NAME

XF86VidModeQueryExtension, XF86VidModeQueryVersion, XF86VidModeSetClientVersion, XF86VidModeGetModeLine, XF86VidModeGetAllModeLines, XF86VidModeAddModeLine, XF86VidModeDeleteModeLine, XF86VidModeModModeLine, XF86VidModeValidateModeLine, XF86VidModeSwitchMode, XF86VidModeSwitchToMode, XF86VidModeLockModeSwitch, XF86VidModeGetMonitor, XF86VidModeGetViewPort, XF86VidModeSetViewPort, XF86VidModeGetDotClocks, XF86VidModeGetGamma, XF86VidModeSetGamma, XF86VidModeGetGammaRamp, XF86VidModeSetGammaRamp, XF86VidModeGetGammaRampSize, XF86VidModeGetPermissions - Extension library for the XFree86-VidMode X extension

SYNOPSIS

```
#include <X11/extensions/xf86vmode.h>

Bool XF86VidModeQueryExtension(
    Display *display,
    int *event_base_return,
    int *error_base_return);

Bool XF86VidModeQueryVersion(
    Display *display,
    int *major_version_return,
    int *minor_version_return);

Bool XF86VidModeSetClientVersion(
    Display *display);

Bool XF86VidModeGetModeLine(
    Display *display,
    int screen,
    int *dotclock_return,
    XF86VidModeModeLine *modeline);

Bool XF86VidModeGetAllModeLines(
    Display *display,
    int screen,
    int *modecount_return,
    XF86VidModeModeInfo ***modesinfo);
```

Bool XF86VidModeAddModeLine(
Display *_display_,
int _screen_,
XF86VidModeModeInfo *_modeline_
XF86VidModeModeInfo *_aftermode_);

Bool XF86VidModeDeleteModeLine(
Display *_display_,
int _screen_,
XF86VidModeModeInfo *_modeline_);

Bool XF86VidModeModModeLine(
Display *_display_,
int _screen_,
XF86VidModeModeLine *_modeline_);

Status XF86VidModeValidateModeLine(
Display *_display_,
int _screen_,
XF86VidModeModeLine *_modeline_);

Bool XF86VidModeSwitchMode(
Display *_display_,
int _screen_,
int _zoom_);

Bool XF86VidModeSwitchToMode(
Display *_display_,
int _screen_,
XF86VidModeModeInfo *_modeline_);

Bool XF86VidModeLockModeSwitch(
Display *_display_,
int _screen_,
int _lock_);

Bool XF86VidModeGetMonitor(
Display *_display_,
int _screen_,
XF86VidModeMonitor *_monitor_);

Bool XF86VidModeGetViewPort(
Display *display,
int screen,
int *x_return,
int *y_return);

Bool XF86VidModeSetViewPort(
Display *display,
int screen,
int x,
int y);

XF86VidModeGetDotClocks(
Display *display,
int screen,
int *flags return,
int *number of clocks return,
int *max dot clock return,
int **clocks return);

XF86VidModeGetGamma(
Display *display,
int screen,
XF86VidModeGamma *Gamma);

XF86VidModeSetGamma(
Display *display,
int screen,
XF86VidModeGamma *Gamma);

XF86VidModeGetGammaRamp(
Display *display,
int screen,
int size,
unsigned short *red array,
unsigned short *green array,
unsigned short *blue array);

XF86VidModeSetGammaRamp(
Display *display,

int *screen*,
int *size*,
unsigned short **red array*,
unsigned short **green array*,
unsigned short **blue array*);

XF86VidModeGetGammaRampSize(
Display **display*,
int *screen*,
int **size*);

ARGUMENTS

display	Specifies the connection to the X server.
screen	Specifies which screen number the setting apply to.
event_base_return	Returns the base event number for the extension.
error_base_return	Returns the base error number for the extension.
major_version_return	Returns the major version number of the extension.
minor_version_return	Returns the minor version number of the extension.
dotclock_return	Returns the clock for the mode line.
modecount_return	Returns the number of video modes available in the server.
zoom	If greater than zero, indicates that the server should switch to the next mode, otherwise switch to the previous mode.
lock	Indicates that mode switching should be locked, if non-zero.
modeline	Specifies or returns the timing values for a video mode.
aftermode	Specifies the timing values for the video mode after which the new mode will added.
modesinfo	Returns the timing values and dotclocks for all of the available video modes.

monitor Returns information about the monitor.

x Specifies the desired X location for the viewport.

x_return Returns the current X location of the viewport.

y Specifies the desired Y location for the viewport.

y_return Returns the current Y location of the viewport.

STRUCTURES

Video Mode Settings:

```
typedef struct {
    unsigned short      hdisplay;       /* Number of display pixels horizontally */
    unsigned short      hsyncstart;     /* Horizontal sync start */
    unsigned short      hsyncend;       /* Horizontal sync end */
    unsigned short      htotal;         /* Total horizontal pixels */
    unsigned short      vdisplay;       /* Number of display pixels vertically */
    unsigned short      vsyncstart;     /* Vertical sync start */
    unsigned short      vsyncend;       /* Vertical sync start */
    unsigned short      vtotal;         /* Total vertical pixels */
    unsigned int        flags;          /* Mode flags */
    int privsize;                       /* Size of private */
    INT32       *private;               /* Server privates */
} XF86VidModeModeLine;

typedef struct {
    unsigned int        dotclock;       /* Pixel clock */
    unsigned short      hdisplay;       /* Number of display pixels horizontally */
    unsigned short      hsyncstart;     /* Horizontal sync start */
    unsigned short      hsyncend;       /* Horizontal sync end */
    unsigned short      htotal;         /* Total horizontal pixels */
    unsigned short      vdisplay;       /* Number of display pixels vertically */
    unsigned short      vsyncstart;     /* Vertical sync start */
    unsigned short      vsyncend;       /* Vertical sync start */
    unsigned short      vtotal;         /* Total vertical pixels */
    unsigned int        flags;          /* Mode flags */
    int privsize;                       /* Size of private */
    INT32       *private;               /* Server privates */
} XF86VidModeModeInfo;
```

Monitor information:

```
typedef struct {
    char*       vendor;             /* Name of manufacturer */
    char*       model;              /* Model name */
    float       EMPTY;              /* unused, for backward compatibility */
    unsigned char nhsync;           /* Number of horiz sync ranges */
    XF86VidModeSyncRange* hsync;    /* Horizontal sync ranges */
    unsigned char nvsync;           /* Number of vert sync ranges */
    XF86VidModeSyncRange* vsync;    /* Vertical sync ranges */
} XF86VidModeMonitor;

typedef struct {
    float       hi;     /* Top of range */
    float       lo;     /* Bottom of range */
} XF86VidModeSyncRange;

typedef struct {
    int type;               /* of event */
    unsigned long serial;       /* # of last request processed by server */
    Bool send_event;            /* true if this came from a SendEvent req */
    Display *display;           /* Display the event was read from */
    Window root;                /* root window of event screen */
    int state;                  /* What happened */
    int kind;                   /* What happened */
    Bool forced;                /* extents of new region */
    Time time;                  /* event timestamp */
} XF86VidModeNotifyEvent;

typedef struct {
    float red;              /* Red Gamma value */
    float green;            /* Green Gamma value */
    float blue;             /* Blue Gamma value */
} XF86VidModeGamma;
```

DESCRIPTION

These functions provide an interface to the server extension *XFree86-VidModeExtension* which allows the video modes to be queried and adjusted dynamically and mode switching to be controlled. Applications that use these functions must be linked with *-lXxf86vm*

MODELINE FUNCTIONS

The *XF86VidModeGetModeLine* function is used to query the settings for the currently selected video mode. The calling program should pass a pointer to a *XF86VidModeModeLine* structure that it has already allocated. The function fills in the fields of the structure.

If there are any server private values (currently only applicable to the S3 server) the function will allocate storage for them. Therefore, if the *privsize* field is non-zero, the calling program should call *Xfree(private)* to free the storage.

XF86VidModeGetAllModeLines returns the settings for all video modes. The calling program supplies the address of a pointer which will be set by the function to point to an array of *XF86VidModeModeInfo* structures. The memory occupied by the array is dynamically allocated by the *XF86VidModeGetAllModeLines* function and should be freed by the caller. The first element of the array corresponds to the current video mode.

The *XF86VidModeModModeLine* function can be used to change the settings of the current video mode provided the requested settings are valid (e.g. they don't exceed the capabilities of the monitor).

To add a mode to the list of available modes, the *XF86VidModeAddModeLine* function can be used. Assuming the settings are valid, the video mode will be added after the existing mode which matches the timings specified by the *aftermode* parameter. To be considered a match, all of the fields of the given *XF86VidModeModeInfo* structure must match, except the *privsize* and *private* fields. If the *aftermode* parameter is zero, the mode will be added after the current mode.

Modes can be deleted with the *XF86VidModeDeleteModeLine* function. The specified mode must match an existing mode. To be considered a match, all of the fields of the given *XF86VidModeModeInfo* structure must match, except the *privsize* and *private* fields. If the mode to be deleted is the current mode, a mode switch to the next mode will occur first. The last remaining mode can not be deleted.

The validity of a mode can be checked with the *XF86VidModeValidateModeLine* function. If the specified mode can be used by the server (i.e. meets all the constraints placed upon a mode by the combination of the server, card, and monitor) the function returns *MODE_OK*, otherwise it returns a value indicating the reason why the mode is invalid (as defined in *xf86.h*)

MODE SWITCH FUNCTIONS

When the function *XF86VidModeSwitchMode* is called, the server will change the video mode to next (or previous) video mode. The *XF86VidModeSwitchToMode* function can be used to switch directly to the specified mode. Matching is as specified in the description of the *XF86VidModeAddModeLine* function above. The *XF86VidModeLockModeSwitch* function can be used to allow or disallow mode switching whether the request to switch modes comes from a call to the *XF86VidModeSwitchMode* or

XF86VidModeSwitchToMode functions or from one of the mode switch key sequences.

Note: Because of the asynchronous nature of the X protocol, a call to *XFlush* is needed if the application wants to see the mode change immediately. To be informed of the execution status of the request, a custom error handler should be installed using *XSetErrorHandler* before calling the mode switching function.

MONITOR FUNCTIONS

Information known to the server about the monitor is returned by the *XF86VidModeGetMonitor* function. The *hsync* and *vsync* fields each point to an array of *XF86VidModeSyncRange* structures. The arrays contain *nhsync* and *nvsync* elements, respectively. The *hi* and *low* values will be equal if a discreate value was given in the *XF86Config* file.

The *vendor*, *model*, *hsync*, and *vsync* fields point to dynamically allocated storage that should be freed by the caller.

VIEWPORT FUNCTIONS

The *XF86VidModeGetViewPort* and *XF86VidModeSetViewPort* functions can be used to, respectively, query and change the location of the upper left corner of the viewport into the virtual screen.

OTHER FUNCTIONS

The *XF86VidModeQueryVersion* function can be used to determine the version of the extension built into the server.

The function *XF86VidModeQueryExtension* returns the lowest numbered error and event values assigned to the extension.

BUGS

The XF86VidModeSetClientVersion, XF86VidModeGetDotClocks, XF86VidModeGetGamma, XF86VidModeSetGamma, XF86VidModeSetGammaRamp, XF86VidModeGetGammaRamp, XF86VidModeGetGammaRampSize, and XF86VidModeGetPermissions functions need to be documented. In the meantime, check the source code for information about how to use them.

SEE ALSO

Xorg(1), xorg.conf(5), XFlush(3), XSetErrorHandler(3), xvidtune(1)

AUTHORS

Kaleb Keithley, Jon Tombs, David Dawes, and Joe Moss