

File created: 21-Sep-88 11:02:16 {EG:PARC:XEROX}<LANNING>LISP>USERS>GENERIC-INIT.;108

changes to: (ADVISE FILES? CL:COMPILE MAKEFILE GETPROMPTWINDOW)
(VARS ROOMS-INIT-COMMANDS MAIL-INIT-COMMANDS UNIX-INIT-COMMANDS LOOPS-INIT-COMMANDS
TEDIT-INIT-COMMANDS PCL-INIT-COMMANDS GENERIC-INITCOMS WHO-LINE-COMMANDS
DISPLAY-CONTROL-INIT-COMMANDS CHAT-INIT-COMMANDS)
(COMMANDS "MORE" "BREAK" "UNBREAK" "CALLS" "DESCRIBE" "EC" "EFF" "FILES?" "IC" "NOTICE" "MAKE"
"SPY")
(FUNCTIONS XCL-USER::LOG-GENERIC-INIT-USER EVAL-AT-GREET NOTICE MAKE OAM DE FILE |PickOneAtRandom|
ATOM-NEIGHBORS LOAD-NOVA-FONTS)

previous date: 19-Sep-88 09:26:30 {EG:PARC:XEROX}<LANNING>LISP>USERS>GENERIC-INIT.;107

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1987, 1988 by Xerox Corporation. All rights reserved.

(RPAQQ **GENERIC-INITCOMS**

(;; Macro to avoid problems when trying to maintain file.

(COMS (P (CL:PROCLAIM ' (GLOBAL *GENERIC-INIT-LOADED*)))
(INITVARS (*GENERIC-INIT-LOADED* NIL))
(FUNCTIONS EVAL-AT-GREET))

;; Silent loads

(COMS (P (CL:PROCLAIM ' (GLOBAL *LOAD-SILENT* PRETTYHEADER))))
(INITVARS (*LOAD-SILENT* NIL))
(VARS (\\ORIGINAL-LOAD-VERBOSE *LOAD-VERBOSE*)
(\\ORIGINAL-PRETTYHEADER PRETTYHEADER)
(*LOAD-VERBOSE* (IF *LOAD-SILENT* THEN NIL ELSE *LOAD-VERBOSE*))
(PRETTYHEADER (IF *LOAD-SILENT* THEN NIL ELSE PRETTYHEADER))))

;; Environment setup

(COMS * COMPUTE-DIRECTORIES-INIT-COMMANDS)
(COMS * ENV-TAILORING-INIT-COMMANDS)
(COMS * PATCH-INIT-COMMANDS)
(COMS ;; Wants to be (COMS * FILECACHE-INIT-COMMANDS), but the FileCrasher doesn't exist in Lyric, yet. Probably never will.

(VARS FILECACHE-INIT-COMMANDS))
(COMS * FONT-SETUP-INIT-COMMANDS)
(DECLARE\ : EVAL@LOADWHEN (NOT *GENERIC-INIT-LOADED*)
DONTVAL@COMPILE DOCOPY (FILES (SYSLOAD FROM LISPUSERS)
LOADMENUITEMS))

;; Machine status

(COMS * WHO-LINE-COMMANDS)
(COMS * VSTATS-INIT-COMMANDS)

;; Display control

(COMS * SCREEN-SETUP-INIT-COMMANDS)
(COMS * ROOMS-INIT-COMMANDS)
(COMS * CHANGE-BACKGROUND-INIT-COMMANDS)
(COMS * DISPLAY-CONTROL-INIT-COMMANDS)
(COMS * IDLE-INIT-COMMANDS)
(COMS * CLOCK-INIT-COMMANDS)

;; Programming stuff

(COMS * PROGRAMMING-INIT-COMMANDS)
(COMS * OLD-UTILS-COMMANDS)
(COMS * WIZARD-INIT-COMMANDS)
(COMS * DINFO-INIT-COMMANDS)
(COMS * PCL-INIT-COMMANDS)
(COMS * LOOPS-INIT-COMMANDS)

;; Documentation

(COMS * TEDIT-INIT-COMMANDS)
(COMS * SKETCH-INIT-COMMANDS)
(COMS * NOTECARDS-INIT-COMMANDS)

;; Communication & Info

(COMS * MAIL-INIT-COMMANDS)
(COMS * CHAT-INIT-COMMANDS)
(COMS * TALK-INIT-COMMANDS)
(COMS * CALENDAR-INIT-COMMANDS)
(COMS * PRINTER-INIT-COMMANDS)
(COMS * DB-INIT-COMMANDS)
(COMS * NFS-INIT-COMMANDS)

;; Files

(COMS * FILE-WATCH-INIT-COMMANDS)
(COMS * FILE-SERVER-INIT-COMMANDS)
(COMS * DIRGRAPHER-INIT-COMMANDS)
(COMS * FB-INIT-COMMANDS)
(COMS * COMPARE-FILES-INIT-COMMANDS)

```

;; Random stuff
(COMS * UNIX-INIT-COMMANDS)
(COMS * DEMOS-INIT-COMMANDS)
(COMS * GAMES-INIT-COMMANDS)

;; Cleanup
(COMS * BACKGROUND-MENU-CLEANUP-INIT-COMMANDS)
(COMS * DO-LOAD-UTILITIES-INIT-COMMANDS)
(COMS ;; Send the Tool Work's a message telling it about this user.
      (FUNCTIONS XCL-USER::LOG-GENERIC-INIT-USER)
      (INITVARS (\\CC-GENERIC-INIT-MSG T))
      (P (EVAL-AT-GREET (CL:UNLESS *GENERIC-INIT-LOADED* (XCL-USER::LOG-GENERIC-INIT-USER))))))
(VARS (*LOAD-VERBOSE* \\ORIGINAL-LOAD-VERBOSE)
      (PRETTYHEADER \\ORIGINAL-PRETTYHEADER)
      (*GENERIC-INIT-LOADED* T))

;; Make the FileManager happy
(DECLARE\ : DONTCOPY (PROP MAKEFILE-ENVIRONMENT GENERIC-INIT)))

```

;; Macro to avoid problems when trying to maintain file.

```

(CL:PROCLAIM ' (GLOBAL *GENERIC-INIT-LOADED*))

(RPAQ? *GENERIC-INIT-LOADED* NIL)

(DEFMACRO EVAL-AT-GREET (&BODY FORMS)
  "Evaluate the forms only when loading the compiled file, and then only when greeting"
  `(CL:EVAL-WHEN (CL:LOAD)
    (CL:UNLESS (OR *GENERIC-INIT-LOADED* (MEMB DFNFLG ' (PROP ALLPROP)))
      ,@FORMS)))

```

;; Silent loads

```

(CL:PROCLAIM ' (GLOBAL *LOAD-SILENT* PRETTYHEADER))

(RPAQ? *LOAD-SILENT* NIL)

(RPAQ \\ORIGINAL-LOAD-VERBOSE *LOAD-VERBOSE*)

(RPAQ \\ORIGINAL-PRETTYHEADER PRETTYHEADER)

(RPAQ *LOAD-VERBOSE* (IF *LOAD-SILENT* THEN NIL ELSE *LOAD-VERBOSE*))

(RPAQ PRETTYHEADER (IF *LOAD-SILENT* THEN NIL ELSE PRETTYHEADER))

```

;; Environment setup

```

(RPAQO COMPUTE-DIRECTORIES-INIT-COMMANDS
  ( ;; Who am I?
    (DECLARE\ : DONTEVAL@COMPILE (VARS (|\\BasicUserName| (LET* ((NAME (USERNAME))
                                                                (REGISTRY-POS (STRPOS "." NAME NIL NIL NIL
                                                                NIL NIL T)))
                                                                (COND (REGISTRY-POS (SUBSTRING NAME 1
                                                                (SUB1 REGISTRY-POS)
                                                                )))
                                                                (T NAME))))
      (LOGINHOST/DIR (COND ((OR (NULL LOGINHOST/DIR)
                                (EQ LOGINHOST/DIR '{DSK})))
                          (CONCAT "{PHYLUM}<" |\\BasicUserName| ">LISP>"))
                      (T LOGINHOST/DIR))))
    (INITVARS (|\\UserHomeDirectory| (U-CASE (PACKFILENAME.STRING 'HOST (FILENAMEFIELD LOGINHOST/DIR
                                                                'HOST)
                                                                'DIRECTORY |\\BasicUserName|)))
      (TEMPDIR (CONCAT |\\UserHomeDirectory| "TEMP>"))
      (HOME-MACHINE-NAME "")
      (PRIVATE-LISPUSERSDIRECTORIES NIL)
      (*CACHE-DIRECTORIES* NIL))))

```

;; Who am I?

```

(DECLARE\ : DONTEVAL@COMPILE

(RPAQ \\BasicUserName|
  (LET* ((NAME (USERNAME))
        (REGISTRY-POS (STRPOS "." NAME NIL NIL NIL NIL NIL T)))
    (COND (REGISTRY-POS (SUBSTRING NAME 1 (SUB1 REGISTRY-POS)))
      (T NAME))))

(RPAQ LOGINHOST/DIR (COND ((OR (NULL LOGINHOST/DIR)
                                (EQ LOGINHOST/DIR '{DSK})))
                          (CONCAT "{PHYLUM}<" |\\BasicUserName| ">LISP>"))

```

```

(T LOGINHOST/DIR))
)
(RPAQ? |\UserHomeDirectory| (U-CASE (PACKFILENAME.STRING 'HOST (FILENAMEFIELD LOGINHOST/DIR 'HOST)
'DIRECTORY |\BasicUserName|)))
(RPAQ? TEMPDIR (CONCAT |\UserHomeDirectory| "TEMP>"))
(RPAQ? HOME-MACHINE-NAME "")
(RPAQ? PRIVATE-LISPUSERSDIRECTORIES NIL)
(RPAQ? *CACHE-DIRECTORIES* NIL)
(RPAQQ ENV-TAILORING-INIT-COMMANDS
(

```

::: Misc environmental tailoring

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (FILES (SYSLOAD NOERROR FROM LISPUSERS)
PAGEHOLD)
(VARS (|MaxValueLeftMargin| 512)
(SCROLLBARWIDTH 20)
(FIXSPELLDEFAULT '\n)
(\\ETHERTIMEOUT 3000)
(EMPRESS#SIDES 2)
(*PRINT-CASE* :DOWNCASE))
:: FileManager defaults
(VARS (CLEANUPOPTIONS '(RC ST))
(COPYRIGHTFLG 'DEFAULT)
(RECOMPILEDEFAULT 'EXPRS)
(*DEFAULT-CLEANUP-COMPILER* 'CL:COMPILE-FILE)
(*DEFAULT-MAKEFILE-ENVIRONMENT* '(:PACKAGE "XCL-USER" :READTABLE "XCL" :BASE 10)))
:: None of us here are system hackers
(VARS (*ORIGINAL-GIVE-AND-TAKE-DIRECTORIES* *GIVE-AND-TAKE-DIRECTORIES*)
(*GIVE-AND-TAKE-DIRECTORIES* (CL:REMOVE-IF #'(CL:LAMBDA (FILE)
(CL:MEMBER (CL:PATHNAME-HOST FILE)
'("ERIS" "ERINYES")
:TEST
#'STRING-EQUAL))
*GIVE-AND-TAKE-DIRECTORIES*)))
:: I don't like being told that I haven't listed files...
(ADVISE FILES?)
:: Load up the extended vmem stuff
(P (EVAL-AT-GREET (CL:WHEN (AND (EQ MAKESYSNAME 'LYRIC)
(EQ (MACHINETYPE)
'DORADO))
(FILESLoad (SYSLOAD FROM LISPUSERS)
EXTENDEDVMEM)
(INSTALL-EXTENDED-VIRTUAL-MEMORY))
(CL:WHEN (CL:FBOUND P 'DESCRIBE-VIRTUAL-MEMORY)
(DESCRIBE-VIRTUAL-MEMORY))))
:: Check greetdates whenever I log back in
(ADDVARS (AFTERLOGOUTFORMS (|for| GREET-DATE |in| GREETDATES |bind| (TODAY _ (DATE))
|when|
(EQ 1 (STRPOS (CAR GREET-DATE)
TODAY NIL NIL NIL NIL (UPPERCASEARRAY)))
|do|
(|printout| T T (CDR GREET-DATE)
T))))))

```

::: Misc environmental tailoring

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(FILESLoad (SYSLOAD NOERROR FROM LISPUSERS)
PAGEHOLD)
(RPAQQ |MaxValueLeftMargin| 512)
(RPAQQ SCROLLBARWIDTH 20)
(RPAQQ FIXSPELLDEFAULT \n)
(RPAQQ \ETHERTIMEOUT 3000)
(RPAQQ EMPRESS#SIDES 2)
(RPAQ *PRINT-CASE* :DOWNCASE)
(RPAQQ CLEANUPOPTIONS (RC ST))

```

```
(RPAQQ COPYRIGHTFLG DEFAULT)
(RPAQQ RECOMPILEDEFAULT EXPRS)
(RPAQQ *DEFAULT-CLEANUP-COMPILER* CL:COMPILE-FILE)
(RPAQQ *DEFAULT-MAKEFILE-ENVIRONMENT* (:PACKAGE "XCL-USER" :READTABLE "XCL" :BASE 10))
(RPAQ *ORIGINAL-GIVE-AND-TAKE-DIRECTORIES* *GIVE-AND-TAKE-DIRECTORIES*)
(RPAQ *GIVE-AND-TAKE-DIRECTORIES* (CL:REMOVE-IF #'(CL:LAMBDA (FILE)
                                         (CL:MEMBER (CL:PATHNAME-HOST FILE)
                                         '("ERIS" "ERINYES")
                                         :TEST
                                         #'STRING-EQUAL))
                                     *GIVE-AND-TAKE-DIRECTORIES*))
```

```
(XCL:REINSTALL-ADVICE 'FILES? :BEFORE '(:LAST (SETQ NOTLISTEDFILES NIL)))
(READWISE FILES?)
```

```
(EVAL-AT-GREET (CL:WHEN (AND (EQ MAKESYSNAME ' :LYRIC)
                             (EQ (MACHINETYPE)
                             'DORADO))
                  (FILESLOAD (SYSLOAD FROM LISPUSERS)
                             EXTENDEDVMEM)
                  (INSTALL-EXTENDED-VIRTUAL-MEMORY))
  (CL:WHEN (CL:FBOUNDP 'DESCRIBE-VIRTUAL-MEMORY)
    (DESCRIBE-VIRTUAL-MEMORY)))
```

```
(ADDTOVAR AFTERLOGOUTFORMS
  (|for| GREET-DATE |in| GREETDATES |bind| (TODAY _ (DATE))
    |when|
    (EQL 1 (STRPOS (CAR GREET-DATE)
                  TODAY NIL NIL NIL NIL (UPPERCASEARRAY)))
    |do|
    (|printout| T T (CDR GREET-DATE)
      T)))
)
```

```
(RPAQQ PATCH-INIT-COMMANDS
  (
```

;; Patches

```
(FNS PURGE-FILE-ADVICE PURGE-ADVICE)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
  ;;
  ;; Start the Lyric-only stuff
  (E (PRINTOUT NIL "(" .PPFTL ' (CL:WHEN (EQ MAKESYSNAME :LYRIC)
    T))
  ;; Add NS fileserv random-access support
  (INITVARS (*NSFILING-RANDOM-ACCESS* T))
  ;; Fix some compiler bogosity
  (ADVISE CL:COMPILE)
  ;; Add a few missing optimizers
  (FILES (NOERROR FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES LISPUSERSDIRECTORIES))
    BASE-OPTIMIZERS)
```

;; The interpreted LET* function is busted

```
(P (/PUTD 'LET* NIL))
```

;; The var \BrushAList is broken in the loadup - it ends in (... . NOBIND) instead of (... . NIL)

; Some people have their system set up to be so paranoid that it
; is always checking vars to see if they are eq to NOBIND, and
; generating an error if they are. Since, in the last loop thru the
; list, the var is indeed bound to NOBIND, we got troubles. The
; following piece of code is structured in a convoluted way to get
; around this problem.

```
(VARS (|\BrushAList| (CL:DO ((X |\BrushAList| (CDR X))
                            (NUMBER-GOOD-BRUSHES 0 (CL:INCF NUMBER-GOOD-BRUSHES))
                            (BRUSHES NIL)
                            (NUMBER-BRUSHES (LENGTH |\BrushAList|)))
  (= NUMBER-GOOD-BRUSHES NUMBER-BRUSHES)
  (CL:NREVERSE BRUSHES))
  (CL:PUSH (CAR X)
    BRUSHES))))
```

;; Advice saved in a file often gets duplicated. This tries to fix it.

```
(ADDVARS (MAKEFILEFORMS (PURGE-FILE-ADVICE FILE)))
(ADVISE MAKEFILE)
```

;; Fix the SEdit hang bug

```

(P (CHANGENAME '|\\seditA0001| 'READP '\\SYSBUFP))
;;
;; End the Lyric-only stuff
(E (PRINTOUT NIL " ") T)))

```

;;; Patches

(DEFINEQ

(PURGE-FILE-ADVICE

; Edited 30-Oct-87 11:08 by sml

```

(LAMBDA (FILE)
  (|for| F |in| (FILECOMSLST (ROOTFILENAME FILE)
    'ADVICE)
    |do| (PURGE-ADVICE F)))

```

(PURGE-ADVICE

; Edited 30-Oct-87 11:07 by sml

```

(LAMBDA (FN)
  (|if| (HASDEF FN 'ADVICE)
    |then| (PUTDEF FN 'ADVICE (LET ((ADVICE (GETDEF FN 'ADVICE))
      (INTERSECTION ADVICE ADVICE)))
    FN
    |else| NIL)))

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(CL:WHEN (EQ MAKESYSNAME :LYRIC)

```

(RPAQ? *NSFILING-RANDOM-ACCESS* T)
(XCL:REINSTALL-ADVICE 'CL:COMPILE :AROUND '(:LAST (LET (COMPILER::*INPUT-STREAM*
  (XCL:INNER))))))

```

```

(READWISE CL:COMPILE)
(FILESLOAD (NOERROR FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES LISPUSERSDIRECTORIES))
  BASE-OPTIMIZERS)
(/PUTD 'LET* NIL)
(RPAQ |\\BrushAList|
  (CL:DO ((X |\\BrushAList| (CDR X))
    (NUMBER-GOOD-BRUSHES 0 (CL:INCF NUMBER-GOOD-BRUSHES))
    (BRUSHES NIL)
    (NUMBER-BRUSHES (LENGTH |\\BrushAList|)))
    ((= NUMBER-GOOD-BRUSHES NUMBER-BRUSHES)
      (CL:NREVERSE BRUSHES))
    (CL:PUSH (CAR X)
      BRUSHES)))

```

```

(ADDTVAR MAKEFILEFORMS (PURGE-FILE-ADVICE FILE))
(XCL:REINSTALL-ADVICE 'MAKEFILE :AFTER '(:LAST (PURGE-FILE-ADVICE FILE)))
  :AROUND
  '(:LAST (LET ((PRETTYFLG (AND (NOT (MEMB 'FAST OPTIONS))
    PRETTYFLG))
    (DECLARE (CL:SPECIAL PRETTYFLG))
    *)
    (:LAST (LET ((|ObjectAlwaysPPFlag| NIL))
      *))))))
(READWISE MAKEFILE)
(CHANGENAME '|\\seditA0001| 'READP '\\SYSBUFP))

```

;; Wants to be (COMS * FILECACHE-INIT-COMMANDS), but the FileCrasher doesn't exist in Lyric, yet. Probably never will.

(RPAQQ FILECACHE-INIT-COMMANDS

;;; File caching stuff

```

(FILE (FROM "{EG:}<Lanning>FileCache>")
  FILECACHE)
(DECLARE\ : DONTEVAL@LOAD
  (P (|if| (BOUNDP 'FCACHE.VERSION.NUMBER)
    |then|
      ;; Set up some initial parameters
      (|for| |propDescr| |in| '((TRUST.CACHELIST T)
        (DELAY.DELETE T)
        (TIMETOVERIFY 300)
        (IO.BLOCK? T))
        |do|
          ; UNDOably, of course
          (UNDOSAVE `(FCACHE.PUTPROP ,(CAR |propDescr|)
            ,(FCACHE.GETPROP (CAR |propDescr|))))
          (FCACHE.PUTPROP (CAR |propDescr|)
            (CADR |propDescr|)))
      ;; Tell the fcache scavenger to ignore some files that I store on the cache partition

```

```
(/NCONC (ASSOC 'DORADO FCACHE.SCAVENGE.IGNORE)
(COPY '(*.SYSOUT UPDATELOOPS.CM)))
```

:: Treat any mail files as private files if I'm not on my normal machine

```
(|if| (NOT (STRING-EQUAL HOME-MACHINE-NAME (ETHERHOSTNAME)))
|then|
(|/push| PRIVATE.FILES '*.MAIL))))
```

:: File cache msg window stuff

```
(INITVARS (FILE-CACHE-MESSAGE-STREAM-REGION (|with| REGION (WINDOWPROP PROMPTWINDOW 'REGION)
(CREATEREGION LEFT (DIFFERENCE BOTTOM HEIGHT)
WIDTH HEIGHT)))
(FILE-CACHE-MESSAGE-STREAM-ICON-POSITION (|with| REGION (WINDOWPROP PROMPTWINDOW 'REGION)
(|create| POSITION XCOORD _ (DIFFERENCE LEFT 75)
YCOORD _ BOTTOM))))
(DECLARE\ : DONTEVAL@LOAD (P (|if| (BOUNDP 'FCACHE.VERSION.NUMBER)
|then|
(FILESLoad (SYSLOAD NOERROR FROM LISPUSERS)
FILECACHMSGWINDOW)
; UNDOably, of course
(UNDOSAVE `(DSPFONT , (DSPFONT (FONTCREATE 'GACHA 8)
*FILE-CACHE-MESSAGE-STREAM*)
, *FILE-CACHE-MESSAGE-STREAM*))
(DSPRESET *FILE-CACHE-MESSAGE-STREAM*)
(SHRINKW *FILE-CACHE-MESSAGE-STREAM*))))))
```

(RPAQQ **FONT-SETUP-INIT-COMMANDS**
(

::: Define printing/fonts the way people like. This includes setting prettyprinting fonts used by the editor and others, Lafite fonts, default flags controlling
::: prettyprinting at the top level, ... This needs to be early, so other utils get the desired fonts.

```
(ALISTS (FONTDEFS GENERIC-INIT))
(INITVARS (\\FONT-PROFILE-NAME 'GENERIC-INIT))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(P (EVAL-AT-GREET (FONTSET \\FONT-PROFILE-NAME T)
(CL:WHEN (BOUNDP 'PROMPTFONT)
(UNDOSAVE `(DSPFONT ,(DSPFONT PROMPTFONT PROMPTWINDOW)
,PROMPTWINDOW))))))
(ADVISE GETPROMPTWINDOW)))
```

::: Define printing/fonts the way people like. This includes setting prettyprinting fonts used by the editor and others, Lafite fonts, default flags controlling
::: prettyprinting at the top level, ... This needs to be early, so other utils get the desired fonts.

(ADDTOVAR **FONTDEFS**

```
(GENERIC-INIT (FONTCHANGEFLG . ALL)
(FILELINELENGTH . 102)
(COMMENTLINELENGTH 80 . 102)
(LAMBDAFONTLINELENGTH . 95)
(FIRSTCOL . 60)
(PRETTYLCOM . 25)
(LISTFILESTR . "
")
(|ObjectDontPPFlag| . T)
(SYSPRETTYFLG . T)
(**COMMENT**FLG)
(FONTPROFILE (DEFAULTFONT 1 (GACHA 10)
(GACHA 8)
(TERMINAL 8))
(BOLDFONT 2 (HELVETICA 10 BRR)
(HELVETICA 8 BRR)
(MODERN 8 BRR))
(LITTLEFONT 3 (HELVETICA 8)
(HELVETICA 6 MIR)
(MODERN 6 MIR))
(BIGFONT 4 (HELVETICA 12 BRR)
(HELVETICA 10 BRR)
(MODERN 10 BRR))
(USERFONT BOLDFONT)
(COMMENTFONT LITTLEFONT)
(LAMBDAFONT BIGFONT)
(SYSTEMFONT)
(CLISPFONT BOLDFONT)
(CHANGFONT)
(PRETTYCOMFONT BOLDFONT)
(TINYFONT LITTLEFONT)
(FONT1 DEFAULTFONT)
(FONT2 BOLDFONT)
(FONT3 LITTLEFONT)
(FONT4 BIGFONT)
(FONT5 5 (HELVETICA 10 BIR)
(HELVETICA 8 BIR)
(MODERN 8 BIR))
(FONT6 6 (HELVETICA 10 BRR)
(HELVETICA 8 BRR))
```

```

(MODERN 8 BRR))
(FONT7 7 (GACHA 12)
(GACHA 12)
(TERMINAL 12))
(FONT8 8 (CREAM 10)
(CREAM 10)
(MODERN 10 MIR))
(FONT9 9 (CREAM 10 BRR)
(CREAM 10 BRR)
(MODERN 10 BIR))
(FONT10 10 (CREAM 12)
(CREAM 12)
(MODERN 12 MIR))
(FONT11 11 (TIMESROMAN 10)
(TIMESROMAN 10)
(CLASSIC 10))
(|\\WindowTitleFont| BIGFONT)
(LAFITETITLEFONT |\\WindowTitleFont|)
(CHAT.FONT FONT7)))

```

(RPAQ? \\FONT-PROFILE-NAME 'GENERIC-INIT)

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

(EVAL-AT-GREET (FONTSET \\FONT-PROFILE-NAME T)
(CL:WHEN (BOUNDP 'PROMPTFONT)
(UNDOSAVE `(DSPFONT ,(DSPFONT PROMPTFONT PROMPTWINDOW)
,PROMPTWINDOW))))

```

```

(XCL:REINSTALL-ADVICE 'GETPROMPTWINDOW :BEFORE '(:LAST (COND
((AND (NULL FONT)
(BOUNDP 'PROMPTFONT))
(SETQ FONT PROMPTFONT))))))

```

(READWISE GETPROMPTWINDOW)
)

(DECLARE\ : EVAL@LOADWHEN

(NOT *GENERIC-INIT-LOADED*)
DOCOPY

```

(FILESLOAD (SYSLOAD FROM LISPUSERS)
LOADMENUITEMS)
)

```

;; Machine status

(RPAQQ WHO-LINE-COMMANDS
(

;; Give us a Who-Line

;; Load and start the who-line

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (FILES (SYSLOAD FROM LISPUSERS)
WHO-LINE)

```

;; Define these now, instead of with an INITVARS, because (i) Who-Line might have been in the sysout, and (ii) you can't define the
;; entries untill the Who-Line code is loaded.

```

(VARS (*WHO-LINE-ANCHOR* '(:JUSTIFY :TOP))
(*WHO-LINE-DISPLAY-NAMES?* T)
(*WHO-LINE-DIRECTORIES* (LIST |\\UserHomeDirectory|))
(*WHO-LINE-ENTRIES* (LIST *WHO-LINE-VMEM-ENTRY* *WHO-LINE-SYMBOL-SPACE-ENTRY*
*WHO-LINE-PACKAGE-ENTRY* *WHO-LINE-READTABLE-ENTRY*
*WHO-LINE-TTY-PROC-ENTRY* *WHO-LINE-TTY-STATE-ENTRY*
*WHO-LINE-DIRECTORY-ENTRY* *WHO-LINE-TIME-ENTRY*)))
(P (EVAL-AT-GREET (INSTALL-WHO-LINE-OPTIONS)
(CL:WHEN (AND (BOUNDP '*WHO-LINE*)
(WINDOWP *WHO-LINE*))
(UNDOSAVE `(CLOSEW ,*WHO-LINE*))))))

```

;; Give us a Who-Line

;; Load and start the who-line

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

(FILESLOAD (SYSLOAD FROM LISPUSERS)
WHO-LINE)

```

(RPAQQ *WHO-LINE-ANCHOR* (:JUSTIFY :TOP))

(RPAQQ *WHO-LINE-DISPLAY-NAMES?* T)

(RPAQ *WHO-LINE-DIRECTORIES* (LIST |\\UserHomeDirectory|))

```
(RPAQ *WHO-LINE-ENTRIES* (LIST *WHO-LINE-VMEM-ENTRY* *WHO-LINE-SYMBOL-SPACE-ENTRY* *WHO-LINE-PACKAGE-ENTRY*
*WHO-LINE-READTABLE-ENTRY* *WHO-LINE-TTY-PROC-ENTRY* *WHO-LINE-TTY-STATE-ENTRY*
*WHO-LINE-DIRECTORY-ENTRY* *WHO-LINE-TIME-ENTRY*))
```

```
(EVAL-AT-GREET (INSTALL-WHO-LINE-OPTIONS)
(CL:WHEN (AND (BOUNDP '*WHO-LINE*')
(WINDOWP *WHO-LINE*))
(UNDOSAVE `(CLOSEW ,*WHO-LINE*))))
)
```

```
(RPAQ VSTATS-INIT-COMMANDS
(
```

;;; Storage stuff

```
(INITVARS (VSTATS.CLOCK.INTERVAL 0)
(VSTATS.MUTIL.INTERVAL NIL)
(VSTATS.POSITION (CREATEPOSITION (DIFFERENCE SCREENWIDTH 147)
0)))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS
"System-Aids"
((SYSLOAD FROM LISPUSERS)
"VStats")
(LET ((VSTATS-ITEM (|for| ITEM |in| |BackgroundMenuCommands|
|thereis| (STRING-EQUAL "VStats"
(CAR ITEM)))))
(CL:WHEN VSTATS-ITEM (/DREMOVE VSTATS-ITEM
|BackgroundMenuCommands|)))
)
((SYSLOAD FROM LISPUSERS)
|Storage|
(SHOWSTORAGE 'ITEM))))
```

;;; Storage stuff

```
(RPAQ? VSTATS.CLOCK.INTERVAL 0)
(RPAQ? VSTATS.MUTIL.INTERVAL NIL)
(RPAQ? VSTATS.POSITION (CREATEPOSITION (DIFFERENCE SCREENWIDTH 147)
0))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(|AddLoadMenuItem| ' "System-Aids" ' ((SYSLOAD FROM LISPUSERS)
"VStats")
' (LET ((VSTATS-ITEM (|for| ITEM |in| |BackgroundMenuCommands| |thereis| (STRING-EQUAL "VStats" (CAR ITEM)))))
(CL:WHEN VSTATS-ITEM (/DREMOVE VSTATS-ITEM |BackgroundMenuCommands|))))
(|AddLoadMenuItem| ' "System-Aids" ' ((SYSLOAD FROM LISPUSERS)
|Storage|
' (SHOWSTORAGE 'ITEM))
)
```

;; Display control

```
(RPAQ SCREEN-SETUP-INIT-COMMANDS
(
```

;;; Screen layout stuff

```
(DECLARE\ :
DONTEVAL@LOAD DONTEVAL@COMPILE (VARS (WINDOWTITLESHAD E GRAYSHAD E))
;; Some interesting background shades
(FILE (SYSLOAD FROM "{FS8:PARC:XEROX}<Foster>Lisp>Users>")
"BITMAP-GALLERY" "GRANITE")
;; Rearrange the screen
(INITVARS (\\REARRANGE-SCREEN (NOT (CL:MEMBER "ROOMS" *MODULES* :TEST #'STRING-EQUAL)))
(P (EVAL-AT-GREET
(CL:WHEN \\REARRANGE-SCREEN
;; Change the background shade
(|GraniteBG|)
;; Fix up the prompt window
(UNDOSAVE `(DSPTEXTURE ,(DSPTEXTURE 42405 PROMPTWINDOW)
,PROMPTWINDOW))
(UNDOSAVE `(DSOPERATION ,(DSOPERATION 'REPLACE PROMPTWINDOW)
,PROMPTWINDOW))
(UNDOSAVE `(WINDOWPROP ,PROMPTWINDOW BORDER ,(WINDOWPROP PROMPTWINDOW 'BORDER)))
(WINDOWPROP PROMPTWINDOW 'BORDER 1)
(UNDOSAVE `(WINDOWPROP ,PROMPTWINDOW TITLE ,(WINDOWPROP PROMPTWINDOW 'TITLE))
```



```
(WINDOWPROP PROMPTWINDOW 'TITLE NIL)
(CLEARW PROMPTWINDOW)

;; Rearrange the screen a bit
(CL:WHEN (OPENWP LOGOW)
  (UNDOSAVE `(OPENW ,LOGOW))
  (CLOSEW LOGOW))
(LET* ((WINDOW-GAP 2)
  (WINDOW-WIDTH (DIFFERENCE (QUOTIENT SCREENWIDTH 2)
    (PLUS WINDOW-GAP WINDOW-GAP))))
  (UNDOSAVE `(SHAPEW ,PROMPTWINDOW ,(WINDOWPROP PROMPTWINDOW 'REGION)))
  (SHAPEW PROMPTWINDOW (LET ((HEIGHT (HEIGHTIFWINDOW (TIMES 4 (FONTPROP (DSPFONT NIL
    PROMPTWINDOW
    )
    'HEIGHT))
    (WINDOWPROP PROMPTWINDOW 'TITLE)
    (WINDOWPROP PROMPTWINDOW 'BORDER))))
    (CREATEREGION
      WINDOW-GAP
      (DIFFERENCE (|if| (AND (BOUNDP '*WHO-LINE*)
        (WINDOWP *WHO-LINE*))
        |then|
        (DIFFERENCE SCREENHEIGHT
          (|fetch| (REGION HEIGHT)
            |of|
            (WINDOWPROP *WHO-LINE*
              'REGION)))
          |else| SCREENHEIGHT)
        (PLUS WINDOW-GAP HEIGHT))
      WINDOW-WIDTH HEIGHT)))
    (UNDOSAVE `(SHAPEW ,|\\TopLevelTtyWindow| ,(WINDOWPROP |\\TopLevelTtyWindow| 'REGION)))
    (SHAPEW |\\TopLevelTtyWindow| (LET ((HEIGHT (HEIGHTIFWINDOW (TIMES 17
      (FONTPROP (DSPFONT
        NIL
        |\\TopLevelTtyWindow|
        )
        'HEIGHT))
      (WINDOWPROP |\\TopLevelTtyWindow|
        'TITLE)
      (WINDOWPROP |\\TopLevelTtyWindow|
        'BORDER))))
      (CREATEREGION WINDOW-GAP
        (DIFFERENCE (|fetch| (REGION BOTTOM)
          |of|
          (WINDOWPROP PROMPTWINDOW
            'REGION))
          (PLUS WINDOW-GAP HEIGHT))
        WINDOW-WIDTH HEIGHT))))))

;; Make the standard icon functions be cute
(INITVARS (\\LOAD-GRID-ICONS T)
  (ENFORCE.ICON.GRID T))
(P (EVAL-AT-GREET (CL:WHEN \\LOAD-GRID-ICONS
  (FILESLOAD (FROM LISPUSERS)
    GRID-ICONS))
  (CL:WHEN (AND \\REARRANGE-SCREEN (BOUNDP 'LOADED-FILES-ICON-WINDOW)
    (WINDOWP LOADED-FILES-ICON-WINDOW))
    (MOVEW LOADED-FILES-ICON-WINDOW 5 100))))))
```

;;; Screen layout stuff

```
(DECLARE\ : DONTVAL@LOAD DONTVAL@COMPILE)
(RPAQ WINDOWTITLESHADE GRAYSHADE)
(FILESLOAD (SYSLOAD FROM "{FS8:PARC:XEROX}<Foster>Lisp>Users>")
  "BITMAP-GALLERY" "GRANITE")
(RPAQ? \\REARRANGE-SCREEN (NOT (CL:MEMBER "ROOMS" *MODULES* :TEST #'STRING-EQUAL)))
(EVAL-AT-GREET (CL:WHEN \\REARRANGE-SCREEN
  ;; Change the background shade
  (|GraniteBG|)
  ;; Fix up the prompt window
  (UNDOSAVE `(DSPTEXTURE ,(DSPTEXTURE 42405 PROMPTWINDOW)
    ,PROMPTWINDOW))
  (UNDOSAVE `(DSOPERATION ,(DSOPERATION 'REPLACE PROMPTWINDOW)
    ,PROMPTWINDOW))
  (UNDOSAVE `(WINDOWPROP ,PROMPTWINDOW BORDER ,(WINDOWPROP PROMPTWINDOW 'BORDER)))
  (WINDOWPROP PROMPTWINDOW 'BORDER 1)
  (UNDOSAVE `(WINDOWPROP ,PROMPTWINDOW TITLE ,(WINDOWPROP PROMPTWINDOW 'TITLE)))
  (WINDOWPROP PROMPTWINDOW 'TITLE NIL)
  (CLEARW PROMPTWINDOW)
  ;; Rearrange the screen a bit
```

```

(CL:WHEN (OPENWP LOGOW)
  (UNDOSAVE `(OPENW ,LOGOW))
  (CLOSEW LOGOW))
(LET* ((WINDOW-GAP 2)
  (WINDOW-WIDTH (DIFFERENCE (QUOTIENT SCREENWIDTH 2)
    (PLUS WINDOW-GAP WINDOW-GAP))))
  (UNDOSAVE `(SHAPEW ,PROMPTWINDOW ,(WINDOWPROP PROMPTWINDOW 'REGION)))
  (SHAPEW PROMPTWINDOW
    (LET ((HEIGHT (HEIGHTIFWINDOW (TIMES 4 (FONTPROP (DSPFONT NIL PROMPTWINDOW)
      'HEIGHT))
      (WINDOWPROP PROMPTWINDOW 'TITLE)
      (WINDOWPROP PROMPTWINDOW 'BORDER))))
      (CREATEREGION WINDOW-GAP
        (DIFFERENCE (|if| (AND (BOUNDP '*WHO-LINE*
          (WINDOWP *WHO-LINE*))
          |then| (DIFFERENCE SCREENHEIGHT
            (|fetch| (REGION HEIGHT)
              |of| (WINDOWPROP *WHO-LINE*
                'REGION)))
            |else| SCREENHEIGHT)
          (PLUS WINDOW-GAP HEIGHT))
        WINDOW-WIDTH HEIGHT)))
  (UNDOSAVE `(SHAPEW ,|\\TopLevelTtyWindow| ,(WINDOWPROP |\\TopLevelTtyWindow|
    'REGION)))
  (SHAPEW |\\TopLevelTtyWindow| (LET ((HEIGHT (HEIGHTIFWINDOW (TIMES 17
    (FONTPROP (DSPFONT NIL
      |\\TopLevelTtyWindow|
    )
    'HEIGHT))
    (WINDOWPROP |\\TopLevelTtyWindow|
      'TITLE)
    (WINDOWPROP |\\TopLevelTtyWindow|
      'BORDER))))
    (CREATEREGION WINDOW-GAP
      (DIFFERENCE (|fetch| (REGION BOTTOM)
        |of| (WINDOWPROP PROMPTWINDOW
          'REGION))
        (PLUS WINDOW-GAP HEIGHT))
      WINDOW-WIDTH HEIGHT))))))

```

(RPAQ? \\LOAD-GRID-ICONS T)

(RPAQ? ENFORCE.ICON.GRID T)

```

(EVAL-AT-GREET (CL:WHEN \\LOAD-GRID-ICONS
  (FILESLOAD (FROM LISPUSERS)
    GRID-ICONS))
  (CL:WHEN (AND \\REARRANGE-SCREEN (BOUNDP 'LOADED-FILES-ICON-WINDOW)
    (WINDOWP LOADED-FILES-ICON-WINDOW))
    (MOVEW LOADED-FILES-ICON-WINDOW 5 100)))
)

```

(RPAQQ ROOMS-INIT-COMMANDS

```

((DECLARE\
  DONTEVAL@LOAD DONTEVAL@COMPILE (INITVARS (USER-SUITE-DIRECTORIES (LIST (CONCAT |\\UserHomeDirectory|
    "ROOMS>"))
    (ROOMSDIRECTORIES (CASE MAKESYSNAME
      (:LYRIC `(|,@*CACHE-DIRECTORIES*
        "{NB:PARC:XEROX}<Rooms>Lyric>"))
      (CL:OTHERWISE `(|,@*CACHE-DIRECTORIES*
        "{NB:PARC:XEROX}<Room
s>Medley>Sources> " "{Pogo:AINorth:XEROX}<Rooms>Medley>Sources>"))))
    (ROOMSUSERSDIRECTORIES (CASE MAKESYSNAME
      (:LYRIC `(|,@*CACHE-DIRECTORIES*
        "{NB:PARC:XEROX}<Rooms
>Lyric>Users> " "{Pogo:AINorth:XEROX}<Rooms>Lyric>Users>"))
      (CL:OTHERWISE `(|,@*CACHE-DIRECTORIES*
        "{NB:PARC:XEROX}<Rooms
>Medley>Users> " "{NB:PARC:XEROX}<Rooms>Lyric>Users> " "{Pogo:AINorth:XEROX}<Rooms>Medley>Users> "
        "{Pogo:AINorth:XEROX}<Rooms>Lyric>Users>")))))
)

```

; Force CL:EVAL instead of the default IL:EVAL, since IL:EVAL doesn't understand some things.

```

(LOADMENUITEMS
  "Screen-Maintanance"
  ((SYSLOAD FROM ROOMS)
    "ROOMS")
  (CL:EVAL '(PROGN (SET (CL:INTERN "*SUITE-DIRECTORIES*" "ROOMS")
    (APPEND USER-SUITE-DIRECTORIES (CL:EVAL (CL:INTERN "*SUITE-DIRECTORIES*"
      "ROOMS"))))
    (|AddLoadMenuItem| "Rooms" `((SYSLOAD FROM "{NB:PARC:Xerox}<CommonLens>CURRENT>"
      "CLens-Rooms"))
    (CL:DOLIST (UTILITY '("Rooms-Intro" "Rooms-Notes"))

```

```
(|AddLoadMenuItem| "Rooms" `((SYSLOAD FROM ROOMS
,UTILITY)))
(CL:DOLIST (UTILITY `("Background-Menu-Buttons"
"Office.suite" "Touchy-Buttons" "Random-Window-Types"
"Snuggle" "Notecards-Window-Types"
,@(IF (NOT (CL:FBOUNDP 'LAFITE))
THEN NIL ELSEIF (NOT (CL:FBOUNDP
'\LAFITE.GETMAILFOLDER))
THEN
'("New-Lafite-Window-Types")
ELSE
'("Lafite-Window-Types")))
"Un-Hide-Tty" "Rooms-Videohax"
,@(IF (EQ MAKESYSNAME :LYRIC)
THEN NIL ELSE '("WallPaper"))))
(|AddLoadMenuItem| "Rooms" `((SYSLOAD FROM ROOMSUSERS
,UTILITY))))))
```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(RPAQ? **USER-SUITE-DIRECTORIES** (LIST (CONCAT |\\UserHomeDirectory| "ROOMS>")))

```
(RPAQ? ROOMSDIRECTORIES (CASE MAKESYSNAME
(:LYRIC `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Rooms>Lyric">"
"{Pogo:AINorth:XEROX}<Rooms>Lyric">"))
(CL:OTHERWISE `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Rooms>Medley>Sources">"
"{Pogo:AINorth:XEROX}<Rooms>Medley>Sources">))))
```

```
(RPAQ? ROOMSUSERSDIRECTORIES (CASE MAKESYSNAME
(:LYRIC `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Rooms>Lyric>Users">"
"{Pogo:AINorth:XEROX}<Rooms>Lyric>Users">"))
(CL:OTHERWISE `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Rooms>Medley>Users">"
"{NB:PARC:XEROX}<Rooms>Lyric>Users">"
"{Pogo:AINorth:XEROX}<Rooms>Medley>Users">"
"{Pogo:AINorth:XEROX}<Rooms>Lyric>Users">))))
```

```
(|AddLoadMenuItem|
'Screen-Maintanance"
'((SYSLOAD FROM ROOMS)
"ROOMS")
'(CL:EVAL '(PROGN (SET (CL:INTERN "*SUITE-DIRECTORIES*" "ROOMS")
(APPEND USER-SUITE-DIRECTORIES (CL:EVAL (CL:INTERN "*SUITE-DIRECTORIES*" "ROOMS"))))
(|AddLoadMenuItem| "Rooms" `((SYSLOAD FROM "{NB:PARC:Xerox}<CommonLens>CURRENT">"
"CLens-Rooms"))
(CL:DOLIST (UTILITY '("Rooms-Intro" "Rooms-Notes"))
(|AddLoadMenuItem| "Rooms" `((SYSLOAD FROM ROOMS)
,UTILITY)))
(CL:DOLIST (UTILITY `("Background-Menu-Buttons"
"Office.suite" "Touchy-Buttons" "Random-Window-Types" "Snuggle"
"Notecards-Window-Types" ,@(IF (NOT (CL:FBOUNDP 'LAFITE))
THEN NIL
ELSEIF (NOT (CL:FBOUNDP
'\LAFITE.GETMAILFOLDER
))
THEN '("New-Lafite-Window-Types")
ELSE '("Lafite-Window-Types")))
"Un-Hide-Tty" "Rooms-Videohax" ,@(IF (EQ MAKESYSNAME :LYRIC)
THEN NIL
ELSE '("WallPaper"))))
(|AddLoadMenuItem| "Rooms" `((SYSLOAD FROM ROOMSUSERS)
,UTILITY))))))
```

(RPAQQ **CHANGE-BACKGROUND-INIT-COMMANDS**

;;; Make it easy to change your background

```
(DECLARE\ :
DONTEVAL@LOAD DONTEVAL@COMPILE
(LOADMENUITEMS
NIL
((SYSLOAD FROM LISPUSERS)
"BackgroundImages" "StarBG" (NIL FROM "{PHYLUM}<Loops>Faces">")
"Dead")
(|push|
|BackgroundMenuCommands|
(LET
((BITMAPS '("Standard texture" . WINDOWBACKGROUNDSHADE)
("Gray shade" . GRAYSHADE)
("Light wall paper" . LIGHTWALLPAPER)
("Medium wall paper" . WALLPAPER)
("Dark wall paper" . DARKWALLPAPER)
("Waves" . WAVE-TEXTURE)
("Tweed" . XCL-USER:*TWEED-BM*)
("Chambray" . XCL-USER:*CHAMBRAY-BM*)
("Canvas" . XCL-USER:*CANVAS-BM*))
```

```

("Corduroy" . XCL-USER::*CORDUROY-BM*)
("Seersucker" . XCL-USER::*SEERSUCKER-BM*)
("Burlap" . XCL-USER::*BURLAP-BM*)
("Mesh" . MESH-TEXTURE)))
(IMAGES (BACKGROUND.FILES))
(MENU-ITEMS '(("Stars" '(STARBG)
              "Fill the background with stars"
              (SUBITEMS ("ET go home" '(|SaucerOff|)
                        "Get rid of the saucer"))))
          ("Granite" '(|GraniteBG|)
              "Fill the background with a pleasing, almost random shade")
          ("Dead" (CHANGEBACKGROUND (EXPANDBITMAP DEAD-BM 2 2))
              "Put a Deadhead on the background"))))
'("Change background"
  '(CHANGEBACKGROUND)
  "Change the background to the default texture"
  (SUBITEMS
    ("Random texture"
      '(LET ((SHADE (|\Pick-One-At-Random| '(@(|for| BM |in| BITMAPS |collect|
        (|if| (CL:SYMBOLP BM)
              |then| BM |else| (CDR BM)))
              ,@(|for| ITEM |in| MENU-ITEMS |collect| (CAR ITEM))
              ,@IMAGES "Rand"))))
        (COND ((FOR ITEM IN MENU-ITEMS THEREIS (AND (CL:CONSP ITEM)
              (STRINGP (CAR ITEM))
              (STRING-EQUAL SHADE (CAR ITEM))))
          FINALLY
            (IF $SVAL THEN (EVAL (CADR ITEM))))))
        ((EQUAL SHADE "Rand")
          (CHANGEBACKGROUND (RAND 0 BLACKSHADE)))
        ((LISTP SHADE)
          (BACKGROUND.FETCH (CAR SHADE)
            (CDR SHADE)))
        ((LITATOM SHADE)
          (CHANGEBACKGROUND (EVALV SHADE)))
        (T (CHANGEBACKGROUND SHADE))))
      "Change the background texture to a randomly selected shade")
    ,@(|for| BM |in| BITMAPS |collect| (|if| (CL:SYMBOLP BM)
      |then|
        '(, BM (CHANGEBACKGROUND , BM))
      |else|
        '(, (CAR BM)
          (CHANGEBACKGROUND , (CDR BM))))))
    ,@MENU-ITEMS
    ,@(|for| IMAGE |in| IMAGES |collect| '(, (CAR IMAGE)
      '(BACKGROUND.FETCH ', (CAR IMAGE)
        ', (CDR IMAGE))))))))))

```

;;; Make it easy to change your background

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
```

```

(|AddLoadMenuItem|
'NIL
'((SYSLOAD FROM LISPUSERS)
  "BackgroundImages" "StarBG" (NIL FROM "{PHYLUM}<Loops>Faces>")
  "Dead")
'(|push|
  |BackgroundMenuCommands|
  (LET
    ((BITMAPS '("Standard texture" . WINDOWBACKGROUNDSHADE)
              ("Gray shade" . GRAYSHADE)
              ("Light wall paper" . LIGHTWALLPAPER)
              ("Medium wall paper" . WALLPAPER)
              ("Dark wall paper" . DARKWALLPAPER)
              ("Waves" . WAVE-TEXTURE)
              ("Tweed" . XCL-USER::*TWEED-BM*)
              ("Chambray" . XCL-USER::*CHAMBRAY-BM*)
              ("Canvas" . XCL-USER::*CANVAS-BM*)
              ("Corduroy" . XCL-USER::*CORDUROY-BM*)
              ("Seersucker" . XCL-USER::*SEERSUCKER-BM*)
              ("Burlap" . XCL-USER::*BURLAP-BM*)
              ("Mesh" . MESH-TEXTURE)))
    (IMAGES (BACKGROUND.FILES))
    (MENU-ITEMS '(("Stars" '(STARBG)
                      "Fill the background with stars"
                      (SUBITEMS ("ET go home" '(|SaucerOff|)
                                "Get rid of the saucer"))))
              ("Granite" '(|GraniteBG|)
                      "Fill the background with a pleasing, almost random shade")
              ("Dead" (CHANGEBACKGROUND (EXPANDBITMAP DEAD-BM 2 2))
                      "Put a Deadhead on the background"))))
  '("Change background"
    '(CHANGEBACKGROUND)
    "Change the background to the default texture"
    (SUBITEMS ("Random texture"

```

```

      (LET ((SHADE (|\Pick-One-At-Random| ' (,@(|for| BM |in| BITMAPS
                                         |collect| (|if| (CL:SYMBOLP BM)
                                                         |then| BM
                                                         |else| (CDR BM)))
                                         ,@(|for| ITEM |in| MENU-ITEMS |collect| (CAR ITEM)
                                         ,@IMAGES "Rand"))))
          (COND
           ((FOR ITEM IN MENU-ITEMS THEREIS (AND (CL:CONSP ITEM)
                                                  (STRINGP (CAR ITEM))
                                                  (STRING-EQUAL SHADE (CAR ITEM)))
            (FINALLY (IF $$VAL
                      THEN (EVAL (CADR ITEM))))
              ((EQUAL SHADE "Rand")
               (CHANGEBACKGROUND (RAND 0 BLACKSHADE)))
              ((LISTP SHADE)
               (BACKGROUND.FETCH (CAR SHADE)
                                   (CDR SHADE)))
              ((LITATOM SHADE)
               (CHANGEBACKGROUND (EVALV SHADE)))
              (T (CHANGEBACKGROUND SHADE))))
            "Change the background texture to a randomly selected shade")
           ,@(|for| BM |in| BITMAPS |collect| (|if| (CL:SYMBOLP BM)
                                                  |then| ` (, BM (CHANGEBACKGROUND , BM))
                                                  |else| ` (, (CAR BM)
                                                  (CHANGEBACKGROUND , (CDR BM))))))
           ,@MENU-ITEMS
           ,@(|for| IMAGE |in| IMAGES |collect| ` (, (CAR IMAGE)
                                                  ' (BACKGROUND.FETCH ' (, (CAR IMAGE)
                                                  ' (, (CDR IMAGE)))))))))
)

```

(RPAQQ DISPLAY-CONTROL-INIT-COMMANDS

```

  ((DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "Screen-Maintanance" ((SYSLOAD FROM LISPUSERS)
                                                                                   "WDWHacks"))
            ((SYSLOAD FROM LISPUSERS)
             "Turbo-Windows"))
            ((SYSLOAD FROM LISPUSERS)
             "Solid-Movew"))
            ((SYSLOAD FROM LISPUSERS)
             "NSDisplaySizes"))
            ((SYSLOAD FROM LISPUSERS)
             "SNAPW-ICON"))))
)

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

(|AddLoadMenuItem| "Screen-Maintanance" ' ((SYSLOAD FROM LISPUSERS)
                                             "WDWHacks"))
(|AddLoadMenuItem| "Screen-Maintanance" ' ((SYSLOAD FROM LISPUSERS)
                                             "Turbo-Windows"))
(|AddLoadMenuItem| "Screen-Maintanance" ' ((SYSLOAD FROM LISPUSERS)
                                             "Solid-Movew"))
(|AddLoadMenuItem| "Screen-Maintanance" ' ((SYSLOAD FROM LISPUSERS)
                                             "NSDisplaySizes"))
(|AddLoadMenuItem| "Screen-Maintanance" ' ((SYSLOAD FROM LISPUSERS)
                                             "SNAPW-ICON"))
)

```

(RPAQQ IDLE-INIT-COMMANDS

```

(
  ;;: The Idle package
  (DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "IdlePatterns" ((SYSLOAD FROM LISPUSERS)
                                                                                   "IdleHax"))
            ((SYSLOAD FROM LISPUSERS)
             "IdleDrain")
            (/LISTPUT IDLE.PROFILE 'DISPLAYFN ' IDLE-DRAIN))
            ((SYSLOAD FROM LISPUSERS)
             "ReadBrush"))
            ((SYSLOAD FROM "{PHYLUM}<Colab>Andes>Users>")
             "Bouncing-Face"))
            ((SYSLOAD FROM LISPUSERS)
             "StarBG")
            (/LISTPUT IDLE.PROFILE 'DISPLAYFN ' |Cosmos|))
            ((SYSLOAD FROM LISPUSERS)
             "Pac-Man-Idle")
            (/LISTPUT IDLE.PROFILE 'DISPLAYFN ' |Pac-Man-Idle|))
            ((SYSLOAD FROM "{QV}<Bagley>Lisp>")
             "Idle-Cost"))
            ((SYSLOAD)
             "ScreenPaper")
            (/LISTPUT IDLE.PROFILE 'DISPLAYFN ' SCREENPAPER))
            ((SYSLOAD FROM PRIVATE-LISPUSERS)

```

```

                                "Idle-Lyrics")
                                (/LISTPUT IDLE.PROFILE 'DISPLAYFN
                                'XCL-USER::IDLE-LYRICS))
(P (EVAL-AT-GREET (/LISTPUT IDLE.PROFILE 'FORGET NIL
                            (/LISTPUT IDLE.PROFILE 'ALLOWED.LOGINS NIL))))))

```

;; The Idle package

```

(DECLARE\ DONTEVAL@LOAD DONTEVAL@COMPILE
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM LISPUSERS)
                                   "IdleHax"))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM LISPUSERS)
                                   "IdleDrain")
' (/LISTPUT IDLE.PROFILE 'DISPLAYFN 'IDLE-DRAIN))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM LISPUSERS)
                                   "ReadBrush"))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM "{PHYLUM}<Colab>Andes>Users>"
                                   "Bouncing-Face"))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM LISPUSERS)
                                   "StarBG")
' (/LISTPUT IDLE.PROFILE 'DISPLAYFN '|Cosmos|))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM LISPUSERS)
                                   "Pac-Man-Idle")
' (/LISTPUT IDLE.PROFILE 'DISPLAYFN '|Pac-Man-Idle|))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM "{QV}<Bagley>Lisp>"
                                   "Idle-Cost"))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD)
                                   "ScreenPaper")
' (/LISTPUT IDLE.PROFILE 'DISPLAYFN 'SCREENPAPER))
(|AddLoadMenuItem| "IdlePatterns" '((SYSLOAD FROM PRIVATE-LISPUSERS)
                                   "Idle-Lyrics")
' (/LISTPUT IDLE.PROFILE 'DISPLAYFN 'XCL-USER::IDLE-LYRICS))
(EVAL-AT-GREET (/LISTPUT IDLE.PROFILE 'FORGET NIL
                  (/LISTPUT IDLE.PROFILE 'ALLOWED.LOGINS NIL))
)

```

(RPAQQ **CLOCK-INIT-COMMANDS**

;; Telling the time

```

;; Standard clock is the Biclock, in the lower-left corner
(INITVARS (BICLOCKINITIALPROPS '(HORIZONTAL LEFT VERTICAL BOTTOM SIZE 95)))
;; Optional clock is CROCK, also in the lower-left corner
(DECLARE\ DONTEVAL@LOAD DONTEVAL@COMPILE (FILES (SYSLOAD NOERROR FROM LISPUSERS)
                                                BICLOCK)
(LOADMENUITEMS NIL (((SYSLOAD FROM LISPUSERS)
                    "Crock")
                    (CL:WHEN (AND (BOUNDP 'BICLOCKWINDOW)
                                   (WINDOWP BICLOCKWINDOW))
                              (CLOSEW BICLOCKWINDOW))))))

```

;; Telling the time

;; Standard clock is the Biclock, in the lower-left corner

```
(RPAQ? BICLOCKINITIALPROPS '(HORIZONTAL LEFT VERTICAL BOTTOM SIZE 95))
```

;; Optional clock is CROCK, also in the lower-left corner

```

(DECLARE\ DONTEVAL@LOAD DONTEVAL@COMPILE
(FILESLOAD (SYSLOAD NOERROR FROM LISPUSERS)
           BICLOCK)
(|AddLoadMenuItem| 'NIL '((SYSLOAD FROM LISPUSERS)
                          "Crock")
' (CL:WHEN (AND (BOUNDP 'BICLOCKWINDOW)
                (WINDOWP BICLOCKWINDOW))
            (CLOSEW BICLOCKWINDOW)))
)

```

;; Programming stuff

(RPAQQ **PROGRAMMING-INIT-COMMANDS**

(

::: Editing code

(FUNCTIONS NOTICE)

:: Saving files

(FUNCTIONS MAKE)

:: For testing optimizers

(FUNCTIONS OAM)

:: Handy exec commands

(COMMANDS "BREAK" "UNBREAK" "CALLS" "DESCRIBE" "EC" "EFF" "FILES?" "IC" "NOTICE" "MAKE" "SPY")

:: never really worked - (loadmenuitems "ProgrammingAids" (((sysload from lispusers) "Step-Command-Menu")))

:::

:: SEDIT stuff

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (P (EVAL-AT-GREET

;; Give us a META key if running on a Dorado

(SELECTQ (MACHINETYPE)
(DORADO (METASHIFT T))
NIL)

:: Reset SEdit so it gets the correct fonts

(CASE MAKESYSNAME
(:LYRIC (SEdit.RESET))
(CL:OTHERWISE (CL:FUNCALL (CL:INTERN
"RESET"
(CL:FIND-PACKAGE
"SEdit")))))

:: Hacking SEdit so it uses better package/readtable defaults

(CASE MAKESYSNAME
(:LYRIC (FILESLOAD (SYSLOAD NOERROR FROM VALUEOF
(APPEND
PRIVATE-LISPUSERSDIRECTORIES
LISPUSERSDIRECTORIES
))
"SEdit-HACK"))
(CL:OTHERWISE (FILESLOAD (SYSLOAD FROM LISPUSERS
)
"seedit-profile"))))

:: Change the SEdit EXPAND behavior to expand definers in a reasonable way

(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
LISPUSERSDIRECTORIES))
"SEdit-Definers")))

:: The Eval&Insert hook

(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
LISPUSERSDIRECTORIES))
"SEdit-Eval")))

:: Exit/Compile hooks

(P (CASE MAKESYSNAME
(:LYRIC (|AddLoadMenuItem| "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
"SEdit-Compile"))))

:: TTY editor stuff

(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"CL-TTYedit")))

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

:: Better WHEREIS facility

(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"New-Where-Is")))

:: Checking out lexical contexts in a break

(P (CASE MAKESYSNAME
(:LYRIC (|AddLoadMenuItem| "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
"Debugger-Context"))))

:: Save-Your-Ass

(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"Checkpoint")))

:: Spy button

(INITVARS (\sPY.BUTTON.POS NIL))
(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
SPY)
(SPY.BUTTON \sPY.BUTTON.POS)))

```

;; Graph calls
(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"GraphCalls")))

;; The Source Manager
(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"Manager")))

;; Better file listing tools
(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"PP-Code-File")
(((SYSLOAD FROM LISPUSERS)
"PrettyFileIndex"))))

;; TEdit executive
(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM LISPUSERS)
"TExec")))

;; Iteration packages
(LOADMENUITEMS "ProgrammingAids" (((SYSLOAD FROM "{Phylum}<Brotsky>rcw">")
"OSS"))))

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
;; Moving between Xerox Lisp and the rest of the world
(P (|AddLoadMenuItem| "ProgrammingAids" (CASE MAKESYSNAME
(:LYRIC ' ((SYSLOAD FROM LISPUSERS)
"Port-CLFile"))
(CL:OTHERWISE ' ((SYSLOAD FROM LISPUSERS)
"XCL-Bridge"))))))))

```

;;; Editing code

```

(CL:DEFUN NOTICE (&REST FILES)
"Notice a set of files, so things on them can be edited"

```

;;; Return 4 values: a list of all files that were noticed, a list of files that were already noticed, a list of files that weren't noticed because they weren't loaded, and a list of files that couldn't be found.

```

(CL:LABELS ((CANONOCAL-FILEMANAGER-NAME (PATH)
"Return the canonical FileManager name of a file"
(CL:INTERN (CL:STRING-UPCASE (CL:PATHNAME-NAME PATH))
(CL:FIND-PACKAGE "IL"))))
(FIND-SOURCE-FILE (FILE-NAME &OPTIONAL (SEARCH-PATH-LIST DIRECTORIES))
"Return the full pathname of the source file"
(OR
(CL:PROBE-FILE FILE-NAME) ; In case we are given enough to find the file
; Check for the original source, in it's original location
(LET ((ORIGINAL-SOURCE-FILE-NAME (CDR (CL:FIRST (GET (CANONOCAL-FILEMANAGER-NAME
(PATHNAME FILE-NAME)
' FILEDATES)))))
(CL:IF ORIGINAL-SOURCE-FILE-NAME
(CL:PROBE-FILE ORIGINAL-SOURCE-FILE-NAME)
NIL)) ; As a last resort, check the list of directories
(CL:FIND-IF #'CL:PROBE-FILE (CL:MAPCAR #'(CL:LAMBDA (DIR)
(CL:MERGE-PATHNAMES FILE-NAME DIR))
SEARCH-PATH-LIST))))))
(FILE-NOTICED-P (PATH)
"Has the file been noticed?"
(CL:MEMBER (CANONOCAL-FILEMANAGER-NAME PATH)
FILELST :TEST 'EQ))
(FILE-LOADED-P (PATH)
"Has the file been loaded?"
(NOT (NULL (GET (CANONOCAL-FILEMANAGER-NAME PATH)
' FILEDATES))))
(NOTICE-FILE (PATH)
"Notice the file"
(LOAD PATH 'PROP)))
(LET ((ALREAD-NOTICED-FILES NIL)
(NOT-LOADED-FILES NIL)
(NOTICED-FILES NIL)
(NOT-FOUND-FILES NIL))
(CL:MAPC #'(CL:LAMBDA (FILE)
(LET ((PATHNAME (FIND-SOURCE-FILE FILE)))
(COND
((NULL PATHNAME)
(CL:PUSH FILE NOT-FOUND-FILES))
((FILE-NOTICED-P PATHNAME)
(CL:PUSH PATHNAME ALREAD-NOTICED-FILES))
((FILE-LOADED-P PATHNAME)
(LOADFROM PATHNAME NIL 'PROP)
(CL:PUSH PATHNAME NOTICED-FILES))
(T (CL:PUSH PATHNAME NOT-LOADED-FILES))))))
FILES)
(CL:VALUES NOTICED-FILES ALREAD-NOTICED-FILES NOT-LOADED-FILES NOT-FOUND-FILES)))

```


:: Saving files

```
(CL:DEFUN MAKE (FILES)
  (LET ((FILES (OR FILES (CL:REMOVE-IF-NOT #'(CL:LAMBDA (FILE-NAME)
                                                (CDR (GET FILE-NAME 'FILE)))
                                                FILELST)))
        (ORIGINAL-DIR *DEFAULT-PATHNAME-DEFAULTS*)
        FILE-DIR ROOPT-FILE)
    (CL:UNWIND-PROTECT
      (CL:DOLIST (FILE FILES)
        (CL:SETQ ROOPT-FILE (CL:PATHNAME-NAME FILE))
        (CL:SETQ ROOPT-FILE (CL:TYPECASE ROOPT-FILE
          (STRING (CL:INTERN (CL:STRING-UPCASE ROOPT-FILE)
                             (CL:FIND-PACKAGE "IL"))))
          (CL:SYMBOL (CL:INTERN (CL:SYMBOL-NAME ROOPT-FILE)
                              (CL:FIND-PACKAGE "IL")))))
        (CNDIR (CL:IF (GET ROOPT-FILE 'FILEDATES)
          (LET ((FILE-DIR (UNPACKFILENAME.STRING (CDR (CL:FIRST (GET ROOPT-FILE 'FILEDATES)))))
                )
            (PACKFILENAME.STRING 'HOST (CL:GETF FILE-DIR 'HOST)
              'DEVICE
              (CL:GETF FILE-DIR 'DEVICE)
              'DIRECTORY
              (CL:GETF FILE-DIR 'DIRECTORY)))
              ORIGINAL-DIR))
          (CL:WHEN (CL:FUNCALL 'CLEANUP ROOPT-FILE)
            (CL:LOAD (PACKFILENAME.STRING 'NAME ROOPT-FILE 'EXTENSION "dfasl"))))
          (CNDIR ORIGINAL-DIR))))))
```

:: For testing optimizers

```
(CL:DEFUN OAM (FORM)
  "Optimize and Macroexpand the form. For use as an Sedit mutator."
  (COMPILER:OPTIMIZE-AND-MACROEXPAND-1 FORM (COMPILER:MAKE-EMPTY-ENV)
    (COMPILER:MAKE-CONTEXT)))
```

:: Handy exec commands

```
(DEFCOMMAND "BREAK" (&REST FNS) "Set a breakpoint on the named functions."
  (EVAL `(BREAK ,@FNS)))

(DEFCOMMAND "UNBREAK" (&REST FNS) "Remove a breakpoint from the named functions."
  (EVAL `(UNBREAK ,@FNS)))

(DEFCOMMAND "CALLS" (FN) "Print out information about what the function calls."
  (COND
    ((NOT (CL:FBOUNDP FN))
      (CL:FORMAT T "~%~S has no function definition" FN))
    ((CL:MACRO-FUNCTION FN)
      (CL:FORMAT T "~%~S is a macro" FN))
    ((CL:SPECIAL-FORM-P FN)
      (CL:FORMAT T "~%~S is a special-form" FN))
    (T (DESTRUCTURING-BIND (CALLS BINDS USES-FREE USES-GLOBAL)
      (CALLS FN)
      (CL:FORMAT T "~%--- ~S ---" FN)
      (LET ((FORMAT-STRING "~%~A:~{ ~S~}"))
        (CL:WHEN (NOT (NULL CALLS))
          (CL:FORMAT T FORMAT-STRING "CALLS" CALLS))
        (CL:WHEN (NOT (NULL BINDS))
          (CL:FORMAT T FORMAT-STRING "BINDS" BINDS))
        (CL:WHEN (NOT (NULL USES-FREE))
          (CL:FORMAT T FORMAT-STRING "SPECIALS USED" USES-FREE))
        (CL:WHEN (NOT (NULL USES-GLOBAL))
          (CL:FORMAT T FORMAT-STRING "GLOBALS USED" USES-GLOBAL))))))
    (CL:VALUES))

(DEFCOMMAND "DESCRIBE" (&REST OBJECTS) "Describe the named objects."
  (CL:MAPC #'(CL:LAMBDA (X)
    (CL:FORMAT T "~&-- ~A --" X)
    (CL:DESCRIBE X))
    OBJECTS)
  (CL:VALUES))

(DEFCOMMAND "EC" (FORM) "Evaluate a compiled version of the form"
  (CL:FUNCALL (PROG2 (CL:FORMAT T "~%Compiling...")
    (CL:COMPILE NIL `(CL:LAMBDA NIL ,FORM))
    (CL:FORMAT T "done.~%"))))
```

```
(DEFCOMMAND "EFF" (&REST PATTERNS-COMMANDS)
  "Edit any uses of any of the patterns on any noticed file.  Args are ..patterns - ..edit comands."
  (LET* ((POSITION (CL:POSITION "-" PATTERNS-COMMANDS :KEY #'(LAMBDA (PATTERN)
    (IF (CL:SYMBOLP PATTERN)
      THEN (CL:SYMBOL-NAME PATTERN)
      ELSE "")))
    :TEST #'STRING-EQUAL))
    (PATTERNS (IF (NULL POSITION)
      THEN PATTERNS-COMMANDS
      ELSE (CL:BUTLAST PATTERNS-COMMANDS (- (LENGTH PATTERNS-COMMANDS)
        POSITION))))
    (EDIT-COMMANDS (IF POSITION
      THEN (CL:SUBSEQ PATTERNS-COMMANDS (1+ POSITION))
      ELSE NIL)))
    (CASE (CL:LENGTH PATTERNS)
      (0 NIL)
      (1 (EDITFROMFILE NIL NIL (CL:FIRST PATTERNS)
        EDIT-COMMANDS))
      (CL:OTHERWISE (EDITFROMFILE NIL NIL `(*ANY* ,@PATTERNS)
        EDIT-COMMANDS))))
    (CL:VALUES))
```

```
(DEFCOMMAND "FILES?" NIL "Tell you about what source files need to be dumped."
  (FILES?)
  (CL:VALUES))
```

```
(DEFCOMMAND "IC" (FN) "Inspect the code for the function."
  (INSPECTCODE (IF (CL:SYMBOLP FN)
    THEN (IF (CCODEP (GETD FN))
      THEN FN
      ELSE (CL:COMPILE NIL (GETD FN)))
    ELSE (CL:COMPILE NIL (IF (CL:MEMBER (CAR FN)
      '(CL:LAMBDA LAMBDA)
      :TEST #'EQ)
      THEN FN
      ELSE `(CL:LAMBDA NIL ,FN))))
    (CL:VALUES))
```

```
(DEFCOMMAND "NOTICE" (&REST FILES) "Notice a set of files, so things on them can be edited"
  (CL:FLET ((TELL-USER (FILES MSG)
    (CL:WHEN FILES
      (CL:FORMAT T "~%~A" MSG)
      (CL:MAPCAR #'(CL:LAMBDA (PATH)
        (CL:FORMAT T "~%~5T~A" (CL:PATHNAME-NAME PATH)))
        FILES))))
    (CL:MULTIPLE-VALUE-BIND (JUST-NOTICED PREVIOUSLY-NOTICED NOT-LOADED NOT-FOUND)
      (CL:APPLY #'NOTICE FILES)
      (TELL-USER JUST-NOTICED "Noticed files")
      (TELL-USER PREVIOUSLY-NOTICED "Previously noticed files")
      (TELL-USER NOT-LOADED "Not loaded, so not noticed files")
      (TELL-USER NOT-FOUND "Could not find files"))
    (CL:VALUES))
```

```
(DEFCOMMAND "MAKE" (&REST FILES) "Save, recompile, and reload the files."
  (MAKE FILES)
  (CL:VALUES))
```

```
(DEFCOMMAND "SPY" (FORM) (CL:UNWIND-PROTECT
  (PROGN (SPY.START)
    (PROG1 (CL:EVAL FORM)
      (SPY.END))
    (SPY.END)
    (SPY.TREE)))
```

:: never really worked - (loadmenuitems "ProgrammingAids" (((sysload from lispusers) "Step-Command-Menu")))

::

:: SEDIT stuff

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(EVAL-AT-GREET

:: Give us a META key if running on a Dorado

```
(SELECTQ (MACHINETYPE)
  (DORADO (METASHIFT T))
  NIL)
```

:: Reset SEdit so it gets the correct fonts

```
(CASE MAKESYSNAME
```

```

    (:LYRIC (SEdit.RESET))
    (CL:OTHERWISE (CL:FUNCALL (CL:INTERN "RESET" (CL:FIND-PACKAGE "SEdit")))))
;; Hacking SEdit so it uses better package/readtable defaults
(CASE MAKESYSNAME
  (:LYRIC (FILESLOAD (SYSLOAD NOERROR FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
    LISPUSERSDIRECTORIES))
    "SEdit-HACK"))
  (CL:OTHERWISE (FILESLOAD (SYSLOAD FROM LISPUSERS)
    "seedit-profile"))))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
  LISPUSERSDIRECTORIES))
  "SEdit-Definers"))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
  LISPUSERSDIRECTORIES))
  "SEdit-Eval"))
(CASE MAKESYSNAME
  (:LYRIC (|AddLoadMenuItem| "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
    "SEdit-Compile"))))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "CL-TTYedit"))
)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "New-Where-Is"))
(CASE MAKESYSNAME
  (:LYRIC (|AddLoadMenuItem| "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
    "Debugger-Context"))))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "Checkpoint"))
(RPAQQ? \sPY.BUTTON.POS NIL)
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  SPY)
  ' (SPY.BUTTON \sPY.BUTTON.POS))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "GraphCalls"))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "Manager"))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "PP-Code-File"))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "PrettyFileIndex"))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "TExec"))
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM "{Phylum}<Brotsky>rcw")
  "OSS"))
)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(|AddLoadMenuItem| "ProgrammingAids" (CASE MAKESYSNAME
  (:LYRIC ' ((SYSLOAD FROM LISPUSERS)
    "Port-CLFile"))
  (CL:OTHERWISE ' ((SYSLOAD FROM LISPUSERS)
    "XCL-Bridge"))))
)
(RPAQQ OLD-UTILS-COMMANDS
  ( ;; Used to be in the seperate file UTILS.
    ;; Making sure that breaks happen
    (FNS |DebugMode|)
    ;; Used to be in Loops
    (FNS SELECTW)
    (FUNCTIONS DE FILE)
    (USERMACROS DE EE FV)
    ;; Print out a doc file
    (FNS |PrintDocFile|)

```

```
;; Just what is says
(FNS |\\Pick-One-At-Random|)
(FUNCTIONS |PickOneAtRandom|)

;; Ways to quit a sysout
(FNS |GoodNight| |NewLisp|)

;; Make the KEY3.CM file on the top partition put you back in you last used partition, and the KEY2.CM restart the vmem, assuming that
;; KEY1.CM starts a fresh sysout
(FNS |RememberLastPartition| |RememberLispState|)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (ADDVARS (BEFORELOGOUTFORMS (|RememberLispState|)
(|RememberLastPartition|))))))
```

;; Used to be in the seperate file UTILS.
 ;; Making sure that breaks happen

(DEFINEQ

(|DebugMode|

```
(LAMBDA (DEBUG-ON-P ALL-EXECS-P)
  (|if| DEBUG-ON-P
    |then| (SETQ NLSETQGAG NIL)
           (SETQ HELPFLAG BREAK!)
           (|if| ALL-EXECS-P
             |then| (PUTASSOC 'HELPFLAG (LIST 'BREAK!)
                             *PER-EXEC-VARIABLES*))
           (SETQ NLSETQGAG T)
           (SETQ HELPFLAG T)
           (|if| ALL-EXECS-P
             |then| (PUTASSOC 'HELPFLAG (LIST T)
                             *PER-EXEC-VARIABLES*))))))
```

; Edited 25-Jan-88 08:19 by sml

)

;; Used to be in Loops

(DEFINEQ

(|SELECTW

```
(LAMBDA NIL
  (|first| (CLRSPROMPT)
           (|printout| PROMPTWINDOW "Move mouse to desired window." T "Then press down the CTRL key or click
           mouse")
           |until| (OR (KEYDOWNP 'CTRL)
                      (NOT (MOUSESTATE UP)))
           |do| NIL |finally| (GETMOUSESTATE)
                             (CLRSPROMPT)
                             (RETURN (WHICHW))))))
```

; Edited 15-Jan-88 09:17 by sml

;;; Let the user select a window

```
(|first| (CLRSPROMPT)
  (|printout| PROMPTWINDOW "Move mouse to desired window." T "Then press down the CTRL key or click
  mouse")
  |until| (OR (KEYDOWNP 'CTRL)
             (NOT (MOUSESTATE UP)))
  |do| NIL |finally| (GETMOUSESTATE)
                    (CLRSPROMPT)
                    (RETURN (WHICHW))))
```

)

(DEFMACRO **DE** (|fn-name| |arg-list| &REST |body|)

;;; Shorthand for defining functions

```
`(DEFINEQ (, |fn-name| , |arg-list| , @|body|)))
```

(DEFMACRO **FILE** (|file-name| &REST |file-package-commands|)

;;; Allows one to create a file giving the commands explicitly e.g. - (FILE FOO (VARS * FUMVARS) (FNS * FNSLIST)) - will create FOOCOMS and make
 ;;; file FOO

```
(LET ((|real-file-name| (U-CASE |file-name|)))
  `(PROGN , (|if| (NULL |file-package-commands|)
    |then| NIL
    |elseif| (AND (LITATOM (CAR |file-package-commands|))
                 (NULL (CDR |file-package-commands|)))
    |then| `(/SETATOMVAL (FILECOMS ', |real-file-name|)
                       (GETATOMVAL ', (CAR |file-package-commands|)))
    |else| `(/SETATOMVAL (FILECOMS ', |real-file-name|)
                       ', |file-package-commands|))
    (RESETFORM (RADIX 10)
               (MAKEFILE ', |real-file-name|))))))
```

(ADDTOVAR **USERMACROS**

```
(FV NIL (E (FREEVARS (\## (ORR (UP 1)
                                NIL))
            T)))
(EE NIL (E (CURSOR T)
            T))
```

```

(E (WINDOWPROP TTYINEDITWINDOW 'TITLE (CONCAT "TtyInEdit of " TYPE " " ATM))
  T)
(COMS (TTED DO.EE EE))
(EE (DUMMY)
  (E (CURSOR T)
    T)
  (E (WINDOWPROP TTYINEDITWINDOW 'TITLE (CONCAT "TtyInEdit of " TYPE " " ATM))
    T)
  (COMS (TTED DO.EE EE))
  (DE NIL (COMSQ (BI 1 -1)
    (E (DEDITE (\## 1))
      T)
      (BO 1))))

```

(ADDTOVAR EDITCOMSA DE EE)

(ADDTOVAR EDITCOMSL EE)

:: Print out a doc file

(DEFINEQ

(|PrintDocFile|

(LAMBDA (UTILITY-NAME PRINT-SERVER)

; Edited 17-Mar-88 16:24 by sml

::: Print out the documentation file for the named package

```

(SETQ PRINT-SERVER (OR PRINT-SERVER (CAR DEFAULTPRINTINGHOST)))
(CL:FLET ((FIND-DOC-SOURCE-FILE NIL (OR (FINDFILE (PACKFILENAME 'NAME UTILITY-NAME 'EXTENSION 'TEDIT)
  NIL DIRECTORIES)
  (FINDFILE (PACKFILENAME 'NAME UTILITY-NAME 'EXTENSION 'TED)
  NIL DIRECTORIES)
  (FINDFILE (PACKFILENAME 'NAME UTILITY-NAME 'EXTENSION 'TXT)
  NIL DIRECTORIES)
  (FINDFILE (PACKFILENAME 'NAME UTILITY-NAME 'EXTENSION 'DOC)
  NIL DIRECTORIES))))

```

```

(|if| (EQ PRINT-SERVER T)
  |then| (LET ((DOC-FILE (FIND-DOC-SOURCE-FILE)))
    (|if| DOC-FILE
      |then| (TEDIT DOC-FILE)
      |else| "No doc file found"))
  |elseif| PRINT-SERVER
    |then| (LET ((DOC-FILE (OR (FINDFILE (PACKFILENAME 'NAME UTILITY-NAME 'EXTENSION
      (SELECTQ (PRINTERTYPE PRINT-SERVER)
        ((PRESS FULLPRESS)
          'PRESS)
        (INTERPRESS 'IP)
          (HELP "Unknown printer type!"))
      NIL DIRECTORIES)
      (FIND-DOC-SOURCE-FILE))))
      (|if| DOC-FILE
        |then| (ADD.PROCESS '(EMPRESS ',DOC-FILE NIL ',PRINT-SERVER)
          (CONCAT "Printing file " DOC-FILE " on printer " PRINT-SERVER)
          |else| "No doc file found"))
        |else| "No printer specified"))))

```

)

:: Just what is says

(DEFINEQ

(|Pick-One-At-Random|

(LAMBDA (|list|)

; Edited 15-Jan-88 09:20 by sml

::: Return a random element of the list

```

(RESETLST
  (RESETSAVE (RANDSET T)
    '(RANDSET , (RANDSET)))
  (CAR (NTH |list| (RAND 1 (LENGTH |list|))))))

```

)

```

(DEFMACRO |PickOneAtRandom| (&REST |elements|)
  '(|Pick-One-At-Random| ', (MAPCAR |elements| 'EVAL))

```

:: Ways to quit a sysout

(DEFINEQ

(|GoodNight|

```

(LAMBDA (|flag| |altoCommandString|)
  (LET ((|stream| (OPENSTREAM '{DSK}REM.CM\;1 'OUTPUT 'OLD/NEW))
    (PRIN1 (OR |altoCommandString| "Q")
      (* |sml| "20-Sep-85 14:43"))

```

```

      (stream|)
    (TERPRI |stream|)
    (CLOSEF |stream|)
  (LOGOUT |flag|)))

```

(|NewLisp|

(LAMBDA NIL

; Edited 15-Jan-88 09:20 by sml

:: Start up a new system, assuming that {DSK}KEY1.CM starts one up.

```

(|if| (MOUSECONFIRM "Do you really want to start up a new system?"
  |then| (|GoodNight| T "@KEY1.CM"))))

```

)

:: Make the KEY3.CM file on the top partition put you back in you last used partition, and the KEY2.CM restart the vmem, assuming that KEY1.CM starts a fresh sysout

(DEFINEQ

(|RememberLastPartition|

(LAMBDA NIL

; Edited 15-Jan-88 09:21 by sml

:: Sets up the KEY3 CM file in the last partition (19 or 5) to put you back in this partition.

```

(SELECTQ (MACHINETYPE)
  (DORADO (|for| |partitionNumber| |in| ' (19 5) |bind| |stream| |key3File|
    |eachtime| (SETQ |key3File| (CONCAT "{DSK}" |partitionNumber| "KEY3.CM;1"))
      (SETQ |stream| (CAR (NLSETQ (GETSTREAM |key3File|))))
      (AND |stream| (CLOSEF? |stream|))
      (SETQ |stream| (CAR (NLSETQ (OPENSTREAM |key3File| 'OUTPUT 'OLD/NEW)))))
    |thereis| (STREAMP |stream|) |finally| (|if| (AND (STREAMP |stream|)
      (OPENP |stream|))
      |then| (|printout| |stream| "// " "This will set you back
        in your last used partition, " FIRSTNAME T
        "// [last used " (DATE)
        "]" T "Par " (DISKPARTITION)
        T)
      (CLOSEF |stream|))))
  NIL))

```

(|RememberLispState|

(LAMBDA NIL

; Edited 15-Jan-88 09:21 by sml

:: Make KEY2.CM restart this lisp if the logout was not FAST...

```

(NLSETQ (|if| (AND (STKPOS 'LOGOUT)
  (EQ (MACHINETYPE)
    'DORADO))
  |then| (|if| (NLSETQ (GETSTREAM '{DSK}KEY2.CM\;1))
    |then| (CLOSEF? (GETSTREAM '{DSK}KEY2.CM\;1)))
  (RESETLST
    (LET ((LOGOUT-ARG (STKARG 1 'LOGOUT))
      (STREAM (OPENSTREAM '{DSK}KEY2.CM\;1 'OUTPUT 'OLD/NEW)))
      (RESETSAVE NIL (LIST (FUNCTION CLOSEF?)
        STREAM))
      (|printout| STREAM "// You did a (LOGOUT")
      (SELECTQ LOGOUT-ARG
        (NIL NIL)
        (|printout| STREAM " " LOGOUT-ARG))
      (|printout| STREAM ") last time [" (DATE)
        "], so this will ")
      (SELECTQ LOGOUT-ARG
        (NIL ?)
        (|printout| STREAM "restart your old"))
        (|printout| STREAM "start a new"))
      (|printout| STREAM " LISP, " FIRSTNAME T)
      (SELECTQ LOGOUT-ARG
        (NIL ?)
        (|printout| STREAM "Lisp")
        (|if| (EQP (REALMEMORYSIZE)
          21845)
          |then| (|printout| STREAM " 52525/c"))
        (|printout| STREAM "@KEY1.CM"))
      (|printout| STREAM T))))))

```

)

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(ADDTOVAR BEFORELOGOUTFORMS (|RememberLispState|)
(|RememberLastPartition|))

)

(RPAQQ WIZARD-INIT-COMMANDS (

;;; Some tools for wizards, or people who occasionally think they are.

;; Find out what other symbols were interned at about the same time as a given symbol. Useful to find out what file defined a symbol.
(FUNCTIONS ATOM-NEIGHBORS))

;;; Some tools for wizards, or people who occasionally think they are.

;; Find out what other symbols were interned at about the same time as a given symbol. Useful to find out what file defined a symbol.

```
(CL:DEFUN ATOM-NEIGHBORS (CL:SYMBOL &OPTIONAL (XCL-USER::NUMBER-OF-NEIGHBORS 8))
  (CL:IF (CL:SYMBOLP CL:SYMBOL)
    (LET ((XCL-USER::ATOM-NUMBER (\\LOLOC CL:SYMBOL))
          (XCL-USER::NEIGHBORS (LIST CL:SYMBOL)))
      (CL:DOTIMES (XCL-USER::I XCL-USER::NUMBER-OF-NEIGHBORS)
        (CL:PUSH (\\VAG2 0 (+ XCL-USER::ATOM-NUMBER 1 XCL-USER::I))
          XCL-USER::NEIGHBORS))
      (CL:SETF XCL-USER::NEIGHBORS (CL:NREVERSE XCL-USER::NEIGHBORS))
      (CL:DOTIMES (XCL-USER::I XCL-USER::NUMBER-OF-NEIGHBORS)
        (CL:PUSH (\\VAG2 0 (- XCL-USER::ATOM-NUMBER 1 XCL-USER::I))
          XCL-USER::NEIGHBORS))
      XCL-USER::NEIGHBORS)
    "Not a symbol"))
```

(RPAQQ **DINFO-INIT-COMMANDS**

;; Set up the on-line documentation

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (VARS (IRM.HOST&DIR (COND ((INFILEP
                                                                    "{DSK}<LISPFILES>HELPSYS>IRM
TOP.TEDIT"
                                                                    "{DSK}<LISPFILES>HELPSYS>")
((AND (BOUNDP 'IRM.HOST&DIR)
      IRM.HOST&DIR)
      IRM.HOST&DIR)
(T
  "{Pallas:AISNorth:XEROX}<Lisp>Lyr
ic>LispUsers>IRM>"))))
```

```
(DINFOMODES ' (GRAPH))
(INITVARS (IRM.FONT (FONTCREATE ' (HELVETICA 10)))
  (IRMWINDOWREGION (LET ((WIDTH (WIDTHIFWINDOW (CL:* 60 (STRINGWIDTH "w" IRM.FONT))))
    (CREATEREGION (- SCREENWIDTH WIDTH 5)
      5 WIDTH (QUOTIENT SCREENHEIGHT 2))))))
(LOADMENUITEMS "ProgrammingAids" ((SYSLOAD FROM LISPUSERS)
  "DInfo" "Helpsys")
  (DINFO (IRM.GET.DINFOGRAPH T)
    IRMWINDOWREGION))
(((SYSLOAD FROM LISPUSERS)
  "LispNerd"))))
```

;; Set up the on-line documentation

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(RPAQ IRM.HOST&DIR (COND ((INFILEP "{DSK}<LISPFILES>HELPSYS>IRMTOP.TEDIT"
  "{DSK}<LISPFILES>HELPSYS>")
  ((AND (BOUNDP 'IRM.HOST&DIR)
        IRM.HOST&DIR)
        IRM.HOST&DIR)
  (T "{Pallas:AISNorth:XEROX}<Lisp>Lyric>LispUsers>IRM>"))))
```

(RPAQQ **DINFOMODES** (GRAPH))

(RPAQ? **IRM.FONT** (FONTCREATE ' (HELVETICA 10)))

(RPAQ? **IRMWINDOWREGION** (LET ((WIDTH (WIDTHIFWINDOW (CL:* 60 (STRINGWIDTH "w" IRM.FONT))))
 (CREATEREGION (- SCREENWIDTH WIDTH 5)
 5 WIDTH (QUOTIENT SCREENHEIGHT 2))))))

```
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "DInfo" "Helpsys")
  ' (DINFO (IRM.GET.DINFOGRAPH T)
    IRMWINDOWREGION))
```

```
(|AddLoadMenuItem| ' "ProgrammingAids" ' ((SYSLOAD FROM LISPUSERS)
  "LispNerd"))
)
```

(RPAQQ **PCL-INIT-COMMANDS**

;; PCL fun and games

```
(INITVARS (PCLDIRECTORIES (CASE MAKESYSNAME
  (:LYRIC ` (,@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<PCL>LYRIC">
    "{PHYLUM}<PCL>LYRIC">))
```

```

(CL:OTHERWISE `(@*CACHE-DIRECTORIES* "{pooh/n}<pooh>pcl>medley>"
               "{NB:PARC:XEROX}<PCL>MEDLEY>"
               "{PHYLUM}<PCL>MEDLEY>"))))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
  (P (EVAL-AT-GREET (CL:UNLESS (CL:FIND-PACKAGE "PCL")
    (|AddLoadMenuItem| "Subsystems" ' ((FROM PCL)
                                     "Load-PCL")
    ' (CL:WHEN (EQ MAKESYSNAME :LYRIC)
              (|AddLoadMenuItem| "ProgrammingAids" ' ((FROM PCL)
                                                         "Clos-Browser"))))))))
)

```

;; PCL fun and games

```

(RPAQ? PCLDIRECTORIES (CASE MAKESYSNAME
  (:LYRIC `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<PCL>LYRIC>" "{PHYLUM}<PCL>LYRIC>"))
  (CL:OTHERWISE `(@*CACHE-DIRECTORIES* "{pooh/n}<pooh>pcl>medley>"
                                           "{NB:PARC:XEROX}<PCL>MEDLEY>" "{PHYLUM}<PCL>MEDLEY>"))))
)

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

(EVAL-AT-GREET (CL:UNLESS (CL:FIND-PACKAGE "PCL")
  (|AddLoadMenuItem| "Subsystems" ' ((FROM PCL)
                                     "Load-PCL")
  ' (CL:WHEN (EQ MAKESYSNAME :LYRIC)
            (|AddLoadMenuItem| "ProgrammingAids" ' ((FROM PCL)
                                                       "Clos-Browser"))))))
)

```

(RPAQ? LOOPS-INIT-COMMANDS

```

(
  ;; Loops initialization
  (INITVARS (LOOPSDIRECTORIES (CASE MAKESYSNAME
    (:LYRIC `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Loops>Lyric>Sources>"
                                     "{PHYLUM}<Loops>Lyric>Sources>"
                                     "{POGO:AINorth:XEROX}<LOOPSCORE>SYSTEM>"))
    (CL:OTHERWISE `(@*CACHE-DIRECTORIES*
                    "{NB:PARC:XEROX}<Loops>Medley>Sources>"
                    "{PHYLUM}<Loops>Medley>Sources>"
                    "{POGO:AINorth:XEROX}<LOOPSCORE>SYSTEM>"))))
    (\\LOOPS-INIT-FORM ' (PROGN (FILESLOAD (FROM "{EG:PARC:XEROX}<Lanning>Loops>"
                                             INITLOOPS))))
  (DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (P (EVAL-AT-GREET (COND ((CL:MEMBER "LOOPS" *FEATURES* :TEST
    #'STRING-EQUAL)
    (EVAL \\LOOPS-INIT-FORM))
    (T (|AddLoadMenuItem|
        "Subsystems"
        ' ((FROM LOOPS)
           "Loops")
        \\LOOPS-INIT-FORM))))))
)

```

;; Loops initialization

```

(RPAQ? LOOPSDIRECTORIES (CASE MAKESYSNAME
  (:LYRIC `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Loops>Lyric>Sources>"
                                     "{PHYLUM}<Loops>Lyric>Sources>"
                                     "{POGO:AINorth:XEROX}<LOOPSCORE>SYSTEM>"))
  (CL:OTHERWISE `(@*CACHE-DIRECTORIES* "{NB:PARC:XEROX}<Loops>Medley>Sources>"
                                           "{PHYLUM}<Loops>Medley>Sources>"
                                           "{POGO:AINorth:XEROX}<LOOPSCORE>SYSTEM>"))))
)

```

```

(RPAQ? \\LOOPS-INIT-FORM ' (PROGN (FILESLOAD (FROM "{EG:PARC:XEROX}<Lanning>Loops>"
                                                 INITLOOPS)))
)

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

(EVAL-AT-GREET (COND
  ((CL:MEMBER "LOOPS" *FEATURES* :TEST #'STRING-EQUAL)
  (EVAL \\LOOPS-INIT-FORM))
  (T (|AddLoadMenuItem| "Subsystems" ' ((FROM LOOPS)
                                         "Loops")
    \\LOOPS-INIT-FORM))))
)

```

;; Documentation

(RPAQ? TEDIT-INIT-COMMANDS

```

(
  ;; TEDIT stuff
  (FUNCTIONS LOAD-NOVA-FONTS)
  (DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
)

```



```
(VARS (TEDIT.DEFAULT.WINDOW (CREATEW (LET ((WIDTH (WIDTHIFWINDOW (TIMES 80 (TEDIT.STRINGWIDTH
" A")))))
(HEIGHT (HEIGHTIFWINDOW (TIMES 35 (FONTPROP
DEFAULTFONT
'HEIGHT)))
T)))
(CREATEREGION (MIN 475 (- SCREENWIDTH (+ WIDTH 10)))
(MAX 85 (- SCREENHEIGHT HEIGHT 100))
WIDTH HEIGHT))
"TEDIT window" NIL T))
; Set up default tabs to be 8 spaces, so we can edit code files.
(TEDIT.DEFAULT.PROPS `(@TEDIT.DEFAULT.PROPS PARALOOKS (TABS (, (TIMES 8 (TEDIT.STRINGWIDTH
" A"))))))))
(P (EVAL-AT-GREET (CL:WHEN (GETD 'TEDIT)
;; Make the ESC key REDO the previous TEdit operation
(UNDOSAVE `(TEDIT.SETSYNTAX ,(CHARCODE ESC)
,(TEDIT.GETSYNTAX (CHARCODE ESC))))
(TEDIT.SETSYNTAX (CHARCODE ESC)
'REDO)
;; Make TEdit close files when shrunk
(FILESLoad (SYSLOAD FROM LISPUSERS)
TEDIT-CLOSE-ON-SHRINK)))
(LOADMENUITEMS "WritingAids" ((SYSLOAD FROM LISPUSERS)
"ProofReader"))
((SYSLOAD FROM LISPUSERS)
"TMAX"))
((SYSLOAD FROM LISPUSERS)
"DictTool"))
((SYSLOAD FROM LISPUSERS)
"TEditDoradoKeys"))
((SYSLOAD FROM LISPUSERS)
"EditKeys"))
((SYSLOAD FROM LISPUSERS)
"VirtualKeyboards")
(LET ((KEYBOARD-FILE (FINDFILE "Code.Keyboard" T PRIVATE-LISPUSERSDIRECTORIES)))
(CL:WHEN KEYBOARD-FILE (VKBD.LOAD-KEYBOARD-FILE KEYBOARD-FILE)))
((SYSLOAD FROM LISPUSERS)
"KeyboardEditor"))
((SYSLOAD FROM LISPUSERS)
"Equations" "Sketch"))
((SYSLOAD FROM LISPUSERS)
"NovaFont")
(LOAD-NOVA-FONTS)))
(COMS (INITVARS (DOBJECTSMENUFONT MENUFONT))
(ALISTS (IMAGEOBJGETFNS DOCOBJ-FILESTAMP-GETFN DOCOBJ-TIMESTAMP-GETFN DOCOBJ-INCLUDE-GETFN)
)
(LOADMENUITEMS "WritingAids" ((SYSLOAD FROM LISPUSERS)
"Doc-Objects"))))
```

;;; TEDIT stuff

```
(CL:DEFUN LOAD-NOVA-FONTS ()
(LET ((NOVA-FONT-HOST "Starfile Public:Parc:Xerox")
(NOVA-FONTS-TO-LOAD '("<VP Applications>VP Optima XSG Fonts>OptimaItalic" "<VP Applications>VP Optima
XSG Fonts>OptimaMedium"))
(NOVA-FONTS-TO-NOTICE '("<VP Applications>Xerox Logo Fonts>XeroxLogo" "<VP Applications>Xerox VP Quartz
Fonts!2>QuartzBIR" "<VP Applications>Xerox VP Quartz Fonts!2>QuartzBRR"
"<VP Applications>Xerox VP Quartz Fonts!2>QuartzMIR" "<VP
Applications>Xerox VP Quartz Fonts!2>QuartzMRR")))
(CL:FLET ((FIND-NOVA-FONT (FONT)
"Find the Novafont file"
(CL:PROBE-FILE (CL:MAKE-PATHNAME :HOST NOVA-FONT-HOST :TYPE "NovaFont" :DEFAULTS FONT)
))
(CL:MAPC #'(CL:LAMBDA (FONT)
(LET ((FONT-FILE (FIND-NOVA-FONT FONT))
(CL:WHEN FONT-FILE
(CL:FORMAT (GETSTREAM PROMPTWINDOW 'OUTPUT)
"~%Loading Novafont ~A"
(CL:PATHNAME-NAME FONT-FILE))
(LOAD-NOVAFONT-FILE FONT-FILE)
(NOTICE-NOVAFONT-FILE FONT-FILE))))
NOVA-FONTS-TO-LOAD)
(CL:MAPC #'(CL:LAMBDA (FONT)
(LET ((FONT-FILE (FIND-NOVA-FONT FONT))
(CL:WHEN FONT-FILE
(CL:FORMAT (GETSTREAM PROMPTWINDOW 'OUTPUT)
"~%Noticing Novafont ~A"
(CL:PATHNAME-NAME FONT-FILE))
(NOTICE-NOVAFONT-FILE FONT-FILE))))
NOVA-FONTS-TO-NOTICE)))
(CL:MAPC #'(CL:LAMBDA (ITEM)
(CL:PUSHNEW ITEM TEDIT.KNOWN.FONTS :TEST 'CL:EQUAL))
'(("XeroxLogo" 'XEROXLOGO)
```

```
("Quartz" 'QUARTZ)
("Optima" 'OPTIMA)))
```

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
```

```
(RPAQ TEDIT.DEFAULT.WINDOW
(CREATEW (LET ((WIDTH (WIDTHIFWINDOW (TIMES 80 (TEDIT.STRINGWIDTH "A"))))
              (HEIGHT (HEIGHTIFWINDOW (TIMES 35 (FONTPROP DEFAULTFONT 'HEIGHT)
              T))))
          (CREATEREGION (MIN 475 (- SCREENWIDTH (+ WIDTH 10))
                          (MAX 85 (- SCREENHEIGHT HEIGHT 100))
                          WIDTH HEIGHT)
            "TEDIT window" NIL T))
```

```
(RPAQ TEDIT.DEFAULT.PROPS `(@TEDIT.DEFAULT.PROPS PARALOOKS (TABS (, (TIMES 8 (TEDIT.STRINGWIDTH "A"))))))
```

```
(EVAL-AT-GREET (CL:WHEN (GETD 'TEDIT)
  ;; Make the ESC key REDO the previous TEdit operation
  (UNDOSAVE `(TEDIT.SETSYNTAX ,(CHARCODE ESC)
                ,(TEDIT.GETSYNTAX (CHARCODE ESC))))
  (TEDIT.SETSYNTAX (CHARCODE ESC)
    'REDO)
  ;; Make TEdit close files when shrunk
  (FILESLOAD (SYSLOAD FROM LISPUSERS)
    TEDIT-CLOSE-ON-SHRINK)))
```

```
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "ProofReader"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "TMAX"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "DictTool"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "TEditDoradoKeys"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "EditKeys"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "VirtualKeyboards")
  '(LET ((KEYBOARD-FILE (FINDFILE "Code.Keyboard" T PRIVATE-LISPUSERSDIRECTORIES)))
    (CL:WHEN KEYBOARD-FILE (VKBD.LOAD-KEYBOARD-FILE KEYBOARD-FILE))))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "KeyboardEditor"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "Equations" "Sketch"))
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "NovaFont")
  '(LOAD-NOVA-FONTS))
```

```
(RPAQ? DOBJECTSMENUFONT MENUFONT)
```

```
(ADDTOVAR IMAGEOBJGETFNS (DOCOBJ-FILESTAMP-GETFN FILE DOC-OBJECTS)
  (DOCOBJ-TIMESTAMP-GETFN FILE DOC-OBJECTS)
  (DOCOBJ-INCLUDE-GETFN FILE DOC-OBJECTS))
```

```
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "Doc-Objects"))
)
```

```
(RPAQ SKETCH-INIT-COMMANDS (
```

;;; SKETCH Stuff

```
(ALISTS (IMAGEOBJGETFNS SKIO.GETFN SKIO.GETFN.2))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "WritingAids"
  (((SYSLOAD FROM LISPUSERS)
    "Sketch")))))
```

;;; SKETCH Stuff

```
(ADDTOVAR IMAGEOBJGETFNS (SKIO.GETFN)
  (SKIO.GETFN.2 FILE SKETCH))
```

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
```

```
(|AddLoadMenuItem| ' "WritingAids" ' ((SYSLOAD FROM LISPUSERS)
  "Sketch"))
```

)

(RPAQQ **NOTECARDS-INIT-COMMANDS**

(

;;; NOTECARDS stuff

```
(INITVARS (|NC.NoteCardsIconPosition| (CREATEPOSITION 891 2))
(NCINITIALGLOBALPARAMS '(|ForceFiling| NIL |MarkersInFileBoxes| NIL
|UseDeletedLinkIconIndicators| NIL))
(NOTECARDSDIRECTORIES `(|,*CACHE-DIRECTORIES* "{QV}<NoteCards>1.3L>"
"NB:PARC:XEROX}<NoteCards>1.3L>"))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "WritingAids" (((SYSLOAD FROM NOTECARDS)
"NoteCards")
(|NoteCards|
|NC.NoteCardsIconPosition|
))))))
```

;;; NOTECARDS stuff

```
(RPAQQ? |NC.NoteCardsIconPosition| (CREATEPOSITION 891 2))
(RPAQQ? NCINITIALGLOBALPARAMS '(|ForceFiling| NIL |MarkersInFileBoxes| NIL |UseDeletedLinkIconIndicators| NIL))
(RPAQQ? NOTECARDSDIRECTORIES `(|,*CACHE-DIRECTORIES* "{QV}<NoteCards>1.3L>" "{NB:PARC:XEROX}<NoteCards>1.3L>"))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(|AddLoadMenuItem| "WritingAids" '(((SYSLOAD FROM NOTECARDS)
"NoteCards")
'(|NoteCards| |NC.NoteCardsIconPosition|))
)
```

;; Communication & Info

(RPAQQ **MAIL-INIT-COMMANDS**

(

;;; LAFITE stuff

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
;; These have to be VARS instead of INITVARS since they come set to default values in the FULL sysout.
(VARS (*NEW-LAFITE-P* (NOT (CL:FBOUND P '\\LAFITE.GETMAILFOLDER)))
(DEFAULTMAILFOLDERNAME 'ACTIVE.MAIL)
(LAFITEDEFAULTHOST&DIR (PACK* '\\UserHomeDirectory| "MAIL>"))
(LAFITEHARDCOPYBATCHFLG T)
(LAFITEMOVETOCONFIRMFLG 'LEFT)
(LAFITESHOWMODEFLG 'ALWAYS)
(LAFITEBROWSERREGION (CREATEREGION 360 5 650 165))
(LAFITEDISPLAYREGION (|with| REGION (WINDOWPROP '\\TopLevelTtyWindow| 'REGION)
(CREATEREGION (MAX LEFT SCROLLBARWIDTH)
175
(WIDTHIFWINDOW (MIN (QUOTIENT (TIMES 2 SCREENWIDTH)
3)
(TIMES 85 (STRINGWIDTH "W"
LAFITEDISPLAYFONT))
))
(DIFFERENCE BOTTOM 175))))
(LAFITESTATUSWINDOWPOSITION (CREATEPOSITION 100 45))
(LAFITEMODEDEFAULT (OR LAFITEMODEDEFAULT 'GV)))
```

;; In latest Lafite. Of course, it doesn't hurt to set them even if they aren't used.

```
(VARS (LAFITE.DONT.DISPLAY.HEADERS '("Return-Path" "Redistributed" "Received" "Message-Id"
"Comments" "Zippy-Says" "X-Mailer" "Organization"
"References"))
(LAFITE.DONT.FORWARD.HEADERS LAFITE.DONT.DISPLAY.HEADERS)
(LAFITE.DONT.HARDCOPY.HEADERS LAFITE.DONT.DISPLAY.HEADERS))
```

;; There are lots of optional mail utilities

```
(LOADMENUITEMS "MailTools" (((SYSLOAD FROM LISPUSERS)
"LafiteTimedDelete"))
(((SYSLOAD FROM LISPUSERS)
"LafiteFind"))
(((SYSLOAD FROM LISPUSERS)
"Maintain"))
(((SYSLOAD FROM LISPUSERS)
"NSMaintain"))
(((SYSLOAD FROM LISPUSERS)
"MailScavenge"))
(((SYSLOAD FROM LISPUSERS)
"Undigestify"))
((FROM LISPUSERS)
"Lafite-Indent"))
(((SYSLOAD FROM LISPUSERS)
"MailShare"))
```

```

((SYSLOAD FROM "{QV}<Briggs>Lisp>"
 "LafiteFolderIcon"))
((SYSLOAD FROM "{ERIS}<Lafite>Sources>"
 "AppendMail"))
(P (EVAL-AT-GREET (CL:UNLESS (NOT *NEW-LAFITE-P*)
 (|AddLoadMenuItem| "MailTools" ' ((SYSLOAD FROM LISPUSERS)
 "LafiteHighlight"))
 (|AddLoadMenuItem| "MailTools" ' ((SYSLOAD FROM "{PHYLUM}<Bobrow>Lisp>"
 "Short-Lafite-Header"))))))
;;
;; Additional text at the start and end of a msg
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
 (P (EVAL-AT-GREET (IF *NEW-LAFITE-P* THEN
 ;; New Lafite running, set the appropriate vars.
 (RPAQ LAFITE.SIGNATURE (OR LAFITE.SIGNATURE
 (CONCAT (CHARACTER (CHARCODE EOL))
 (CHARACTER (CHARCODE EOL))
 "----- "
 (OR (CADDR (CAR INITIALSLST))
 "me")
 (CHARACTER (CHARCODE EOL)))))
 (RPAQ LAFITE.GV.FROM.FIELD
 (OR LAFITE.GV.FROM.FIELD (CONCAT (OR (CADR (CAR INITIALSLST))
 " ")
 " "
 (USERNAME NIL NIL T)
 " <"
 (COND ((CL:FBOUNDP 'FULLUSERNAME)
 (FULLUSERNAME))
 (T (USERNAME)))
 ">"))))
 ELSE
 ;; Use the hack I wrote
 (RPAQ? LAFITEMSGTAGSTRING (CONCAT (CHARACTER (CHARCODE EOL))
 (CHARACTER (CHARCODE EOL))
 "----- "
 (OR (CADDR (CAR INITIALSLST))
 "me")
 (CHARACTER (CHARCODE EOL))))
 (RPAQ? LAFITEMSGHEADERSTRING
 (CONCAT "From: " (OR (CADR (CAR INITIALSLST))
 " ")
 " "
 (USERNAME NIL NIL T)
 " <"
 (COND ((CL:FBOUNDP 'FULLUSERNAME)
 (FULLUSERNAME))
 (T (USERNAME)))
 ">"
 (CHARACTER (CHARCODE EOL))))
 (FILESLOAD (NOERROR FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
 LISPUSERSDIRECTORIES)
 "Tailor-Lafite-Msg")))))
;; Private DL support
(INITVARS (LAFITEDLDIRECTORIES NIL))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (P (EVAL-AT-GREET (CL:UNLESS *NEW-LAFITE-P*
 (|AddLoadMenuItem| "MailTools"
 ' ((SYSLOAD FROM LISPUSERS)
 "LafitePrivateDL"))))))
;; Patches to Lafite to put the MOVE-TO folder first in the title.
(P (EVAL-AT-GREET (CL:WHEN (AND (CL:FBOUNDP 'LAFITE)
 (NOT *NEW-LAFITE-P*))
 (FILESLOAD (SYSLOAD NOERROR FROM "{Phylum}<Bobrow>Lisp>"
 "LafiteTitlePatch")
 (CHANGENAME 'LA.OPENTEMPFILE 'DSK 'SCRATCH)
 (LAFITEMODE 'GV))))
 (COMS
;;; COMMON-LENS stuff
(INITVARS (\\USE-LENS? NIL)
 (USER::*LENS-USER-PARAMETERS* `((:DEFAULT-HOST-AND-DIR LAFITEDEFAULTHOST&DIR)
 (:ICON-POS (CREATEPOSITION 98 8))
 (:MAILWEB-POS (CREATEPOSITION 194 7))
 (:MAILWEB-ICON-POS (CREATEPOSITION 194 7))
 (:MAILWEB-FONT (FONTCREATE ' (HELVETICA 8)))
 (:FEEDBACK-P NIL)
 (:FILING-RULE "Filing Rules")
 (:ROOM-NAME ' ("Mail")))))
 (VARS (USER::LENS-LOADER-DIR "{NB:PARC:Xerox}<CommonLens>Current>"))
 (P (EVAL-AT-GREET (COND ((CL:MEMBER "Common-Lens" *MODULES* :TEST #'STRING-EQUAL)
 ; Common-Lens already loaded

```

```

      (CL:FUNCALL (CL:INTERN "LOAD-LENS-PATCHES" "CLENS"))
      ((AND \\USE-LENS? (NOT (NULL (CL:FIND-PACKAGE "PCL"))))
       ; Want to use Common-Lens, so load it now
      (CL:LOAD (CL:MAKE-PATHNAME :NAME "LOAD-CLENS" :DEFAULTS
                                USER::LENS-LOADER-DIR))
      (T
       ; Give the user the option of using it later
       (|AddLoadMenuItem| "MailTools" '( (FROM VALUEOF USER::LENS-LOADER-DIR)
                                         "LOAD-CLENS")
        ' (IGNORE-ERRORS (LAFITE 'OFF)
          (CL:FUNCALL (CL:INTERN "COMMON-LENS" (CL:FIND-PACKAGE
                                                "CLENS"))
                    :ON))))))
;; Turn on mailer (no, no, not Norman -- but he is easy to turn on. Or so I've heard.)
(INITVARS (\\TURN-ON-MAILER (EQUAL (ETHERHOSTNUMBER HOME-MACHINE-NAME)
                                  (ETHERHOSTNUMBER))))
(P (EVAL-AT-GREET (COND ((NOT (GETD 'LAFITE))
                          ; Don't turn on mail if it doesn't exist
                          NIL)
                        ((NOT \\TURN-ON-MAILER)
                          ; Don't turn on if the user doesn't want to
                          NIL)
                        ((AND \\USE-LENS? (NOT (NULL (CL:FIND-PACKAGE "CLENS"))))
                          ; Use COMMON-LENS if it's here and the user wants it
                          (CL:FUNCALL (CL:INTERN "COMMON-LENS" (CL:FIND-PACKAGE "CLENS"))
                                      :ON))
                        (T
                         ; Use Lafite
                         (LAFITE 'ON))))))

```

;;; LAFITE stuff

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(RPAQ *NEW-LAFITE-P* (NOT (CL:FBOUNDP '\\LAFITE.GETMAILFOLDER)))
(RPAQQ DEFAULTMAILFOLDERNAME ACTIVE.MAIL)
(RPAQ LAFITEDEFAULTHOST&DIR (PACK* |\\UserHomeDirectory| "MAIL>"))
(RPAQQ LAFITEHARDCOPYBATCHFLG T)
(RPAQQ LAFITEMOVETOCONFIRMFLG LEFT)
(RPAQQ LAFITESHOWMODEFLG ALWAYS)
(RPAQ LAFITEBROWSERREGION (CREATEREGION 360 5 650 165))
(RPAQ LAFITEDISPLAYREGION
  (|with| REGION (WINDOWPROP |\\TopLevelTTYWindow| 'REGION)
    (CREATEREGION (MAX LEFT SCROLLBARWIDTH)
                  175
                  (WIDTHIFWINDOW (MIN (QUOTIENT (TIMES 2 SCREENWIDTH)
                                                3)
                                       (TIMES 85 (STRINGWIDTH "w" LAFITEDISPLAYFONT))))
                  (DIFFERENCE BOTTOM 175))))
(RPAQ LAFITESTATUSWINDOWPOSITION (CREATEPOSITION 100 45))
(RPAQ LAFITEMODEDEFAULT (OR LAFITEMODEDEFAULT 'GV))
(RPAQQ LAFITE.DONT.DISPLAY.HEADERS ("Return-Path" "Redistributed" "Received" "Message-Id" "Comments"
                                     "Zippy-Says" "X-Mailer" "Organization" "References"))
(RPAQ LAFITE.DONT.FORWARD.HEADERS LAFITE.DONT.DISPLAY.HEADERS)
(RPAQ LAFITE.DONT.HARDCOPY.HEADERS LAFITE.DONT.DISPLAY.HEADERS)
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
                                     "LafiteTimedDelete"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
                                     "LafiteFind"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
                                     "Maintain"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
                                     "NSMaintain"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
                                     "MailScavenge"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
                                     "Undigestify"))
(|AddLoadMenuItem| ' "MailTools" ' ((FROM LISPUSERS)

```

```

        "Lafite-Indent"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM LISPUSERS)
    "MailShare"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM "{QV}<Briggs>Lisp>"
    "LafiteFolderIcon"))
(|AddLoadMenuItem| ' "MailTools" ' ((SYSLOAD FROM "{ERIS}<Lafite>Sources>"
    "AppendMail"))
(EVAL-AT-GREET (CL:UNLESS (NOT *NEW-LAFITE-P*)
    (|AddLoadMenuItem| "MailTools" ' ((SYSLOAD FROM LISPUSERS)
        "LafiteHighlight"))
    (|AddLoadMenuItem| "MailTools" ' ((SYSLOAD FROM "{PHYLUM}<Bobrow>Lisp>"
        "Short-Lafite-Header"))))
)

```

;;
 ;; Additional text at the start and end of a msg

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

(EVAL-AT-GREET (IF *NEW-LAFITE-P*
  THEN ;; New Lafite running, set the appropriate vars.
    (RPAQ LAFITE.SIGNATURE (OR LAFITE.SIGNATURE (CONCAT (CHARACTER (CHARCODE EOL))
        (CHARACTER (CHARCODE EOL))
        "----- "
        (OR (CADDR (CAR INITIALSLST))
            "me")
        (CHARACTER (CHARCODE EOL)))))
    (RPAQ LAFITE.GV.FROM.FIELD
        (OR LAFITE.GV.FROM.FIELD (CONCAT (OR (CADDR (CAR INITIALSLST))
            " "
            (USERNAME NIL NIL T)
            " <"
            (COND ((CL:FBOUNDP 'FULLUSERNAME)
                (FULLUSERNAME))
                (T (USERNAME))))
            ">"))))
  ELSE ;; Use the hack I wrote
    (RPAQ? LAFITEMSGTAGSTRING (CONCAT (CHARACTER (CHARCODE EOL))
        (CHARACTER (CHARCODE EOL))
        "----- "
        (OR (CADDR (CAR INITIALSLST))
            "me")
        (CHARACTER (CHARCODE EOL)))))
    (RPAQ? LAFITEMSGHEADERSTRING
        (CONCAT "From: " (OR (CADDR (CAR INITIALSLST))
            " ")
            " "
            (USERNAME NIL NIL T)
            " <"
            (COND ((CL:FBOUNDP 'FULLUSERNAME)
                (FULLUSERNAME))
                (T (USERNAME))))
            ">"
            (CHARACTER (CHARCODE EOL))))
    (FILESLOAD (NOERROR FROM VALUEOF (APPEND PRIVATE-LISPUSERSDIRECTORIES
        LISPUSERSDIRECTORIES))
        "Tailor-Lafite-Msg")))
)

```

;; Private DL support

```

(RPAQ? LAFITEDLDIRECTORIES NIL)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(EVAL-AT-GREET (CL:UNLESS *NEW-LAFITE-P*
    (|AddLoadMenuItem| "MailTools" ' ((SYSLOAD FROM LISPUSERS)
        "LafitePrivateDL"))))
)

```

;; Patches to Lafite to put the MOVE-TO folder first in the title.

```

(EVAL-AT-GREET (CL:WHEN (AND (CL:FBOUNDP 'LAFITE)
    (NOT *NEW-LAFITE-P*))
    (FILESLOAD (SYSLOAD NOERROR FROM "{Phylum}<Bobrow>Lisp>"
        "LafiteTitlePatch")
    (CHANGENAME 'LA.OPENTEMPFILE 'DSK 'SCRATCH)
    (LAFITEMODE 'GV)))
)

```

;;; COMMON-LENS stuff

(RPAQ? \USE-LENS? NIL)

(RPAQ? USER::*LENS-USER-PARAMETERS*
`((:DEFAULT-HOST-AND-DIR LAFITEDEFAULTHOST&DIR)
(:ICON-POS (CREATEPOSITION 98 8))
(:MAILWEB-POS (CREATEPOSITION 194 7))
(:MAILWEB-ICON-POS (CREATEPOSITION 194 7))
(:MAILWEB-FONT (FONTCREATE '(HELVETICA 8)))
(:FEEDBACK-P NIL)
(:FILING-RULE "Filing Rules")
(:ROOM-NAME '("Mail"))))

(RPAQ USER::LENS-LOADER-DIR "{NB:PARC:Xerox}<CommonLens>Current>")

(EVAL-AT-GREET (COND
((CL:MEMBER "Common-Lens" *MODULES* :TEST #'STRING-EQUAL)
; Common-Lens already loaded
(CL:FUNCALL (CL:INTERN "LOAD-LENS-PATCHES" "CLENS")))
(AND \USE-LENS? (NOT (NULL (CL:FIND-PACKAGE "PCL"))))
; Want to use Common-Lens, so load it now
(CL:LOAD (CL:MAKE-PATHNAME :NAME "LOAD-CLENS" :DEFAULTS USER::LENS-LOADER-DIR)))
(T
; Give the user the option of using it later
(|AddLoadMenuItem| "MailTools" ' ((FROM VALUEOF USER::LENS-LOADER-DIR)
"LOAD-CLENS")
' (IGNORE-ERRORS (LAFITE 'OFF)
(CL:FUNCALL (CL:INTERN "COMMON-LENS" (CL:FIND-PACKAGE "CLENS")
:ON))))))

;; Turn on mailer (no, no, not Norman -- but he is easy to turn on. Or so I've heard.)

(RPAQ? \TURN-ON-MAILER (EQUAL (ETHERHOSTNUMBER HOME-MACHINE-NAME)
(ETHERHOSTNUMBER)))

(EVAL-AT-GREET (COND
((NOT (GETD 'LAFITE))
; Don't turn on mail if it doesn't exist
NIL)
((NOT \TURN-ON-MAILER)
; Don't turn on if the user doesn't want to
NIL)
((AND \USE-LENS? (NOT (NULL (CL:FIND-PACKAGE "CLENS"))))
; Use COMMON-LENS if it's here and the user wants it
(CL:FUNCALL (CL:INTERN "COMMON-LENS" (CL:FIND-PACKAGE "CLENS")
:ON))
; Use Lafite
(T (LAFITE 'ON))))))

(RPAQQ CHAT-INIT-COMMANDS

(

;;; CHAT stuff

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (ALISTS (NETWORKLOGININFO LILY SYMBOLICS))
(VARS (CHAT.ALLHOSTS (SORT `(|IBID:PARC:XEROX| |PARC CHS:PARC:XEROX| ERINYES ERIS PHYLUM QV
, @(|for| HOST |in| DEFAULTPRINTINGHOST |when|
(EQ (PRINTERTYPE HOST)
' INTERPRESS)
|collect| HOST))))
(CLOSECHATWINDOWFLG T)
(CHAT.WINDOW.SIZE (LET ((N-CHARS-WIDE 80)
(N-CHARS-HIGH 24))
(CONS (MIN (QUOTIENT (TIMES SCREENWIDTH 2)
3)
(WIDTHIFWINDOW (TIMES N-CHARS-WIDE (STRINGWIDTH "A"
CHAT.FONT))))
(MIN (QUOTIENT (TIMES SCREENHEIGHT 2)
3)
(HEIGHTIFWINDOW (TIMES N-CHARS-HIGH (FONTPROP
CHAT.FONT
' HEIGHT))
T))))))
(DEFAULTCHATHOST (FILENAMEFIELD LOGINHOST/DIR 'HOST)))
(LOADMENUITEMS NIL ((SYSLOAD FROM LISPUSERS)
"TCPChat"))))

;;; CHAT stuff

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(ADDTovar NETWORKLOGININFO (LILY (LOGIN "1" USERNAME CR PASSWORD CR))
(SYMBOLICS (LOGIN)))

(RPAQ CHAT.ALLHOSTS (SORT `(|IBID:PARC:XEROX| |PARC CHS:PARC:XEROX| ERINYES ERIS PHYLUM QV
, @(|for| HOST |in| DEFAULTPRINTINGHOST |when| (EQ (PRINTERTYPE HOST)

```

' INTERPRESS)
|collect| HOST)))
(RPAQQ CLOSECHATWINDOWFLG T)
(RPAQ CHAT.WINDOW.SIZE
(LET ((N-CHARS-WIDE 80)
(N-CHARS-HIGH 24))
(CONS (MIN (QUOTIENT (TIMES SCREENWIDTH 2)
3)
(WIDTHIFWINDOW (TIMES N-CHARS-WIDE (STRINGWIDTH "A" CHAT.FONT))))
(MIN (QUOTIENT (TIMES SCREENHEIGHT 2)
3)
(HEIGHTIFWINDOW (TIMES N-CHARS-HIGH (FONTPROP CHAT.FONT 'HEIGHT)
T))))))

```

```

(RPAQ DEFAULTCHATHOST (FILENAMEFIELD LOGINHOST/DIR 'HOST))

```

```

(|AddLoadMenuItem| 'NIL ' ((SYSLOAD FROM LISPUSERS)
"TCPChat"))
)

```

(RPAQQ TALK-INIT-COMMANDS

```

(;; Talk program initialization commands
(INITVARS (TALK.DEFAULT.REGION (CREATEREGION 575 0 500 500))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS NIL ((SYSLOAD FROM LISPUSERS)
"Talk"))))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS NIL ((SYSLOAD COMPILED NOERROR FROM
"{PHYLUM}<Wickberg>Lisp>"
FING)
(FINGW))))))

```

;; Talk program initialization commands

```

(RPAQ? TALK.DEFAULT.REGION (CREATEREGION 575 0 500 500))

```

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

```

(|AddLoadMenuItem| 'NIL ' ((SYSLOAD FROM LISPUSERS)
"Talk"))
)

```

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```

```

(|AddLoadMenuItem| 'NIL ' ((SYSLOAD COMPILED NOERROR FROM "{PHYLUM}<Wickberg>Lisp>"
FING)
' (FINGW))
)

```

(RPAQQ CALENDAR-INIT-COMMANDS

```

(
;;; Calendar
(INITVARS (CALDAYDEFAULTREGION (CREATEREGION 32 200 375 100))
(CALDEFAULTALERTDELTA -10)
(CALDEFAULTHOST&DIR (CONCAT |\\UserHomeDirectory| "CALENDAR>"))
(CALFONT (FONTCREATE '(HELVETICA 18)))
(CALUPDATEONSHRINKFLG T)
(CALKEEPEXPIREDREMS T)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS NIL
((SYSLOAD FROM LISPUSERS)
"Calendar")
(PROGN (CALLOADFILE "CalReminders")
(LET ((FILE (FINDFILE "Calmanac88" T
LISPUSERSDIRECTORIES)))
(CL:WHEN FILE (CALLOADFILE FILE)))
(CALENDAR 'THISYEAR))))))

```

;;; Calendar

```

(RPAQ? CALDAYDEFAULTREGION (CREATEREGION 32 200 375 100))

```

```

(RPAQ? CALDEFAULTALERTDELTA -10)

```

```

(RPAQ? CALDEFAULTHOST&DIR (CONCAT |\\UserHomeDirectory| "CALENDAR>"))

```

```

(RPAQ? CALFONT (FONTCREATE '(HELVETICA 18)))

```

```

(RPAQ? CALUPDATEONSHRINKFLG T)

```

```

(RPAQ? CALKEEPEXPIREDREMS T)

```

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

```



```

{MEDLEY}<lispusers>GENERIC-INIT.;1

(|AddLoadMenuItem| 'NIL ' ((SYSLOAD FROM LISPUSERS)
                          "Calendar")
  ' (PROGN (CALLOADFILE "CalReminders")
          (LET ((FILE (FINDFILE "Calmanac88" T LISPUSERSDIRECTORIES))
                (CL:WHEN FILE (CALLOADFILE FILE)))
            (CALENDAR 'THISYEAR)))
)

(RPAQQ PRINTER-INIT-COMMANDS
  ( ;; Printer menu
    (INITVARS (PRINTERMENU.POSITION (CREATEPOSITION (DIFFERENCE SCREENWIDTH 125)
                                                    5)))
    (LOADMENUITEMS NIL (((SYSLOAD FROM LISPUSERS)
                          "PrinterMenu")
                        (PRINTERMENU)))
    ;; Hardcopying graphs
    (LOADMENUITEMS NIL (((SYSLOAD FROM LISPUSERS)
                          "HGraph"))))

 ;; Printer menu
(RPAQ? PRINTERMENU.POSITION (CREATEPOSITION (DIFFERENCE SCREENWIDTH 125)
                                             5))

(|AddLoadMenuItem| 'NIL ' ((SYSLOAD FROM LISPUSERS)
                          "PrinterMenu")
  ' (PRINTERMENU))

 ;; Hardcopying graphs
(|AddLoadMenuItem| 'NIL ' ((SYSLOAD FROM LISPUSERS)
                          "HGraph"))

(RPAQQ DB-INIT-COMMANDS
  ( ;; Information gathering database stuff
    (INITVARS (|*Address-Book-Pos*| (CREATEPOSITION 19 181))
              (|*Phone-Directory-Pos*| (CREATEPOSITION 19 181)))
    (ADDVARS (PHONELISTFILES "{PHYLUM}<KSA>ROLODEX.TEDIT")
              (LOADMENUITEMS NIL (((FROM LISPUSERS)
                                    "Phone-Directory")
                                  ((FROM LISPUSERS)
                                   "AddressBook"))))
    ;; Bibliographic lookup
    (LOADMENUITEMS "WritingAids" (((FROM LISPUSERS)
                                     "Find-Citation"))))

 ;; Information gathering database stuff
(RPAQ? |*Address-Book-Pos* (CREATEPOSITION 19 181))
(RPAQ? |*Phone-Directory-Pos* (CREATEPOSITION 19 181))
(ADDTOVAR PHONELISTFILES "{PHYLUM}<KSA>ROLODEX.TEDIT")

(|AddLoadMenuItem| 'NIL ' ((FROM LISPUSERS)
                          "Phone-Directory"))

(|AddLoadMenuItem| 'NIL ' ((FROM LISPUSERS)
                          "AddressBook"))

 ;; Bibliographic lookup
(|AddLoadMenuItem| ' "WritingAids" ' ((FROM LISPUSERS)
                                     "Find-Citation"))

(RPAQQ NFS-INIT-COMMANDS ( ;; Add NFS support
                          (LOADMENUITEMS "FileAids" (((SYSLOAD FROM "{NB:PARC:XEROX}<NFS>")
                                                         "PARC-NFS"))))

 ;; Add NFS support
(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM "{NB:PARC:XEROX}<NFS>")
                                     "PARC-NFS"))

 ;; Files
(RPAQQ FILE-WATCH-INIT-COMMANDS
  (

```

;; FileWatcher

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
  (INITVARS (|FW-Properties| `(FONT (GACHA 8)
    ALL-FILES? NIL POSITION
    , (CREATEPOSITION SCREENWIDTH
      (CL:IF (AND (BOUNDP '*WHO-LINE*)
        (WINDOWP *WHO-LINE*))
        (DIFFERENCE SCREENHEIGHT (|fetch|
          (REGION HEIGHT)
          |of|
          (WINDOWPROP *WHO-LINE*
            'REGION)))
        SCREENHEIGHT))
    ANCHOR TOP-RIGHT SHADE ,GRAYSHADE INTERVAL 1000 FILTERS
    ({DSK1}SYSDIR {CORE}<*>ROLODEX.* {CORE}<*>*PHONE*.TXT)))
  (LOADMENUITEMS "FileAids" (((SYSLOAD FROM LISPUSERS)
    "FileWatch")
    (FILEWATCH))))))
```

;; FileWatcher

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
  (RPAQ? |FW-Properties|
    `(FONT (GACHA 8)
      ALL-FILES? NIL POSITION , (CREATEPOSITION SCREENWIDTH (CL:IF (AND (BOUNDP '*WHO-LINE*)
        (WINDOWP *WHO-LINE*))
        (DIFFERENCE SCREENHEIGHT
          (|fetch| (REGION HEIGHT)
          |of|
          (WINDOWPROP *WHO-LINE*
            'REGION)))
        SCREENHEIGHT))
      ANCHOR TOP-RIGHT SHADE ,GRAYSHADE INTERVAL 1000 FILTERS ({DSK1}SYSDIR {CORE}<*>ROLODEX.*
        {CORE}<*>*PHONE*.TXT)))
```

```
(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
  "FileWatch")
  ' (FILEWATCH))
)
```

(RPAQQ **FILE-SERVER-INIT-COMMANDS**

```
((LOADMENUITEMS "FileAids" (((SYSLOAD FROM LISPUSERS)
  "NSProtection")
  (NSPROTECTION)))
```

;; The ArchiveTool

```
(LOADMENUITEMS "FileAids" (((SYSLOAD FROM LISPUSERS)
  "ArchiveTool")
  ((SYSLOAD FROM LISPUSERS)
  "ArchiveBrowser")
  ((SYSLOAD FROM LISPUSERS)
  "NSAllocation"))))
```

```
(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
  "NSProtection")
  ' (NSPROTECTION))
```

;; The ArchiveTool

```
(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
  "ArchiveTool"))
```

```
(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
  "ArchiveBrowser"))
```

```
(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
  "NSAllocation"))
```

(RPAQQ **DIRGRAPHER-INIT-COMMANDS**

```
(; DirGrapher
  (INITVARS (DG.FILE-INFO-ATTRIBUTES FB.DEFAULT.INFO)
    (DG.DEFAULT-DIR |\\UserHomeDirectory|)
    (DG.VERTICAL-HORIZONTAL-OPTION 'HORIZONTAL)
    (DG.BACKGROUND-DIRECTORIES ` (, DG.DEFAULT-DIR "{NB:PARC:XEROX}<Colab>" "{Phylum}<KSA>"
      "{Pogo:AISNorth:XEROX}<Loops>"))
  (DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "FileAids" (((SYSLOAD FROM LISPUSERS)
    "DirGrapher")))))
```

;; DirGrapher

```
(RPAQ? DG.FILE-INFO-ATTRIBUTES FB.DEFAULT.INFO)
```

```

{MEDLEY}<lispusers>GENERIC-INIT.;1

(RPAQ? DG.DEFAULT-DIR |\\UserHomeDirectory|)

(RPAQ? DG.VERTICAL-HORIZONTAL-OPTION 'HORIZONTAL)

(RPAQ? DG.BACKGROUND-DIRECTORIES `(,DG.DEFAULT-DIR "{NB:PARC:XEROX}<Colab>" "{Phylum}<KSA>"
                                     "{Pogo:AISNorth:XEROX}<Loops>"))

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
                                   "DirGrapher"))
)

(RPAQQ FB-INIT-COMMANDS ( ;; FileBrowser tailoring
                          (DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (VARS (FB.DEFAULT.INFO ' (SIZE CREATIONDATE)))
                                    (LOADMENUITEMS "FileAids" (((SYSLOAD FROM LISPUSERS)
                                                                "Resize-FileBrowser")))))

;; FileBrowser tailoring

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(RPAQQ FB.DEFAULT.INFO (SIZE CREATIONDATE))

(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
                                   "Resize-FileBrowser"))
)

(RPAQQ COMPARE-FILES-INIT-COMMANDS ( ;; Comparing files and the like
                                     (LOADMENUITEMS "FileAids" (((SYSLOAD FROM LISPUSERS)
                                                                "CompareDirectories"))
                                     (((SYSLOAD FROM LISPUSERS)
                                      "CompareText"))
                                     (((SYSLOAD FROM LISPUSERS)
                                      "CompareSources")))))

;; Comparing files and the like

(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
                                   "CompareDirectories"))

(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
                                   "CompareText"))

(|AddLoadMenuItem| ' "FileAids" ' ((SYSLOAD FROM LISPUSERS)
                                   "CompareSources"))

;; Random stuff

(RPAQQ UNIX-INIT-COMMANDS ( ;; Some of UNIXish commands
                             (VARIABLES *UNIX-DIR-STACK*)
                             (FUNCTIONS DO-CD PRINT-DIRECTORY-STACK)
                             (COMMANDS "CD" "DIRS" "LS" "MORE" "POPD" "PUSHD" "PWD")))

;; Some of UNIXish commands

(DEFGLOBALVAR *UNIX-DIR-STACK* NIL
  "The directory stack used in the exec commands PUSHD and friends.")

(CL:DEFUN DO-CD (DIRECTORY)
  (IF (STRING-EQUAL DIRECTORY "..")
      THEN (LET* ((CURRENT (DIRECTORYNAME T))
                  (PARENT (SUBSTRING CURRENT 1 (STRPOS ">" CURRENT -2 NIL NIL NIL NIL T))))
            (CNDIR PARENT))
      ELSE (CNDIR DIRECTORY))
  (DIRECTORYNAME T))

(CL:DEFUN PRINT-DIRECTORY-STACK ()
  (CL:FORMAT T "~%~A" (DIRECTORYNAME T))
  (FOR DIR IN *UNIX-DIR-STACK* DO (CL:FORMAT T " ~A" DIR))
  (CL:VALUES))

(DEFCOMMAND "CD" (&OPTIONAL DIRECTORY) "Connect to a directory"
  (DO-CD DIRECTORY)
  (SETQ *UNIX-DIR-STACK* NIL)
  (CL:FORMAT T "~%~A" (DIRECTORYNAME T))
  (CL:VALUES))

```

```
(DEFCOMMAND "DIRS" NIL "Print out the directory stack used by PUSHD and friends."
  (PRINT-DIRECTORY-STACK)
  (CL:VALUES))
```

```
(DEFCOMMAND "LS" (&OPTIONAL (DIRSPEC "**")) "List files matching the spec"
  (LET (( (FILING.ENUMERATION.DEPTH 1)
        (DIRECTORY DIRSPEC ' (P)))
    (CL:VALUES))
```

```
(DEFCOMMAND "MORE" (FILE) "Type the contents of a file"
  (CL:FUNCALL #'SEE FILE)
  (CL:VALUES))
```

```
(DEFCOMMAND "POPD" (DIRECTORY) "Connect to the previous directory"
  (IF (NULL *UNIX-DIR-STACK*)
    THEN (CL:FORMAT T "~%popd: Directory stack empty.")
    ELSE (DO-CD (POP *UNIX-DIR-STACK*)
            (PRINT-DIRECTORY-STACK))
    (CL:VALUES))
```

```
(DEFCOMMAND "PUSHD" (DIRECTORY) "Connect to a directory, remember the current one on the directory stack."
  (CL:PUSH (DIRECTORYNAME T)
    *UNIX-DIR-STACK*)
  (DO-CD DIRECTORY)
  (PRINT-DIRECTORY-STACK)
  (CL:VALUES))
```

```
(DEFCOMMAND "PWD" NIL "Print out the currently connected directory."
  (CL:FORMAT T "~%~A" (DIRECTORYNAME T))
  (CL:VALUES))
```

(RPAQQ **DEMOS-INIT-COMMANDS**
(

;;; Giving demos and the like

```
(INITVARS (|SlideFiles| ' ("{PHYLUM}<KSA>Talks>*.Tedit;"))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "Demos" (((SYSLOAD FROM LISPUSERS)
  "SlideProjector")))
  (((SYSLOAD FROM LISPUSERS)
  "Magnifier")))
  (((SYSLOAD FROM LISPUSERS)
  "Big")))
  ((LOAD FROM "{PHYLUM}<KSA>"
  "Demo")))))
```

;;; Giving demos and the like

```
(RPAQ? |SlideFiles| ' ("{PHYLUM}<KSA>Talks>*.Tedit;"))
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
(|AddLoadMenuItem| ' "Demos" ' ((SYSLOAD FROM LISPUSERS)
  "SlideProjector"))
(|AddLoadMenuItem| ' "Demos" ' ((SYSLOAD FROM LISPUSERS)
  "Magnifier"))
(|AddLoadMenuItem| ' "Demos" ' ((SYSLOAD FROM LISPUSERS)
  "Big"))
(|AddLoadMenuItem| ' "Demos" ' (LOAD FROM "{PHYLUM}<KSA>"
  "Demo"))
)
```

(RPAQQ **GAMES-INIT-COMMANDS**
(

;;; R and R

```
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE (LOADMENUITEMS "Games" (((FROM "{PHYLUM}<Bobrow>Lisp>"
  "BlackBox")))
  (((FROM " {FS8:PARC:XEROX}<Poster>Lisp>"
  "Go")))
  (((SYSLOAD FROM LISPUSERS)
  "Qix")
  (ADD.PROCESS ' (QIX.GROW)))
  (((SYSLOAD FROM LISPUSERS)
  "FaceInvader")))
  (((SYSLOAD FROM LISPUSERS)
```

```

"Donz"))
(( (SYSLOAD FROM LISPUSERS)
  "Doctor"))
(( (SYSLOAD FROM LISPUSERS)
  "Hanoi"))
(( (SYSLOAD FROM LISPUSERS)
  "Life"))
(( (SYSLOAD FROM LISPUSERS)
  "Solitaire"))
(( (SYSLOAD FROM "{FS8:PARC:XEROX}<Foster>Lisp>")
  "RandomWord"))))

```

;; R and R

```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(|AddLoadMenuItem| ' "Games" ' ((FROM "{PHYLUM}<Bobrow>Lisp>")
  "BlackBox"))

(|AddLoadMenuItem| ' "Games" ' ((FROM "{FS8:PARC:XEROX}<Foster>Lisp>")
  "Go"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "Qix")
  ' (ADD.PROCESS ' (QIX.GROW)))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "FaceInvader"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "Donz"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "Doctor"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "Hanoi"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "Life"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM LISPUSERS)
  "Solitaire"))

(|AddLoadMenuItem| ' "Games" ' ((SYSLOAD FROM "{FS8:PARC:XEROX}<Foster>Lisp>")
  "RandomWord"))
)

```

;; Cleanup

```

(RPAQQ BACKGROUND-MENU-CLEANUP-INIT-COMMANDS
  (

```

;; Clean up the background menu

```

(FUNCTIONS MOVE-BACKGROUND-ITEM-UNDER)
(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE
  (P (EVAL-AT-GREET
    ;; Put the HARDCOPY item under the SNAP command
    (MOVE-BACKGROUND-ITEM-UNDER "Hardcopy" "Snap")
    ;; Put the ArchiveTool under the FileBrowser
    (MOVE-BACKGROUND-ITEM-UNDER "ArchiveTool" "FileBrowser")
    ;; Add a menu item for logging out
    (/NCONC1 |BackgroundMenuCommands|
      ' ("LOGOUT & Power-Off" ' (AND (MOUSECONFIRM "Really (LOGOUT)?"
        (|GoodNight|))
        "Logout of LISP"
        (SUBITEMS ("LOGOUT T & Power-Off" ' (AND (MOUSECONFIRM "Really (LOGOUT
          T)?"
            (|GoodNight| T))
            "Logout without saving VMem")))))
    ;; Hide some system-level commands under a single top-level command
    (OR (FOR ITEM IN |BackgroundMenuCommands| THEREIS (STRING-EQUAL "System" (CAR ITEM)))
      (/NCONC1 |BackgroundMenuCommands| (LIST "System" NIL NIL (LIST 'SUBITEMS))))
      (FOR LABEL IN ' ("SaveVM" "Idle" "AR Edit" "DumpCache" "Set Directories" "LOGOUT &
        Power-Off")
        DO
          (MOVE-BACKGROUND-ITEM-UNDER LABEL "System"))))
    ;; Clear the background menu cache
    (VARS (|BackgroundMenu| NIL))))

```

:: Clean up the background menu

```
(CL:DEFUN MOVE-BACKGROUND-ITEM-UNDER (LABEL-TO-MOVE PARENT-LABEL)
  (LET ((ITEM-TO-MOVE (FOR ITEM IN |BackgroundMenuCommands| THEREIS (STRING-EQUAL LABEL-TO-MOVE (CAR ITEM))))
    (PARENT-ITEM (FOR ITEM IN |BackgroundMenuCommands| THEREIS (STRING-EQUAL PARENT-LABEL (CAR ITEM))))
    (COND
      ((OR (NULL PARENT-ITEM)
           (NULL ITEM-TO-MOVE))
        NIL)
      ((NULL (CDDDR PARENT-ITEM))
        ; No subitems yet
        (/NCONC1 PARENT-ITEM `(SUBITEMS ,ITEM-TO-MOVE))
        (/DREMOVE ITEM-TO-MOVE |BackgroundMenuCommands|))
      (T
        ; Already has subitems
        (/NCONC1 (CADDR PARENT-ITEM)
                 ITEM-TO-MOVE)
        (/DREMOVE ITEM-TO-MOVE |BackgroundMenuCommands|))))))
```

(DECLARE\ : DONTEVAL@LOAD DONTEVAL@COMPILE

(EVAL-AT-GREET

```
:: Put the HARDCOPY item under the SNAP command
(MOVE-BACKGROUND-ITEM-UNDER "Hardcopy" "Snap")
:: Put the ArchiveTool under the FileBrowser
(MOVE-BACKGROUND-ITEM-UNDER "ArchiveTool" "FileBrowser")
:: Add a menu item for logging out
(/NCONC1 |BackgroundMenuCommands| '("LOGOUT & Power-Off" '(AND (MOUSECONFIRM "Really (LOGOUT)?"
                                                                    (|GoodNight|))
                                                                    "Logout of LISP"
                                                                    (SUBITEMS ("LOGOUT T & Power-Off" '(AND (MOUSECONFIRM
                                                                    "Really (LOGOUT
                                                                    T)?"
                                                                    (|GoodNight| T))
                                                                    "Logout without saving VMem")))))
:: Hide some system-level commands under a single top-level command
(OR (FOR ITEM IN |BackgroundMenuCommands| THEREIS (STRING-EQUAL "System" (CAR ITEM)))
    (/NCONC1 |BackgroundMenuCommands| (LIST "System" NIL NIL (LIST 'SUBITEMS))))
(FOR LABEL IN '("SaveVM" "Idle" "AR Edit" "DumpCache" "Set Directories" "LOGOUT & Power-Off")
  DO (MOVE-BACKGROUND-ITEM-UNDER LABEL "System")))
```

(RPAQQ |BackgroundMenu| NIL)

(RPAQQ **DO-LOAD-UTILITIES-INIT-COMMANDS** (;

```
:: Load the users specified utilities
(INITVARS *LOAD-UTILITY-OPTIONS* NIL)
(P (EVAL-AT-GREET (CL:UNLESS *GENERIC-INIT-LOADED*
                            (CL:MAPC #'(CL:LAMBDA (UTILITY)
                                          (|PickLoadUtilityItem| UTILITY
                                                                NIL T))
                                          *LOAD-UTILITY-OPTIONS*))))))
```

:: Load the users specified utilities

(RPAQ? *LOAD-UTILITY-OPTIONS* NIL)

(RPAQ? NIL NIL)

```
(EVAL-AT-GREET (CL:UNLESS *GENERIC-INIT-LOADED*
                        (CL:MAPC #'(CL:LAMBDA (UTILITY)
                                          (|PickLoadUtilityItem| UTILITY NIL T))
                        *LOAD-UTILITY-OPTIONS*)))
```

:: Send the Tool Work's a message telling it about this user.

```
(CL:DEFUN XCL-USER::LOG-GENERIC-INIT-USER ()
  "If another person uses Generic-Init, let me know."
  (IGNORE-ERRORS (LET ((XCL-USER:ME "Lanning")
                      (XCL-USER::USER (CL:IF \\CC-GENERIC-INIT-MSG
                                             (USERNAME)
                                             "")))
    (COND
      ((STRING-EQUAL XCL-USER::USER XCL-USER:ME)
        ; Don't bother if the user is me.
        NIL)
      ((NOT (CL:FBOUNDP 'LAFITE.SENDMESSAGE))
        ; Can't send message if LAFITE isn't loaded.
        NIL))
```

```
(T (LAFITE.SENDMESSAGE (CL:FORMAT NIL "Subject: ~A~%To: ~A~%Cc: ~A~@{~%~%~A~}"
    "Generic-Init" XCL-USER::ME XCL-USER::USER "This is to
    let you know that I am using Generic-Init (again).
    Thanks for making it available.))
  (IF (EQ MAKESYSNAME :MEDELY)
    THEN (LAFITE.SENDMESSAGE (CL:FORMAT NIL "Subject: ~A~%To: ~A~%From:
    ~A~@{~%~%~A~}" "!!!Stop using
    Generic-Init!!!" (USERNAME)
    "Generic-Init (lanning.pa)" "Welcome to
    Medley." "Did you know that Generic-Init is no
    longer supported in Medley?" "I thought not."
    "Instead, please start using VANILLA-INIT
    (another failure in the quest for a good
    name)." "You should edit your personal INIT
    file to reflect this change." "" "Have a nice
    day." "--smL"))))))))
```

(RPAQ? \CC-GENERIC-INIT-MSG T)

(EVAL-AT-GREET (CL:UNLESS *GENERIC-INIT-LOADED* (XCL-USER::LOG-GENERIC-INIT-USER)))

(RPAQ *LOAD-VERBOSE* \ORIGINAL-LOAD-VERBOSE)

(RPAQ PRETTYHEADER \ORIGINAL-PRETTYHEADER)

(RPAQQ *GENERIC-INIT-LOADED* T)

:: Make the FileManager happy

(DECLARE\ : DONTCOPY

(PUTPROPS GENERIC-INIT MAKEFILE-ENVIRONMENT (:PACKAGE "IL" :READTABLE "XCL"))
)

(PUTPROPS GENERIC-INIT COPYRIGHT ("Xerox Corporation" 1987 1988))

FUNCTION INDEX

ATOM-NEIGHBORS	23	MOVE-BACKGROUND-ITEM-UNDER	38	PURGE-FILE-ADVICE	5
DebugMode 	20	NewLisp 	22	RememberLastPartition 	22
DO-CD	35	NOTICE	16	RememberLispState 	22
GoodNight 	21	OAM	17	SELECTW	20
LOAD-NOVA-FONTS	25	PRINT-DIRECTORY-STACK	35	\Pick-One-At-Random 	21
XCL-USER::LOG-GENERIC-INIT-USER ..	38	PrintDocFile 	21		
MAKE	17	PURGE-ADVICE	5		

COMMAND INDEX

"BREAK"	17	"DESCRIBE"	17	"EFF"	18	"LS"	36	"NOTICE"	18	"PWD"	36
"CALLS"	17	"DIRS"	36	"FILES?"	18	"MAKE"	18	"POPD"	36	"SPY"	18
"CD"	35	"EC"	17	"IC"	18	"MORE"	36	"PUSHD"	36	"UNBREAK"	17

MACRO INDEX

DE	20	EVAL-AT-GREET	2	FILE	20	PickOneAtRandom 	21
----------	----	---------------------	---	------------	----	------------------------	----

ADVICE INDEX

FILES?	4	GETPROMPTWINDOW	7
--------------	---	-----------------------	---

PROPERTY INDEX

GENERIC-INIT	39
--------------------	----

VARIABLE INDEX

Address-Book-Pos 	33	DEMOS-INIT-COMMANDS	36
CACHE-DIRECTORIES	3	DG.BACKGROUND-DIRECTORIES	35
DEFAULT-CLEANUP-COMPILER	4	DG.DEFAULT-DIR	35
DEFAULT-MAKEFILE-ENVIRONMENT ..	4	DG.FILE-INFO-ATTRIBUTES	34
GENERIC-INIT-LOADED	2, 39	DG.VERTICAL-HORIZONTAL-OPTION ..	35
GIVE-AND-TAKE-DIRECTORIES	4	DINFO-INIT-COMMANDS	23
USER::*LENS-USER-PARAMETERS*	31	DINFOMODES	23
LOAD-SILENT	2	DIRGRAPHER-INIT-COMMANDS	34
LOAD-UTILITY-OPTIONS	38	DISPLAY-CONTROL-INIT-COMMANDS ..	13
LOAD-VERBOSE	2, 39	DO-LOAD-UTILITIES-INIT-COMMANDS ..	38
NEW-LAFITE-P	29	DOCUMENTSMENUFONT	26
ORIGINAL-GIVE-AND-TAKE-DIRECTORIES	4	EDITCOMSA	21
Phone-Directory-Pos 	33	EDITCOMSL	21
PRINT-CASE	3	EMPRESS#SIDES	3
UNIX-DIR-STACK	35	ENFORCE.ICON.GRID	10
WHO-LINE-ANCHOR	7	ENV-TAILORING-INIT-COMMANDS	3
WHO-LINE-DIRECTORIES	7	FB-INIT-COMMANDS	35
WHO-LINE-DISPLAY-NAMES?	7	FB.DEFAULT.INFO	35
WHO-LINE-ENTRIES	8	FILE-SERVER-INIT-COMMANDS	34
AFTERLOGOUTFORMS	4	FILE-WATCH-INIT-COMMANDS	33
BACKGROUND-MENU-CLEANUP-INIT-COMMANDS	37	FILECACHE-INIT-COMMANDS	5
BackgroundMenu 	38	FIXSPELLDEFAULT	3
BEFORELOGOUTFORMS	22	FONT-SETUP-INIT-COMMANDS	6
BICLOCKINITIALPROPS	14	FONTDEFS	6
CALDAYDEFAULTREGION	32	FW-Properties 	34
CALDEFAULTALERTDELTA	32	GAMES-INIT-COMMANDS	36
CALDEFAULTHOST&DIR	32	HOME-MACHINE-NAME	3
CALENDAR-INIT-COMMANDS	32	IDLE-INIT-COMMANDS	13
CALFONT	32	IMAGEOBJGETFNS	26
CALKEEPEXPIREDREMS	32	IRM.FONT	23
CALUPDATEONSHRINKFLG	32	IRM.HOST&DIR	23
CHANGE-BACKGROUND-INIT-COMMANDS ..	11	IRMWINDOWREGION	23
CHAT-INIT-COMMANDS	31	LAFITE.DONT.DISPLAY.HEADERS	29
CHAT.ALLHOSTS	31	LAFITE.DONT.FORWARD.HEADERS	29
CHAT.WINDOW.SIZE	32	LAFITE.DONT.HARDCOPY.HEADERS	29
CLEANUPOPTIONS	3	LAFITEBROWSERREGION	29
CLOCK-INIT-COMMANDS	14	LAFITEDEFAULTHOST&DIR	29
CLOSECHATWINDOWFLG	32	LAFITEDISPLAYREGION	29
COMPARE-FILES-INIT-COMMANDS	35	LAFITELDDIRECTORIES	30
COMPUTE-DIRECTORIES-INIT-COMMANDS ..	2	LAFITEHARDCOPYBATCHFLG	29
COPYRIGHTFLG	4	LAFITEMODEDEFAULT	29
DB-INIT-COMMANDS	33	LAFITEMOVETOCONFIRMFLG	29
DEFAULTCHATHOST	32	LAFITESHOWMODEFLG	29
DEFAULTMAILFOLDERNAME	29	LAFITESTATUSWINDOWPOSITION	29


```
{MEDLEY}<lispusers>GENERIC-INIT.;1
```

USER::LENS-LOADER-DIR	31	\sPY.BUTTON.POS	19
LOGINHOST/DIR	2	TALK-INIT-COMMANDS	32
LOOPS-INIT-COMMANDS	24	TALK.DEFAULT.REGION	32
LOOPSDIRECTORIES	24	TEDIT-INIT-COMMANDS	24
MAIL-INIT-COMMANDS	27	TEDIT.DEFAULT.PROPS	26
MaxValueLeftMargin 	3	TEDIT.DEFAULT.WINDOW	26
NC.NoteCardsIconPosition 	27	TEMPDIR	3
NCINITIALGLOBALPARAMS	27	UNIX-INIT-COMMANDS	35
NETWORKLOGINFO	31	USER-SUITE-DIRECTORIES	11
NFS-INIT-COMMANDS	33	USERMACROS	20
NOTECARDS-INIT-COMMANDS	27	VSTATS-INIT-COMMANDS	8
NOTECARDSDIRECTORIES	27	VSTATS.CLOCK.INTERVAL	8
OLD-UTILS-COMMANDS	19	VSTATS.MUTIL.INTERVAL	8
PATCH-INIT-COMMANDS	4	VSTATS.POSITION	8
PCL-INIT-COMMANDS	23	WHO-LINE-COMMANDS	7
PCLDIRECTORIES	24	WINDOWTITLESHADE	9
PHONELISTFILES	33	WIZARD-INIT-COMMANDS	22
PRETTYHEADER	2,39	\\BasicUserName 	2
PRINTER-INIT-COMMANDS	33	\\CC-GENERIC-INIT-MSG	39
PRINTERMENU.POSITION	33	\\ETHERTIMEOUT	3
PRIVATE-LISPUSERSDIRECTORIES	3	\\FONT-PROFILE-NAME	7
PROGRAMMING-INIT-COMMANDS	15	\\LOAD-GRID-ICONS	10
RECOMPILEDEFAULT	4	\\LOOPS-INIT-FORM	24
ROOMS-INIT-COMMANDS	10	\\ORIGINAL-LOAD-VERBOSE	2
ROOMSDIRECTORIES	11	\\ORIGINAL-PRETTYHEADER	2
ROOMSUSERSDIRECTORIES	11	\\REARRANGE-SCREEN	9
SCREEN-SETUP-INIT-COMMANDS	8	\\TURN-ON-MAILER	31
SCROLLBARWIDTH	3	\\USE-LENS?	31
SKETCH-INIT-COMMANDS	26	\\UserHomeDirectory 	3
SlideFiles 	36		
