



UEFI 2.3 Self-Certification Test (SCT) Getting Started Guide

*Revision 1.0
January, 2011*

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 - 2011 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

Contents

Contents	v
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 5
2.1	Setup Development System 5
2.1.1	Install Tools 5
2.2	Download Source Code 5
2.2.1	Downloading UEFI SCT 5
2.3	Downloading Third-party Libraries 6
2.3.1	Downloading the Tcl-Tk Library 6
2.3.2	Downloading the Winpcap Library 6
2.3.3	Downloading EDKII and EFI Shell 7
3	Building the UEFI SCT Agent 9
3.1	Building the UEFI SCT Agent 9
3.1.1	IA32 Build Tip 9
3.1.2	X64 Build Tip 10
3.1.3	IPF Build Tip 10
4	Building the EMS 11
4.1	Building the EMS 11
4.2	Runtime Configuration File 11
5	Installing the UEFI SCT Agent 13
5.1	Overview 13
5.2	Installing the UEFI SCT Agent 13
5.2.1	Installing the UEFI SCT Agent on an IA32 Platform 13
5.2.2	Installing the UEFI SCT Agent on an Itanium-Based Platform 14
5.2.3	Installing the UEFI SCT Agent on an EM64T-Based Platform 14

Tables

Table 1. Runtime Configuration Files 11

1

Introduction

1.1 Summary

This document provides detailed instructions for building the UEFI SCT and creating a 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 EFI Developer Kit II (EDKII) and the EFI Shell are required. The EDKII and the EFI Shell can be downloaded from <http://sourceforge.net/apps/mediawiki/tianocore/index.php>.

1.2 Overview

This document does not try to explain the UEFI SCT, the EFI Shell, or EDK II. That information is available at <http://sourceforge.net/apps/mediawiki/tianocore/index.php>. 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 5.
- 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, EM64T or Intel® Itanium Family processor 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) In the online help version of this specification, 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).

Plain Monospace In code, words in a **Plain Monospace** typeface that is a dark red color but is not bold or italicized indicate pseudocode or example code. These code segments typically occur in one or more separate paragraphs.

See the master Framework glossary in the Framework Interoperability and Component Specifications help system for definitions of terms and abbreviations that are used in this document or that might be useful in understanding the descriptions presented in this document.

See the master Framework references in the Interoperability and Component Specifications help system for a complete list of the additional documents and specifications that are required or suggested for interpreting the information presented in this document.

The Framework Interoperability and Component Specifications help system is 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 EDKII and the UEFI SCT source tree, the following tools must be installed on the development system:

- Microsoft Windows XP® operating system

For 32-bit Intel architecture (IA32) platform development:

- Microsoft Visual Studio .NET® 2003 (7.1)
- Microsoft Visual Studio ® 2005 (8.0)

For Intel Itanium® processor family development:

- Microsoft® C/C++ Optimizing Compiler Version 13.10.2240.8 for IA64 in the Microsoft Windows Server® 2003 DDK (Build 3790).

When compiling for targets based on Itanium architecture, use **NMAKE.EXE**, **LINK.EXE** and **LIB.EXE** from the Microsoft Windows Server 2003 DDK (Build 3790).

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 EDKII and the EFI Shell source code.

2.2.1.1 Downloading Using a Link

Directions for downloading UefiSctPackAll source using a link:

1. Login to www.uefi.org/specs/ and download the SCT source packages.

2. Under Supplemental Test Tools, select the most recent Self Certification Test (SCT).
3. 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 8.4.13-threaded or higher, and Winpcap should be 3.0. To get third-party Tcl-Tk and Winpcap libraries, follow the steps below. Refer to the [UEFI SCT Release Package release notes](#) for specific links or versions other than those listed in the subsections below.

2.3.1 Downloading the Tcl-Tk Library

1. Download the Tcl-Tk installation file from the official website, <http://www.tcl.tk>. Use the following download link: <http://downloads.activestate.com/ActiveTcl/Windows/8.4.13/ActiveTcl8.4.13.0.261555-win32-ix86-threaded.exe>.
2. Install the downloaded setup file 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.
3. Copy into C:\Test\Ems\Lib\Tcl\Include\ all content under C:\Tcl\include.
4. Copy into C:\Test\Ems\Lib\Tcl\Lib\ the .lib files under C:\Tcl\Lib.

2.3.2 Downloading the Winpcap Library

1. Download Winpcap installation file and Winpcap Developer's Pack from the official website <http://www.winpcap.org/>. Use one of the following download links: <http://www.winpcap.org/archive/3.0-WinPcap.exe> and <http://www.winpcap.org/archive/3.0-WpdPack.zip>
2. Install the downloaded setup file into your OS.
3. Extract the 3.0-WpdPack.zip file. In this example, the extraction folder is C:\WpdPack, but you can choose any location you like.
4. Copy into C:\Test\Ems\Lib\WpdPack\Include\ all content under C:\WpdPack\Include.
5. Copy into C:\Test\Ems\Lib\WpdPack\Lib\ the .lib files under C:\WpdPack\Lib

2.3.3 Downloading EDKII and EFI Shell

In order to build the UEFI SCT, download EDKII and the EFI Shell from <http://sourceforge.net/apps/mediawiki/tianocore/index.php>.

2.3.3.1 Directions for Downloading EDKII Using the Web Site:

1. Download BaseTools binary stable version - BaseTools(Binaries).zip from [http://sourceforge.net/projects/edk2/files/Releases/EDK II Core \(v 0.90\)/](http://sourceforge.net/projects/edk2/files/Releases/EDK II Core (v 0.90)/).
2. Extract BaseTools(Binaries).zip to C:\Test, then the folders named BaseTools and Conf. One batch file named edksetup.bat will be located in C:\Test .
3. Download EdkCompatibilityPkg stable version - EdkCompatibilityPkg 0.90.zip from [http://sourceforge.net/projects/edk2/files/Releases/EDK II Core \(v 0.90\)/](http://sourceforge.net/projects/edk2/files/Releases/EDK II Core (v 0.90)/) .
4. Extract EdkCompatibilityPkg 0.90.zip to C:\Test\EdkCompatibilityPkg.
5. Replace the section of DDK3790 (between line 819 and line 926) in BaseTools\Conf\tools_def.template with the content in SctPkg\DDK3790.txt (between line 46 and line 153).

2.3.3.2 Directions for Downloading EFI Shell Source Using the Web Site:

1. Download Shellpkg latest stable version from http://sourceforge.net/apps/mediawiki/tianocore/index.php?title=EFI_Shell.
2. Put it to C:\Test\EdkCompatibilityPkg\Other\Maintained\Application\Shell

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/SctPackageIPF will be created under C:\Test\Build\UefiSct\DEBUG_DDK3790 when the UEFI SCT is built successfully.

3.1.1 IA32 Build Tip

Run Visual Studio 2005 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_DDK3790\SctPackageIA32 directory.

1. cd C:\Test
2. edksetup.bat
3. set efi_source=C:\Test
4. set edk_source=C:\Test\EdkCompatibilityPkg
5. copy SctPkg\Tools\Bin\GenBin.exe BaseTools\Bin\Win32\
6. build -p SctPkg\UEFI\UEFI_SCT.dsc -t DDK3790 -a IA32
7. cd Build\UefiSct\DEBUG_DDK3790
8. ..\..\..\SctPkg\CommonGenFramework.bat uefi_sct IA32 InstallIA32.efi

3.1.2 X64 Build Tip

Run the Visual Studio 2005 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 directory C:\Test\Build\UefiSct\DEBUG_DDK3790\SctPackageX64.

1. cd C:\Test
2. edksetup.bat
3. set efi_source=C:\Test
4. set edk_source=C:\Test\EdkCompatibilityPkg
5. copy SctPkg\Tools\Bin\GenBin.exe BaseTools\Bin\Win32\
6. build -p SctPkg\UEFI\UEFI_SCT.dsc -t DDK3790 -a X64
7. cd Build\UefiSct\DEBUG_DDK3790
8. ..\..\..\SctPkg\CommonGenFramework.bat uefi_sct X64 InstallX64.efi

3.1.3 IPF Build Tip

Run the Visual Studio 2005 Command Prompt to go to the command line environment. The following commands can be used to build the UEFI SCT IPF binary. If the build is successful, one set of executable image file ***.efi** will be created in the directory C:\Test\Build\UefiSct\DEBUG_DDK3790\SctPackageIPF.

1. cd C:\Test
2. edksetup.bat
3. set efi_source=C:\Test
4. set edk_source=C:\Test\EdkCompatibilityPkg
5. copy SctPkg\Tools\Bin\GenBin.exe BaseTools\Bin\Win32\
6. build -p SctPkg\UEFI\UEFI_SCT.dsc -t DDK3790 -a IPF
7. cd Build\UefiSct\DEBUG_DDK3790
8. ..\..\..\SctPkg\CommonGenFramework.bat uefi_sct IPF InstallIPF.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 ActiveTcl8.4.13-threaded, winpcap3.0 and co-released libnet for building. Refer to the [UEFI SCT Release Package release notes](#) for latest specific links or versions.

Run Visual Studio .NET 2003 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. Table 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. If you do not have the built-in shell, for EFI1.10 sample code or a Tiano implementation, you can configure the shell environment.

Follow the steps in the *EFI Shell Getting Started* document to build the image file shell.efi.

1. Copy the shell.efi to the target machine.
2. Add a boot option to the shell.efi just added.
3. Boot to the specified shell environment, and perform the installation steps from the subsections below, according to different target platforms.

The UEFI SCT Agent can be installed on the following platforms:

- IA32 Platform
- Itanium-Based Platform
- EM64T-Based Platform

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 or IDE-CD.
2. Put the USB or IDE-CD into the USB port or the IDE-CD drive and boot the system to the EFI Shell environment.
3. In EFI Shell environment, change the current drive and directory to the installation CD or USB device drive and root directory.
4. Run `installIA32.efi` and follow the instructions on the screen.

5.2.2 Installing the UEFI SCT Agent on an Itanium-Based Platform

To install the UEFI SCT Harness from the installation CD:

1. Copy the contents of the IPF build directory SctPackage to a USB device or IDE-CD.
2. Put the USB or IDE-CD into the USB port or the IDE-CD drive and boot the system to the EFI Shell environment.
3. In EFI Shell environment, change the current drive and directory to the installation CD or USB device drive and root directory.
4. Run `installIPF.efi` and follow the instructions on the screen.

5.2.3 Installing the UEFI SCT Agent on an EM64T-Based 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 or IDE-CD.
2. Put the USB or IDE-CD into the USB port or the IDE-CD drive and boot the system to the EFI Shell environment.
3. In EFI Shell environment, change the current drive and directory to the installation CD or USB device drive and root directory.
4. Run `installX64.efi` and follow the instructions on the screen.