NAME

Xcomposite - X Composite Extension library

SYNOPSIS

#include <X11/extensions/Xcomposite.h>

Bool XCompositeQueryExtension(Display ****dpy*, int ****event_base_return***, int ****error_base_return***);**

Status XCompositeQueryVersion(Display ****dpy*, int ****major_version_return***,**
int ****minor_version_return*);**

int XCompositeVersion(void);

void XCompositeRedirectWindow(Display ****dpy*, Window** *window***, int** *update***);**

void XCompositeRedirectSubwindows(Display ****dpy*, Window** *window***, int** *update***);**

void XCompositeUnredirectWindow(Display ****dpy*, Window** *window***, int** *update***);**

void XCompositeUnredirectSubwindows(Display ****dpy*, Window** *window***, int** *update***);**

XserverRegion XCompositeCreateRegionFromBorderClip(Display ****dpy*, Window** *window***);**

Pixmap XCompositeNameWindowPixmap(Display ****dpy*, Window** *window***);**

Window XCompositeGetOverlayWindow(Display ****dpy*, Window** *window***);**

void XCompositeReleaseOverlayWindow(Display ****dpy*, Window** *window***);**

DESCRIPTION

The composite extension provides several related mechanisms:

Per-hierarchy storage

The rendering of an entire hierarchy of windows is redirected to off-screen storage. The pixels of that hierarchy are available whenever it is viewable. Storage is automatically reallocated when the top level window changes size. Contents beyond the geometry of the top window are not preserved.

Automatic shadow update

When a hierarchy is rendered off-screen, the X server provides an automatic mechanism for

presenting those contents within the parent window. The implementation is free to make this update lag behind actual rendering operations by an unspecified amount of time. This automatic update mechanism may be disabled so that the parent window contents can be completely determined by an external application.

Composite Overlay Window

Version 0.3 of the protocol adds the Composite Overlay Window, which provides compositing managers with a surface on which to draw without interference. This window is always above normal windows and is always below the screen saver window. It is an InputOutput window whose width and height are the screen dimensions. Its visual is the root visual and its border width is zero. Attempts to redirect it using the composite extension are ignored. This window does not appear in the reply of the QueryTree request. It is also an override redirect window. These last two features make it invisible to window managers and other X11 clients. The only way to access the XID of this window is via the CompositeGetOverlayWindow request. Initially, the Composite Overlay Window is unmapped.

Parent window clipping

Version 0.4 of the protocol modifies the semantics of parent window clipping in the presence of manual redirected children. With this version, the area in the parent covered by manual redirected children is left in the parent clip list instead of being removed as in older versions.

Per-hierarchy storage may be created for individual windows or for all children of a window. Manual shadow update may be selected by only a single application for each window; manual update may also be selected on a per-window basis or for each child of a window. Detecting when to update may be done with the Damage extension.

The off-screen storage includes the window contents, its borders and the contents of all descendants.

ARGUMENTS

display

Pointer to the **Display** structure returned from **XOpenDisplay** for the connection to the X server.

event_base_return

Pointer to integer where the base value for Composite Extension events will be stored.

error_base_return

Pointer to integer where the base value for Composite Extension errors will be stored.

major_version_return

Pointer to integer where the major version of the Composite Extension supported by the X server

will be stored.

minor_version_return
Pointer to integer where the minor version of the Composite Extension supported by the X server will be stored.

window
Specifies the window ID to operate on.

update
Specifies the mode for updating the window contents. Must be either **CompositeRedirectAutomatic** or **CompositeRedirectManual**.

FUNCTIONS

XCompositeQueryExtension

XCompositeQueryExtension determines if the Composite Extension is available on the given display. It returns **True** if the extension is supported, otherwise **False**. If the extension is present, the base values for events and errors are returned, and can be used to decode incoming event and error values.

XCompositeQueryVersion

XCompositeQueryVersion determines if the X Server supports a version of the X Composite Extension which is compatible with the client library. A non-zero Status is returned if a compatible version of the extension is supported, otherwise a zero Status is returned. If the extension is supported, the major and minor version numbers are returned to indicate the level of Composite Extension support. No other XComposite functions (except XCompositeQueryExtension) may be called before this function. If a client violates this rule, the effects of all subsequent XComposite calls that it makes are undefined.

XCompositeVersion

XCompositeVersion returns the version of the X Composite library. The version number is encoded as: (major * 10000) + (minor * 100) + revision

For example, version 1.4.6 would be encoded as the integer 10406.

XCompositeRedirectWindow

XCompositeRedirectWindow requests the X server to direct the hierarchy starting at *window* to off-screen storage. The *update* argument specifies whether the contents are mirrored to the parent window automatically or not. Only one client at a time may specify an update type of **CompositeRedirectManual**, another attempt will result in a BadAccess error. When all clients enabling redirection terminate, the redirection will automatically be disabled.

The root window may not be redirected. Doing so results in a BadMatch error. Specifying an invalid window id will result in a BadWindow error.

XCompositeRedirectSubwindows

XCompositeRedirectSubwindows requests the X server to redirect hierarchies starting at all current and future children of *window* as in **XCompositeRedirectWindow**. If *update* is **CompositeRedirectManual**, then painting of the window background during window manipulation and ClearArea requests is inhibited.

XCompositeUnredirectWindow

XCompositeUnredirectWindow requests the X server to terminate redirection of *window*. If the specified window was not selected for redirection by the current client, a BadValue error results.

XCompositeUnredirectSubwindows

XCompositeUnredirectWindow requests the X server to terminate redirection of all children of *window*. If the specified window was not selected for sub-redirection by the current client, a BadValue error results.

XCompositeCreateRegionFromBorderClip

XCompositeCreateRegionFromBorderClip creates a region containing the "usual" border clip value; that is the area of the window clipped against siblings and the parent. This region can be used to restrict rendering to suitable areas while updating only a single window. The region is copied at the moment the request is executed; future changes to the window hierarchy will not be reflected in this region.

XCompositeNameWindowPixmap

XCompositeNameWindowPixmap creates and returns a pixmap id that serves as a reference to the off-screen storage for *window*. This pixmap will remain allocated until freed, even if the window is unmapped, reconfigured or destroyed. However, the window will get a new pixmap allocated

each time it is mapped or resized, so this function will need to be reinvoked for the client to continue to refer to the storage holding the current window contents. Generates a BadMatch error if *window* is not redirected or is not visible.

The X server must support at least version 0.2 of the Composite Extension for **XCompositeNameWindowPixmap**.

XCompositeGetOverlayWindow

XCompositeGetOverlayWindow returns the window ID of the Composite Overlay Window for the screen specified by the argument *window*. This function notifies the X server that the client wishes to use the Composite Overlay Window of this screen. If this Composite Overlay Window has not yet been mapped, it is mapped by this request.

The Composite Overlay Window for a particular screen will be unmapped when all clients who have called this function have either called **XCompositeReleaseOverlayWindow** for that screen, or terminated their connection to the X server.

The X server must support at least version 0.3 of the Composite Extension for **XCompositeGetOverlayWindow**.

XCompositeReleaseOverlayWindow

This request specifies that the client is no longer using the Composite Overlay Window on the screen specified by the argument *window*. A screen's Composite Overlay Window is unmapped when there are no longer any clients using it.

The X server must support at least version 0.3 of the Composite Extension for **XCompositeReleaseOverlayWindow**.

AUTHORS

Keith Packard <keithp@keithp.com>

Extension specification and implementation

Deron Johnson <deron.johnson@sun.com>

Overlay Window specification and implementation