The Moblin.org Open Source Project

http://www.moblin.org

Derek S. Speed, Open Source Technology Center, Intel Corporation

Introduction

A new category of devices is emerging in the market that delivers the ability to access the rich content available on the Internet, but without the complexity often associated with PCs. These devices come in a variety of form factors, from Mobile Internet Devices (MIDs) to portable media players (PMPs) to Personal Navigation Devices (PNDs) to digital set top boxes (STB) and netbooks.

The Moblin.org project was launched by Intel in July 2007 with the goal of driving the new technologies required to enable this new class of devices. The initial focus of the Moblin.org project is on enabling the MID software ecosystem, but will expand over time to include additional product categories.

This paper will describe the motivation behind the Moblin project, its goals and objectives, key projects and technologies, and how to get involved.

The Moblin Vision

In 2006, as Intel was launching its first generation of Ultra Mobile PCs, device manufacturers voiced their desire for additional operating system choices that met the following high level requirements:

- Available from a number of vendors
- Low cost, as these devices are sold at consumer price points
- Delivers a rich Internet and multimedia experience to end-users
- Customizable to allow deployment on a variety of devices and manufacturer specific branding
- Excellent support for wireless networking technologies (WiFi, Bluetooth, GPS, 3G, WiMax)
- Tuned for low power consumption, long battery life and efficient use of system resources such as memory and storage
- System software and built-in applications that were consumer friendly, intuitive and easy to use
- Leverages the x86 software ecosystem in terms of software availability and developer mindshare
- Enables compelling new usage models and user interface paradigms

From these requirements, it was clear that a Linux*-based operating system was a good starting point - in fact, several successful mobile and consumer electronics products had been created using Linux as the core operating system. However, there was no concerted effort in the open source community to proactively drive the new technologies required for this category of devices.

And so, the idea behind the Moblin.org project was born. The purpose of the Moblin.org project is to:

- Drive the new technologies required to fill the gaps in using Linux for MIDs and other consumer oriented devices
- Jumpstart the software ecosystem required to support these new devices.

What Are The Key Problems That Moblin Is Trying To Solve?

Because the first generation of MIDs were based on standard mobile PC hardware, it was easy for the Intel engineering team to bring up a desktop/notebook PC Linux distribution on a MID platform to do some early experimenting to help define the scope of the Moblin project. After doing this, a few issues quickly became apparent:

- The standard Linux desktop user interfaces (UI), such as GNOME and KDE, are better suited for experienced computer users than novices. The fact that Linux was running "under the hood" needed to be transparent to end-users.
- Application GUIs written for a standard PC Linux distribution did not work well on a MID whose primary input method is a touch screen and whose primary output method is a landscape orientation screen that is much shorter than it is wide (800x480 is typical). On-screen buttons were too big or located in the wrong place on the screen. In many cases, you couldn't even see the buttons because they were drawn off-screen.
- The memory and disk footprint was too large. A typical desktop Linux distribution might be several gigabytes in size and use 512 MB or more memory. Because they are cost sensitive, MIDs typically have 512 MB memory or less and their on-board flash memory storage is typically around 4-8 GB. If too much storage was used, there would be no room left for user content and run-time performance would be sluggish.
- Power efficiency was sub-optimal, mostly due to lack of attention to this important metric as opposed to any fundamental issue with Linux.
- Software development for Linux is typically self-hosted, meaning that the software is edited, compiled, tested and debugged on the same system. Because MIDs use less powerful processors that are optimized for low power, they are not as well suited to software development tasks as more powerful multi-core notebook or desktop workstations.

This early prototyping helped identify the key projects required on Moblin.org – a "MID appropriate" application framework, consumer friendly applications and user interface, power optimization, platform configuration tools, and development environment.

At the same time, the Moblin team recognized that embedded Linux operating systems had a number of advantages:

- The operating system is highly modular and allows the platform developer fine-grained control over which components are installed on the target platform. The downside is that platform developers need to have a detailed understanding of which of the hundreds of software packages are required to enable the end-user usage models they have in mind.
- Because embedded OSes "grew up" on platforms with significant constraints in terms of processing power, memory, and storage, they are typically highly optimized.
- The software development model for embedded Linux is a host/target model – most development (code editing, compilation, builds) is done on the host, then the resulting OS image is downloaded to the target and is debugged remotely from the host PC.

The Intel Moblin team felt that it was possible to achieve the benefits of both desktop and embedded Linux distributions.

One of the first problems the team tackled was how to make it easy for a platform developer to configure a target Linux OS image without having to understand, at a detailed level, which software packages need to be

in the target device image. The solution was to create what would become one of the key Moblin projects, "Moblin Image Creator".

Moblin Image Creator Overview

Moblin Image Creator (MIC) is a tool that is targeted at two types of developers: platform software developers who are responsible for creating the core platform software that runs on a device and developers who are creating software components that add value on top of the core software stack. Developers in the second category typically focus on the creation of either middleware/libraries or end-user applications.

If you are a platform software developer, MIC allows you to easily create an image for a target device that contains all of the software components needed to enable specific usage models. You do not need to know which of the hundreds of open source software components are required to enable specific usage models – you simply choose from a set of pre-defined feature sets using MIC and easily create an image that can be downloaded onto a target device.

If you are a middleware or application developer, MIC is used to create a development environment that includes all of the core components that are required to compile, test and debug software in an isolated (chroot) environment.

MIC runs on a host development PC running Linux. All of the development work is done on the host PC until the point that you want to run the image on the real target hardware. At that point, a device specific image is burned onto removable media (USB pen drive, DVD, etc.), transferred from the host to the target and loaded onto it. Because both the host and target devices are based on the x86 instruction set, cross-compilation and emulation is not required.

To understand what MIC does, it is important to understand some fundamental terms:

Project - a project is a platform-specific build environment with one or more targets. When you create a project, MIC creates a full Linux file system in the directory you specify. Chrooting into this directory creates an isolated environment where you can develop software for your device without impacting the root file system of your development machine. The environment includes tools such as apt so you can install packages of the tools you need. The project directory includes one or more targets which are located in the <project location>/targets directory.

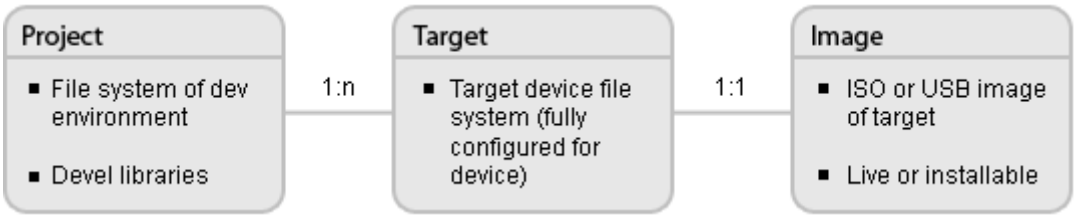
Platform - the platform is your target device. You select a platform when you create a project. This determines the kernel, system configuration, and device drivers that are made available as part of the targets you create within a project.

Target - the target is a platform-specific Linux file system, created in the /targets directory of your project. You create an image from a target. One or more targets can exist for each project. Like projects, you can also chroot into this environment to configure it or copy applications from your project before making an image.

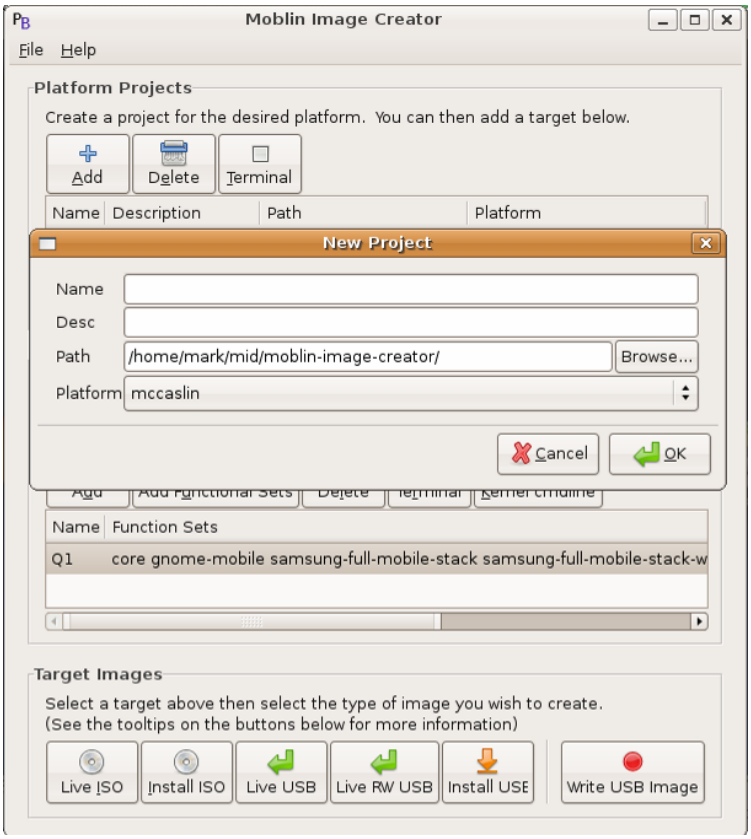
Function/Feature Set (fset) - a functional (or feature) set is a group of packages representing some functional area that Moblin Image Creator can install on the target. A number of pre-defined fsets are included with MIC from the minimal "core" fset (boots the platform up to a command prompt) to a full MID stack fset (including browser, media player and application launcher/home screen). An fset can have dependencies on other fsets. If you select to install an fset that depends on another fset, the prerequisite fset will automatically be selected and installed on the target first. Platform developers are free to create their own custom fsets for their unique requirements.

Image - an image is a large (around 350-600+ MB depending on the number of fsets included) binary file created from a target. Several types of images can be created including live images (where the OS is run from removable media, not installed on the device) and install images (where the software stack is loaded onto the built-in flash or disk drive on the device). Once the image has been created, it can be installed on the target device or run from removable media such as a USB pen drive or DVD-ROM.

The diagram below shows the relationship between projects, platforms, targets, fsets and images:

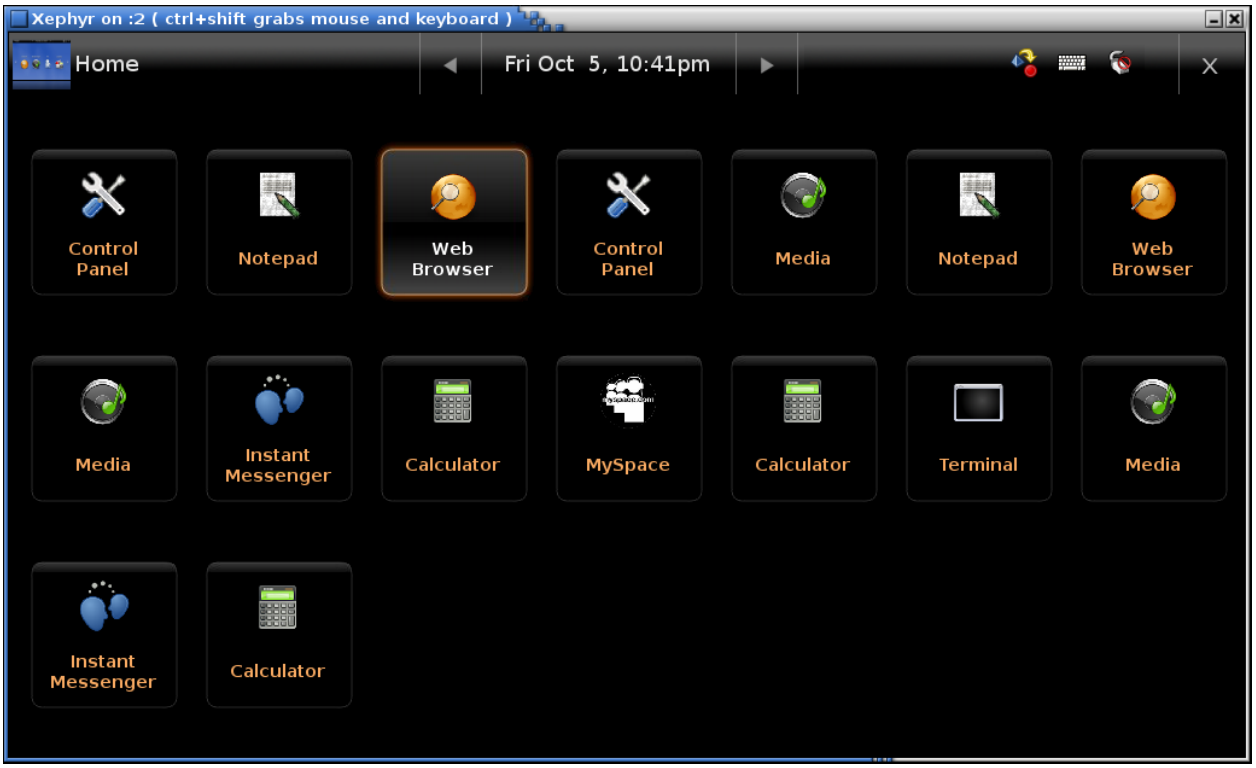


Moblin Image Creator offers both a GUI and a command line interface. The image below shows the GUI interface.



After creating a target, platform developers can write, build, test and debug their software on the host PC. By using MIC in conjunction with the xephyr tool, the screen output that would be displayed on the target device can be displayed on the local workstation, eliminating the need to have a real target device for most categories of software development. Of course, software that is device specific, such as the kernel, device drivers, hardware accelerated graphics or multimedia codecs and power management software, needs to be run and debugged on the actual target device.

The image below shows the application launcher/home screen running under xephyr.



Moblin.org Technologies

The Moblin core Linux stack is based on proven technologies from Linux and other open source communities that were chosen by the Moblin team to enable the target usage model for MIDs and other consumer oriented devices. These technologies include:

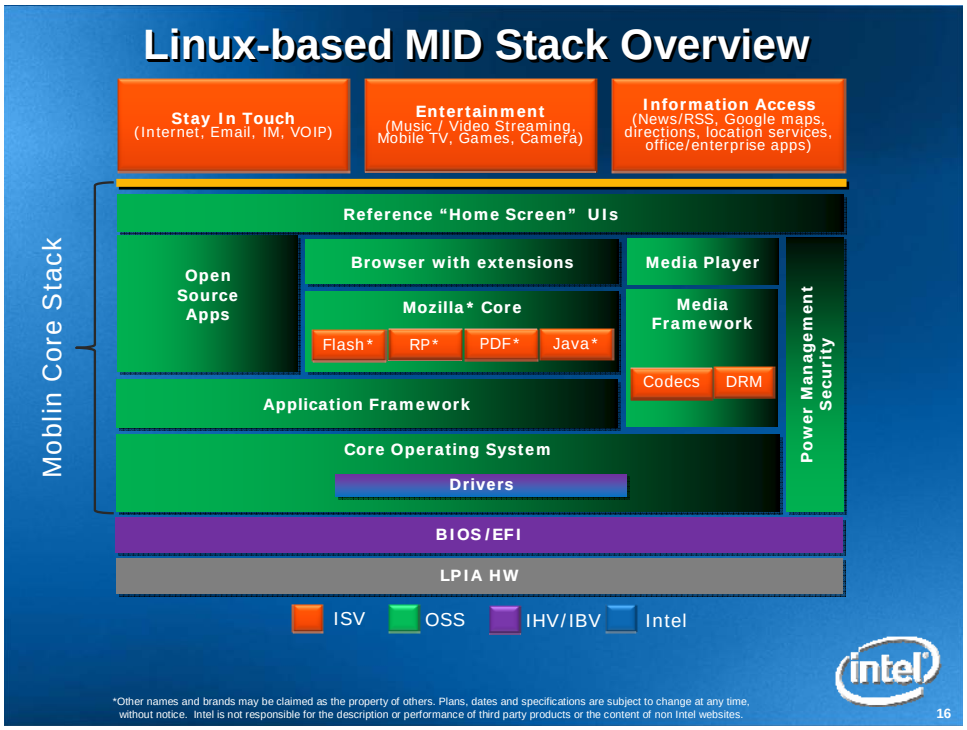
Graphics, windowing, multimedia: X Window System, OpenGL, GStreamer, Helix, ALSA, Video for Linux, Matchbox window manager

Application framework: GTK, Cairo, Pango, BlueZ, D-Bus, Avahi, Evolution Data Server (EDS)

Core operating system: Linux kernel, chipset drivers, platform specific device drivers, core libraries, complete networking stack, USB stack, platform power management, etc.

Web browser: Mozilla-based browser with full support for available Linux plug-ins such as Adobe* Flash*

The diagram below shows a high level view of Moblin software architecture.



The boxes shown in green are open source software components that in many cases have been optimized by Intel and others in the Moblin community – these components are included when a target device image is built using Moblin Image Creator. The boxes shown in orange are optional software components, some of which are closed source, which may be licensed by OEMs from independent software vendors (ISVs) to create a complete platform.

Moblin.org Projects

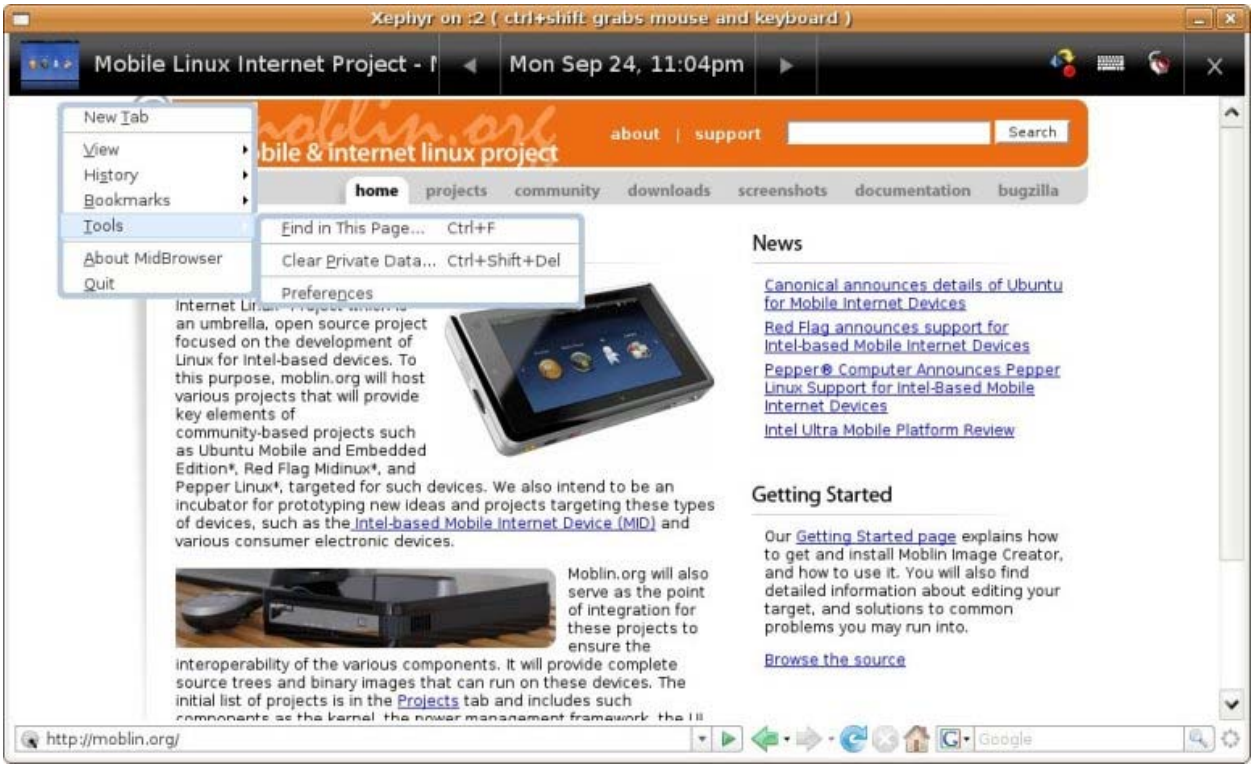
Moblin.org hosts several open source projects that are focused on enabling key technologies required for MIDs and other consumer oriented devices. These projects include:

- **Moblin Image Creator**
- **Kernel** Platform-specific kernel patches and device drivers are part of the kernel project, along with information on adding your own patches and building a kernel.
- **UI Framework** The UI Framework project is focused on the Mobile Internet Device home screen interface and its underlying framework, which is GTK-based. Customizability is a key aspect of the UI Framework project, with a focus on easy reorganization and extension of the UI.
- **Power Policy Manager** Power management is a critical aspect of Linux for mobile and consumer electronics devices. The power policy management project is focused on extending and enhancing existing Linux power management capabilities, and providing a comprehensive, robust, and extensible power management framework.

- **Connection Manager** Mobile Internet Devices are expected to provide a rich set of communication options, from wireless LAN to personal area network technologies, such as Bluetooth*. The Connection Manager project addresses challenges such as network configuration and profile-driven dynamic network selection.

- **Browser** A full-featured browser is at the heart of delivering an uncompromised Internet experience on mobile and consumer electronics devices. The browser project focuses on extending a Mozilla-based browser core with features such as a simple, finger-driven UI.

The following picture shows the Moblin browser running on a MID.



- **Media Player.** The multimedia project is home to multimedia application development that supports audio and video playback and photo viewing, along with content management through a mobile-optimized interface. The multimedia application, Moblin Media, supports both the Helix* and GStreamer* multimedia frameworks.

The following picture shows the Moblin Media Player playing a video on a MID.

- **Camera.** The camera project focuses on allowing the end-user to capture, manage, and share both pictures and videos using a built-in camera.

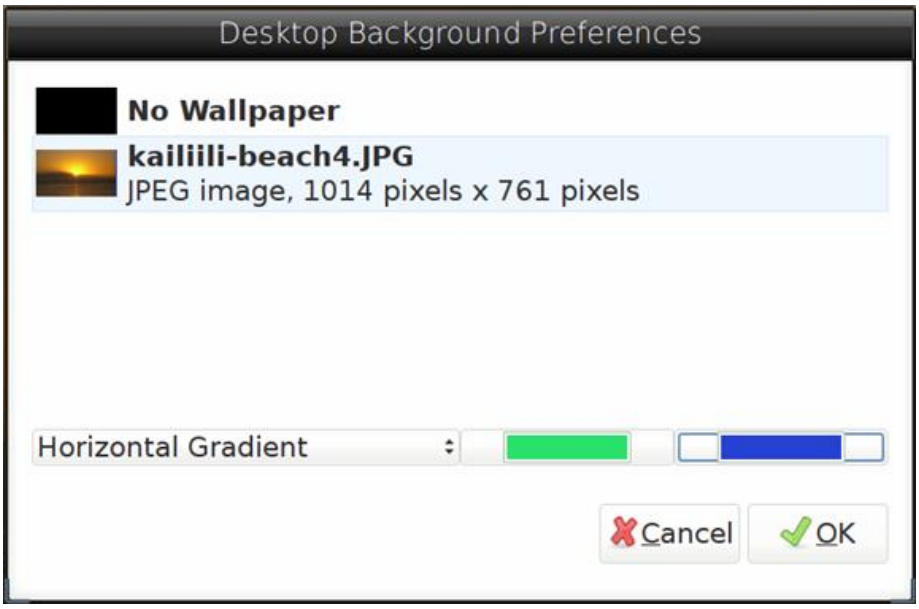
- **Chat.** The chat project is built on Telepathy's framework and specification, and inherits many of its benefits. It provides a robust framework to support popular messaging protocols.

The following picture shows Moblin Chat running on a MID.

- **Applets.** The focus of moblin-applets is on the creation and maintenance of configuration utilities for the Moblin desktop environment. Configuration utilities are pieces of software which provide the

ability to change the settings for one or more features of a mobile device. Each supports an independent GUI, usually in the form of a dialog.

The following picture shows one of the Moblin applets, the background preferences applet, running on a MID.



- **SDK.** Software developers use the SDK to create software components and applications that are optimized for Moblin compatible platforms. It includes tools like Moblin Image Creator, documentation and sample code.

- **Moblin Compatibility.** The goal of the Moblin compatibility project is to define a set of common APIs, libraries and other components that are required to enable the key usage models for MIDs and other devices. It builds upon the work done in the Linux Standard Base project. The Moblin Compatibility project includes a Moblin Compatibility Specification and a set of tools that allow OSVs and ISVs to test whether their OS and applications comply with the spec.

For detailed project descriptions or to get involved, visit: http://www.moblin.org.project.php

Moblin's Role in the MID Software Ecosystem

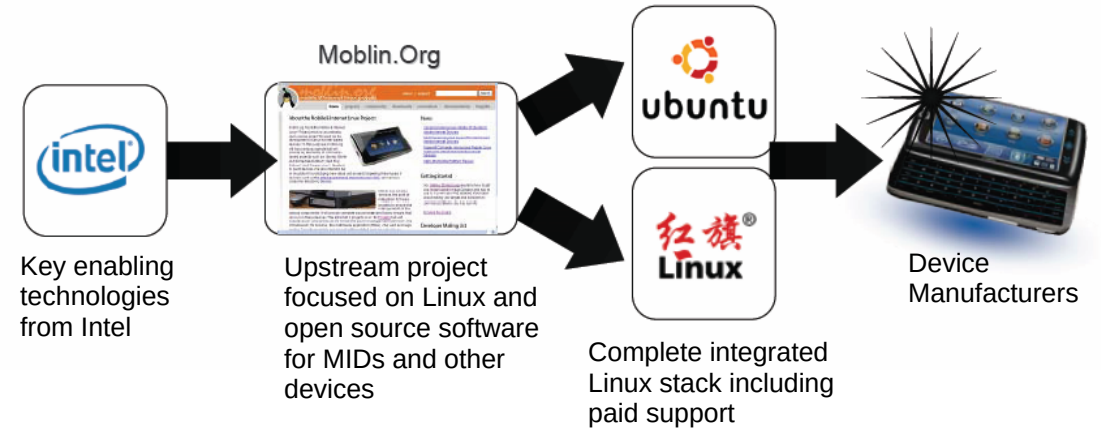
Moblin plays three critical roles in the MID software ecosystem:

1) **Creating a community of developers who are focused on creating and enhancing the core technology required for Linux-based MIDs and other devices.** When Moblin was first created, this was the primary objective of the project. This core technology is used by Linux operating system vendors who are creating distributions targeted for MIDs.

2) **Creating a community of developers who are focused on creating new software and services that innovate and add value on top of the Moblin core Linux stack.** With the launch of the first generation of MIDs based on the new Intel® Centrino® Atom™ processor technology, the role of the Moblin project is expanding to include this critical function.

3) **Defining a set of standards and tools to make it easier for operating system vendors (OSVs) and ISVs to ensure binary compatibility across Linux distributions from different vendors.**

The diagram below is a visual representation of where Moblin fits in the device ecosystem.



For OEMs with significant internal Linux expertise, using the Moblin stack directly is also a potential option. Those OEMs with less internal Linux expertise will want to take advantage of the skills of a Linux operating system vendor like Canonical or Red Flag who are creating products based on the Moblin technologies.

Why Moblin Is Unique in the Mobile Linux Ecosystem

There are many organizations, consortiums and companies that are working to create a mobile Linux platform. Most of them are focused on Linux for mobile phones. These organizations and projects include the LiMo Foundation and Google's Android project.

What distinguishes Moblin from these other initiatives is:

- **Moblin is a completely open source project.** If you want to contribute to or use Moblin components, there is no legal agreement to sign or membership fees to pay. Contributions are welcome from large corporations and individual developers. As with other successful open source projects, code and contributions are what talks in the Moblin community.

- **Moblin taps into the vast Linux/x86 software ecosystem .** x86 architecture is the platform of choice for many Linux and open source software developers. Over the years, these developers have created tens of thousands of software components for Linux/x86 platforms. Because Moblin is based on proven Linux technologies, the vast majority of software components can be used as is with no modifications required – just install the binary and run it. For software components that have a GUI component and need to be optimized for a MID, moving to Moblin is a port, not a re-write. In contrast, other environments require developers to learn new APIs and/or languages. This high degree of compatibility with the existing Linux/x86 ecosystem allows OEMs to reduce their time to market while still taking advantage of the latest and greatest development in Linux and other open source software.

- **Moblin delivers a rich Internet experience.** The Moblin web browser is based on Mozilla* technology, the same technology used in the award-winning Firefox* web browser. The Moblin web browser supports popular web technologies such as HTML 4.01, XHTML 1.0/1.1, Cascading Style Sheets (CSS) 1.x and 2.x, and JavaScript 1.7. Because much of the rich content on the Internet requires the use of browser plug-ins such as Adobe* Flash*, Moblin's web browser allows end-users to experience the web in way not possible on other mobile platforms. Content from popular web sites like YouTube*, Facebook*, NBC.com, and

CNN.com is rendered with the same level of fidelity as on a desktop system, enabling OEMs to deliver on the message of providing the "best Internet experience in your pocket".

- **Moblin technologies are not limited to mobile phones.** Most of the other mobile Linux initiatives are narrowly focused on mobile phone platforms. While MIDs are one of the key platforms supported by Moblin, the Moblin vision is much broader, embracing MIDs, in-vehicle infotainment systems, and other consumer electronics devices.

Moblin Compatible Linux Distributions

Canonical*, the sponsor of the Ubuntu project, has created a Linux distribution focused on Mobile Internet Devices called Ubuntu Mobile that incorporates technologies from the Moblin project. More details on the Ubuntu Mobile product can be found at http://www.ubuntu.com/mobile Developers should refer to the Ubuntu wiki at https://wiki.ubuntu.com/MobileAndEmbedded

Red Flag* Software has also created a Linux distribution focused on MIDs called MIDINUX* that incorporates technologies from the Moblin project. More details on MIDINUX can be found at http://www.moblin.org/news/RedFlagPR_0603Final.pdf

It is expected that other major Linux vendors will create Moblin compatible distributions in the future, providing OEMs with even more choices.

Intel Software Development Tools for MIDs

One of the great things for developers about Linux is the wide variety of tools that are available. In addition to excellent open source tools like MIC, gcc, and PowerTOP, Intel has created a suite of tools that are focused on helping developers reduce their time to market and get the most performance possible out of their MID software. This set of world class tools for Linux-based MIDs includes:

- **The Intel® C++ compiler for Linux with new support added for the Intel® Atom™ processor** The Intel C++ compiler for Linux is gcc-compatible and uses advanced techniques, such as profile guided optimization, to deliver world class performance.

- **Two new debuggers** – one targeted at application debugging and a JTAG-based debugger targeted at low level software and hardware bring-up. These new debuggers give unique visibility into the platform and are fully OS aware.

- **Intel® Integrated Performance Primitives (IPP)**, a set of highly tuned libraries in 15 application domains including imaging, audio and video, signal processing and cryptography. Intel IPP also includes a set of sample code to help jump start your development.

- **Intel® VTune™ Performance Analyzer** which allows developers to identify and resolve hard to find performance bottlenecks in their code.

For the first time ever, these tools are being offered free of charge with community support or with support from Intel for a fee. For more details, please see http://www.intel.com/software/products/mid

Don't Just Sit There – Start Moblin' Into Action!

Hopefully this paper has gotten you interested enough in the Moblin project that you are thinking about how to get involved. Here are some specific things you can do:

1.) Go to the Moblin web site and explore what's there – http://www.moblin.org

2.) Subscribe to the Moblin developer mailing list at http://www.moblin.org/mail-list_subscribe.php and start participating.

3.) Download the Moblin SDK at http://www.moblin.org/SDK.php The SDK includes Moblin Image Creator, extensive documentation and sample code to help you get started. Use MIC to see how easy it is to create a target image and run it on your Linux PC. Start porting your favorite software to Moblin and let other developers know about it through the mailing list.

4.) Get the Intel development tools at http://www.intel.com/software/products/mid to increase your development productivity.

Conclusion

Moblin.org is a key Intel-sponsored open source project that is focused on creating a community of software developers dedicated to pushing the envelope for software on MIDs and other consumer friendly devices. Open source software provides an excellent foundation for these devices. Through its use of well proven open source technologies, Moblin provides a unique platform where open source developers can leverage their knowledge and experience to help create a new class of devices that will enable end-users to carry the full Internet in their pocket. We encourage you to help us make this vision reality by joining us at http://www.moblin.org

About The Author

Derek Speed is a Senior Technologist in the Intel Open Source Technology Center (OTC) where he is responsible for technical collateral development and evangelism for Intel sponsored open source projects, including LessWatts.org and Moblin.org. Prior to his current role in the OTC, Derek managed the engineering team in OTC that was responsible for the initial launch of the Moblin.org project. Before joining the OTC, he held a number of engineering management positions in the Cellular and Handheld Group, supporting Linux and other mobile operating systems on Intel® XScale™ microprocessors. He joined Intel in 1998 from Digital Equipment Corporation where he spent 15 years in a variety of software engineering and technical marketing management roles. Derek graduated from Worcester Polytechnic Institute (WPI) in 1983 with a Bachelor of Science degree in Electrical Engineering (with distinction) and a minor in Music.