

AN IN-DEPTH AND COMPREHENSIVE LOOK AT THE MENTOR EMBEDDED LINUX DEVELOPMENT PLATFORM

The platform built upon shared Freescale and Mentor Embedded technologies, enables developers to create innovative and highly customized applications supporting Freescale's QorIQ and PowerQUICC processors.

The unique strategic alliance signed between Freescale Semiconductor and Mentor Embedded, has resulted in the release of a commercial Linux® Platform for Freescale's QorIQ and PowerQUICC processors. This product, the Mentor Embedded™ Linux platform, is a highly streamlined and fully integrated solution that assists development teams as they move from evaluating on reference platforms to developing on custom-designed hardware. Coupled with a Freescale reference board, the Mentor Embedded Linux platform provides a smooth transition from evaluation to development with no restarts enabling customers to develop customized applications with no vendor lock-in and with more reliability, while accelerating design development time. This continuity of workflow is a compelling competitive advantage offered exclusively by Mentor Embedded Linux.

INTRODUCING MENTOR EMBEDDED LINUX

Mentor Embedded Linux offers developers a new approach to benefit from the vast world of open source software without the trade-offs associated with existing options. Traditional commercial Linux vendors bind developers to a limited set of customization options preventing projects from fully adapting the commercial product to their project requirements. The “roll-your-own” Linux option introduces business risk by distracting development teams from their value-add contributions.

Mentor Embedded Linux brings a new option that fully empowers developers to create their own unique product platform with the commercial support from Mentor Graphics. Mentor Embedded Linux maximizes developer productivity with powerful developer tools and a new approach to harnessing the power of open source and Linux for engineering effectiveness.

THE MENTOR EMBEDDED LINUX OFFERING

Mentor Embedded Linux comes with a ready-to-use root filesystem, including a Linux kernel, for each reference board supported by the product. This reference distribution contains all of the required components for preparing your own unique product platform to meet your product's requirements. Figure 1 represents the major components that comprise Mentor Embedded Linux, including the traditional items that developers expect when they work with a commercial partner – support, warranty and indemnification coverage, BSPs, compilers, and community originated

source code. Within the Mentor Embedded Linux offering are the primary developer tools Mentor Embedded™ System Builder and Mentor Embedded™ EDGE Developer Suite – key technologies that distinguish this product from other competitive offerings on the market today.

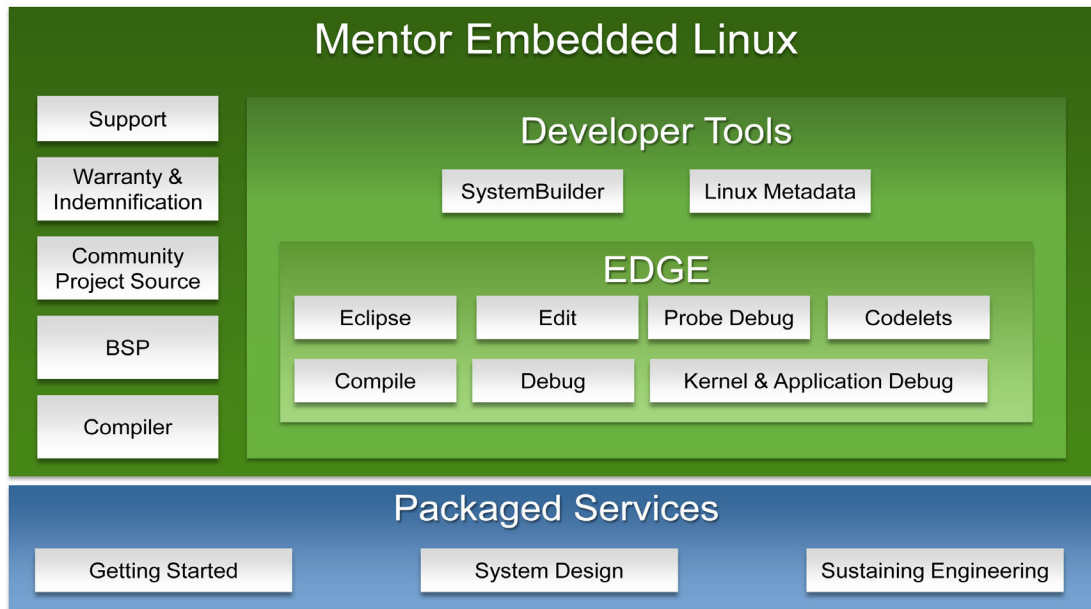


Figure 1: The complete Mentor Embedded Linux offering.

DEVELOPER TOOLS

Mentor Embedded Linux provides developers with a complete developer tools experience that goes far beyond the minimum command line compilers and debuggers.

SYSTEM BUILDER

A classic distribution of a typical development environment is set up for approximately 250 packages. The development engineer has to craft this down just to get the runtime desired. Further, when handed over to the application developer, it could possibly employ a library which is not intended to be used in the platform. The performance could be impacted due to the application being built on top of a larger system rather than the exclusive runtime of that product.

To address these obstacles, Mentor Embedded Linux offers a set of integrated, open source software components built into the easy-to-use System Builder development environment for board-level port, driver, and application development. System Builder is a command line build and customization tool used to prepare cross compilers, the glibc C library, pre-configured file system images, and product-specific supporting libraries.

In addition, System Builder contains a set of metadata used to cross-compile and package software packages into a Platform Development Kit (PDK) or Application Development Kit (ADK). System Builder uses “recipes” and “collections” to define what to build and how to build it.

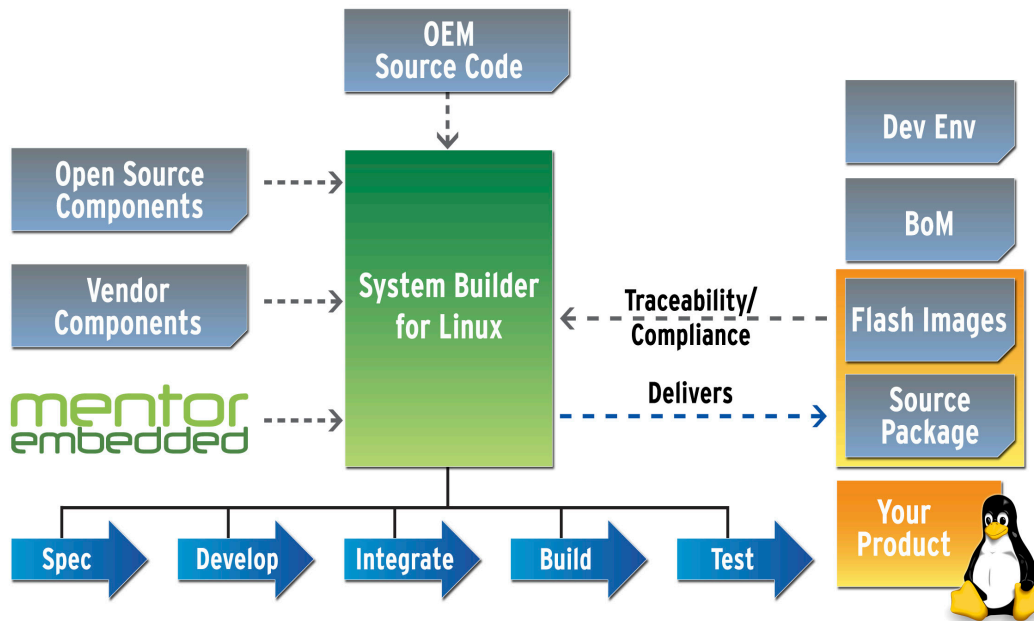


Figure 2: The central role of System Builder during Linux platform development.

As indicated in Figure 2, System Builder also allows users to acquire and integrate embedded Linux platform code from Mentor Embedded’s software IP collection or other project repositories, such as kernel.org. A complete hardware specific Linux system is provided as a pre-integrated collection in build system. Kernel source, including configuration files for unique hardware, is also included.

Key features include:

- Fully customizable, industry standard, command line build system for customizing a unique product platform
- Pre-built binaries to avoid tedious hours of waiting to get to work
- May be rebuilt from source using a preconfigured build system at any time
- Kernel preconfigured to hardware
- Complete hardware specific Linux system, integrated into the build system

-
- License flagging and binary-to-source matching captured in the software Bill of Materials (BoM)

Benefits:

- Development teams can reach product goals quicker and with less risk by relying upon an industry standard software workflow enhanced by Mentor Embedded's performance, stabilization, and compliance features.
- Users enjoy commercial product support without proprietary vendor lock-in.
- Ensures repeatable product builds that always produce best-practice, open-source compliance documents.

INTEGRATING APPLICATIONS AND CUSTOMIZATIONS

Simply rebuilding Linux from source isn't enough. Developers need the power to customize and create their own unique product platform. This is where the Mentor Embedded Linux System Builder can help. Unlike alternatives that just re-build source per the original vendor specification, System Builder opens the door to developer customization.

These customizations include:

- Adding software packages to the device images
- Reconfiguring Flash device storage
- Patching or reconfiguring the Linux kernel
- Patching or reconfiguring Linux software packages
- Modifying system configuration files
- Building an application from source
- Including an application's pre-built binaries

Further, System Builder can be used as either the master build system or as a component of a larger pre-existing workflow.

KEEP APPLICATION DEVELOPERS ON-TASK AND EFFECTIVE WITH ADKS

Once the product platform has been created, it is time to accelerate application development. The first task is to provide application developers with the necessary developer tools and a compatible Integrated Development Environment (IDE). System Builder can produce an ADK that ensures that application developers don't have to waste time learning how to build their own compilers or suffer from incompatible tools.

The following items are included in the ADK and are built to run a specific hardware platform.

- Linux kernel
- Filesystem images
- Host-runnable command line toolchain for one processor architecture
- Support libraries

-
- Firmware (U-Boot, etc)

The ADK can be used at the command line, so no System Builder training is required; or it can be imported to EDGE to automatically configure the IDE and build projects compatible with a developer's unique product platform.

Used either way, System Builder will speed application development by keeping teams focused on their value-add contribution.

BUILT FROM AND COMPATIBLE WITH INDUSTRY STANDARDS

System Builder is based upon and compatible with the industry standard OpenEmbedded (OE) project. The OpenEmbedded project, founded in 1999, is focused on providing cross development tools and package metadata which can be used by developers to build Linux distributions for their embedded devices. Unlike other proprietary build systems, System Builder offers more open source technology options and never locks the user into a proprietary workflow.

Mentor Embedded adds value to OpenEmbedded technology by introducing new features such as pre-built binaries, the BOM tool, and license tagging. Mentor Embedded also enhances the opportunity for success by adding a Mentor Embedded Linux project setup tool that eliminates tedious configuration steps that can hamstring developers who are new to OpenEmbedded technology. By stabilizing the OpenEmbedded metadata and mirroring sources, Mentor Embedded assures customers that their builds will be successful and repeatable for years to come.

Mentor Embedded engineers are also leading contributors to both the OpenEmbedded metadata and the bitbake tool at the core of OpenEmbedded.

TOOLS TO DRIVE OPEN SOURCE LICENSE COMPLIANCE

The usage of open source technology within embedded system projects can deliver superior results, but introduces new license obligations that mandate an organized and repeatable compliance process built into every workflow. Mentor Embedded Linux is delivered with all of the necessary tools to automate this compliance process. All that is missing are the development team/organization's internal policies and procedures. Together with these policies, the compliance features delivered by the Mentor Embedded Linux System Builder can result in a low overhead, yet effective compliance program that avoids costly oversights.

The Mentor Embedded Linux System Builder component provides four key features to aid in the open source compliance program:

- **Software BoM Creation:** Fully documents packages, versions, licenses, upstream source, patches, and dependencies for both development and production builds.
- **License Auditing:** Flags open source licenses under a user's direction and highlights compliance obligations in adherence with a company's specified open source compliance policies.
- **Source Package:** Builds comprehensive device project sources and creates source packages tailored to meet licensing disclosure requirements.

-
- **License Traceability:** Helps support tracking and verification of matched sources and binaries for redistribution, license compliance, maintenance, and end-product customer support.

COMPLETE HOST OPERATING SYSTEM SUPPORT

Developers using OpenEmbedded technology for the first time often suffer from aborted builds due to missing or incompatible host support packages. The Mentor Embedded Linux System Builder ensures that all required host components are provided in a compatible version for the supported host platforms. This enhancement ensures successful and repeatable builds on supported host platforms.

In summary, System Builder provides quality without sacrificing schedule, resources, and cost. It presents a cohesive approach to a system thorough understanding as it evolves from small build to exactly what is needed to meet end-product requirements. In addition to System Builder, Mentor Embedded Linux also includes a special version of the EDGE Developer Suite introducing new Linux-specific technologies to enhance developer effectiveness.

EDGE DEVELOPER SUITE

The Mentor Embedded EDGE™ Developer Suite, an IDE built upon the Eclipse™ framework, provides tools to speed Linux-based product development giving developers insight into their application that cannot be obtained using traditional command line debugging tools. Developers need, first and foremost, a powerful debug solution and the EDGE Developer Suite delivers Linux-specific features that can tackle debug challenges in applications that include both traditional Linux userspace applications and Linux kernel components.

Capabilities include:

- Users can look inside their application as it runs
- Assists multi-thread application comprehension
- Automates repetitive tasks
- Custom register views
- No file system burden

LINUX DEBUG AGENT

Unlike traditional enterprise computing applications, most embedded software applications are not just simple userspace applications. Debugging these applications the traditional way requires developers to juggle multiple debug connections and agents that don't necessarily cooperate and do not present a complete picture to the developer. The Linux Debug Agent transcends these limitations to give developers deeper insight into how their application is interacting with both userspace and Linux kernel software. This is a kernel-resident module whose responsibility is to help the system fully debug both multiple applications in userspace, kernel modules, and the supporting kernel functionality that the applications are employing. This provides a single connection to the target system and the ability to see all the applications.

EDGE DEBUGGER

EDGE Debugger is a powerful debugging environment with Linux-specific enhancements. It goes beyond standard debug features – including advanced scripting, I/O visibility, and Linux awareness capabilities – to help embedded software engineers reduce the large amounts of time spent debugging. Seamlessly integrated within the Eclipse-based EDGE IDE, EDGE Debugger offers a powerful debug experience to see inside both the Linux kernel and applications on real target.

Capabilities include:

- Advanced Interface
- Unparalleled Customization via Codelets
- Flexible Data Visualization via Channel Viewers
- Straight-Forward Target Customization
- Configurable Register View

A fundamental requirement for embedded software development is the ability to connect a debugger running on a host computer (like a PC) to the target system. EDGE Debugger connects to processors via the standardized JTAG debug logic IP macrocell and supports the CodeWarrior USB TAP host target interface for debugging the Power Architecture.

LINUX KERNEL INSIGHT

The EDGE Debugger includes a special library of Codelets that can provide developers with insight into the internal operation of the Linux kernel. The Linux Codelet Library contains a set of prepared Codelets that can interpret kernel data structures and render those for viewing by the developer in the EDGE IDE. The Codelets render these data structures when the mouse cursor is held over a variable in the editor.

EDGE IDE

EDGE IDE is a unified workspace for embedded. It's an integrated development environment (IDE) that brings together all aspects of software development for Linux applications into a single, unified workspace.

Capabilities include:

- Built-in Advanced Editor
- Project Manager
- Wizard-Based Project Creation
- Flexibility and Extensibility

EDGE COMPILER

Every embedded developer needs a compiler toolkit – a set of tools that takes embedded C, C++, and assembly language source code and performs all the processing to generate a final binary image. Projects built on Mentor

Embedded Linux are no different. System Builder creates the correct compiler toolkit that corresponds to each unique product platform. The EDGE IDE uses this compiler toolkit to help developers create their application.

LINUX ADKs FROM SYSTEM BUILDER

EDGE IDE can import ADKs built by System Builder and use the bundled compilers and settings to automatically and correctly setup application projects. Difficult to maintain project settings are now handled automatically by this innovative feature. EDGE Compiler represents a complete code development solution for embedded developers.

MENTOR EMBEDDED LINUX SOFTWARE IP

Mentor Embedded Linux comes with a ready-to-use root filesystem (including a Linux kernel) for each reference board supported by the product. This reference distribution contains all of the required components for preparing a unique product platform to meet specific product requirements. System Builder can be used to customize this reference platform as needed.

Using System Builder developers can:

- Review the original source code and patches applied to create Mentor Embedded Linux.
- Add or remove packages from the reference root file system images.
- Rebuild all of the source code that contributes to reference configuration.
- Add software from outside Mentor Embedded Linux by pulling recipes from the OpenEmbedded project.

Mentor Embedded Linux includes a minimal filesystem starting point that can be used as the baseline for adding just what the product platform requires to be successful. To aid developers as they add to this baseline filesystem, “control knobs” are available as a shortcut to selecting the appropriate packages. While developers are still responsible for ensuring that the corresponding kernel drivers are available filesystem control knobs save time and reduce guesswork when adding features to a design.

Available control knobs include:

- **audio:** Adds required sound support utilities
- **bluetooth:** Adds Bluetooth support utilizing the BlueZ project
- **host-diskdrive:** Adds utilities for managing disks and managing the integrity of disk filesystems
- **framebuffer:** Adds support utilities for the Linux framebuffer devices
- **host-usb:** Adds utilities to manage USB devices and work with filesystems commonly found on USB mass storage devices.
- **mtdev:** Adds the Linux Memory Technology Devices utilities
- **pci:** Adds basic utilities for querying PCI devices
- **touchscreen:** Adds touchscreen calibration utilities

- **wifi:** Adds necessary tools for connecting to wireless networks
- **print-server:** Adds utilities for serving printers to remote devices via the Samba file and print services daemon
- **nas:** Adds utilities for connecting to remote filesystems using either Samba or NFS
- **native-development:** Basic required tools for on device software development
- **router:** Adds utilities for providing routing, firewall, and NTP services
- **x11-minimal:** Adds basic X11 environment

(Please note: BSP support for individual control knobs will vary depending on peripheral and device driver support.)

CUSTOMER SUPPORT

The full resources of the Mentor Graphics customer support organization stands behind Mentor Embedded Linux. Support and maintenance are crucial to customers, so the Mentor Embedded Linux product includes comprehensive support for development tools and core open source technologies. Twice a year, users will receive access to stable and compatible open source technology refreshes, component version updates, and cumulative defect resolutions roll-ups. Additional consultation is available through Mentor Embedded's Professional Services, which can assist in expert development in all phases of open source development.

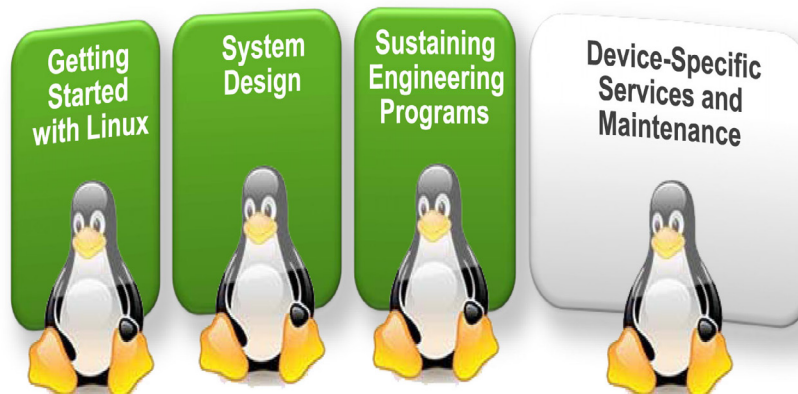


Figure 3: Mentor Embedded Professional Services offers a full complement of packaged and custom services to support developers building their device with Mentor Embedded Linux.

PACKAGED PRACTICES/PROFESSIONAL SERVICES

To provide end-to-end support for Mentor Embedded Linux, Mentor Embedded Professional Services offers a full range of packaged practices and custom services to complement and support the development effort (Figure 3). These packaged practices are offered exclusively by Mentor Embedded and are supported by the experienced team of Mentor open source technologists. These packages include Getting Started, System Design, and Sustaining Engineering.

GETTING STARTED: BUILDING A DEVICE ON LINUX

How can developers best use the powerful tools provided in Mentor Embedded Linux? What are the unique challenges and opportunities of the current engineering workflow and development infrastructure when a team begins to develop on Linux? Mentor Embedded Professional Services offers three days of a senior Linux expert evaluation and consultation at your development site anywhere in the world. Mentor provides a hands-on overview of Linux development, reviews unique requirements of the device, and maps them to a workflow and processes. Following the on-site visit, developers receive a full written report detailing findings and recommendations.

SYSTEM DESIGN

When creating a next generation device, or migrating from another operating system, it is often best to initiate a full architectural review of the software planning at an early phase of development. Mentor can provide an on-site consulting engagement with a senior system architect to identify and evaluate the design of record, and suggest the best directions and technologies for the product under development. The resulting written report details findings and recommendations to help determine both overall structure of system architecture and the specifics required to build an efficient, high quality device.

SUSTAINING ENGINEERING PROGRAMS

Product lifecycle management can be particularly challenging once a device has been shipped. Changes to software to repair a security hole or fix a newly-discovered bug must be first identified, then resolved and delivered to all of these devices wherever they are now located. A sustaining engineering package from Mentor Embedded can provide a valuable link in this process. Mentor starts by finding the defects that are specifically relevant to the device software build, then reports and triages those defects, with regular monthly notification and defect resolution tiered to severity and the level of sustaining engineering purchased.

SUMMARY

Mentor Embedded Linux offers a powerful platform and highly streamlined development approach that permits developers to take full advantage of open source software. Built upon Mentor Embedded and Freescale technology, Mentor Embedded Linux provides Software IP, comprehensive developer tools, and stellar support and maintenance. The platform is a fully integrated vendor independent solution with no lock-in. Customers can move seamlessly from evaluation to product development not having to restart each time. Mentor Embedded Linux maximizes developer productivity with powerful developer tools, a new approach to harnessing the benefits of open source along with award winning support and maintenance. Mentor Embedded Linux is an unrivaled, flexible solution that enables the delivery of a highly customized end product.

The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Visit www.mentor.com/embedded for the latest product information.

©2010 Mentor Graphics Corporation. All rights reserved.

This document contains information that is proprietary to Mentor Graphics Corporation and may be duplicated in whole or in part by the original recipient for internal business purposes only, provided that this entire notice appears in all copies. In accepting this document, the recipient agrees to make every reasonable effort to prevent unauthorized use of this information.

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777 USA
Phone: 503.685.7000
Fax: 503.685.1204
Sales and Product Information
Phone: 800.547.3000

Silicon Valley
Mentor Graphics Corporation
1001 Ridder Park Drive
San Jose, California 95131 USA
Phone: 408.436.1500
Fax: 408.436.1501
North American Support Center
Phone: 800.547.4303

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
Room 1001, 10F
International Trade Building
No. 333, Section 1, Keelung Road
Taipei, Taiwan, ROC
Phone: 886.2.87252000
Fax: 886.2.27576027

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Garden
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo
Japan 140-0001
Phone: +81.3.5488.3030
Fax: +81.3.5488.3021

