Sons of Thunder Software
SOFTWARE DEVELOPMENT AND CONSULTING SERVICES

Products Services Developer Resources Contact STS About STS

livecode

Things to Be Aware of When Developing For Linux

Unlike Mac OS X and Windows, the Linux environment is very different and can be unfamiliar or even downright scary for Revolution developers who wish to deploy applications to Linux. As Linux' popularity grows, and certain distributions ("distros") become more prominent than others, Revolution developers who are developing for non-Linux platforms may wish to start developing for Linux. The following is provided for Rev developers who are "Linux newbies". - Editor

The Linux Environment

The thing to understand first is the environment in which almost all the other applications on a Linux system have been written in. They will almost all, unlike yours, be Open Source. Many will have been written in either the Gnome or the KDE development environments. Even if written in other environments, they will use open source components. If written in Python or other popular languages they will probably use the KDE or Gnome bindings. They will then often take their look and feel from the Gnome or KDE desktop settings. They will also have access to all the shared libraries, and will use them.

The environment in which they run is highly layered and structured. The window manager is distinct from the desktop environment, and distinct from the log-in manager, and they all are interchangeable and communicate via structured protocols. Unlike either Windows or MacOS, you can choose independently:

- which distribution you will use (Ubuntu, Fedora, Mandriva...)
- within that, window manager you will use (metacity, kwin, ICE, openbox, Fluxbox...)
- within that, which desktop environment you will use (KDE, Gnome, Xfce...)
- and within that, which theme you use (Clear Looks, Platinum...)

Some things are windowmanagers but have functionality now associated with full desktop environments. FVWM, WindowMaker and Enlightenment are in this category.

So, you might choose a theme (icon set and colors) for KDE that makes it look just like Gnome. It will look the same, but will of course still have all the KDE menus and default selections. There are innumerable themes, including ones that emulate or approximate Mac and Windows styles. There is no particular way that any distro looks, or even that most Desktop Environments look. OSNews some time ago had a discussion in which people were invited to post their desktops. There were hundreds of posts, and screenshots, and almost all were wildly different, even when based on KDE.

Almost all applications will have been packaged for the distro in which they are included, and will be available for download from the distribution's repositories. For instance, they will be Debian packages, or RPM packages or slackware packages. They will carry indications of what libraries they need, and the installer will check for the presence of these and then add them automatically if not found during installation.

For example, here are the requirements for Xfe, a file manager:

- Xfe is written in C++ and built using the FOX graphical toolkit library, therefore:
- For versions of Xfe prior to 0.70, you need the FOX library 1.0.x.
- For versions of Xfe prior to 0.80, you need the FOX library 1.2.x.
- For versions of Xfe prior to 0.98, you need the FOX library 1.4.x.
- For versions 0.98+, you need the FOX library 1.6.x.

Install Xfe on Debian or Mandriva, and you'll find that the right Fox toolkit comes with it. You'll also likely find that an entry is made in the KDE and Gnome menus, and that your alternative window manager picks up the Gnome and KDE menu entries and uses them, even if you are not running either.

"Standard" Linux Applications vs. Apps Written in Revolution

Your application is likely not anything like what is described above, and you are probably not distributing under the GPL, and in any case, you are using a closed source tool, Revolution. So unlike almost all the other apps on the system, yours will be self contained. Your app will thus not be packaged as an RPM or Deb file. You will thus not find your app on installation automatically finding its way into the system menus, or appearing as a choice for certain file types.

In terms of what is out there, Gnome is reported to account for around 35% of desktops, KDE most of the remaining ones, though Xfce is rising in popularity. You can assume that anyone running the less mass market desktops, Fluxbox, WindowMaker, FVWM and so on, is expert and able to configure for himself. It is a common mistake amoung newcomers to Linux to think Ubuntu is a sort of universal standard. It isn't. Its very popular, but so are Mandriva, Fedora, Red Hat, Debian, Suse, PCLinux - in no particular order.

Suggestions

1. **Ubuntu <> Linux**

   Do not assume that because your app works as you want on Ubuntu out of the box, it works on Linux. It probably will, but it needs verifying. Do not assume that because Ubuntu runs Gnome as a default, Gnome is all you need worry about. Ubuntu also comes in a KDE and Xfce flavor, Kubuntu and Xubuntu. You need to verify that it works how you want on at least Xfce and KDE, and it will be prudent to check an RPM distro like Suse or Fedora or Mandriva as well as Ubuntu.

2. **No Universal "Linux" Look and Feel**

   Don't try to match the look and feel of any particular distribution: there is no such thing. The look and feel doesn't vary with the distro so much as with the window manager and desktop environment and theme. And even people using the same window manager and the same distribution may turn out to have quite different kinds of icons and different looks for folders, because they have chosen a different theme. Unlike Mac or Windows, the look and design of desktop objects is basically down to the user, not to the window manager, and especially not down to the distribution. Even in Gnome, the most restrictive environment, you cannot assume that the now usual form with a top and bottom toolbar/panel is universal. Lots of people like Gnome but hate the top panel, and just delete it. And anyway, you are going to have both Gnome and KDE users and they will be using apps from both stables, so they will be used to things looking different.

   What to do? Pick a well behaved Linux application - Gedit or Kate, the Gnome and KDE editors are good choices, and emulate how it behaves. The result will be acceptable to any Linux user, and the look will resemble at least some of his other applications.

3. **Test Multiple Distros/Environments**

   Do test on multiple distros and multiple desktop environments to make sure that everything is visible and in the right place. Fonts will vary unpredictably (see below). I've found on a tiny scale that its a good idea to leave more space than your development environment suggests you need around objects, because their positioning can vary just a bit.

4. **Accommodate Multiple Users**

   Make sure you accomodate multiple users. You have to decide in the installation whether you are installing for one user, or for all users. If for one user, put your app in /home/user. If for multiple users you should probably put it in /opt.

5. **Store User Data In The Right Place**

   You'll be tempted to think there is such a place as 'where the application is installed', and that perhaps you should keep your user data there. Like, where is the programs folder? This would be a mistake. There is no such place in Linux, except for your application and one or two others.

   There are two reasons:

   - One reason, which is both the glory and the shame of Linux (the shame because it leads to 'dependency hell') is that Linux uses shared libraries to a very high degree. So if you take a typical Linux app, lets say part of the KDE Office suite, and you look for the database component, Kexi, you will not find anything that you can really consider the program Kexi and where it is installed. You'll find a whole bunch of things scattered all over the system portion of the file structure.

     If you try to install Kexi with no KDE installed on a bare, non-KDE Debian or Ubuntu type system, where it gets all the dependencies for you, you'll be surprised to discover that you're downloading over 100Mb. But Kexi itself is quite small. Then you'll find that the rest of KOffice consumes very little extra space. Its because you've already got the KDE libraries which are used by all the K apps. Now, your application will be closed source and self contained, so it will be in one place. But it is unusual in this.

   - The second reason is the accomodation of a multi user environment. No well-behaved Linux application will store user data in **/opt**, even if that's where it installs itself. There are two reasons. First, routine backups will only back up **/home**. So that's where all user data should go. The second is that you have to allow for multiple users to create identically named files, including preferences. So you cannot put them all in one place.

     It follows that all user data belongs in **/home/<userName>**, even if you are installing your program in /opt in order for it to be usable by everyone who logs in. Rev itself on Linux is not well behaved in this sense. It expects as a default to store your work in the Revolution folder. It should expect to store them in **/home/<userName>**, and Open from the IDE File menu should immediately offer you **/home/<userName>** as a default, rather than **/opt/revolution**, where the app is installed.

6. **File Navigation: Hide the Hidden Files**

   [If your application provides file/folder navigation (like an Open or Save As dialog box), keep in mind that there is no OS-inherent file browser in Linux, unlike Mac OS and Windows. So you need to provide a stack as a file browser for your Linux apps, and Rev provides one when you build a standalone (you check off "Unix File Selector" in the Standalone Application Settings window) which you may be tempted to use. However you may wish to modify the one that Rev provides, or create one of your own for the reason that follows. - Editor]

   In the file navigation part of your app, don't make users step through all the dot files. **/home/user** is full of files which start with dot, as in **/home/peter/.kde**. When you are offering to save a file, or open a file, don't display these. Its just confusing and irritating. Rev is a good example of what not to do. You fire it up, and choose "Open". It offers you everything in the Revolution folder. However, you do not want that, because the app you are working on is on the desktop. So you go up a level. Now you see **/opt**. Up again and you see the file system. Down to **/home/peter**. Now, before you can see the files on the desktop, you have 30 or 40 dot files to step through, and somewhere at the bottom is Desktop. Don't do it to people! Any Linux app, eg Kate or OpenOffice or Gedit, will show how this should be done.

7. **Accomodate Multiple Desktops**

   All Linux desktops will come with multiple virtual work spaces. Gnome or KDE will install out of the box with 4. People need to be able to use them - if it makes sense for your app. This is something that rev on Linux fails dramatically at. If you have lots of different windows in an application open at once, you should be able to put one or two on a different workspace. For instance, if I'm doing a complicated spreadsheet, I often have the Open Office document open on one desktop and the help open on another, and flip between them. With Rev, you can have half a dozen free desktops, but all crammed into one window you'll have the dictionary, the tool palette, the application, the property inspector, the script editor, all overlapping and getting in each others way. I've no idea how to write Rev apps so that they behave properly in desktops, but if you are going to have this many possible windows, you have to allow people to manage them as they are used to doing. To see how this should work, use OpenOffice. Note the little icon and popup menu at the very top left, above the File menu. Open up a document and also help. Then move help to a different desktop. Notice how you can use the icons on the reduced image of the desktop in the bottom toolbar (or the Gnome default is the top one) to move open windows from one desktop to another. You can do this with rev windows too. The problem is, as soon as you use them, back they all come to the same desktop!

8. **Be Very Wary Of Fonts**

   Be very wary of fonts. You cannot assume that a Linux install has all the same fonts you have in OSX or Windows. If you are going to use the MS fonts, you have to make sure they are installed. And alas, how to install fonts varies from distro to distro. Kfontviewer is the common denominator, but Gnome installations like Ubuntu will not have it out of the box, they'll have to get it first. Mandriva has its own font management package. The safest thing is going to be only use fonts which you are sure come out of the box for all distros, but even this is not really safe. You cannot assume that because a font is installed, it is available to Rev.

   The issue is paths. Even if you supply fonts, or use the ones you know must be there, you can't be sure, unless the path is set right, that your user and thus your app, can find them. I have found only one reliable way to test this on 2.6.1. If you fire up a Linux version of Rev on your target distro, then place a field with text in it, then set the text to the font you want to use, from the IDE, if it displays right, you are going to be OK with the compiled version on that distro. The result, on Debian, at the moment, is that only two fonts are usable across the full range of sizes. Probably this will be fixed in the next release, but caution is still indicated.

Posted 3/24/2007 by Peter Alcibiades to the Use-Revolution List

Print this tip

Copyright ©1997-2013 Sons of Thunder Software, Inc. All rights reserved.
Send all comments to webmaster@sonsothunder.com.