

Standards

By Jon "maddog" Hall

Recently, I was talking to a college professor who was complaining about the multitude of ways in which Linux works on various distributions. He was complaining that one distribution of Linux put a particular file in one place, and another distribution put a file in another place. He said it was hard for him to remember where each distribution put each file.

If this were a "normal user," I would have simply advised staying with one particular distribution instead of going from distro to distro, but in the professor's case, he could not enjoy that luxury because his students were allowed to use different distributions. This type of inconsistency also affects systems administrators, and even regular users as they encounter different distributions of GNU/Linux in various places.

One of the advantages of GNU/Linux is that you can tailor it to meet your needs. For example, you can choose Gnome, KDE, LXCE, or other graphical desktops, or you can use it with no graphics at all by working from the command line. This flexibility is seen by most hard-core Linux people as a feature.

However, the different sets of interfaces that occur on different distributions – and even different configurations of distributions – do create problems for people who are trying to create courses, document graphical features, or tell new users how to use GNU/Linux.

Recently KDE, Gnome, and Ubuntu have all experimented in what some consider to be a radical departure from the "traditional" desktop, aiming toward something they hope is more intuitive and easier for people to use. In doing this, they have alienated some of the users who preferred the older style of desktop.

To many people, one of the great selling points in Apple's favor is the consistency of their interface and that, if you buy it from Apple, you know you will get (for every release of code) one interface that works the same. I have heard the same for Microsoft, although Apple usually gets the acclamations for consistency and ease of use.

Much of Apple's "ease of

use" comes from consistency in both their own operating system and in the applications developed by their application developers. Unless the application has the look and feel of the OS, it is not "blessed" by Apple to be an "Apple App" and might not end up in an "Apple Store." This consistency across Apple applications also includes things like data formats, data conversions, and other low-level issues that allow applications to work well with each other as well as with the operating system.

Many years ago, it was recognized that a set of standard binary interfaces was needed (an Application Binary Interface, or ABI) that allowed application developers to develop one set of binaries and deliver them to various GNU/Linux distributions. This effort evolved into the Linux Standard Base Project, and although many do not consider it perfect, it certainly has helped keep divergence from happening at an ABI level in the different distributions.

Perhaps it is time for the Free and Open Source Software community to come together and formulate the "Look and Feel" of GNU/Linux. Perhaps it is time for a "standard interface" that people can count on in every distribution that would act in a particular way. These standards would not preclude various GUIs, but they would give people who use GNU/Linux something they could count on to be there at all times, and people writing books and manuals on applications would have standard interfaces to include in their descriptions.

In the past, an interesting product called "Looking Glass" from the Visix company had the capability of looking like any number of different interfaces (Windows, Apple, Motif, and one or two others). A set of widget APIs allowed application developers to code their applications for whatever platform the application was running on.

Even more interesting was that the same binary would look like an Apple application when it displayed on a Mac, like a Microsoft application when it displayed on Windows, and like a Unix application when it displayed on a Unix system. With a client-server architecture, the application could display on three different platforms from one server and have the "look and feel" of three different OSs.

With this type of technology, an agreement on a standard set of system admin tools and more data format and exchange consistency between applications would go a long way toward making GNU/Linux "easier to use." ■■■

