



***UEFI 2.4 Errata B
Self-Certification Test (SCT) II
Getting Started Guide***

***Revision 1.3
December, 2014***

The material contained herein is not a license, either expressly or impliedly, to any intellectual property owned or controlled by any of the authors or developers of this material or to any contribution thereto. The material contained herein is provided on an "AS IS" basis and, to the maximum extent permitted by applicable law, this information is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses and of lack of negligence, all with regard to this material and any contribution thereto. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The Unified EFI Forum, Inc. reserves any features or instructions so marked for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE SPECIFICATION AND ANY CONTRIBUTION THERETO. IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR ANY CONTRIBUTION THERETO BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS DOCUMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Copyright 2007 – 2014 Unified EFI, Inc. All Rights Reserved

Revision History

Revision	Revision History	Date
1.0	Initial release for UEFI 2.3 Specification	6/15/10
1.1	Formatting and grammatical editing	1/4/11
1.2	Mantis 948 refreshed to align with current UEFI 2.3.1 Specification.	8/17/12
1.3	Mantis 1298 refreshed to align with current UEFI 2.4 Errata B Specification.	12/23/14

Contents

1	Introduction	1
1.1	Summary	1
1.2	Overview	1
1.3	Conventions Used in This Document.....	1
1.3.1	Pseudocode Conventions	2
1.3.2	Typographic Conventions	2
2	Quick Start Guide	3
2.1	Setup Development System	3
2.1.1	Install Tools.....	3
2.2	Download Source Code.....	3
2.2.1	Downloading UEFI SCT	3
2.3	Downloading Third-party Libraries	3
2.3.1	Downloading the Tcl-Tk Library	4
2.3.2	Downloading the Winpcap Library	4
2.3.3	Downloading EDKII and EFI Shell	4
3	Building the UEFI SCT Agent	5
3.1	Building the UEFI SCT Agent	5
3.1.1	IA32 Build Tip	5
3.1.2	X64 Build Tip	5
4	Building the EMS	7
4.1	Building the EMS	7
4.2	Runtime Configuration File.....	7
5	Installing the UEFI SCT Agent	9
5.1	Overview	9
5.2	Installing the UEFI SCT Agent	9
5.2.1	Installing the UEFI SCT Agent on an IA32 Platform	9
5.2.2	Installing the UEFI SCT Agent on an X64 Platform	9

Tables

Table 1. Runtime Configuration Files	7
--	---

1

Introduction

1.1 Summary

This document provides detailed instructions for building the UEFI SCT and creating an UEFI SCT Package including UEFI SCT agent and EMS. UEFI SCT is backward compatible with EFI SCT; however, it introduces new UEFI Management Side (EMS) application on the operating system to perform unified remote management for SCT testing. In order to build the UEFI SCT, the UDK2014 are required. The UDK2014 can be downloaded from http://tianocore.sourceforge.net/wiki/UDK2014_Archive.

1.2 Overview

This document does not try to explain the UEFI SCT, the UDK2014. That information about UDK2014 is available at http://tianocore.sourceforge.net/wiki/UDK2014_Archive. Additional documentation is available at www.uefi.org.

Instead, this document specifies how to download the code, build it and run it.

- *Users installing pre-built binaries may skip ahead to Chapter **Error! Reference source not found.***
- *Users wanting to use the EMS, or wanting to add debug information to the SCT or IHV-SCT, need to follow the full instructions to debug and build the tools.*

Note: *Changing the actual functionality of the test could prevent accurate and consistent results.*

The UEFI SCT includes the following additional documentation:

UEFI SCT User Guide - A user's guide detailing how to use the UEFI SCT application on an IA32, X64 or ARM target test platform.

UEFI SCT Case Specification - The descriptions for all checkpoints and assertions in the UEFI SCT. This document can be used in conjunction with the GUID definitions and the assertions of the UEFI SCT.

1.3 Conventions Used in This Document

This document uses the typographic and illustrative conventions described below.

1.3.1 Pseudocode Conventions

Pseudocode helps to describe algorithms in a more concise form. None of the algorithms in this document are intended to be compiled directly. The pseudocode is presented at a level corresponding to the surrounding text.

In describing variables, a *list* is an unordered collection of homogeneous objects. A *queue* is an ordered list of homogeneous objects. Unless otherwise noted, assume the ordering is First in First out (FIFO).

This document presents Pseudocode in a C-like format, using C conventions where appropriate. This coding style, particularly the indentation style, is used for readability. It does not necessarily comply with an implementation of the *Extensible Firmware Interface Specification* or with any of the architecture specifications associated with EDKII.

1.3.2 Typographic Conventions

- *Plain text* The normal text typeface is used for the vast majority of the descriptive text in a specification.
- *Plain text (blue)* Any *plain text* that is underlined and in blue indicates an active link to the cross-reference. Click on the word to follow the hyperlink. Note that these links are not active in the PDF of the specification.
- **Bold** In text, **Bold** typeface identifies a processor register name. In other instances, **bold** typeface can be used as a running head within a paragraph.
- *Italic* In text, Italic typeface can be used as emphasis to introduce a new term or to indicate a manual or specification name.
- **BOLD Monospace** Computer code, example code segments, and all prototype code segments use a **BOLD Monospace** typeface with a dark red color. These code listings normally appear in one or more separate paragraphs, though words or segments can also be embedded in a normal text paragraph.
- *Italic Monospace* In code or in text, words in *Italic Monospace* indicate placeholder names for variable information that must be supplied (i.e., arguments).

The Framework Interoperability and Component Specifications are available at the following URL:

<http://www.intel.com/technology/framework/spec.htm>

2

Quick Start Guide

2.1 Setup Development System

This section describes the steps that are necessary initially to set up the local system in preparation for building a platform (build tip) in the UEFI SCT source tree.

In general, the project is as compiler-neutral as possible. In the future, adding support for more compiler types and for a Linux-hosted development environment will require help from external developers.

2.1.1 Install Tools

To build the UEFI SCT source tree, the following tools must be installed on the development system:

- *Microsoft Windows 7® operating system*
- *Microsoft Visual Studio ® 2008*

2.2 Download Source Code

2.2.1 Downloading UEFI SCT

The first download the source code from www.uefi.org. You must register and use the login information that you receive. To build the UEFI SCT, you must also download the UDK2014 source code.

2.2.1.1 Downloading Using a Link

Directions for downloading UefiSctPackAll source using a link:

1. Login to <http://www.uefi.org/testtools> and download the SCT packages.
2. Extract the Zip file to a location such as C:\Test on your system, and then the folders SCTPkg and EMS will be located as the subfolder of C:\Test.

2.3 Downloading Third-party Libraries

To implement remote validation, EMS uses three third-party libraries such as Tcl-Tk, Winpcap and Libnet. The Libnet library is released with the UEFI SCT.

Use an ActiveTcl version of 8.4.19.5-threaded, and WpdPack should be 3.0. Refer to the *UEFI SCT Release Package release notes* for specific versions other than those listed in the subsections below.

2.3.1 Downloading the Tcl-Tk Library

1. Install the ActiveTcl8.4.19.5.294332-win32-ix86-threaded.exe into your OS. For example, you can install ActiveTcl into your C:\Tcl. In this example, the setup folder is C:\Tcl, though you can choose any location.
2. Copy into C:\Test\Ems\Lib\Tcl\Include\ all content under C:\Tcl\include.
3. Copy into C:\Test\Ems\Lib\Tcl\Lib\ the .lib files under C:\Tcl\lib.

2.3.2 Downloading the Winpcap Library

1. Install the WinPcap_4_1_3.exe into your OS.
2. Extract the 3.0-WpdPack.zip file. In this example, the extraction folder is C:\WpdPack, but you can choose any location you like.
3. Copy into C:\Test\Ems\Lib\WpdPack\Include\ all content under C:\WpdPack\Include.
4. Copy into C:\Test\Ems\Lib\ WpdPack\Lib\ all content under C:\WpdPack\Lib

2.3.3 Downloading EDKII and EFI Shell

In order to build the UEFI SCT, download UDK2014 from http://tianocore.sourceforge.net/wiki/UDK2014_Archive.

2.3.3.1 Directions for Downloading EDKII Using the Web Site:

1. Extract BaseTools(Windows).zip to C:\Test, then the two folders named BaseTools and Conf. One batch file named edksetup.bat will be located in C:\Test .
2. Extract UDK2014.MyWorkspace.zip to C:\Test

3

Building the UEFI SCT Agent

3.1 Building the UEFI SCT Agent

This section describes the steps that are necessary to configure and build the UEFI SCT Agent. The SCT Package is produced when building the UEFI SCT Agent. This package needs to be installed onto a Target test platform that is based on UEFI or "Framework". The directory SctPackageIA32/SctPackageX64 will be created under C:\Test\Build\UefiSct\DEBUG_VS2008x86 when the UEFI SCT is built successfully.

3.1.1 IA32 Build Tip

Run Visual Studio 2008 Command Prompt to go to the command line environment. The following commands can be used to build the UEFI SCT IA32 binary. If the build is successful, one set of executable image file *.efi will be created in the C:\Test\Build\UefiSct\DEBUG_VS2008x86\SctPackageIA32 directory.

1. Execute "cd C:\Test"
2. Execute "edksetup.bat"
3. Execute "copy SctPkg\Tools\Bin\GenBin.exe BaseTools\Bin\Win32\"
4. Execute "build -p SctPkg\UEFI\UEFI_SCT.dsc -t VS2008x86 -a IA32"
5. Execute "cd Build\UefiSct\DEBUG_VS2008x86"
6. Execute "..\..\..\SctPkg\CommonGenFramework.bat uefi_sct IA32 InstallIA32.efi"

3.1.2 X64 Build Tip

Run the Visual Studio 2008 Command Prompt to go to the command line environment. The following commands can be used to build the UEFI SCT X64 binary. If the build is successful, one set of executable image file *.efi will be created in the C:\Test\Build\UefiSct\DEBUG_VS2008x86\SctPackageX64 directory.

1. Execute "cd C:\Test"
2. Execute "edksetup.bat"
3. Execute "copy SctPkg\Tools\Bin\GenBin.exe BaseTools\Bin\Win32\"
4. Execute "build -p SctPkg\UEFI\UEFI_SCT.dsc -t VS2008x86 -a X64"

5. Execute "cd Build\UefiSct\DEBUG_VS2008x86"
6. Execute "..\..\..\SctPkg\CommonGenFramework.bat uefi_sct X64
InstallX64.efi"

4

Building the EMS

4.1 Building the EMS

This section describes the steps that are necessary to set up and build the EMS (UEFI Management Side). The Ems package will need to be built based on three third-party libraries: Tcl-Tk, Winpcap and Libnet. Default Ems packages use ActiveTcl8.4.19.5-threaded, winpcap4.1.3 and WpdPack3.0 and co-released libnet for building. Refer to the *UEFI SCT Release Package release notes* for latest specific versions.

Run Visual Studio 2008 Command Prompt to go to the command line environment. The following commands can be used to build the EMS operating system application. If the build is successful, an Ems.exe executable file is generated under the bin directory. After that, start the application by running "Ems.exe Main.tcl" under the bin directory.

1. cd c:\test\ems\src
2. set ems_dir=c:\test\ems
3. nmake
4. cd ..\bin
5. Ems Main.tcl

4.2 Runtime Configuration File

EMS provides one configuration file primarily used by the Ems application to configure the test environment and describe the test methodology. **Error! Reference source not found.** 1 lists the configuration files that can be used.

Table 1. Runtime Configuration Files

Type of Configuration File	Description
Init.conf	<p>Defines default the configuration for an EMS operating system application.</p> <p>Below is an example of the configuration item:</p> <pre>CASE_ROOT_DIR = c:\ VERBOSE_LEVEL = DEFAULT COMMUNICATION_TYPE = MNP TARGETMAC = 00:00:00:00:00:00</pre> <p>It is located in the bin directory.</p>

5

Installing the UEFI SCT Agent

5.1 Overview

UEFI SCT contains the UEFI SCT agent in the EFI side and EMS in the operating system side. This section describes the steps that are necessary to install UEFI SCT in the EFI side.

5.2 Installing the UEFI SCT Agent

UEFI SCT agent is a shell application, so the EFI Shell environment is required to run UEFI SCT agent. Please boot to the specified shell environment, and perform the installation steps from the subsections below, according to different target platforms.

5.2.1 Installing the UEFI SCT Agent on an IA32 Platform

1. Copy the contents of the IA32 build directory SctPackage to a USB device.
2. Put the USB device into the USB port and boot the system to the EFI Shell environment.
3. In EFI Shell environment, change the current drive and directory to the USB device drive and SctPackageIA32 directory.
4. Run `installIA32.efi` and follow the instructions on the screen.

5.2.2 Installing the UEFI SCT Agent on an X64 Platform

To install the UEFI SCT Harness from the installation CD:

1. Copy the contents of the X64 build directory SctPackage to a USB device.
2. Put the USB device into the USB port and boot the system to the EFI Shell environment.
3. In EFI Shell environment, change the current drive and directory to the USB device drive and SctPackageX64 directory.
4. Run `installX64.efi` and follow the instructions on the screen.