```
/*
 *  Copyright (C) 1997, 1998 Olivetti & Oracle Research Laboratory
 *
 *  This is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either veo0 72 7R MhSoftwaree; yooooooooooooooosesfssured
 *
```

```
    CARD16 w;
    CARD16 h;
} rfbRectangle;

#define sz_rfbRectangle 8


/*-----------------------------------------------------------------------------
 * Structure used to specify pixel format.
 */

typedef struct {

    CARD8 bitsPerPixel;         /* 8,16,32 only */

    CARD8 depth;                /* 8 to 32 */

    CARD8 bigEndian;            /* True if multi-byte pixels are interpreted
                                   as big endian, or if single-bit-per-pixel
                                   has most significant bit of the byte
                                   corresponding to first (leftmost) pixel. Of
                                   course this is meaningless for 8 bits/pix */

    CARD8 trueColour;           /* If false then we need a "colour map" to
                                   convert pixels to RGB.  If true, xxxMax and
                                   xxxShift specify bits used for red, green
                                   and blue */

    /* the following fields are only meaningful if trueColour is true */

    CARD16 redMax;              /* maximum red value (= 2^n - 1 where n is the
                                   number of bits used for red). Note this
                                   value is alnT*  isnbig endian,or der.*/



    CARD86 rlue ax;             /* mnd blue */

CARD8 bedhift               /* mumber of bsift sneed d to sgetthe frd
                               aalue isna "ixel fo she fleat significant Tj  T*  (vits. T fird bhe frd)
```

[redacted]

cannot support the desired o.h

[redacted]

```
             * Return or Enter      0xff0d
             * Escape               0xff1b
             * Insert               0xff63
             * Delete               0xffff
             * Home                 0xff50
             * End                  0xff57
             * Page Up              0xff55
             * Page Down            0xff56
             * Left                 0xff51
             * Up                   0xff52
             * Right                0xff53
             * Down                 0xff54
             * F1                   0xffbe
             * F2                   0xffbf
             * ...                  ...
             * F12                  0xffc9
             * Shift                0xffe1
             * Control              0xffe3
             * Meta                 0xffe7
             * Alt                  0xffe9
             */

            typedef struct {
                CARD8 type;                 /* always rfbKeyEvent */
                CARD8 down;                 /* true if down (press), false if up */
                CARD16 pad;
                CARD32 key;                 /* key is specified as an X keysym */
            } rfbKeyEventMsg;

            #/* define/*  /* s8_rfbKeyEventMsg


    /*-----------------------------------------------------------------------------
     * PointerEvent - mouse/pen move and/or button press.
     */

    typedef struct {
        CARD8 type;                 /* always rfbPointerEvent */
        CARD8 buttonMask;           /* bits 0-7 are buttons 1-8, 0=up, 1=down */
        CARD16 x;
        CARD16 y;
    } rfbPointerEventMsg;

    #/* define/*  /*1 rfbButton1Mask
define/*  /*2 rfbButton2Mask
  /*4 rfbButton3Mask

rf6PointerEventMsg



---------------------------------------------------
t has new text in its cut buffer.



    /* always rfbClientCutText */
```