCFILENAME      - Function arglist: (MAILFILE)
PROMPTFORFILENAME
                 - Function arglist: (WINDOW DEFAULT PROMPT)
LAFITEPROFILE.NAME
                 - Variable value: LAFITE
LAFITETEMPFILEHOSTNAME
                 - Variable value: CORE
LAFITE.READ.NAME.FIELD
                 - Function arglist: (STREAM ARGS)
FB.FETCHFILENAME
                 - Function arglist: (ENTRY)
FB.RENAMECOMMAND
                 - Function arglist: (PREFIX FILEENTRY
WINDOW)
```

```
FB.STARTOFNAME  - Function arglist: (FILENAME SPEC)
VTYPENAME       - Function arglist: (DATUM)
DISPLAY/NAME    - Function arglist: (ND)
                - Property list: (CODE {CCODEP}#65,43610)
FONTNAMELIST    - Function arglist: (FONTDESC)
                - Property list: (CODE {CCODEP}#65,43524)
AR.USERNAME     - Function arglist: NIL
AR.GET.FILENAME
                - Function arglist: (NUM PUTFLG)
AR.SUBMIT.NUM.FILE.NAME
                - Variable value:
{PHYLUM}<LISPARS>LISPARS.NUM
AR.FILENAME     - Function arglist: (ARN)
AR.SUBMIT.FILE.NAME
                - Variable value:
{PHYLUM}<LISPARS>LISPARS.SUBMIT
AR.INFO.FILE.NAME
                - Variable value:
{PHYLUM}<LISPARS>LISPARS.TDS
AR.INDEX.DEFAULT.FILE.NAME
                - Variable value: {PHYLUM}<LISPARS>AR.INDEX
HASHFILENAME    - Function arglist: (HASHFILE)
SKETCH.ELEMENT.TYPE.NAMES
                - Variable value: (MAP SKIMAGEOBJ GROUP
TEXTBOX
BOX --)
SKETCH.ELEMENT.NAMEP
                - Function arglist: (X)
SK.UNDO.NAME    - Function arglist: (HISTEVENT)
SK.GET.HARDCOPY.FILENAME
                - Function arglist: (SKW)
```

APROPOS can be very handy to go wandering around the system looking for a partially remembered variable or function name or for discovering what variables effect, e.g., FONTS.

Note the success of APROPOS at these tasks depends on the variables and functions be named in a sensible manner. A font-related variable named SIXTH-BASE would never be found by the (APROPOS 'FONT) function call. Such is not always the case in Interlisp-D.

Homework

Start reviewing the Lisp programming covered in the first six or so sessions.