Portland

John Cherry, OSDL Desktop Linux Initiative Manager, introduces the project that's unifying the free software desktop

An end to the desktop war?

An application vendor determines that he wants to target the Linux operating system to introduce his application to the Linux market. That's the easy part. While his application may run nicely in a Linux environment, now he needs to decide how to handle the installation and runtime capabilities that may differ from Linux distribution to Linux distribution and from desktop environment to desktop environment. Will he need to limit his application to a subset of the Linux distributions or desktop environments? For Independent Software Vendors (ISVs), Linux has not traditionally been viewed as a unified, single platform. For Linux on the desktop to be successful, this perception must change. It's as simple as that. An ISV must be able to create or port his application for "Linux."

The Portland Project represents the power of the open source community to define a need – such as developing unified Linux desktop capabilities – and collaborate to address it in a way that moves both the market and technology forward. The Portland Project, coming up on the one-year anniversary of its inception in Portland, Oregon, is the result of the first Open Source Development Labs (OSDL) Desktop Architects Meeting. In less than one year, Portland has brought together disparate members of the desktop Linux community to address a common problem. The technical successes have been fast and unprecedented; Portland beta is today being widely tested and has show up in applications such as Google Earth. You can check it out at: <portland.freedesktop.org/wiki/>

Work on the Portland project is closely aligned with the LSB efforts of the Free Standards Group (FSG) around the desktop. The LSB 3.2 roadmap for the upcoming year includes improved desktop integration around freedesktop.org standards. Standards that are considered for inclusion in LSB at this point include the Desktop Entry Specification <www.freedesktop.org/wiki/Standards/desktop-entry-spec>, Desktop Menu Specification <www.freedesktop.org/wiki/Standards/menu-spec>, Desktop Basedir Specification <www.freedesktop.org/wiki/Standards/basedir-spec> and the Icon Theme Specification <www.freedesktop.org/wiki/Standards/icon-theme-spec>. The Portland installation utilities provide application developers with easy to use tools that will help them follow these standards.

THE LAST FIVE PER CENT

Waldo Bastian, leader of the Portland Project, explains that the project addresses the last five percent of the differences between desktop environments such as KDE and GNOME. Ninety percent of the functionality that an application requires comes from the X Windowing System. The other ten percent consists of true window manager functionality (approximately five percent) on the one hand and integration functionality (approximately five percent) on the other. The window manager functionality has been standardised over the years; first by the Inter-Client Communication Conventions Manual (ICCCM) in the early '90s, and later extended by the XDG Extended Window Manager Hints (EWMH) specification on freedesktop.org. As a result the majority of window managers on Linux provide consistent functionality to applications. The integration functionality is provided by the desktop environment and includes the differences between desktop environments. The Portland project attempts to address the differences by providing common high-level desktop integration interfaces.

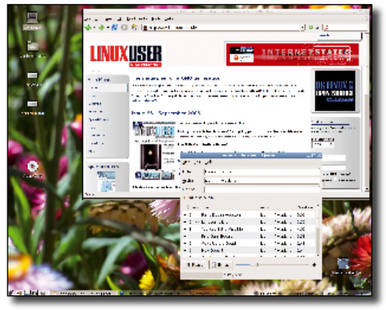
Portland intends to deliver two sets of interfaces. One set of interfaces will be offered in the form of a set of command line tools. The other set of interfaces will be offered in the form of one or more libraries that applications can link with. The library approach is considered to be the more usable one for tasks that require more complex interaction. The set of command line tools can be grouped into tools for use at installation time, and tools for use at runtime. Some tools may be useful in both situations. The Portland Project team is driving to release the 1.0 version of a set of command line tools (xdg-utils) as this issue of LinuxUser & Developer goes to press. These command line tools include tools for both installation and runtime. Since April, the project has produced three beta releases. The team is working with popular desktop environments like GNOME and KDE to enhance its interfaces in order to support the needs of ISVs around the world.

Portland intends to deliver two sets of interfaces. One set will be offered in the form of a set of command line tools. The other set will be offered in the form of one or more libraries that applications can link with

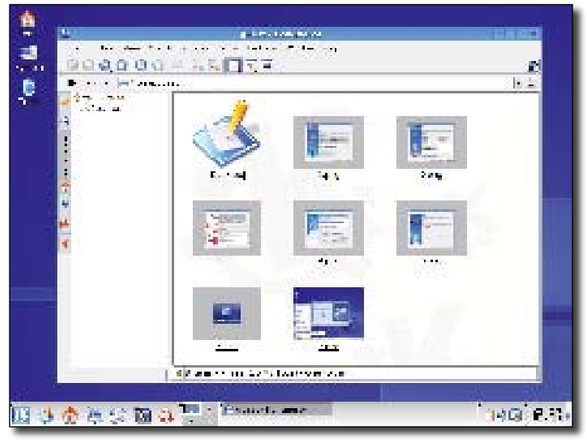
INSTALLATION TOOLS

For tasks best handled at installation time, command line tools are the most suitable. The typical installation use case is an RPM based archive that gets installed system-wide by the root user. Such installation may be part of an automated install process and should not require any user interaction. A secondary use case is the case where an RPM based archive gets installed by an end user in the home directory of the user, for that particular user only. For the design of the installation tools, a primary concern is to make sure that the tools integrate well with RPM-based package management.

It is the intention of the Portland Project that any resulting installation tools will be included in Linux distributions and standardised as part of the LSB effort. In addition, application developers can include the same installation tools along with their applications to use as fallback for use on older distributions that do not yet include the Portland tools.



The GNOME 2.14 desktop. The Portland Project aims to solve the last 5% of free software desktop incompatibility



The KDE 3.4 desktop. Portland will provide both command-line tools and libraries to bridge the free desktop gap

Currently the following installation tools have been developed as part of Portland:

xdg-icon-resource – command line tool for (un)installing icons
xdg-desktop-menu – command line tool for (un)installing desktop menu items
xdg-desktop-icon – command line tool for (un)installing icons to the desktop
xdg-mime – command line tool for querying information about file type handling

and adding descriptions for new file types

RUNTIME TOOLS

The Portland runtime tools are for integration tasks that are relatively straightforward and that do not require complex interaction with the application. The advantages of command line tools are that they can be used from all kinds of applications, regardless of programming language and toolkit used. They can also be used from shell scripts. The presence of a command line tool is easy to detect.

It is the intention of the Portland project that any resulting runtime tools will be included in Linux distributions and standardised as part of the LSB effort. Just as with the installation tools, application developers can include the Portland runtime tools along with their applications to use as fallback for use on older distributions that do not include the Portland tools yet.

For the first version of the Portland runtime tools the following commands are being considered:

xdg-open – opens a file or URL in the user's preferred application
xdg-email – command line tool for sending mail using the user's preferred e-mail composer
xdg-screensaver – Enable, disable, and suspend the screensaver

So what is next? An extensive test framework has been developed that people will be able to run on their favourite distributions: <portland.freedesktop.org/wiki/TestSuite>

BRING FORTH YOUR APPS!

We encourage people to try this out, and to report any problems that they encounter. We are also looking to engage with application developers to hear whether xdg-utils meets their needs with regard to desktop integration. Last but not least, we want to go over the freedesktop.org specifications in order to bring them to 1.0 status. The current specifications have largely been implemented in the two major desktop environments for some time already. However, they still contain odd bits and pieces that never managed to gain widespread adoption. We would like to remove those parts to ensure that we are left with specifications that are truly widely adopted in their entirety, and give application developers a set of guidelines that they can rely on. See: <freedesktop.org/wiki/Standards>

The Portland Project is a success on many fronts. There is not much doubt that it will accelerate the availability of desktop applications for Linux. The Project improves interoperability issues for ISVs whose applications must work regardless of Linux distributor or desktop environment. Together with Linux's growing desktop market share, more and more ISVs are expected to make their applications available for Linux in the near future, resulting in further acceleration of Linux adoption on the desktop. It also gives PC makers and enterprise application companies, for example Dell and SAP respectively, added incentive to increase the development and general availability of Linux-based products.

The Portland Project represents the power of the open source community to define a need, and collaborate to address it in a way that moves both the market and technology forward