Subject: Re: How to find the top-most window the mouse is over (Linux)?
Posted by James Thomas on Wed, 24 Jan 2007 11:25:13 GMT
View Forum Message <> Reply to Message

Thanks, but unfortunately I've tried those and they don't solve my problem. I should have been more specific.

I'm implementing dragging of data within the app between different windows/ctrls and when dragging is in progress I want the mouse pointer set to an icon indicating the type of data, even when over a ctrl that has no drag-drop interface. The only way (I know of) to achieve this is by overloading CursorImage on the source control and using SetCapture to route all CursorImage events to it. This means that GetMouseCtrl (or a mouse hook) will always return the source ctrl and not the one the mouse is actually over.

I have also tried this using vaious other methods, but I like this implementation for two further reasons:
1) While dragging all mouse events can be caught by the drag aware control, which avoids certain 'state' problems.
For instance if the user starts a drag with the left mouse button down and then releases the button while the mouse is outside any application window (where there is no way to get notified of the event) the app still needs to cancel the drag-drop process.
2) It provides a nice interface between the 'source' and 'target' controls.

However, this means I need a way of determining which control is under the mouse when the left mouse button is released even when it is not actually recieving the event. I accomplish this in Windows using the API call above and everything works perfectly, but I am unable to make the XWindows API work similarly.

Am I going about this wrong? Or perhaps there is a better way of implementing this that I haven't thought of? For instance, if there was a way to either hook onto CursorImage events or set the default mouse pointer (as far I can tell this currently impossible - Image::Arrow() is specified directly all over the place) it would solve one my problems and I could work around the rest.

I hope this makes sense. If I have time today I might make a test app to illustrate the problem.