**TPM 2.0 Provisioning Tool**

**User Manual**

December 11, 2015

# Disclaimer

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE MATERIALS, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR ACCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. *Intel and its suppliers further do not warranty the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. Intel may make changes to these materials, or the products described therein, at any time without notice. Intel makes no commitment to update the Materials.*

Copyright © 2013- Intel Corporation. All rights reserved. Intel, the Intel logo, and the Intel Xeon logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

# Notice

This tool is provided as sample code to illustrate the type of tool needed to do provisioning on a TPM 2.0 device and to evaluate the TPM 2.0 family of devices. This code and script files have NOT been thoroughly tested. The license (below) allows you to modify and distribute this tool as needed. Intel recommends that you thoroughly test and validate this tool or any derivative before using in a production environment.

# Getting Started

UEFI Provisioning:

To provision a TPM2 in UEFI using this package you will need the capability of booting to a UEFI shell; either a shell internal to the BIOS or one from a USB stick.

Step 1: Copy the TPM2 Prov files directory to a USB stick.

Step 2: Boot the platform to a UEFI shell, either the BIOS internal shell or boot UEFI from a USB stick.

Step 3: Basic Provisioning using the sample SHA256 definitions

1. For TXT indexes run Tpm2TxtProv.nsh SHA256 Example
2. If you need to create the platform owner index run Tpm2PoProv.nsh SHA256 Example
3. For the SGX index run Tpm2SgxiProv.nsh SHA256 Example
4. For the PPI index run Tpm2PpiProv.nsh SHA256 Example

Note: SHA256 is currently the recommended algorithm, however SHA384, 512, and SM3 definitions are also available in the package. The set of “Example” definitions are for development use only and you must either modify the set of Example definitions to use your own authorization policy or use the set of “Default” definitions (by substituting “Default” for “Example” in the commands above. Default indexes are permanent and cannot be modified or deleted.

Windows Provisioning:

To provision a TPM2 in Windows using this package you will need the capability of booting to a Windows 8.1 or later system, or a Windows system with the TBS driver manually installed.

Step 1: Copy the TPM2 Prov files directory and TPM Prov server directory to a working directory in on the target machine.

Step 2: Open a windows command prompt and navigate to the provisioning files working directory.

Step 3: Launch TPMProvServer.exe with administrator privileges. Note that this tool can communicate with either TPM 1.2 or 2.0.

Step 4: Since Windows provisioning scripts have not been written, you will need to open the .nsh scripts to see which TPM2ProvTool commands must be executed from the command prompt or create your own DOS script (\*.bat file).

# Tool Overview

The Tpm2ProvTool is a command line tool designed to be used with a batch (\*.nsh) file. The complexity of TPM2.0 poses many challenges. So TPM provisioning is split between two tools:

* A Windows based WinTpm2Tool that provides a GUI environment to create definition files
* The UEFI Tpm2ProvTool that uses those definition files to send commands to the TPM

Many TPM operations require multiple commands be sent to the TPM (i.e., the Tpm2ProvTool must be invoked multiple times). Thus the TpmProv tool is designed to maintain context between invocations. Therefore the tool provides storage for:

* 10 Handles – When issuing an instruction that returns a handle (such as loading a key), you indicate a handle index (0-9) in the command line. When issuing a command that references that object (such as issuing a Tpm2PolicySigned assertion) you specify that handle # in the command line.
* 5 Sessions (and associated authValue) – when creating a session you specify the session index (0-4) in the command line and when issuing an instruction that references that session (such as issuing a Tpm2PolicyXXX assertion), you specify that session # in the command line.
* 8 Digests – when you issue a command which returns a digest (such as Tpm2PolicyGetDigest) you indicate the digest index (0-7) in the command line. The def files generated by the WinTpm2Tool (such as a PolicyOR.pDef) can reference those digests.

EXAMPLE: When issuing instructions to the Tpm2Prov tool; handles, sessions, and digest are referenced by an index number. For example you would start a session with the following instruction to start a policy session using session context #3:

Tpm2Provtool StartSession MyFirstPolicySession.sDef 3

And then issue the following instruction to issue an assertion against that session:

Tpm2Provtool AssertPolicy Password.pDef 3

The Tpm2ProvTool is designed to be used with a batch (\*.nsh) file and the commandline has the following general syntax:

Tpm2ProvTool <instruction> [<arg2> [<arg3> […]]]

Where

[arg] indicates an optional argument

<option1 | option2 | option3> indicates a selection of one of the options

Arg2 is typically a def filename when one is required and the remaining arguments specify session, handle, and/or digest indexes and output filenames.

# Instruction Set

## List of Instructions

Here is an alphabetical list of the Tpm2ProvTool instructions. Following the list is the syntax and definition.

* AssertPolicy
* Create
* CreatePrimary
* ExportValue – Saves a value stored in the config file
* FlushSession
* GetCapability
* GetRandom
* HashStart
* HashUpdate
* HashComplete
* HierarchyChangeAuth
* ImportValue – Reads a value into the config file
* Load – loads a key or sealed blob
* LoadExternal
* NvChangeAuth
* NvCompareData – Verifies that the Data is correct
* NvDefineSpace – Creates an NV Index
* NvRead – Displays the NV index data
* NvReadPublic – Displays the NV index attributes
* NvVerifyDef – Verifies that the attributes match
* NvWrite – writes the NV index data
* NvWriteLock – locks the NV index
* NvUndefineSpace – deletes an NV index
* NvUndefineSpecial– deletes a PH NV index
* PcrRead – Reads PCRs and saves to a file
* PolicyRestart
* SavePolDigest
* SavePolDigestDLF
* SelfTest
* SetPrimaryPolicy
* ShutdownClear
* ShutdownState
* StartSession
* StartUpClear
* StartUpState
* StorePolDigest
* TPMOwnerClear
* TPMStatus

## Syntax and Details

Tpm2ProvTool AssertPolicy <PolicyDefFile.pDef> <Session #> [<auth Session #>]  
Issues a TPM2\_PolicyXXX() command to the TPM for the specified session and requires a Policy Def file built by the WinTpm2 tool. The second session number is only required for policy commands that require authorization (e.g., TPM2\_PolicyNV). Currently, the following assertions are supported:   
 **Authorize  
 Command Code  
 NameHash  
 NV   
 OR  
 Secret  
 Signed**

*Tpm2ProvTool* Create <KeyDef.kDef> <session#> <Parent-handle#> < OutPrivate.prv> < OutPublic.pub>  
Creates a TPM object (key or sealed blob). It requires session for hierarchy authorization and a Key Def file created by the WinTpm2 tool. The Parent key must first be loaded (with its handle is stored in the config file at the specified index – see the *Load* instruction). This instruction stores the public key and encrypted private key to the respective files.

*Tpm2ProvTool* CreatePrimary <KeyDef.kDef> <session#> < handle#> < OutPublic.pub>   
Creates a TPM object (key or sealed blob). Requires session for hierarchy authorization and a Key Def file created by the WinTpm2 tool. Stores the public key to the specified file and the key is automatically loaded and its handle is saved in the tool’s config file at the referenced handle number.

*Tpm2ProvTool* ExportValue [AUTH | DIGEST | KEY | DATA] <index> < output Data file>  
Stores a value saved in the tool’s config file to the specified output file.

*Tpm2ProvTool* FlushSession <Session #>  
Destroys the specified session

*Tpm2ProvTool* GetCapability < Capability> <Property> <PropertyCount >   
Sends the Tpm2GetCatability() command – not fully implemented. Since there are a few hundred possible combinations for input and output for this command, it currently takes the three capability parameters as inputs and displays the raw TPM2 response. Decoding the capability information and individual fields of each type of response is an exercise currently left for the user.

*Tpm2ProvTool* GetRandom <dataSize> <outputFilename>   
Instructs the TPM to generate a random value of the specified size and store it in the specified file.

Hash Sequence instructions: These commands setup and send data to the TPM to be hashed and then saves the hash digest and the TPM generated ticked in the specified files. The HashStart command saves the TPM generated handle in the tool’s config file and uses it for the HashUpdate and HashComplete commands. The HashStart is followed by one or more HashUpdate instructions that provide the data to be hashed. And the HashComplete() instruction returns the digest and ticket.  
 *Tpm2ProvTool* HashStart <HashDefFile.hDef>   
 *Tpm2ProvTool* HashUpdate <DataFilename> <DataSize>  
 *Tpm2ProvTool* HashComplete <ResultFileName> <TicketFileName>

*Tpm2ProvTool* HierarchyChangeAuth [PH | SH | EH] <filename.dat> <Session #>  
This instruction changes the authValue of the specified hierarchy and requires a session for hierarchy authorization.

*Tpm2ProvTool* ImportValue <input data file> [AUTH | DIGEST | KEY | DATA] <index>  
This command imports a value into the tool’s config file

*Tpm2ProvTool* Load <keyPrivate.prv> <keyPublic.pub> <ParentHandle#> <Session#> <Handle#>  
This command loads a key (or sealed blob) into the TPM. It requires a session for the parent authorization and the public and private keys saved by the Create instruction. Note if only the public key needs to be loaded – use the *LoadExternal* instruction.

Tpm2ProvTool LoadExternal <PH |SH | EH | NULL> [<keyPrivate.prv>] <keyPublic.pub> <Handle#> <KeyNameFile>  
This command loads an unprotected key into the TPM. It requires the public and (optionally) private keys. Note that if the private key is provided, then the hierarchy must be NULL.

*Tpm2ProvTool* NVChangeAuth <IndexDefFile.iDef> <Session #> <CFG\_Auth #>  
Changes the authValue of the index specified in the Index Def File to a value stored in the tool’s config file. It requires a session for index write authorization.

Tpm2ProvTool NvCompareData <IndexDefFile.iDef> <Session #>  
Compares the data of the index specified in the Index Def File to the data specified in that file. It requires a session for index read authorization.

*Tpm2ProvTool* NvDefineSpace <IndexDefFile.iDef> <Session #>  
Creates an index specified by the Index Def File created by the WinTpm2 tool. It requires a session for hierarchy authorization.

*Tpm2ProvTool* NvRead <IndexDefFile.iDef> <Session #>  
Displays the data of the NV Index specified by the Index Def File created by the WinTpm2 tool. It requires an authorization session to read the NV Index.

*Tpm2ProvTool* NvReadPublic <IndexDefFile.iDef>   
Displays the NV Public information for the index specified in the Index Def file created by the WinTpm2 tool.

*Tpm2ProvTool* NvVerifyDef <IndexDefFile.iDef>   
Compares the NV Public information for the index specified in the Index Def file created by the WinTpm2 tool to the definition specified in the Index Def file.

*Tpm2ProvTool* NvWrite <IndexDefFile.iDef> <Session #>  
Writes the data of the NV Index specified in the Index Def file with the data specified in the Index Def File. It requires a session for index write authorization.

*Tpm2ProvTool* NVWriteLock <IndexdefFile.iDef> <Session #>  
Locks the data of the NV Index specified in the Index Def file. It requires a session for index write authorization.

*Tpm2ProvTool* NvUndefineSpace <IndexDefFile.iDef> <Session #>  
Deletes the index specified in the Index Def File and requires a session for hierarchy authorization.

Tpm2ProvTool NvUndefineSpecial <IndexDefFile.iDef> <Hierarchy Auth Session #> <Index Auth Session #>  
Deletes the index specified in the Index Def File and requires a session for hierarchy authorization and a session for index authorization (which must be a policy session).

*Tpm2ProvTool* PcrRead [<BANK ID> [PCR#] ]   
Displays the specified PCR value. If PCR# is omitted, it will display all PCRs for the specified bank and if BANK ID is omitted, will display all PC R values of all banks. BANK ID is a TPM 2 Alg ID (e.g., 0x0004 for SHA1, 0x000B for SHA256)

*Tpm2ProvTool* PolicyRestart <Session #>  
Resets the session to start a new set of assertions.

Tpm2ProvTool SavePolDigest <Session #> <outputFilename>  
Performs the Tpm2PolicyGetDigest(session#) and saves the digest in the specified output file.

Tpm2ProvTool SavePolDigestDLF <DigestListFile.dlf> <Session #> <DLF index>  
Performs the Tpm2PolicyGetDigest(session#) and saves the digest in the specified Digest List File at the specified index.

*Tpm2ProvTool* SelfTest  
Invokes the TPM self test

*Tpm2ProvTool* SetPrimaryPolicy <PolicyFile.pSet> <Session #>   
Sets the authPolicy for the hierarchy specified in the pSet file created by the WinTpm2 tool, It requires a session for hierarchy authorization.

*Tpm2ProvTool* ShutdownClear  
Issues a normal TPM shutdown command

*Tpm2ProvTool* ShutdownState  
Issues a SAVE STATE TPM shutdown command

*Tpm2ProvTool* StartSession <SessionDefFile.sDef> <Session #>  
Sets up a session and issues the Tpm2StartSession() command. It requires a Session Def file created by the WinTpm2 Tool. Note that when this instruction is used to setup a PW session, no command is sent to the TPM, but it does allocate a session structure in the tool’s config file.

*Tpm2ProvTool* StartUpClear  
Issues a normal TPM Startup command

*Tpm2ProvTool* StartUpState  
Issues a RESTORE TPM Startup command

*Tpm2ProvTool* StorePolDigest <Session #> <index>  
Performs the Tpm2PolicyGetDigest(session#) and saves the digest in the tool’s config file at the specified index.

*Tpm2ProvTool* TPMOwnerClear <PH | LOCKOUT> <Session#>  
Clears TPM ownership and requires an authorization session for either Platform Hierarchy authorization or LockoutAuth

*Tpm2ProvTool* TPMStatus  
Displays the TPM status

# Sample Batch Files

The following are examples of how the tool can be used

## Calculate platformPolicy Digest

For this example the platformPolicy consists of the following branch

* PolicyOR(zeroDigestSha256, MyPhSecretSha256)

You will need the following:

* Create a hash digest of a password/passphrase (PhSecretSha256.hash)
* Policy Definition file: PhSecretORSha256.pDef :: Create Policy 🡪OR; Digest Count = 2; Digest0 = Import ZeroDigestSha256.hash; digest1 = Read from File PhSecretSha256.hash
* Session Definition file: TrialSha256.sDef :: Create Session🡪Type=TRIAL

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# TPM 2.0 Platform authPolicy (platformPolicy) Calculator (SHA256)

# Last update: Dec 30, 2013

# Sample script to generate PhAuthPolicySha256.hash

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo -OFF

echo \*\*\*\*\*\* Generating Platform Hierarchy platformPolicy Digest \*\*\*\*\*\*

echo First, start a trial session using session #0

Tpm2ProvTool StartSession TrialSha256.sDef 0 > PhAuthPolicyGen.log

echo Issue assertion(s)

Tpm2ProvTool AssertPolicy PhSecretORSha256.pDef 0 >> PhAuthPolicyGen.log

# If the branch requires multiple assertions, they would go here

echo Save policy digest to file

Tpm2ProvTool SavePolDigest 0 PhAuthPolicySha256.hash >> PhAuthPolicyGen.log

echo Flush the session

Tpm2ProvTool FlushSession 0 >> PhAuthPolicyGen.log

## Calculate authPolicy Digest for an NV Index

For this example the authPolicy consists of the following branches

* PolicyOR(zeroDigest, MySecretSha256) – using the WinTpm2Tool, create MySecret.pDef
* PolicyOR(zeroDigest, MySecretSha256) and TPM2\_PolicyCommandCode(NV\_UndefineSpaceSpecial) – using the WinTpm2Tool, create UnDefSpaceSpecial.pDef
* PolicyNvWritten(writtenSet=0) – using the WinTpm2Tool, create NvWritten.pDef

You will need the following:

* Policy Definitions (pDef):
  + MySecrete.pDef = Create Policy 🡪OR; Digest Count = 2; Digest0 = Import NullDigest.hash; digest1 = Read from File MySecret.hash

You will also need to create a session definition for a Trial Policy Session named Trial.sDef and PolicyOR assertion named MyIndexOR.pdef.

# First, start a trial session using session #1

Tpm2ProvTool StartSession Trial.sDef 1

# Issue assertion(s)

Tpm2ProvTool AssertPolicy MySecret.pDef 1

# If the branch requires multiple assertions, they would go here

# Save branch digest in Digest 0

Tpm2ProvTool StorePolDigest 1 0

# Restart the session for the next branch calculation

Tpm2ProvTool PolicyRestart 1

# Issue assertion(s)

Tpm2ProvTool AssertPolicy UnDefSpaceSpecial.pDef 1

# Save branch digest in Digest 1

Tpm2ProvTool StorePolDigest 1 1

# Restart the session for the next branch calculation

Tpm2ProvTool PolicyRestart 1

# Issue assertion(s)

Tpm2ProvTool AssertPolicy NvWritten.pDef 1

# Save branch digest in Digest 2

Tpm2ProvTool StorePolDigest 1 2

# Done calculating branch digests – no need to Restart the session

# Now issue PolicyOR

Tpm2ProvTool AssertPolicy MyIndexOR.pDef 1

# Now get the digest and save it to a file

Tpm2ProvTool SavePolDigest 1 MyIndexAuthPolicy.hash

# Release session

Tpm2ProvTool FlushSession 1

## Provisioning an NV Index

Use the WinTpm2Tool to create an Index definition named MyIndex.iDef – assumes index AuthPolicy above and attributes similar to the PS Index.

@echo –OFF

Echo Example TPM provisioning for TPM2 Provisioning Tool

# Test if PS index already exists

Tpm2ProvTool NvReadPublic MyIndex.iDef

if %lasterror% == 0 goto VERIFY

:CREATE

echo Creating the Index

# First start a session to authorize hierarchy authorization

Tpm2ProvTool StartSession PhPolicy.sDef 2

Tpm2ProvTool AssertPolicy PhSecret.pDef 2

# Now Create the Index

Tpm2ProvTool NvDefineSpace MyIndex.iDef 2

if NOT %lasterror% == 0 goto ERROR

:WRITE

echo Writing the Index

# We need a session to authorize writing the index

Tpm2ProvTool StartSession MyIndexPolicy.sDef 3

Tpm2ProvTool AssertPolicy MySecret.pDef 3

Tpm2ProvTool AssertPolicy MyIndexOR.pDef 3

# And write the index

Tpm2ProvTool NvWrite MyIndex.iDef> 3

if NOT %lasterror% == 0 goto ERROR

GOTO DONE

:VERIFY

echo Index already exists, so let’s check if it is correct

# First check index attributes

Tpm2ProvTool NvVerifyDef MyIndex.iDef

if NOT %lasterror% == 0 goto DELETE

# Establish a PW session for read authorization

Tpm2ProvTool StartSession EmptyPW.sDef 4

Echo verifying Index Data

Tpm2ProvTool NvCompareData MyIndex.iDef 4

if NOT %lasterror% == 0 goto WRITE

goto DONE

:DELETE

echo Index exists, but is not correct and must be deleted

# First start a session to authorize hierarchy authorization

Tpm2ProvTool StartSession PhPolicy.sDef 2

Tpm2ProvTool AssertPolicy PhSecret.pDef 2

# Start a session to authorized index deletion

Tpm2ProvTool StartSession MyIndexPolicy.sDef 3

Tpm2ProvTool AssertPolicy UnDefSpecial.pDef 3

Tpm2ProvTool AssertPolicy MyIndexOR.pDef 3

Echo Deleting index

Tpm2ProvTool UndefineSpecial <MyIndex.iDef> 2 3

if NOT %lasterror% == 0 goto ERROR

Tpm2ProvTool FlushSession 2

Tpm2ProvTool FlushSession 3

goto CREATE

:ERROR

echo \*\*\*\*\*\*\*\*\* Error \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\* Error \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo \*\*\*\*\*\*\*\*\* Error \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

:DONE

Tpm2ProvTool FlushSession 2

Tpm2ProvTool FlushSession 3

Tpm2ProvTool FlushSession 4

:EOF

# License

**IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING ANY OF THE CODE OR OTHER MATERIALS PROVIDED WITH THIS LICENSE AGREEMENT.**

YOU MAY NOT COPY, INSTALL OR USE ANY OF THE MATERIALS LICENSED UNDER THIS AGREEMENT UNTIL YOU CAREFULLY READ AND AGREE TO ALL THE FOLLOWING TERMS AND CONDITIONS.

BY COPYING, INSTALLING OR OTHERWISE USING ANY OF THE MATERIALS PROVIDED HEREWITH, YOU AGREE TO ALL TERMS AND CONDITIONS SET FORTH BELOW. IF YOU DO NOT AGREE, YOU MUST NOTIFY YOUR INTEL CONTACT PERSON, REFRAIN FROM ALL USE OF ANY LICENSED MATERIALS AND RETURN ALL LICENSED MATERIALS TO HIM/HER IMMEDIATELY.

INTEL® TRUSTED EXECUTION TECHNOLOGY TOOLKIT

ENABLING TOOLS LICENSE AGREEMENT

1.0 DEFINITIONS

1.1 “**End User**” means a third party licensee to whom you license your Licensee Program (either directly or through your usual distribution channels and methods) for use on Intel-based platforms pursuant to a license agreement that complies with the requirements of Exhibit B.

1.2 “**Intel**” means Intel Corporation and its majority-owned global subsidiaries.

1.3 “**Licensee**”, “you” and “your” refers to you, the licensee under this Agreement.

1.4 “**Licensee Programs**” means software programs developed by Licensee or on Licensee’s behalf using any of the licensed Materials, as further described in Subsection 2.1(a) below.

1.5 "**Materials**" are defined as the Object Code, Source Code, Third Party Software, documentation, license key codes, APIs and any other materials (including any updates, error corrections and upgrades thereto) provided by Intel to Licensee under this Agreement.

1.6 “**Object Code**” means computer programming code provided by Intel to Licensee under this Agreement in binary form suitable for machine execution by a processor without the intervening steps of interpretation or compilation.

1.7 “**Source Code**” means computer programming code provided by Intel to Licensee under this Agreement in human-readable format.

1.8 “**Third Party Software**” is defined as third party software programs and associated information included in the Materials, which may be subject to certain additional or different third party license terms (reference Exhibit A hereto and the ReadMe and text files provided with the Materials).

1.9 “**Intel** **TXT**” refers to Intel’s proprietary Trusted Execution Technology (which is Intel’s trademark, all rights reserved).

2.0 LICENSE AND LICENSE RESTRICTIONS

2.1 LIMITED COPYRIGHT LICENSE: Subject to Licensee’s compliance with all terms and conditions of this Agreement, Intel grants to Licensee a limited, non-exclusive, non-transferable license during the term of this Agreement under Intel’s copyrights to:

a. use the Materials for the limited purposes (the “**Purposes**”) of developing, using, distributing to End Users, and supporting derivative tools, utilities and related materials (the “**Licensee Programs**”) for use on Intel-based platforms that (i) verify that BIOS and platform conform to the applicable Intel TXT requirements, (ii) provision Trusted Platform Modules (TPMs) for Intel TXT, (iii) create and manage launch control policies for Intel TXT, (iv) evaluate and manage platform configuration regarding any aspect of Intel TXT, and/or (v) otherwise promote appropriate use of TXT;

b. make, and distribute internally to your employees and contractors who have agreed in writing to comply with the terms of this Agreement, copies of the Materials (including the Source Code) solely for use in support of the Purposes;

c. make, and distribute externally to your End Users, copies of the Materials (excluding the Source Code, which you are not permitted to redistribute externally), independently and/or as incorporated into your Licensee Programs, solely for use in support of the Purposes; and

d. modify and create derivative works of the Materials, and distribute such derivative works to End Users subject to Intel’s rights in the Materials as specified herein (again, you are not permitted to redistribute externally any of the Source Code), solely for use in support of the Purposes.

2.2 LICENSE RESTRICTIONS: Licensee may not:

a. use, copy or distribute any of the Materials except as expressly permitted by this Agreement;

b. rent, lease or otherwise make available any of the Materials to any third party, except as expressly permitted by this Agreement;

c. assign this Agreement or transfer any of the Materials without the express written consent of Intel;

d. modify, adapt or translate any of the Materials, in whole or in part, except as provided in this Agreement;

e. decompile, disassemble or reverse engineer any of the Materials;

f. remove or in any manner alter any product identification, proprietary, trademark, copyright or other notices contained in the Materials;

g. attempt to modify or tamper with the normal function of a license manager that regulates usage of the Materials; or

h. use Intel's name, logo or any Intel trademark without Intel’s written permission.

3.0 OWNERSHIP: The Materials are and shall remain the property of Intel or its third party suppliers. Licensee understands and agree that no license under any Intel patent, copyright (except as expressly described in Section 2.1 above), trade secret or other intellectual property right is granted or conferred upon Licensee either expressly, by implication, inducement, estoppel or otherwise, and that any further license under such intellectual property rights must be express and in writing. Title in and to any permitted derivative works of the Materials shall be held by Licensee subject to Intel’s underlying ownership of the Materials.

4.0 FEEDBACK: Licensee is not obligated to provide Intel with comments or suggestions regarding the Materials. However, should Licensee provide Intel with any feedback, input, designs, comments or suggestions for the modification, correction, improvement or enhancement of any of the Materials (collectively, “**Feedback**”), then Licensee hereby grants to Intel a non-exclusive, perpetual, irrevocable, worldwide, royalty-free license under Licensee’s intellectual property rights in such Feedback, including the right to sublicense it, and the rights to use and disclose it in any manner Intel chooses, and to display, perform, create derivative works, copy, make, sell and otherwise dispose of Intel’s and its sublicensees’ products embodying such Feedback in any manner and via any media Intel chooses without reference to the source of such Feedback.

5.0 AUDITS: Intel reserves the right, not more than once per year, to conduct an audit of Licensee’s activities relating to this Agreement upon not less than five (5) business days prior written notice and during Licensee’s normal business hours solely to verify Licensee’s compliance with the terms and conditions of this Agreement. If any such audit discloses substantial material breaches by Licensee, without limiting any other remedies Intel may have, Licensee shall reimburse Intel for the reasonable costs of such audit (including Intel employee time).

6.0 DISCLAIMER OF WARRANTIES

NEITHER INTEL NOR ITS SUPPLIERS MAKE ANY REPRESENTATION OR WARRANTY OR CONDITION OF ANY KIND, WHETHER EXPRESS OR IMPLIED (EITHER IN FACT OR BY OPERATION OF LAW), WITH RESPECT TO ANY OF THE LICENSED MATERIALS. INTEL AND ITS SUPPLIERS EXPRESSLY DISCLAIM ALL EXPRESS AND IMPLIED WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. WITHOUT LIMITING THE FOREGOING, INTEL AND ITS SUPPLIERS DO NOT WARRANT THAT THE SOURCE CODE OR OBJECT CODE IS ERROR-FREE OR THAT ITS OPERATION WILL BE SECURE, ERROR-FREE OR UNINTERRUPTED, AND HEREBY DISCLAIM ANY AND ALL LIABILITY ON ACCOUNT THEREOF. ALL MATERIALS ARE LICENSED ON AN "AS-IS" BASIS AND NEITHER INTEL NOR ITS SUPPLIERS WILL PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES HEREUNDER. INTEL AND ITS SUPPLIERS MAY PROVIDE ANY CORRECTIONS, UPDATES, ENHANCEMENTS OR EXTENSIONS AT THEIR SOLE DISCRETION.

7.0 LIMITATION OF LIABILITY

IN NO EVENT WILL INTEL OR ITS SUPPLIERS BE LIABLE IN ANY WAY FOR: (A) ANY REPRESENTATION OR WARRANTY MADE TO ANY THIRD PARTY BY LICENSEE OR ITS ANY AGENT; (B) ANY FAILURE OF ANY OF THE MATERIALS TO PERFORM AS PLANNED; OR (C) ANY DAMAGE OR LIABILITY RESULTING IN ANY WAY FROM ANY USE OF THE MATERIALS.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, INTEL AND ITS SUPPLIERS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, IN RELATION TO THIS AGREEMENT.

IN NO EVENT SHALL INTEL’S AGGREGATE LIABILITY FOR ANY CLAIMS UNDER OR RELATED TO THIS AGREEMENT EXCEED ONE THOUSAND DOLLARS ($1,000).

.

8.0 CONFIDENTIALITY: Licensee acknowledges that Intel considers the Source Code Intel’s confidential trade secret, and shall treat it as such in accordance with Licensee’s confidentiality agreement with Intel or, if no such agreement has been agreed by Intel and Licensee, in accordance with Intel’s standard confidentiality agreement generally used for similar development licenses at the time the Materials are provided to Licensee.

9.0 TERM, TERMINATION AND EFFECT OF TERMINATION

9.1 TERM: This Agreement shall commence when Licensee copies, installs or otherwise uses any of the Materials provided by Intel hereunder, and shall remain in effect until terminated by either party.

9.2 TERMINATION: Either Intel or Licensee may terminate this Agreement at any time, with or without cause, on thirty (30) days written notice. This Agreement shall also terminate automatically and immediately upon any breach by Licensee of any of the license restrictions in Subsection 2.2. Any such termination shall be without prejudice to any other remedy available to either party, at law or in equity.

9.3 EFFECT OF TERMINATION: Upon termination of the Agreement for any reason, Licensee shall (a) cease all distribution and other use of the Materials, (b) at Intel's option, either return to Intel or destroy the original and all full or partial copies of the Materials, and (c) certify in writing to Intel that they have been destroyed. In addition, in the event Licensee breaches any of the license restrictions in Subsection 2.2, upon written request by Intel, Licensee shall cease all distribution and other use of any derivative works of the Materials, and provide written certification of same to Intel. Irrespective of the basis of termination, however, Licensee may retain one (1) copy of the Materials solely for Licensee’s own internal use in supporting its products.

9.4 SURVIVAL: Those provisions of this Agreement, which by virtue of their nature and surrounding circumstances, reasonably should survive termination of this Agreement shall so survive (including, without limitation, Subsection 2.2, Sections 3.0 through 8.0, Subsections 9.3 and 9.4 and 11.0).

10.0 EXPORT COMPLIANCE: Licensee shall not export, either directly or indirectly, any of the Materials or any product, service or technical data or system incorporating any of the Materials or derivative works thereof in violation of any U.S. export law or without first obtaining all licenses and other approvals required by the U.S. Department of Commerce and any other agency or department of the U.S. government.

11.0 U.S. GOVERNMENT RESTRICTED RIGHTS: The Materials, technical data and computer software covered by this Agreement is a “Commercial Item,” as such term is defined by the FAR 2.101 (48 C.F.R. 2.101) and is “commercial computer software” and “commercial computer software documentation” as specified under FAR 12.212 (48 C.F.R. 12.212) or DFARS 227.7202 (48 C.F.R. 227.7202), as applicable. This commercial computer software and related documentation is provided to end users for use by and on behalf of the U.S. Government, with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Use for or on behalf of the U.S. Government is permitted only if the party acquiring or using this software is properly authorized by an appropriate U.S. Government official. This use by or for the U.S. Government clause is in lieu of, and supersedes, any other FAR, DFARS, or other provision that addresses Government rights in the computer software or documentation covered by this license. All copyright licenses granted to the U.S. Government are coextensive with the technical data and computer software licenses granted herein. The U.S. Government shall only have the right to reproduce, distribute, perform, display, and prepare derivative works as needed to implement those rights.

12.0 GENERAL

12.1 GOVERNING LAW AND VENUE: Any claim arising under or relating to this Agreement shall be governed by the internal substantive laws of the State of Delaware or federal courts located in Delaware, without regard to principles of conflict of laws. Each party hereby agrees to jurisdiction and venue in the courts of the State of Delaware for all disputes and litigation arising under or relating to this Agreement.

12.2 SEVERABILITY: If any paragraph, provision, or clause in this Agreement shall be found or held to be invalid or unenforceable in any jurisdiction in which this Agreement is being performed, the remainder of this Agreement shall be valid and enforceable and the parties shall use good faith to negotiate a substitute, valid and enforceable provision which most nearly effects the parties' intent in entering into this Agreement.

12.3 ENTIRE AGREEMENT: This is the complete and exclusive Agreement between the parties relating to this subject matter. No amendment shall be effective unless in writing signed by authorized representatives of both parties.

**EXHIBIT A**

**TO INTEL TRUSTED EXECUTION TECHNOLOGY TOOLKIT**

**ENABLING TOOLS LICENSE AGREEMENT**

(**MATERIALS TO BE SUPPLIED BY INTEL)**

The Materials to be supplied to Licensee by Intel under the Agreement include the following and such other Materials as Intel may identify to Licensee:

**1. Intel TXT Tool Kit**

(tools to help Licensee or Licensee’s customer develop, test, debug and verify that BIOS and platform conform to Intel Trusted Execution Technology requirements)

**2. Intel TXT Factory TPM Provisioning Tool**

(sample source code and binary object code to help Licensee develop a tool to provision Trusted Platform Modules (TPMs) for Intel-based platforms using Intel Trusted Execution Technology)

**3. Intel TXT Launch Control Policy Generation Tool for Windows**

(sample source code and binary object code to help Licensee develop tools to create and manage launch control policies for Intel-based platforms using Intel Trusted Execution Technology)

The Third Party Software provided with the Materials include the following and such other Third Party Software as Intel may identify to Licensee:

a. **Expat XML Parser** (http://expat.sourceforge.net/)

b. **Linux Launch Control Policy (LCP) Tools** (http://www.openssl.org/source/license.html)

c. **Cryptographic software written by Eric Young and Tim Hudson** **(crypto/asn1/asn1\_mac.h)**

Copyright 1995-1998 Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)), All rights reserved.

**EXHIBIT B**

**TO INTEL TRUSTED EXECUTION TECHNOLOGY TOOLKIT**

**ENABLING TOOLS LICENSE AGREEMENT**

(**MINIMUM REQUIRED TERMS FOR END USER LICENSES)**

Licensee must include, in all its license agreements with End Users to whom Licensee provides any Licensee Program, terms at least as protective of Intel’s rights and interests as the terms listed below. This requirement applies only to Licensee Programs and related Licensee materials that include any Intel Materials that were provided to Licensee under the Agreement.

1. End User may not decompile, disassemble or reverse engineer any of the Materials (except as may be mandated by applicable law, e.g., in limited circumstances to permit interoperability);

2. End User may not remove or in any manner alter any Intel or third party product identification or any trademark, copyright or other proprietary notices or license terms contained in the Materials;

3. End User may not use Intel's name, logo or any Intel trademark without Intel’s written permission;

4. End User may not use any of the Materials with any platform that is not Intel-based;

5. End User may not export any of the Materials in violation of any law, regulation, order or other restriction of the United States government and its agencies or any foreign government;

6. Licensee must disclaim, on behalf of Intel and all its third party licensors who provided any component of the Materials, all warranties (express and implied), liability (including, without limitation, all direct, indirect, incidental, consequential and special damages of any kind) and support obligations to End User relating to the Materials or Licensee Programs, to the maximum extent permitted by applicable law;

7. End User must agree that (i) it receives only RESTRICTED RIGHTS in the Materials (as that terms is understood in government procurement), (ii) Intel and its licensors retain ownership of the Materials, and (iii) End User receives no right or license under any Intel patents, trademarks or trade secrets, and no implied licenses under any Intel copyrights not expressly granted; and

8. End User must agree that its license to the Materials may be terminated if End User violates any license restriction relating to any Materials or any other term relating to Intel or its third party licensors.

The XML parser was obtained from <http://expat.sourceforge.net/> with under the following license

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Other components were developed by Intel

This is “beta” version software - Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

# Disclaimer

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE MATERIALS, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR ACCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. *Intel and its suppliers further do not warranty the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. Intel may make changes to these materials, or the products described therein, at any time without notice. Intel makes no commitment to update the Materials.*

Copyright © 2013- Intel Corporation. All rights reserved. Intel, the Intel logo, and the Intel Xeon logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.