

## KIOXIA Software-Enabled Flash™ Technology: See How a Simple SDK Accelerates Application Development

Software-Enabled Flash technology brings unparalleled control of flash memory to storage developers through a new, software-defined, open source application programming interface (API). The API, in conjunction with purpose-built Software-Enabled Flash hardware, enables applications to define their desired latency outcomes, performance isolation requirements, data placement strategies and thin provisioning specifications using calls to this low-level API.

The Software-Enabled Flash Software Developer's Kit (SDK) builds on the API (Figure 1) to provide, among other features, a customizable application-aware flash translation layer (FTL). Specialized applications with unique data placement requirements can use this reference FTL to implement their own customized FTL to maximize flash memory endurance while tuning the flash storage for optimal application performance.

The SDK includes source code for components and utilities that simplify and accelerate application prototyping and deployment. It is built to work seamlessly with Software-Enabled Flash hardware to be used as a code base from which storage developers can build and deploy their own applications. All components are provided in open source code form, empowering developers with the freedom to modify, extend and integrate the code within their own environments.

### SDK Components

The Software-Enabled Flash SDK consists of four major components:

- *Reference FTL implementation*
- *Reference Virtual Machine (VM) input/output (I/O) driver*
- *Command Line Interface (CLI)*
- *Benchmarking utility*

These four components provide a springboard for storage application development and allow Software-Enabled Flash technology to be evaluated without writing any code.

#### Reference FTL Implementation:

The reference FTL is a base upon which storage developers can build their own application-specific FTL. It manages the logical to physical data mapping to Software-Enabled Flash storage in the same manner as on-drive FTL modules in standard SSDs. It is built around a set of independent building blocks (Figure 2) that can be used in application development and customer-specific FTLs:

- **Instrumentation:** *Implements an extensible monitoring framework for Software-Enabled Flash applications*
- **Persistence:** *Handles the persisting of data and metadata in an application-agnostic manner*
- **Garbage Collection:** *Handles ad-hoc and automatic flash memory garbage collection, wear leveling and data integrity algorithms and fully utilizes Software-Enabled Flash hardware offload capabilities*
- **Superblock Management:** *Handles the lifecycle of individual flash memory super blocks*
- **Logical-to-Physical Translation (L2P):** *Implements a time and space efficient indirection table allowing for high performance and memory aware logical block address (LBA)-like access to native Software-Enabled Flash data*

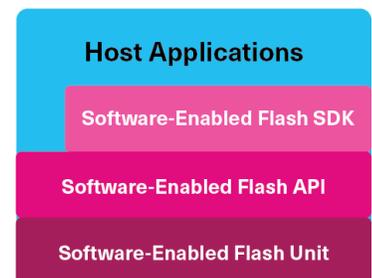


Figure 1: Software-Enabled Flash software stack

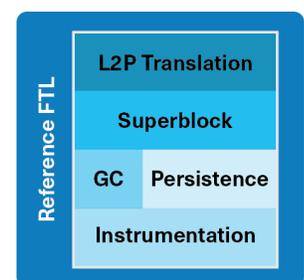


Figure 2: Software-Enabled Flash reference FTL building blocks

Each reference FTL layer has a documented set of software interfaces that can be mixed and matched with existing SDK code and new developer algorithms. For example, a storage developer could create a log-based database with a custom garbage collection algorithm to perform on-drive database compaction using Software-Enabled Flash technology offload capabilities. Alternatively, a cloud storage application could be developed using the L2P translation layer to implement a pseudo-file system, minimizing the number of indirections required to access data for high performance applications.

**Reference VM I/O Driver:**

Many workloads, including containerized ones, are run in VMs. The Software-Enabled Flash SDK VM I/O driver works with the open source QEMU<sup>1</sup> virtualization system to enable these use cases without requiring a change to any single line of code in the application. It also provides the VM hypervisor with extensive control over isolation, and latency outcomes to specify Quality of Service (QoS) levels.

The reference VM I/O driver utilizes the reference FTL implementation to provide an emulated block device for VMs. Devices that utilize standard block, an emulated NVMe<sup>®</sup> interface or a Zoned Namespace (ZNS) protocol are supported simultaneously. All of the control that Software-Enabled Flash technology provides, such as workload isolation and latency management, is supported at the hypervisor level for these devices (Figure 3).

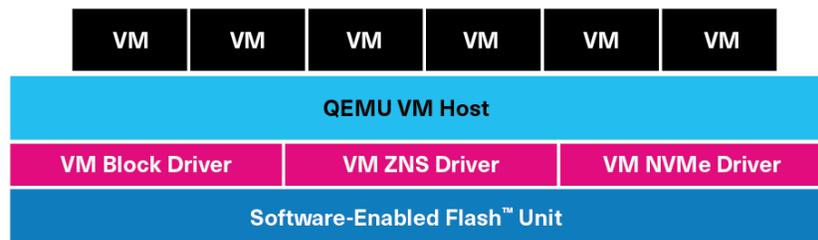


Figure 3: Reference VM I/O driver block diagram

**Command Line Interface (CLI):**

This scriptable command-line interface can be added to server deployment recipes for provisioning Software-Enabled Flash devices before their first use. The CLI enables the device to be configured and monitored, assigned physical workload isolation and provisioned for desired default latency profiles, while enabling FTL options such as overprovisioning to be adjusted. For complicated, programmatic configurations, the CLI can be used within server-side automation scripts.

**Benchmarking Utility:**

The Software-Enabled Flash SDK extends the open source Flexible I/O<sup>2</sup> (FIO) test tool with an engine that is capable of accessing Software-Enabled Flash storage directly. This enables rapid evaluation of features such as isolation and queuing configurations for application workloads prior to code development. Virtualized workloads can also be tested using the SDK reference VM I/O driver on the virtualization host using FIO in the VMs.

## Source Code and Licensing

The SDK is written in standard C which enables it to be included directly into C and C++ applications. This also allows for the creation of simple programs or codes that wrap around other programming languages such as C++, Go<sup>™</sup>, Rust<sup>™</sup> or the Java<sup>®</sup> platform.

The SDK is provided as source code and patches under various open source software licenses. Developers may examine, optimize and extend the functionality of the code. With open source licenses the improvements developers make to the SDK may, in general, be shared throughout the open source community, enhancing the product for all users.

## Accelerating Application Development

Software-Enabled Flash technology is a technology that brings software-defined to flash storage, enabling full flash memory capabilities, performance and scalability to be delivered into the data center and cloud applications. The SDK builds on this new technology by providing proven, easy-to-use and redistributable code for upgrading current applications or building new ones. The combination of a reference FTL, reference VM I/O driver, CLI and industry respected testing tool can help speed up development of Software-Enabled Flash applications and enable these applications to maximize the value of flash memory.

Software-Enabled Flash technology is an open source project that aligns the unique and untapped capabilities of flash memory with the specific requirements of cloud applications and hyperscale data center environments. More information is available at the KIOXIA Software-Enabled Flash technology home page, <https://softwareenabledflash.com>, with white papers, presentations and demonstrations. The open source API definition and specification documents are downloadable from the [KIOXIA repositories on the GitHub® site](#).

### NOTES:

<sup>1</sup> QEMU is an acronym for Quick Emulator and defined as a software module that supports full virtualization by providing emulation of various hardware devices. QEMU is a component of the hypervisor platform.

<sup>2</sup> Flexible I/O (FIO) is a free and open source disk I/O tool used both for benchmark and stress/hardware verification. The software displays a variety of I/O performance results, including complete I/O latencies and percentiles.

### Trademarks:

Java is a registered trademark of Oracle and/or its affiliates. GitHub is a registered trademark of GitHub, Inc. Go is a trademark of Google LLC. NVMe is a registered trademark of NVM Express, Inc. PCIe is a registered trademark of PCI-SIG. Rust is a trademark of the Mozilla Foundation in the U.S. and other countries. All other company names, product names and service names may be trademarks or registered trademarks of their respective companies.

### Disclaimers:

© 2021 KIOXIA America, Inc. All rights reserved. Information in this tech brief, including product specifications, tested content, and assessments are current and believed to be accurate as of the date that the document was published, but is subject to change without prior notice. Technical and application information contained here is subject to the most recent applicable KIOXIA product specifications.