

# ProtectServer 2 Migration Guide



THE  
DATA  
PROTECTION  
COMPANY

## Document Information

Product Version: 5.0

Document Part Number: 007-012737-001, Rev. B

Release Date: November 2014

Copyright © 2014 SafeNet, Inc. All rights reserved.

## Revision History

Revision	Date	Reason
A	27 October 2014	Beta release
B	11 November 2014	GA release

## Trademarks

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of SafeNet, Inc.

## Disclaimer

SafeNet makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, SafeNet reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon SafeNet to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

# CONTENTS

<b>Preface</b> .....	<b>4</b>
Customer Release Notes .....	4
Audience.....	4
Document Conventions .....	4
Notifications .....	4
Command Syntax and Typeface Conventions .....	5
Obtaining Technical Support .....	6
<b>Introducing ProtectServer 2</b> .....	<b>7</b>
Overview.....	7
Functionality Modules.....	7
Serial Devices .....	8
Software Changes.....	8
FM SDK (formerly PPO) is now included with the PTK software .....	8
Installation Directories .....	8
Environment Variables.....	8
DVD Directory Structure .....	9
<b>FM Migration</b> .....	<b>11</b>
Supported Hardware and Software .....	11
Summary of Changes and Enhancements .....	11
FM SDK Toolkit.....	12
Toolchain .....	12
Makefiles.....	13
Memory Endian Issues .....	14
Emulation Mode Enhancements.....	14
FM Certificates.....	15
FM Debug Logging Using printf .....	15
Compile-Time Checking .....	16
The clock() Function Uses the ANSI/ISO Standard for Returning CPU Time .....	16
The integers.h Header File Removed From \$(FMSDK)/include.....	16
Obsolete FM SDK Copy Function Removed .....	16
Support Removed for libfmhost .....	16
Failed FMs .....	16
Migrating Your FMs .....	17

# Preface

This document describes how to migrate from the legacy ProtectServer HSMs (PSI and PSI-E) to the upgraded ProtectServer 2 HSMs (PSI2 and PSI-E2). It contains the following chapters:

- “Introducing ProtectServer 2
- “FM Migration” on page 11

## Customer Release Notes

The Customer Release Notes (CRN) document provides important information about this release that is not included in the customer documentation. It is strongly recommended that you read the CRN to fully understand the capabilities, limitations, and known issues for this release. You can view or download the latest version of the CRN for this release at the following location:

[http://www.securedbysafenet.com/releasenotes/ptk/crn\\_ptk\\_5-0.pdf](http://www.securedbysafenet.com/releasenotes/ptk/crn_ptk_5-0.pdf)

## Audience

This document is intended for personnel responsible for maintaining your organization's security infrastructure. This includes Luna HSM users and security officers, the key manager administrators, and network administrators. It is assumed that the users of this document are proficient with security concepts.

All products manufactured and distributed by SafeNet, Inc. are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

## Document Conventions

This section provides information on the conventions used in this document.

### Notifications

This document uses notes, cautions, and warnings to alert you to important information that may help you to complete your task, or prevent personal injury, damage to the equipment, or data loss.

### Notes

Notes are used to alert you to important or helpful information. They use the following format:

---

**Note:** Take note. Notes contain important or helpful information.

---

## Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. They use the following format:

---

### CAUTION

Exercise caution. Caution alerts contain important information that may help prevent unexpected results or data loss.

---

## Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. They use the following format:

---

### WARNING!

Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.

---

## Command Syntax and Typeface Conventions

**Table 1: Syntax and Typeface Conventions**

Convention	Description
<b>bold</b>	The bold attribute is used to indicate the following: <ul style="list-style-type: none"> <li>• Command-line commands and options (Type <b>dir /p</b>.)</li> <li>• Button names (Click <b>Save As</b>.)</li> <li>• Check box and radio button names (Select the <b>Print Duplex</b> check box.)</li> <li>• Dialog box titles (On the <b>Protect Document</b> dialog box, click <b>Yes</b>.)</li> <li>• Field names (<b>User Name</b>: Enter the name of the user.)</li> <li>• Menu names (On the <b>File</b> menu, click <b>Save</b>.) (Click <b>Menu &gt; Go To &gt; Folders</b>.)</li> <li>• User input (In the <b>Date</b> box, type <b>April 1</b>.)</li> </ul>
<i>italic</i>	In type, the italic attribute is used for emphasis or to indicate a related document. (See the <i>Installation Guide</i> for more information.)
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.
[ optional ] [ <optional> ]	Represent optional keywords or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.
{ a   b   c }	Represent required alternate keywords or <variables> in a command line description. You

Convention	Description
{ <a>   <b>   <c> }	must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.
[ a   b   c ] [<a>   <b>   <c>]	Represent optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.

## Obtaining Technical Support

If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or SafeNet support. SafeNet support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between SafeNet and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

**Table 2: SafeNet Technical Support Contacts**

Contact Method	Contact Information
<b>Address</b>	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017 USA
<b>Phone</b>	United States (800) 545-6608, (410) 931-7520
	Australia and New Zealand +1 410-931-7520
	China (86) 10 8851 9191
	France 0825 341000
	Germany 01803 7246269
	India +1 410-931-7520
	United Kingdom 0870 7529200, +1 410 931-7520
<b>Web</b>	<a href="http://www.safenet-inc.com">www.safenet-inc.com</a>
<b>Support and Downloads</b>	<a href="http://www.safenet-inc.com/Support">www.safenet-inc.com/Support</a> Provides access to the SafeNet Knowledge Base and quick downloads for various products.
<b>Technical Support Customer Portal</b>	<a href="https://serviceportal.safenet-inc.com">https://serviceportal.safenet-inc.com</a> Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the SafeNet Knowledge Base.

# CHAPTER 1

## Introducing ProtectServer 2

This chapter provides an introduction to the new ProtectServer 2 HSMs. The information in this chapter provides a high-level overview of what is new or changed, and how these differences may impact you when you migrate from a legacy ProtectServer HSM to a new ProtectServer 2 HSM. It contains the following sections:

- “Overview” on page 7
- “Functionality Modules” on page 7
- “Serial Devices” on page 8
- “Software Changes” on page 8

### Overview

ProtectServer 2 is a major upgrade to the ProtectServer product line. ProtectServer2 introduces the PSE2 and PSI-E2 HSMs as direct replacements for the legacy PSE and PSI-E HSMs, which have been declared end of sale, and are no longer available for purchase.

Although the new PSE2 and PSI-E2 HSMs are functionally equivalent to their legacy counterparts, the underlying hardware is significantly different. The major hardware change is to the embedded cryptographic engine used on the HSMs. The legacy PSE/PSI-E HSMs incorporate the K5 cryptographic engine. The new PSE2/PSI-E2 HSMs incorporate the more modern K6 cryptographic engine.

Although every effort has been made to mitigate the impact of these hardware changes, the introduction of a new cryptographic engine impacts the following:

- Functionality modules (FMs). The processor used on the ProtectServer 2 HSMs is different from the processor used on the legacy ProtectServer HSMs. As a result, you must rebuild your FMs to run on the new hardware.
- Serial devices. The serial port on the ProtectServer HSMs has been replaced on the ProtectServer 2 HSMs with a USB port and a USB-to-serial cable. Any serial devices that were previously attached to a ProtectServer HSM will continue to work on a ProtectServer 2 HSM.

In addition to these changes, ProtectServer 2 also includes some software fixes/enhancements, as described in “Software Changes” on page 8.

### Functionality Modules

The K5 cryptographic engine is based on the ARM processor. The K6 cryptographic engine is based on the PowerPC processor. As a result, any FMs built for the PSE/PSI-E (K5) HSMs will not run on the new PSE2/PSI-E2 (K6) HSMs. You must rebuild your existing FMs to run on the PowerPC (K6) platform. This requires a Linux workstation or VM and some changes to your source files, as described in “FM Migration” on page 11.

## Serial Devices

The new PSE2/PSI-E2 HSMs do not include a serial port. The serial port on these HSMs has been replaced with a USB port that provides serial access to the HSMs via the included USB-to-serial cable.

## Software Changes

The software changes introduced in this release primarily affect the FM SDK, as detailed in “FM Migration” on page 11. Any additional changes are described in the following sections.

### FM SDK (formerly PPO) is now included with the PTK software

In previous releases, the FM SDK (PPO) was provided on a separate CD. The FM SDK software is now bundled with the PTK software on the software DVD. Documentation for all of the software is provided on the PTK documentation DVD.

### Installation Directories

The installation directories have been modified to conform to SafeNet standard conventions, as follows:

<b>Linux</b>	/opt/safenet/protecttoolkit5 /opt/safenet/fm-toolchain
<b>Windows</b>	C:\Program Files\SafeNet\Protect Toolkit 5

### Environment Variables

Environment configuration for the PTK-C SDK and FM SDK has been simplified in this release as follows. Manual setting of environment variables is no longer required.

<b>Linux</b>	A configuration script ( <b>setvars.sh</b> ) is now included with PTK-C to configure your development environment. You would typically run this script each time you open a new shell. See the installation documentation for more information.
<b>Windows</b>	The runtime environment is automatically configured as part of the installation process. The FM SDK installation directory includes a configuration batch ( <b>fmsdkvars.bat</b> ) file to configure your FM development environment. You would typically run this batch file each time you open a new shell. See the installation documentation for more information.

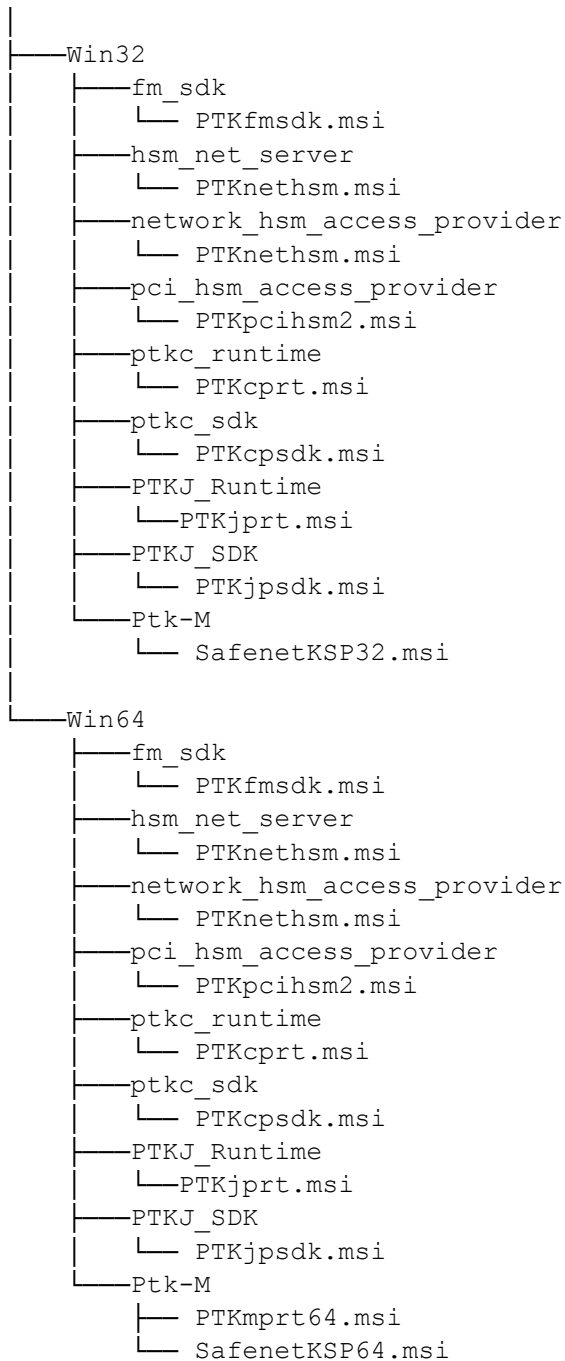


## DVD Directory Structure

```

├──<part_number>_sw_license_agreement.pdf
├──<part_number>_sw_license_agreement.txt
├──autorun.inf
├──firmware
│   └──<firmware_upgrade_files>
├──SDKs
│   ├── safeNet-install.sh
│   ├── Linux
│   │   ├── fm_sdk
│   │   │   └── PTKfmsdk-<version>.i386.rpm
│   │   ├── fm_toolchain
│   │   │   └── fm-toolchain-ppc440e-<version>.i686.rpm
│   │   ├── hsm_net_server
│   │   │   └── PTKnetsrv-<version>.i386.rpm
│   │   ├── network_hsm_access_provider
│   │   │   └── PTKnethsm-<version>.i386.rpm
│   │   ├── pci_hsm_access_provider
│   │   │   └── PTKpcihs2-<version>.i386.rpm
│   │   ├── ptkc_runtime
│   │   │   └── PTKcppt-<version>.i386.rpm
│   │   ├── ptkc_sdk
│   │   │   └── PTKcpsdk-<version>.i386.rpm
│   │   ├── ptkj_runtime
│   │   │   └── PTKjprov-<version>.i386.rpm
│   │   └── ptkj_sdk
│   │       └── PTKjpsdk-<version>.i386.rpm
│   ├── Linux64
│   │   ├── fm_sdk
│   │   │   └── PTKfmsdk-<version>.x86_64.rpm
│   │   ├── fm_toolchain
│   │   │   └── fm-toolchain-ppc440e-<version>.i686.rpm
│   │   ├── hsm_net_server
│   │   │   └── PTKnetsrv-<version>.x86_64.rpm
│   │   ├── network_hsm_access_provider
│   │   │   └── PTKnethsm-<version>.x86_64.rpm
│   │   ├── pci_hsm_access_provider
│   │   │   └── PTKpcihs2-<version>.x86_64.rpm
│   │   ├── ptkc_runtime
│   │   │   └── PTKcppt-<version>.x86_64.rpm
│   │   ├── ptkc_sdk
│   │   │   └── PTKcpsdk-<version>.x86_64.rpm
│   │   ├── ptkj_runtime
│   │   │   └── PTKjprov-<version>.x86_64.rpm
│   │   └── ptkj_sdk
│   │       └── PTKjpsdk-<version>.x86_64.rpm

```



# CHAPTER 2

## FM Migration

This chapter describes the changes and enhancements introduced in ProtectServer 2 to the FM development process. It provides guidance and recommends best practices for developing new FMs or migrating your existing FMs to work on the new ProtectServer 2 hardware. This chapter contains the following sections:

- “Supported Hardware and Software” on page 11
- “Summary of Changes and Enhancements” on page 11
- “Migrating Your FMs” on page 17

### Supported Hardware and Software

FM, HOST or Cryptoki applications built or recompiled using the ProtectServer 2 FM SDK are supported on the PSI-E2 and PSE2.

Recompiling your existing FM, HOST, and Cryptoki application source code should require makefile modifications only. FMs must be compiled on Linux instead of Windows.

Platform support is as follows:

<b>Embedded FM development</b>	Linux only
<b>FM HOST applications</b>	Windows and Linux

The minimum required installation for building FM HOST applications with PTK 5 on Windows is Microsoft Visual C++ (the 2005, 2008, and 2010 runtime redistributables must all be installed), PTKcpsdk, and PTKfmsdk. Compilers other than Microsoft Visual C++ may work, but our examples and configuration makefile (<FMDIR>\cfgbuild.mak) are built for Microsoft Visual C++.

### Summary of Changes and Enhancements

This section provides a summary of the changes and enhancements made to the FM SDK and toolchain. In most cases, your existing FMs should continue to work after recompiling on the new platform.

The most significant change to the FM development process in ProtectServer 2 is that FMs must now be compiled on Linux instead of Windows. This will require moving your source to a Linux workstation or VM. Recompiling in the new environment should not require any source changes.

If your existing FMs use the makefile syntax used in the samples provided in previous ProtectServer releases, you must update your makefiles to use the syntax defined in the samples provided with ProtectServer 2.

Some of the APIs used in previous ProtectServer releases have been changed or deprecated, however the header files provided with ProtectServer 2 include backwards compatibility for the old APIs. Warnings for the deprecated APIs will be displayed at compile time.

## FM SDK Toolkit

The FM SDK cross-compiler is now Linux-only.

There are no longer separate FM and HOST toolkits. All FM-SDK toolkits are the same, except that the cross compiler and embedded libraries will only be available on Linux hosts, and are no longer supported on Windows XP.

## Toolchain

The major changes to the toolchain are as follows:

- GCCFMDIR is no longer required
- gnumake is now make, and you can now use the native system make
- C99 support
- additional function support
- the tool naming convention has changed
- some of the tools in the toolchain use newer versions

### \$GCCFMDIR No Longer Required

\$GCCFMDIR is not required for PSI-E2 FM building since the toolchain is non-relocatable.

---

**Best practice:** Ensure that \$GCCFMDIR is not defined.

---

### Builds Use make Instead of gnumake

In ProtectServer 2, **gnumake** becomes simply **make**, and the host's native **make** can be used. If the host's native **make** is older than 3.82, **make** can also be used from the fm-toolchain from the following location:

```
/opt/safenet/fm-toolchain/fmsdk-ppc440e-1.0/sysroots/i686-fmsdk-linux/usr/bin/make
```

In previous releases, we provided GNU make (renamed to **gnumake**). The **gnumake** command was used to build FMs. This has changed as follows:

<b>Linux</b>	The <b>gnumake</b> command is no longer available. If you use an automated build that calls <b>gnumake</b> you must update your build scripts to call <b>make</b> instead of <b>gnumake</b> .
<b>Windows</b>	The <b>gnumake</b> command is provided in <fm_install_dir>/bin. It is simply a renamed version of the <b>make</b> command.

### C99 Support

The FM SDK supports a subset of the ISO C 99 standard library as defined by ISO/IEC 9899:1999. In general, floating point math, multibyte characters, localization, and I/O APIs are not supported. **printf** and **vprintf** are exceptions, and are redirected to the logging channel.

In addition to the standard library, you can also use C99 language features not present in ANSI C (C89/90). C99 **stdint.h** types are now supported by the FM SDK toolchain, however sized types from the proprietary

**Integers.h** remain used for FM SDK published APIs to maintain continuity and compatibility with pre-C99 versions.

Due to the change to C99, and the default C99 locale, the **strftime %x** result is a different format than in PPO3. It is now **mm/dd/yy** instead of **Ddd Mmm dd yyy**.

See the *FM SDK Programming Guide* for more information.

---

**Best practice:** Update your code to take advantage of the features offered by C99.

---

## Functions added to the FM SDK in ProtectServer 2

The following functions have been added:

### ctype.c:

isblank

### stdio.h

printf, vprintf, vsscanf  
snprintf (moves from non-standard to C99 variant).

### stdlib.h

atoll, labs, lldiv, strtoll, strtoull

## Updated Tool Versions

Tool	PPO 3.0 version	FM SDK 5.0 version
gcc	2.95.3	4.6.1
gnu make	3.78.1	3.82
binutils	2.11.2	2.21.1
C standard	C89	C99

## New Tool Naming Convention

The tool naming convention changes from `<toolname>-fm` to `<arch>-fm-<toolname>`. For example **gcc-fm** becomes **powerpc-fm-linux-gcc**.

## Makefiles

The emulation mode makefile syntax used in ProtectServer 2 is different than in previous releases. See the makefiles in the **samples** directory for the new syntax. You must change your existing makefiles to conform to the new syntax.

If your existing makefiles include **cfgbuild.mak**, you should be able to recompile existing non-emulation code without making any changes, other than changing the path separators, as described below. If you only used **cfgbuild.mak** as an example, changes will be required in your makefiles for the new toolchain. See the makefiles in the **samples** directory, and the **cfgbuild.mak** file.

## Path Separators

When building for PSI-E2 on Linux hosts, all path separators in your makefiles must use forward slashes (/). Windows-style backslashes (\) will not work.

---

**Best practice:** Change all backslashes (\) to forward slashes (/) in your makefiles.

---

## The cfgbuild.mak File

For both FMs and HOST applications, **cfgbuild.mak** is now a supported file and not simply an example. The **cfgbuild.mak** file has been moved to \$(FMDIR)/ from \$(FMDIR)/**samples**. A wrapper is provided in \$(FMDIR)/**samples** for compatibility.

The default libraries provided by the FM SDK are now automatically included by **cfgbuild.mak** and should not be specified within the FM's makefile directly.

---

**Best practice:** Include **cfgbuild.mak** in all of your makefiles to set up the compiler, link flags, and link the FM toolchain.

---

## Memory Endian Issues

The ProtectServer 2 processors are big endian, whereas the PSI-E and PSG processors are little endian. ProtectServer 2 consolidates all methods of endian byte order manipulation into the **endyn.h** header file.

---

**Best practice:** Use the endian macros provided in the PTK-C header file **endyn.h** to encode all messages in network byte order. By using the endian macros on both host and FM, endian differences between host and HSM are not an issue.

---

## FM\_MAKE\_VERSION Macro

The samples included in previous ProtectServer releases did not use the FM\_MAKE\_VERSION(major,minor) macro. The ProtectServer 2 samples now use this macro. Not using the macro may result in FM version numbers being displayed reversed in **ctconf** and **ctfm**.

---

**Best practice:** Use the FM\_MAKE\_VERSION(major,minor) macro to define the version number passed to DEFINE\_FM\_HEADER().

---

## Emulation Mode Enhancements

Emulation mode has been enhanced to support C99 and Cryptoki (Cprov) function patching of any application that is run against the emulated **cryptoki** wrapper built with the emulated FM.

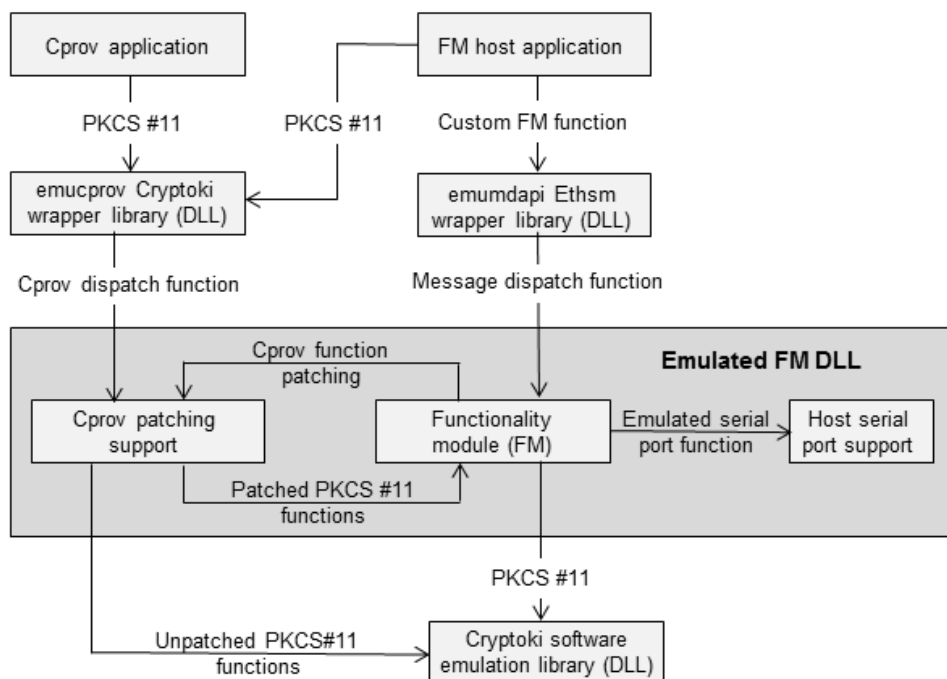
Unlike Protect Processing 3.0 and earlier releases, applications do not have to be recompiled with the emulated libraries, they simply have to have the **emucprov** and **emumdapi** wrapper libraries (with standard **cryptoki.so** and **ethsm.so** naming) in the library search path ahead of the real **cryptoki** library.

---

**Best practice:** Ensure that the **emucprov** and **emumdapi** wrapper libraries appear before the **cryptoki** library in your library search path.

---

The following diagram illustrates the functionality provided by the FM SDK in emulation mode. See the *FM SDK Programming Guide* for more information.



## FM Certificates

By default, MKFM will not sign with a 512-bit certificate. It is recommended that you create your FM certificates using RSA 2048 instead of RSA512. For example:

<b>ProtectServer</b>	<code>ctcert c -s0 -k -trsa -z512 -lfm</code>
<b>ProtectServer 2</b>	<code>ctcert c -s0 -k -trsa -z2048 -lfm</code>

MKFM now uses SHA-512 instead of SHA-1. To continue using a legacy 512-bit certificate for signing with a SHA-1 hash, you can use the **-3** option of the MKFM command, although this is not recommended.

---

**Best practice:** Create your FM certificates using RSA 2048.

---

## FM Debug Logging Using printf

Historically, debug logging has been via a simulated serial port 0 and the **dbgprint** routines. These methods are maintained for backwards compatibility.

As a simpler alternative to these methods, ProtectServer 2 adds support for standard C **printf** to write debug messages to the **hsmtrace** log.

---

**Best practice:** Update your code to use **printf** instead of serial port 0 and the **dbgprint** routines.

---

## Compile-Time Checking

The `_SFNT_FM_` compiler define has been added to enable a compile time check for an FM build. `_SFNT_FM_` is also set for emulation builds, which also have `FM_EMU` and `EMUL` set.

## The `clock()` Function Uses the ANSI/ISO Standard for Returning CPU Time

The `clock()` function now follows the ANSI/ISO standard of returning CPU time. This differs from Microsoft Windows `clock()` which returns wall time. Elapsed time checks should now use the more accurate `THR_BeginTiming` and `THR_UpdateTiming` APIs. Although these APIs existed in PPO, `clock()` on the PSG and K5 behaved like Windows, returning wall time instead of CPU time.

---

**Best practice:** Use the `THR_BeginTiming` and `THR_UpdateTiming` APIs to perform elapsed time checks.

---

## The `integers.h` Header File Removed From `$(FMSDK)/include`

The `integers.h` header file is no longer included in both `$(FMSDK)/include` and `$(CPROVIDIR)/include`. It is now only included in `$(CPROVIDIR)/include` since PTK-C is required in order to use the FM SDK.

## MKFM and CTFM Disable (d) Flag is Now Delete

The tools used to load and manage your FMs, such as MKFM and CTFM are the same as in previous releases with the exception of the "Disable" (d) flag. In previous releases, this flag disabled, but did not remove the FM. In this release, this flag will fully remove an FM rather than simply disabling it.

## Obsolete FM SDK Copy Function Removed

The `Copy` function, which is obsolete and previously deprecated, has been removed from both the `Cipher` object and the `Hash` object.

---

**Best practice:** Remove all instances of the `CipherObj copy` and `HashObj copy` functions from your code.

---

## Support Removed for `libfmhost`

The `libfmhost` library is no longer supported for HOST applications. The `fmdisp.h` header remains and has wrappers around the supported MD APIs for migration purposes. If your application uses these legacy APIs, your compile output will include deprecated warnings.

## Failed FMs

Failed FMs are now deleted instead of being disabled.



## Migrating Your FMs

---

Migrating your existing FMs should require only makefile changes and recompilation. It is recommended that you take advantage of the enhancements to the FM SDK introduced in ProtectServer 2 when migrating your source.

### To migrate an FM

1. Install the PTK-C SDK and FM SDK on a workstation or VM running a supported Linux OS.

---

**Note:** By default, the FM SDK is set to operate in emulation mode. You cannot install FMs while in emulation mode. To install an FM you must change the operating mode to hardware mode. The operating mode is specified when you run the installation script, and can be changed by re-running the script.

---

2. Update your makefiles, as outlined in “Makefiles” on page 13.
3. Review the list of changes and enhancements introduced in the ProtectServer 2 FM SDK, as detailed in “Summary of Changes and Enhancements on page 11, to determine whether any additional changes are required. You may also want to update your code at this time to implement some of the enhancements.
4. Build your source.
5. If necessary, fix any compile errors. See “Summary of changes and enhancements” on page 12 for a list of the changes introduced in ProtectServer 2 that may be the source of the errors.