
Subject: Re: [SOLUTION FOUND!] TheIDE doesn't restore window state correctly on Linux

Posted by [sergeynikitin](#) on Tue, 21 Jun 2011 03:26:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Solution found! Please apply patch!

File TopWinX11.cpp
svn 3546

function
void TopWindow::Open(Ctrl *owner) (line 235-338)

Near end of function I add Line

SyncState0();

result function will be:

```
void TopWindow::Open(Ctrl *owner)
{
    GuiLock __;
    if(dokeys && (!GUI_AKD_Conservative() || GetAccessKeysDeep() <= 1))
        DistributeAccessKeys();
    UsrLogT(3, "OPEN " + Desc(this));
    LLOG("OPEN " << Name() << " owner: " << UPP::Name(owner));
    IgnoreMouseUp();
    bool weplace = owner && center == 1 || center == 2 || !GetRect().IsEmpty();
    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();
    if(!weplace) {
        LLOG("SyncCaption");
        SyncCaption0();
    }
    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++) {
        // X11 tries to move our window, so ignore the first set of ConfigureNotify
        // and move the window into position after FocusIn - but not if we want WM to
        // place the window
        if(weplace)
            while(XCheckTypedWindowEvent(Xdisplay, top->window, ConfigureNotify, &e)) {
                if(e.xconfigure.window != top->window)
                    ProcessEvent(&e);
            }
        if(XCheckTypedWindowEvent(Xdisplay, top->window, FocusIn, &e)) {
            ProcessEvent(&e);
            if(e.xfocus.window == top->window)
                break;
        }
        Sleep(10);
    }
}
if(weplace) {
    WndSetPos0(GetRect());
    LLOG("SyncCaption");
}

```

```

    SyncCaption0();
}
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)
{
    GuiLock __;
    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(!sOpen() && top)
    top->owner = owner;

int version = 5;
XChangeProperty(Xdisplay, GetWindow(), XAtom("XdndAware"), XA_ATOM, 32,
    0, (byte *)&version, 1);
SyncState0(); // ##### This Added line! #####
FixIcons();
}

```

After this correction, function Maximize work perfectly.
And function SerializePlacement(s) fully work.

PS

I'm not shure about place of call SyncState0() may be we must place it a bit early.
