Subject: Re: Toggling between FullScreen and not
Posted by mirek on Wed, 27 Feb 2008 11:49:16 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 27 February 2008 04:29I downloaded the upp-svn_SVN.165_i386.deb, but had no luck. Note that going to the fullscreen and back to normal size is not the problem but opening a context menu while in fullscreen mode is. I.e. the context menu does not pop up when in fullscreen mode although the context menu works correctly when window is in its normal size.

// Tom

Well, after some investigation, I have fixed all X11 issues by changing the fullscreen implementation from popup to using _NET_WM_STATE_FULLSCREEN.

Now it seems to work much better

Please try:

```
void TopWindow::SyncCaption()
{
 LLOG("SyncCaption");
 SyncTitle();
 if(IsOpen() && GetWindow()) {
  unsigned long wina[6];
  int n = 0;
  Window w = GetWindow();
  if(tool)
   wina[n++] = XAtom("_NET_WM_WINDOW_TYPE_TOOLBAR");
  if(GetOwner())
   wina[n++] = XAtom("_NET_WM_WINDOW_TYPE_DIALOG");
  wina[n++] = XAtom("_NET_WM_WINDOW_TYPE_NORMAL");
  XChangeProperty(Xdisplay, GetWindow(), XAtom("_NET_WM_WINDOW_TYPE"),
XAtom("ATOM"), 32,
                  PropModeReplace, (const unsigned char *)wina, n);
  n = 0;
  if(topmost)
   wina[n++] = XAtom("_NET_WM_STATE_ABOVE");
  if(state == MAXIMIZED) {
   wina[n++] = XAtom("_NET_WM_STATE_MAXIMIZED_HORZ");
   wina[n++] = XAtom("_NET_WM_STATE_MAXIMIZED_VERT");
  }
  if(fullscreen)
   wina[n++] = XAtom("_NET_WM_STATE_FULLSCREEN");
  XChangeProperty(Xdisplay, GetWindow(), XAtom("_NET_WM_STATE"), XAtom("ATOM"), 32,
                  PropModeReplace, (const unsigned char *)wina, n);
  wm_hints->flags = InputHint|WindowGroupHint|StateHint;
  wm_hints->initial_state = NormalState;
```

```
  wm_hints->input = XTrue;
  Ctrl *owner = GetOwner();
  wm_hints->window_group = owner ? owner->GetWindow() : w;
  if(!icon.IsEmpty()) {
   Size isz = icon.GetSize();
   int len = 2 + isz.cx * isz.cy;
   Buffer<unsigned long> data(len);
   unsigned long *t = data;
   *t++ = isz.cx;
   *t++ = isz.cy;
   for(int y = 0; y < isz.cy; y++) {
    const RGBA *q = icon[y];
    for(int x = isz.cx; x--;) {
     *t++ = ((dword)q->a << 24) |
           (dword)q->b | ((dword)q->g << 8) | ((dword)q->r << 16);
     q++;
    }
   }
   XChangeProperty(Xdisplay, w, XAtom("_NET_WM_ICON"), XA_CARDINAL, 32,
PropModeReplace,
                   (const unsigned char *)~data, len);
  }
  XSetWMHints(Xdisplay, w, wm_hints);
 }
}

void TopWindow::Open(Ctrl *owner)
{
 if(dokeys && (!GUI_AKD_Conservative() || GetAccessKeysDeep() <= 1))
  DistributeAccessKeys();
 UsrLogT(3, "OPEN " + Desc(this));
 LLOG("OPEN " << Name() << " owner: " << UPP::Name(owner));
 IgnoreMouseUp();
 if(fullscreen)
  SetRect(0, 0, Xwidth, Xheight);
 else
  CenterRect(owner);
 LLOG("Open NextRequest " << NextRequest(Xdisplay));
 Create(owner, false, false);
 xminsize.cx = xmaxsize.cx = Null;
 title2.Clear();
 LLOG("SyncCaption");
 SyncCaption();
 LLOG("SyncSizeHints");
 size_hints->flags = 0;
 SyncSizeHints();
 Rect r = GetRect();
 size_hints->x = r.left;
```

```
size_hints->y = r.top;
size_hints->width = r.Width();
size_hints->height = r.Height();
size_hints->win_gravity = StaticGravity;
size_hints->flags |= PPosition|PSize|PWinGravity;
if(owner) {
 ASSERT(owner->IsOpen());
 LLOG("XSetTransientForHint");
 XSetTransientForHint(Xdisplay, GetWindow(), owner->GetWindow());
}
LLOG("XSetWMNormalHints");
XSetWMNormalHints(Xdisplay, GetWindow(), size_hints);
Atom protocols[2];
protocols[0] = XAtom("WM_DELETE_WINDOW");
protocols[1] = XAtom("WM_TAKE_FOCUS");
LLOG("XSetWMProtocols");
XSetWMProtocols(Xdisplay, GetWindow(), protocols, 2);
String x = GetExeTitle().ToString();
const char *progname = ~x;
class_hint->res_name = (char *)progname;
class_hint->res_class = (char *)progname;
XSetClassHint(Xdisplay, GetWindow(), class_hint);
LLOG("WndShow(" << visible << ")");
WndShow(visible);
if(visible) {
 XEvent e;
 LLOG("XWindowEvent");
 XWindowEvent(Xdisplay, top->window, VisibilityChangeMask, &e);
 ignoretakefocus = true;
 SetTimeCallback(500, THISBACK(EndIgnoreTakeFocus));
 LLOG("SetWndFocus");
 SetWndFocus();
 for(int i = 0; i < 50; i++) {
  if(XCheckTypedWindowEvent(Xdisplay, top->window, FocusIn, &e)) {
   ProcessEvent(&e);
   if(e.xfocus.window = top->window)
    break;
  }
  Sleep(10);
 }
}
LLOG(">Open NextRequest " << NextRequest(Xdisplay));
LLOG(">OPENED " << Name());
PlaceFocus();
StateH(OPEN);
Vector<int> fe = GetPropertyInts(top->window, XAtom("_NET_FRAME_EXTENTS"));
if(fe.GetCount() >= 4 &&
   fe[0] >= 0 && fe[0] <= 16 && fe[1] >= 0 && fe[1] <= 16 && //fluxbox returns wrong numbers -
```

```
quick&dirty workaround
    fe[2] >= 0 && fe[2] <= 64 && fe[3] >= 0 && fe[3] <= 48)
 {
  windowFrameMargin.left = max(windowFrameMargin.left, fe[0]);
  windowFrameMargin.right = max(windowFrameMargin.right, fe[1]);
  windowFrameMargin.top = max(windowFrameMargin.top, fe[2]);
  windowFrameMargin.bottom = max(windowFrameMargin.bottom, fe[3]);
 }
 if(IsOpen() && top)
  top->owner = owner;

 int version = 5;
 XChangeProperty(Xdisplay, GetWindow(), XAtom("XdndAware"), XA_ATOM, 32,
     0, (byte *)&version, 1);
 FixIcons();
}
```

Mirek
