

File created: 2-Apr-87 17:06:05 {ERIS}<LISPUSERS>LYRIC>COMMWINDOW.;3

changes to: (VARS REMOTE-CURSOR COMMWINDOWCOMS)  
(COURIERPROGRAMS COMMWINDOW)  
(FNS CLOSE-FRAME START-GET-BITS SEND-BITS FRAME-EVENT MAKE-FRAME)  
(FUNCTIONS \PILOTBITBLT)

previous date: 2-Apr-87 16:54:24 {ERIS}<LISPUSERS>LYRIC>COMMWINDOW.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
::  
:: Copyright (c) 1986, 1990, 1987 by Xerox Corporation. All rights reserved.

(RPAQQ **COMMWINDOWCOMS**  
(

::: Viewer end

(FNS CLOSE-FRAME GET-BITS START-GET-BITS)  
(FILES COURIERSERVE)

::: Sender end

(FNS SEND-BITS SEND-TILE LISTEN-TO-VIEWER MAPTILES SHUT-DOWN-VIEWER CHANGE-SENDER-UPDATE-MODE)  
(FUNCTIONS INCR \PILOTBITBLT)

::: Controlling update schemes

(INITVARS (COMM.DEFAULT.TRANSMIT.TYPE 'SQUARE)  
(COMM.SEND.UNCHANGED.TILES T)  
(COMM.UPDATE.MOUSE.POSITION 'Sender))  
(GLOBALVARS COMM.DEFAULT.TRANSMIT.TYPE COMM.UPDATE.MOUSE.POSITION COMM.SEND.UNCHANGED.TILES)

::: Pruning out unchanged screen tiles

(FNS PACKET-EQUAL GET-CACHED-PACKET PUT-CACHED-PACKET)

::: Low level packet exchange code

(CONSTANTS COMM.BAND.PACKET.TYPE COMM.CURSOR.PACKET.TYPE COMM.CURSOR.CLOSE.PACKET.TYPE  
COMM.SHUT.DOWN.PACKET.TYPE)  
(VARIABLES MAX-PACKET-BITS)  
(RECORDS COMM.XFER.PACKET)

::: Packing and unpacking bitmaps into etherpackets

(FNS BMTOPACKET PACKETTOBM)

::: Displaying the viewing machine's cursor

(VARS REMOTE-CURSOR)  
(INITVARS (CURSORICON NIL))

::: Manipulating the frame that outlines the region being viewed

(INITVARS (\*FRAME-SHADE\* GRAYSHADE))  
(FNS FRAME-EVENT MAKE-FRAME MOVE-FRAME SHAPE-FRAME SET-FRAME-TITLE)

::: Changing the system parameters

(FNS MAKE-MENUS-WINDOW MODE-MENU)  
(VARS COMM-MODES)

::: Initialization

(P (COURIER.START.SERVER))

::: Unused stuff, as far as I can tell

(FNS FASTBITBLT)

::: System file dependencies

```
(DECLARE%: DONTCOPY DOEVAL@COMPILE (FILES (LOADCOMP)
                                           LLDISPLAY LLEATHER LLNS))
(COURIERPROGRAMS COMMWINDOW)))
```

;; Viewer end

(DEFINEQ

**(CLOSE-FRAME**

```
[LAMBDA (FRAME)
  (MAPC FRAME 'CLOSEW)])
```

; Edited 2-Apr-87 16:50 by Masinter

**(GET-BITS**

```
[LAMBDA (RECEIVE-SOCKET WINDOW)
  (RESETLST
```

; Edited 24-Nov-86 13:16 by sml

```
(RESETSAVE NIL (LIST 'CLOSENSOCKET RECEIVE-SOCKET))
[LET ((BBT (create PILOTBBT))
      (STREAM (GETSTREAM WINDOW 'OUTPUT))
      (SCRATCHX 0)
      (SCRATCHY 0)
      SPREAD SCRATCH SEENX SEENY CURSORUP CURLFT CURBTM CURX CURY X Y DATA WORDLEFT WINDOWBOTTOMLINE
      (CURSORCOVEREDIMAGE (BITMAPCREATE 16 16))
      (TRACKING-CURSOR? NIL))
  ; CURLFT and CURBTM are the left and bottom of the cursor
  ; bitmap positions, adjusted for hot spot.
```

(bind CP PACKET THISWIDTH THISHEIGHT while T
do (COND

```
((SETQ PACKET (GETXIP RECEIVE-SOCKET 3000))
 (SELECTC (fetch (COMM.XFER.PACKET PACKET-TYPE) of PACKET)
          (COMM.CURSOR.PACKET.TYPE) ; ignore data, just move cursor
```

```
)
 (COMM.BAND.PACKET.TYPE
  (SETQ SPREAD (fetch (COMM.XFER.PACKET SPREAD) of PACKET))
  (SETQ X (fetch (COMM.XFER.PACKET DATA) of PACKET))
  (SETQ Y (fetch (COMM.XFER.PACKET DATA) of PACKET))
  (SETQ THISWIDTH (fetch (COMM.XFER.PACKET THISWIDTH) of PACKET))
  (SETQ THISHEIGHT (fetch (COMM.XFER.PACKET THISHEIGHT) of PACKET))
  (COND
```

```
((AND CURSORUP (<= (- X 16)
                    CURLFT
                    (+ X THISWIDTH 16))
 (<= (- Y 16)
      CURBTM
      (+ Y THISHEIGHT 16)))
 (BITBLT CURSORCOVEREDIMAGE 0 0 STREAM CURLFT CURBTM)
 (SETQ CURSORUP NIL)))
```

```
[COND ((OR (> THISWIDTH SCRATCHX)
          (> THISHEIGHT SCRATCHY))
```

;; make sure scratch bitmap is big enough

```
(SETQ SCRATCH (BITMAPCREATE (SETQ SCRATCHX (MAX SCRATCHX THISWIDTH))
                             (SETQ SCRATCHY (MAX SCRATCHY THISHEIGHT))
```

```
(PACKETTOBM BBT (fetch (COMM.XFER.PACKET BITS) of PACKET)
             THISWIDTH THISHEIGHT SCRATCH 0 0 SPREAD)
 (BITBLT SCRATCH 0 0 WINDOW X Y THISWIDTH THISHEIGHT))
```

(COMM.SHUT.DOWN.PACKET.TYPE ;; Shut down the listener

```
(CLOSEW WINDOW)
(RETURN)
```

```
(PRINTOUT PROMPTWINDOW "Odd packet" (fetch (COMM.XFER.PACKET PACKET-TYPE)
                                             of PACKET)))
```

```
(SETQ SEENX (fetch (COMM.XFER.PACKET CURSORX) of PACKET))
(SETQ SEENY (fetch (COMM.XFER.PACKET CURSORY) of PACKET))
(RELEASE.XIP PACKET)
```

```
(COND ([AND (KEYDOWNP 'LSHIFT)
            (<= 0 (SETQ X (LASTMOUSEX WINDOW))
                 (WINDOWPROP WINDOW 'WIDTH))
            (<= 0 (SETQ Y (LASTMOUSEY WINDOW))
                 (WINDOWPROP WINDOW 'HEIGHT))
```

;; Tell the sender to track our cursor.

```
(SETQ CP (\FILLINXIP \XIPT.EXCHANGE RECEIVE-SOCKET (fetch XIPSOURCEHOST of PACKET)
          (fetch XIPSOURCECKET of PACKET)
          (fetch XIPSOURCECNET of PACKET)
          NIL)) ; send more than we need just to see if it fixes the problem
```

```
(XIPAPPEND.WORD CP 0)
(XIPAPPEND.WORD CP COMM.CURSOR.PACKET.TYPE) ; turn into a cursor ack
```

```
(XIPAPPEND.WORD CP X)
(XIPAPPEND.WORD CP Y)
(CL:ASSERT (AND (= (fetch (COMM.XFER.PACKET PACKET-TYPE) of CP)
                   COMM.CURSOR.PACKET.TYPE)
```



```

(PARTNERHOST (fetch NSHOSTNUMBER PARTNERADDRESS))
(PARTNERNET (fetch NSNET PARTNERADDRESS))
(PARTNERCALL (COURIER.OPEN PARTNERADDRESS))
(PACKET NIL)
(VIEWER-RETURNED-VALUE (COURIER.CALL PARTNERCALL 'COMMWINDOW 'START-GET-BITS (WINDOWPROP
(CAR FRAME)
'FRAME-REGION)
(USERNAME)))
(PARTNERSOCKET (CAR VIEWER-RETURNED-VALUE))
(PARTNERUSERNAME (CADR VIEWER-RETURNED-VALUE))
(BBT (create PILOTBBT))
(RESETSAVE NIL (LIST 'CLOSESOCKET SENDSOCKET))
(CLOSEF PARTNERCALL) ; close SPP connection, needs no more RPC
(DISCARDXIPS SENDSOCKET)
(RESETSAVE NIL (LIST 'SHUT-DOWN-VIEWER SENDSOCKET PARTNERHOST PARTNERNET PARTNERSOCKET))
(SET-FRAME-TITLE FRAME (CONCAT "Displaying region on " PARTNERUSERNAME "'s display"))
(while T
  do (DESTRUCTURING-BIND (L B W H)
    (WINDOWPROP (CAR FRAME)
'FRAME-REGION)
(MAPC FRAME (FUNCTION TOTOPW))
(MAPTILES MAX-PACKET-BITS W H L B
(FUNCTION (LAMBDA (X Y THIS-WIDTH THIS-HEIGHT SPREAD)
(LISTEN-TO-VIEWER SENDSOCKET L B)
(SEND-TILE X Y L B THIS-WIDTH THIS-HEIGHT SPREAD SENDSOCKET
PARTNERHOST PARTNERNET PARTNERSOCKET PACKET]))))

```

(SEND-TILE

```

[LAMBDA (X Y FRAME-LEFT FRAME-BOTTOM THIS-WIDTH THIS-HEIGHT SPREAD SENDSOCKET PARTNERHOST PARTNERNET
PARTNERSOCKET PACKET) ; Edited 24-Nov-86 14:45 by sml

```

;;; Send a tile to the receiver

```

(SETQ PACKET (\FILLXIP \XIPT.EXCHANGE SENDSOCKET PARTNERHOST PARTNERSOCKET PARTNERNET))
(replace EPREQUEUE of PACKET with 'FREE)
(XIPAPPEND.WORD PACKET (OR SPREAD (SETQ SPREAD 0)))
(XIPAPPEND.WORD PACKET COMM.BAND.PACKET.TYPE)
;; Reserve space for the cursor pos, to be filled in later on
(XIPAPPEND.WORD PACKET 0)
(XIPAPPEND.WORD PACKET 0)
(XIPAPPEND.WORD PACKET X)
(XIPAPPEND.WORD PACKET Y)
(XIPAPPEND.WORD PACKET THIS-WIDTH)
(XIPAPPEND.WORD PACKET THIS-HEIGHT)
(BMTOPACKET NIL (SCREENBITMAP)
(+ FRAME-LEFT X)
(+ FRAME-BOTTOM Y)
THIS-WIDTH THIS-HEIGHT (fetch (COMM.XFER.PACKET BITS) of PACKET)
SPREAD)
(add (fetch XIPLength PACKET)
(IQUOTIENT (+ 7 (TIMES THIS-WIDTH THIS-HEIGHT)
8))
[CL:ASSERT (with COMM.XFER.PACKET PACKET (AND (EQ DATA X)
(EQ DATA Y)
(EQ THISWIDTH THIS-WIDTH)
(EQ THISHEIGHT THIS-HEIGHT)
[if [OR COMM.SEND.UNCHANGED.TILES (NOT (PACKET-EQUAL PACKET (GET-CACHED-PACKET X Y PARTNERHOST PARTNERNET
PARTNERSOCKET)
then (PUT-CACHED-PACKET PACKET X Y PARTNERHOST PARTNERNET PARTNERSOCKET)
else ;; There has been no change in the bits, so don't bother to send them
(replace (XIP XIPLength) of PACKET with (CONSTANT (PLUS \XIPOVLEN (TIMES 2 (INDEXF (FETCH (
COMM.XFER.PACKET
DATALOC)
OF T])
;; Set in the cursor pos and send the packet
(replace (COMM.XFER.PACKET CURSORX) of PACKET with (LOGAND (- LASTMOUSEX FRAME-LEFT)
65535))
(replace (COMM.XFER.PACKET CURSORY) of PACKET with (LOGAND (- LASTMOUSEY FRAME-BOTTOM)
65535))
(SENDXIP SENDSOCKET PACKET)
(BLOCK])

```

(LISTEN-TO-VIEWER

```

[LAMBDA (SENDSOCKET FRAME-LEFT FRAME-BOTTOM) ; Edited 24-Nov-86 13:13 by sml

```

;; Update the display of the viewers cursor

```

(bind CURSORPACKET while (SETQ CURSORPACKET (GETXIP SENDSOCKET 0))
do (SELECTC (fetch (COMM.XFER.PACKET PACKET-TYPE)
CURSORPACKET) ; got an ack
(COMM.CURSOR.PACKET.TYPE
(MOVEW CURSORICON (+ (fetch (COMM.XFER.PACKET CURSORX)

```



(MIN PACKETHEIGHT (- H Y))

(SHUT-DOWN-VIEWER

[LAMBDA (SENDSOCKET PARTNERHOST PARTNERNET PARTNERSOCKET) ; Edited 24-Nov-86 11:40 by smL

;;; Send a shut-down packet to the receiver

;; Beware, this may fail on a noisy line, so we do it twice, just to be safer.

(to 2 do (LET ((PACKET (\FILLINXIP \XIPT.EXCHANGE SENDSOCKET PARTNERHOST PARTNERSOCKET PARTNERNET)))
(replace EPQUEUE of PACKET with 'FREE)
(XIPAPPEND.WORD PACKET 0)
(XIPAPPEND.WORD PACKET COMM.SHUT.DOWN.PACKET.TYPE)
(SENDXIP SENDSOCKET PACKET]))

(CHANGE-SENDER-UPDATE-MODE

[LAMBDA (NEW-MODE) ; Edited 24-Nov-86 12:49 by smL

(SETQ COMM.DEFAULT.TRANSMIT.TYPE NEW-MODE))

)

(DEFMACRO INCR (VAR START HEIGHT REPEATS UNTIL &REST FORMS)

(CL:DO ((REPEAT-COUNT 0 (+ REPEAT-COUNT 1)))
(>= REPEAT-COUNT ,REPEATS))
(CL:DO [(,VAR (+ ,START (CL:\* REPEAT-COUNT ,HEIGHT))
(+ ,VAR (CL:\* ,REPEATS ,HEIGHT)
(,UNTIL)
,@FORMS))

(DEFMACRO \PILOTBITBLT (XCL-USER::TABLE &OPTIONAL (XCL-USER::N 0))

(CL:ASSERT (EQL XCL-USER::N 0))
\((OPCODES PILOTBITBLT)
,XCL-USER::TABLE 0))

;;; Controlling update schemes

(RPAQ? COMM.DEFAULT.TRANSMIT.TYPE 'SQUARE)

(RPAQ? COMM.SEND.UNCHANGED.TILES T)

(RPAQ? COMM.UPDATE.MOUSE.POSITION 'Sender)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS COMM.DEFAULT.TRANSMIT.TYPE COMM.UPDATE.MOUSE.POSITION COMM.SEND.UNCHANGED.TILES)

)

;;; Pruning out unchanged screen tiles

(DEFINEQ

(PACKET-EQUAL

[LAMBDA (PACKET1 PACKET2) ; Edited 24-Nov-86 14:36 by smL

;; Are the data parts of two packets equal?

(AND (type? ETHERPACKET PACKET1)
(type? ETHERPACKET PACKET2)
(EQ (fetch (XIP XIPLength) of PACKET1)
(fetch (XIP XIPLength) of PACKET2))
(LET ((DATA-BYTES (DIFFERENCE (fetch (XIP XIPLength) of PACKET1)
\XIPOVLEN)))
(AND (for I from 0 to (LRSH DATA-BYTES 1) bind (DATA1 \_ (fetch (COMM.XFER.PACKET BITS) of PACKET1))
(DATA2 \_ (fetch (COMM.XFER.PACKET BITS) of PACKET2))
always (EQ (\GETBASE DATA1 I)
(\GETBASE DATA2 I)))
(OR (ZEROP (LOGAND 1 DATA-BYTES))
(EQ (\GETBASEBYTE PACKET1 (SUB1 (fetch (XIP XIPLength) of PACKET1)))
(\GETBASEBYTE PACKET2 (SUB1 (fetch (XIP XIPLength) of PACKET1))))

(GET-CACHED-PACKET

[LAMBDA (X Y PARTNERHOST PARTNERNET PARTNERSOCKET) ; Edited 24-Nov-86 14:41 by smL

;; Make sure the cursor pos in the packet is smashed to zero, and that the packet has actually been sent

NIL))

(PUT-CACHED-PACKET

[LAMBDA (PACKET X Y PARTNERHOST PARTNERNET PARTNERSOCKET) ; Edited 24-Nov-86 13:28 by smL

T))

)

::: Low level packet exchange code

```
(DECLARE%: EVAL@COMPILE
(RPAQQ COMM.BAND.PACKET.TYPE 1321)
(RPAQQ COMM.CURSOR.PACKET.TYPE 2925)
(RPAQQ COMM.CURSOR.CLOSE.PACKET.TYPE 2926)
(RPAQQ COMM.SHUT.DOWN.PACKET.TYPE 4246)
(CONSTANTS COMM.BAND.PACKET.TYPE COMM.CURSOR.PACKET.TYPE COMM.CURSOR.CLOSE.PACKET.TYPE
COMM.SHUT.DOWN.PACKET.TYPE)
)
```

(CL:DEFCONSTANT **MAX-PACKET-BITS** (CL:\* 400 8))

(DECLARE%: EVAL@COMPILE

```
[ACCESSFNS COMM.XFER.PACKET ((COMPACT (fetch (XIP XIPCONTENTS) of DATUM))
(BLOCKRECORD COMPACT ((SPREAD WORD)
(PACKET-TYPE WORD)
(CURSORS WORD)
(CURSORY WORD)
(DATA WORD)
(DATA WORD)
(THISWIDTH WORD)
(THISHEIGHT WORD)
(DATALOC 64 WORD)))
(ACCESSFNS COMM.XFER.PACKET ((BITS (LOC (FETCH (COMM.XFER.PACKET DATALOC) OF DATUM)
)
```

::: Packing and unpacking bitmaps into etherpackets

(DEFINEQ

**(BMTOPACKET**

```
[LAMBDA (BBT SOURCEBITMAP SOURCELEFT SOURCEBOTTOM WIDTH HEIGHT PACKETLOC SPREAD)
; Edited 24-Nov-86 10:48 by sml
```

:: copy bitmap to packet

```
(CL:ASSERT (AND (BITMAP SOURCEBITMAP)
(<= 0 SOURCELEFT (- (BITMAPWIDTH SOURCEBITMAP)
WIDTH 1))
(<= 0 SOURCEBOTTOM (- (BITMAPHEIGHT SOURCEBITMAP)
HEIGHT 1))
(< 0 WIDTH)
(< 0 HEIGHT)))
(PILOTBITBLT (create PILOTBBT
smashing (OR BBT (create PILOTBBT))
PBTWIDTH _ WIDTH PBTHEIGHT _ HEIGHT PBTFLAGS _ 0 PBTDESTBPL _ WIDTH PBTDESTBIT _ 0
PBTUSEGRAY _ NIL PBTSOURCEBPL _ (CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of SOURCEBITMAP
)
)
16
(+ SPREAD 1))
PBTSOURCEBIT _ SOURCELEFT PBTDISJOINT _ T PBTSOURCE _
(\ADDBASE (fetch (BITMAP BITMAPBASE) of SOURCEBITMAP)
(CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of SOURCEBITMAP)
(- (fetch (BITMAP BITMAPHEIGHT) of SOURCEBITMAP)
HEIGHT SOURCEBOTTOM)))
PBTDEST _ PACKETLOC PBTOPERATION _ 0 PBTSOURCE TYPE _ 0)
0))
```

**(PACKETTOBM**

```
[LAMBDA (BBT PACKETLOC WIDTH HEIGHT DESTBITMAP DESTLEFT DESTBOTTOM SPREAD)
; Edited 24-Nov-86 10:48 by sml
```

:: Do a bitblt from a packet into a bitmap. Inverts BMTOPACKET.

```
(CL:ASSERT (AND (BITMAP DESTBITMAP)
(<= 0 DESTLEFT (- (BITMAPWIDTH DESTBITMAP)
WIDTH -1))
(<= 0 DESTBOTTOM (- (BITMAPHEIGHT DESTBITMAP)
(CL:* HEIGHT (CL:1+ SPREAD))
-1))
(< 0 WIDTH)
(< 0 HEIGHT)))
(PILOTBITBLT (create PILOTBBT
smashing (OR BBT (create PILOTBBT))
PBTWIDTH _ WIDTH PBTHEIGHT _ HEIGHT PBTFLAGS _ 0 PBTDESTBPL _
(CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of DESTBITMAP)
)
16
(CL:1+ SPREAD))
```

```

PBTDESTBIT _ DESTLEFT PBTUSEGRAY _ NIL PBTSOURCEBPL _ WIDTH PBTSOURCEBIT _ 0
PBTDISJOINT _ T PBTSOURCE _ PACKETLOC PBTDEST _
(\ADDBASE (fetch (BITMAP BITMAPBASE) of DESTBITMAP)
  (CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of DESTBITMAP)
    (- (fetch (BITMAP BITMAPHEIGHT) of DESTBITMAP)
      HEIGHT DESTBOTTOM)))
PBTOPERATION _ 0 PBTSOURCECTYPE _ 0)

```

0])

)

;;; Displaying the viewing machine's cursor

```
(RPAQQ REMOTE-CURSOR )
```

```
(RPAQ? CURSORICON NIL)
```

;;; Manipulating the frame that outlines the region being viewed

```
(RPAQ? *FRAME-SHADE* GRAYSHADE)
```

(DEFINEQ

**(FRAME-EVENT**

```

[LAMBDA (WINDOW) ; Edited 2-Apr-87 16:53 by Masinter
  (if (KEYDOWNP 'RIGHT)
    then (CLOSE-FRAME (WINDOWPROP WINDOW 'FRAME))
    else (LET [(FRAME (WINDOWPROP WINDOW 'FRAME)
              (if (SHIFTDOWNP 'SHIFT)
                  then [SHAPE-FRAME FRAME (LET [(REGION (WINDOWPROP WINDOW 'FRAME-REGION)
                                                       (with REGION REGION (\SETCURSORPOSITION LEFT BOTTOM)
                                                       (GETREGION 32 32 REGION NIL NIL)
                                                       (MOVE-FRAME WINDOW])

```

**(MAKE-FRAME**

```

[LAMBDA (REGION VIEWER-NAME) ; Edited 2-Apr-87 16:46 by Masinter
  (LET (FRAME)
    [with REGION REGION (SETQ FRAME (LIST (CREATEW (LIST (- LEFT 8)
                                                       (- BOTTOM 8)
                                                       8
                                                       (+ HEIGHT 8 8))
                                                       NIL 0)
                                           (CREATEW (LIST LEFT (- BOTTOM 8)
                                                       (+ WIDTH 8)
                                                       8)
                                                       NIL 0)
                                           (CREATEW (LIST (+ LEFT WIDTH)
                                                       BOTTOM 8 (+ HEIGHT 8))
                                                       NIL 0)
                                           (CREATEW (LIST LEFT (+ BOTTOM HEIGHT)
                                                       WIDTH
                                                       (HEIGHTIFWINDOW 8 T 0)
                                                       "Viewed region" 0)
    (for X in FRAME do (DSPTEXTURE *FRAME-SHADE* X)
                      (DSPRESET X)
                      (WINDOWPROP X 'FRAME-REGION REGION)
                      (WINDOWPROP X 'MINISIZE '(8 . 8))
                      (WINDOWPROP X 'FRAME FRAME)
                      (WINDOWPROP X 'RIGHTBUTTONFN (FUNCTION FRAME-EVENT))
                      (WINDOWPROP X 'BUTTONEVENTFN (FUNCTION FRAME-EVENT)))
    FRAME])

```

**(MOVE-FRAME**

```

[LAMBDA (W) ; (* lmm "17-Nov-86 02:11")
  (with REGION (WINDOWPROP W 'FRAME-REGION)
    (SHAPE-FRAME (WINDOWPROP W 'FRAME)
      (GETBOXREGION WIDTH HEIGHT LEFT BOTTOM]))

```

**(SHAPE-FRAME**

```

[LAMBDA (FRAME REGION) ; Edited 24-Nov-86 13:23 by sml
  [with REGION REGION (PROGN (SHAPEW (CAR FRAME)
                                     (LIST (- LEFT 8)
                                             (- BOTTOM 8)
                                             8
                                             (+ HEIGHT 8 8)))
    (SHAPEW (CADR FRAME)
      (LIST LEFT (- BOTTOM 8)
            (+ WIDTH 8)
            8))
    (SHAPEW (CADDR FRAME)
      (LIST (+ LEFT WIDTH)

```



```

      BOTTOM 8 (+ HEIGHT 8)))
    (SHAPEW (CADDR FRAME)
      (LIST LEFT (+ BOTTOM HEIGHT)
        WIDTH
        (HEIGHTIFWINDOW 8 (WINDOWPROP (CADDR FRAME)
          'TITLE)
          (WINDOWPROP (CADDR FRAME)
            'BORDER])

```

```

    (for X in FRAME do (CLEARW X)
      (WINDOWPROP X 'FRAME-REGION REGION))

```

**(SET-FRAME-TITLE**

```

[LAMBDA (FRAME TITLE)
  (WINDOWPROP (CAR (LAST FRAME))
    'TITLE TITLE))

```

; Edited 24-Nov-86 13:07 by sml

)

:: Changing the system parameters

(DEFINEQ

**(MAKE-MENUS-WINDOW**

```

[LAMBDA (MENUS TITLE POSITION)

```

; Edited 24-Nov-86 10:40 by sml

:: Make sure all the menu fields are filled in and up to date

```

(for MENU in MENUS do (UPDATE/MENU/IMAGE MENU))

```

:: Create a window big enough to hold all the menus, and put the menus in it

```

(LET* ((MENU-GAP 5)
  [INSIDE-WINDOW-WIDTH (PLUS MENU-GAP (for MENU in MENUS sum (PLUS MENU-GAP (fetch (MENU IMAGEWIDTH)
    of MENU)
  [INSIDE-WINDOW-HEIGHT (PLUS MENU-GAP MENU-GAP (for MENU in MENUS largest (fetch (MENU IMAGEHEIGHT)
    of MENU)
    finally (RETURN $$EXTREME]
  (CONTROL-WINDOW (CREATEW (if POSITION
    then (CREATEPOSITION (fetch XCOORD of POSITION)
      (fetch YCOORD of POSITION))
    else (GETBOXREGION (WIDTHIFWINDOW INSIDE-WINDOW-WIDTH)
      (HEIGHTIFWINDOW INSIDE-WINDOW-HEIGHT)
      NIL NIL NIL "Position the Mode Menu"))
    TITLE)))
  (for MENU in MENUS bind (LEFT _ MENU-GAP) do (ADDMENU MENU CONTROL-WINDOW
    (CREATEPOSITION LEFT
      (QUOTIENT (DIFFERENCE INSIDE-WINDOW-HEIGHT
        (fetch (MENU IMAGEHEIGHT)
          of MENU))
        2)))
    (add LEFT (fetch (MENU IMAGEWIDTH) of MENU)
      MENU-GAP))
  CONTROL-WINDOW])

```

**(MODE-MENU**

```

[LAMBDA NIL
  (LET ((UPDATE-MENU (create MENU
    CENTERFLG _ T
    MENUTITLEFONT _ BOLDFONT
    WHENSELECTEDFN _ [FUNCTION (LAMBDA (ITEM MENU BUTTON)
      (MNUDESELECT COMM.DEFAULT.TRANSMIT.TYPE MENU)
      (MNUSELECT ITEM MENU)
      (CHANGE-SENDER-UPDATE-MODE ITEM]
    TITLE _ "Update method"
    ITEMS _ COMM-MODES))

```

; Edited 24-Nov-86 16:52 by sml

```

[MOUSE-POS-UPDATE-MENU (create MENU
  CENTERFLG _ T
  MENUTITLEFONT _ BOLDFONT
  WHENSELECTEDFN _ [FUNCTION (LAMBDA (ITEM MENU BUTTON)
    (MNUDESELECT COMM.UPDATE.MOUSE.POSITION
      MENU)
    (MNUSELECT ITEM MENU)
    (SETQ COMM.UPDATE.MOUSE.POSITION ITEM]
  TITLE _ "Update near cursor?"
  ITEMS _ '(Sender Viewer NIL]

```

```

[SEND-UNCHANGED-TILES-MENU (create MENU
  CENTERFLG _ T
  MENUTITLEFONT _ BOLDFONT
  WHENSELECTEDFN _ [FUNCTION (LAMBDA (ITEM MENU BUTTON)
    (MNUDESELECT COMM.SEND.UNCHANGED.TILES
      MENU)
    (MNUSELECT ITEM MENU)
    (SETQ COMM.SEND.UNCHANGED.TILES ITEM]
  TITLE _ "Send unchanged tiles?"
  ITEMS _ '(T NIL]

```

```

(LIGHTNING-MENU (create MENU

```

```

CENTERFLG _ T
MENUTITLEFONT _ BOLDFONT
WHENSELECTEDFN _ [FUNCTION (LAMBDA (ITEM MENU BUTON)
                          (MENUDESELECT \ETHERLIGHTNING MENU)
                          (MENUSELECT ITEM MENU)
                          (SETQ \ETHERLIGHTNING ITEM])

TITLE _ "Ether Lightning"
ITEMS _ ' (NIL 3 6 1 4 7 2 5 8)
MENUROWS _ 3))

```

```

;;
;; Bring up a window with all the menus, at a location the user specifies
;;

```

```

(MAKE-MENUS-WINDOW (LIST UPDATE-MENU MOUSE-POS-UPDATE-MENU SEND-UNCHANGED-TILES-MENU LIGHTNING-MENU)
                  "Send-Bits mode menu")

```

```

;;
;; Highlight the current values, so the user can see what the current values are.
;;

```

```

(MENUSELECT COMM.DEFAULT.TRANSMIT.TYPE UPDATE-MENU)
(MENUSELECT COMM.UPDATE.MOUSE.POSITION MOUSE-POS-UPDATE-MENU)
(MENUSELECT COMM.SEND.UNCHANGED.TILES SEND-UNCHANGED-TILES-MENU)
(MENUSELECT \ETHERLIGHTNING LIGHTNING-MENU)

```

)

```

(RPAQQ COMM-MODES (SQUARE RECTANGLE HORIZONTAL VERTICAL H3))

```

;;; Initialization

```

(COURIER.START.SERVER)

```

;;; Unused stuff, as far as I can tell

```

(DEFINEQ

```

**(FASTBITBLT**

```

[LAMBDA (BBT SOURCEBITMAP SOURCELEFT SOURCEBOTTOM WIDTH HEIGHT DESTBITMAP DESTLEFT DESTBOTTOM)
      (* Imm "17-Nov-86 03:55")

```

;; copy bitmap to bitmap

```

(CL:ASSERT (AND (BITMAPP SOURCEBITMAP)
                (BITMAPP DESTBITMAP)
                (<= 0 SOURCELEFT (- (BITMAPWIDTH SOURCEBITMAP)
                                   WIDTH 1))
                (<= 0 SOURCEBOTTOM (- (BITMAPHEIGHT SOURCEBITMAP)
                                       HEIGHT 1))
                (<= 0 DESTLEFT (- (BITMAPWIDTH DESTBITMAP)
                                   WIDTH 1))
                (<= 0 DESTBOTTOM (- (BITMAPHEIGHT DESTBITMAP)
                                    HEIGHT 1))
                (< 0 WIDTH)
                (< 0 HEIGHT)))

```

```

(PILOTBITBLT (create PILOTBBT smashing (OR BBT (create PILOTBBT))
              PBTWIDTH _ WIDTH PBTHEIGHT _ HEIGHT PBTFLAGS _ 0 PBTDESTBPL _
              (CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of DESTBITMAP)
                    16)
              PBTDESTBIT _ DESTLEFT PBTUSEGRAY _ NIL PBTSOURCEBPL _
              (CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of SOURCEBITMAP)
                    16)
              PBTSOURCEBIT _ SOURCELEFT PBTDISJOINT _ T PBTSOURCE _
              (\ADDBASE (fetch (BITMAP BITMAPBASE) of SOURCEBITMAP)
                    (CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of SOURCEBITMAP)
                          (- (fetch (BITMAP BITMAPHEIGHT) of SOURCEBITMAP)
                             HEIGHT SOURCEBOTTOM)))
              PBTDEST _ (\ADDBASE (fetch (BITMAP BITMAPBASE) of DESTBITMAP)
                    (CL:* (fetch (BITMAP BITMAPRASTERWIDTH) of DESTBITMAP)
                          (- (fetch (BITMAP BITMAPHEIGHT) of DESTBITMAP)
                             HEIGHT DESTBOTTOM)))
              PBTOPERATION _ 0 PBTSOURCECTYPE _ 0)

```

0])

)

;;; System file dependencies

```

(DECLARE%: DONTCOPY DOEVAL@COMPILE

```

```

(FILESLoad (LOADCOMP)
           LLDISPLAY LLEther LLNS)

```

)

{MEDLEY}<lispusers>COMMWINDOW.;1

(COURIERPROGRAM **COMMWINDOW** (1337 1)

**TYPES**

[(REGION (RECORD (LEFT INTEGER)  
(BOTTOM INTEGER)  
(WIDTH INTEGER)  
(HEIGHT INTEGER)))  
(USERNAME STRING)  
(RESPONSE (RECORD (SOCKET LONGINTEGER)  
(CORRESPONDENT USERNAME])

**PROCEDURES**

((START-GET-BITS 1 (REGION USERNAME)  
RETURNS  
(RESPONSE)  
REPORTS  
(REMOTEERROR)  
IMPLEMENTEDBY START-GET-BITS))

**ERRORS**

((ERROR 1 (STRING))  
(USE.COURIER 2 NIL)))

(PUTPROPS **COMMWINDOW COPYRIGHT** ("Xerox Corporation" 1986 1900 1987))

---

**FUNCTION INDEX**

BMTOPACKET .....	7	MAKE-FRAME .....	8	SEND-BITS .....	3
CHANGE-SENDER-UPDATE-MODE .....	6	MAKE-MENUS-WINDOW .....	9	SEND-TILE .....	4
CLOSE-FRAME .....	2	MAPTILES .....	5	SET-FRAME-TITLE .....	9
FASTBITBLT .....	10	MODE-MENU .....	9	SHAPE-FRAME .....	8
FRAME-EVENT .....	8	MOVE-FRAME .....	8	SHUT-DOWN-VIEWER .....	6
GET-BITS .....	2	PACKET-EQUAL .....	6	START-GET-BITS .....	3
GET-CACHED-PACKET .....	6	PACKETTOBM .....	7		
LISTEN-TO-VIEWER .....	4	PUT-CACHED-PACKET .....	6		

---

**VARIABLE INDEX**

*FRAME-SHADE* .....	8	COMM.SEND.UNCHANGED.TILES .....	6	REMOTE-CURSOR .....	8
COMM-MODES .....	10	COMM.UPDATE.MOUSE.POSITION .....	6		
COMM.DEFAULT.TRANSMIT.TYPE .....	6	CURSORICON .....	8		

---

**CONSTANT INDEX**

COMM.BAND.PACKET.TYPE .....	7	COMM.CURSOR.PACKET.TYPE .....	7	MAX-PACKET-BITS .....	7
COMM.CURSOR.CLOSE.PACKET.TYPE .....	7	COMM.SHUT.DOWN.PACKET.TYPE .....	7		

---

**MACRO INDEX**

INCR .....	6	\PILOTBITBLT .....	6
------------	---	--------------------	---

---

**COURIERPROGRAM INDEX**

COMMWINDOW .....	11
------------------	----

---

**RECORD INDEX**

COMM.XFER.PACKET .....	7
------------------------	---

---