

Intel® VPro™ Technology Platform

Setup and Configuration Application User Guide

June 2007

Revision 3.0.91

Intel Confidential



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

This document contains information on products in the design phase of development.

All products, platforms, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice. All dates specified are target dates, are provided for planning purposes only and are subject to change.

This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local sales office that you have the latest datasheet before finalizing a design.

Intel® Active Management Technology requires the computer system to have an Intel(R) AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. With regard to notebooks, Intel AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see <http://www.intel.com/technology/manage/iamt/>

Intel, vPro and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2007, Intel Corporation. All rights reserved.



Contents

1	Introduction	7
1.1	What is Setup and Configuration?	7
1.2	Setup Types	7
1.3	Secure Communications and Authentication Options.....	7
2	Intel® AMT Enterprise Setup and Configuration TLS-PSK Flow.....	9
2.1	Introduction: TLS-PSK vs. Remote Configuration.....	9
2.2	TLS-PSK Modes.....	10
2.3	Setup and Configuration Network Layout	11
2.3.1	Intel® AMT Setup and Configuration Application (SCA)	12
2.3.2	Intel® AMT Machine	12
2.3.3	DHCP Server	12
2.3.4	DNS Server	12
2.4	Configuring the SCA	12
2.5	Factory Mode Setup.....	13
2.5.1	Host Name	15
2.5.2	TCP/IP Settings.....	15
2.5.3	SCA Server Address ("Provisioning Server")	16
2.5.4	Setup Type ("Provision Model")	16
2.5.5	Virtual Local Area Network (VLAN) Settings.....	16
2.5.6	PID-PPS	17
2.5.7	Other Settings	18
2.5.8	Exit Intel® AMT Configuration.....	18
2.6	Using a USB Storage Device for Factory Mode Setup	18
2.6.1	Requirements	18
2.6.2	Preparation	18
2.6.3	Initializing a Platform.....	19
2.6.4	Moving to Setup Mode.....	19
2.7	Preparing Intel® AMT for Future Configuration	19
2.8	Setup and Configuration Application Flow (PSK).....	19
2.9	Setup Mode "Hello" Messages.....	21
3	Restoring Intel® AMT to Factory Mode	25
4	Remote Configuration.....	27
4.1	Overview of Remote Configuration Flow.....	27
4.1.1	Initial Conditions.....	27
4.1.2	Acquiring a Server Certificate.....	28
4.1.3	Steps leading to the start of Setup and Configuration	28
4.2	The Remote Configuration "Hello" Message	29
4.3	Remote Configuration Setup and Configuration Process.....	29
4.4	Intel® AMT Release 3.0 Additional Features	30
4.4.1	Simplified One-Touch.....	30
4.4.2	Bare Metal Setup and Configuration	31
4.4.3	USB Key Support for Remote Configuration.....	31
4.4.4	Requirements	31



	4.4.5	Preparation	32
	4.4.6	Initializing a Platform	32
	4.4.7	Moving to Setup Mode	32
	4.5	Remote Configuration Local Agent	33
	4.6	Parameters in CONF.XML that support Remote Configuration	34
5		Installing and Running the Sample SCA	35
	5.1	Sample SCA Installation Folders Layout	35
	5.2	Obtaining a Certificate for the Sample SCA	38
	5.3	Issuing a Management Console (Client) Certificate	39
	5.4	Changing Certificate Properties	41
	5.5	GETCFG.BAT	41
	5.6	Intel® AMT Device Configuration Parameters	42
	5.7	Required Setup Parameters	45
	5.8	SCA Command Usage	45
	5.9	Known Issues	45
6		Configuration Server Components	49
	6.1	ConfigurationServer.exe Application	49
	6.2	The SOAP Module	49
	6.3	The Socket Module	50
	6.4	Script Plug-in and Configuration Files	50
	6.5	Crypto Provider	50
7		Configuration Server Batch Scripts	51
	7.1	Certificate Management Scripts	51
		7.1.1 CHECKCA.BAT	51
		7.1.2 ROOTCA_GEN.BAT	51
		7.1.3 SUBCA_REQ.BAT	51
		7.1.4 SUBCA_SIGN.BAT	51
		7.1.5 CLEAN.BAT	52
		7.1.6 CERTGEN.BAT	52
		7.1.7 GENCERTCHAIN.BAT	52
		7.1.8 CHECKCS.BAT	52
	7.2	Configuration and Management Scripts	53
		7.2.1 GETCFG.BAT	53
		7.2.2 PROVEND.BAT	53
		7.2.3 create_usb_file.bat	53
	7.3	*.CONF.xml File Format	54
	7.4	PSK.REPOSITORY.XML File Format	60
8		Issuing Certificates and Certificate Authority	61
	8.1	CA Trust Relations	61
	8.2	Certificate Enrollment	62
	8.3	Certificate and Key Format	62
	8.4	VeriSign Certificate Chain Format	62
		Example:	63
Appendix A		: Using an Enterprise CA to Sign the Sample SCA Certificate	65



Figures

Figure 5-1. Internet Explorer* Alert:	46
Figure 5-2. Mozilla* Alert:	47



Revision History

Revision Number	Description	Revision Date
3.0.91		June 2007

§



1 Introduction

The Setup and Configuration Application (SCA) is a computer program that can be used to configure the Intel® AMT device.

Topics covered by this Guide:

1. The enterprise setup and configuration process required by Intel AMT
2. How to use the SCA
3. How to configure the SCA
4. The internal elements of the SCA

1.1 What is Setup and Configuration?

Setup and Configuration is the process that makes Intel AMT features accessible to management applications. Intel AMT devices are by default delivered in an unconfigured state. Before management applications can access an Intel AMT device, the device must be populated with various configuration settings such as usernames, passwords, network parameters, Transport Layer Security (TLS) certificates, and keys necessary for secure communications.

1.2 Setup Types

Intel AMT supports two setup types (also known as provisioning modes or models): **Small Business** and **Enterprise**. An OEM sets the appropriate default setup type as part of a factory procedure when building the Intel AMT flash image. The Small Business setup, which does not support TLS-based communication, is used when sufficient infrastructure is not available to support the recommended Enterprise setup. Refer to the *Small Business Configuration UserGuide* for a detailed description on how to perform a Small Business Setup.

Enterprise setup is designed to serve the needs of large organizations. When supported with the proper network infrastructure services, enterprise setup can provide automated one-touch setup and configuration for Intel AMT platforms.

Releases 2.2, 2.6 and 3.0 introduce Remote Configuration. This feature reduces the effort of deploying an Intel AMT platform by removing the need to send IT personnel to initiate setup on a platform while maintaining a secure setup and configuration process. This feature was formerly known as Zero-Touch Configuration, or ZTC. Several of the functions described in the appendix use this nomenclature.

1.3 Secure Communications and Authentication Options

Intel AMT supports Transport Layer Security (TLS), and, starting with Intel AMT Release 2.0, there is a mutual authentication option. TLS and mutual authentication are optional. A critical portion of the setup and configuration activity is the exchange of secret keys and installation of certificates that are required to implement TLS and mutual authentication. Please note the following:



- Intel AMT Release 1.0 or later releases operating in Legacy Mode (making it compatible with Intel AMT Release 1.0) performs the configuration process by exchanging sensitive data in an unsecure manner with a configuration server. Therefore, such Intel AMT devices should be configured on an isolated network.
- An Intel AMT device supporting Release 2.0 or later can be initialized with a public identifier and a private key (a PID/PPS pair). The configuration server must have these two values as well as the internal UUID of the Intel AMT device for the configuration process to start. The secure handshake done using this information allows the setup and configuration process to take place on an open enterprise network.
- TLS requires that each Intel AMT device has a signed certificate that is traceable to a Certificate Authority. The setup and configuration application implements the process required to request, sign, and install a **server certificate** in an Intel AMT device.
- Mutual authentication requires that an Intel AMT device have a **trusted root certificate** installed. This certificate will be used to validate clients that attempt to access the Intel AMT device. This includes both remote applications (generally referred to as management consoles), and applications running on the local host processor that communicate with Intel AMT, for example, an anti-virus application.
- Release 2.2, 2.6 and 3.0 support "Remote Configuration". This feature allows setup and configuration of an Intel AMT device without having to install a PID/PPS pair. Platforms that support Remote Configuration always use mutual authentication during setup and configuration. They have one or more pre-installed **root certificate hashes** used to authenticate the setup and configuration application. The Intel AMT device sends a **self-signed certificate** used by the setup and configuration server to establish a secured connection with the Intel AMT device. The protocol used is **PKI-CH** (Public Key Infrastructure – Certificate Hash). See [Remote Configuration](#) for a detailed description.

This document discusses each of these issues in detail.

§



2 Intel® AMT Enterprise Setup and Configuration TLS-PSK Flow

2.1 Introduction: TLS-PSK vs. Remote Configuration

Enterprise Setup and Configuration is a sequence of steps used to configure an Intel AMT device in a secure manner. The process requires a Setup and Configuration server on a platform external to the Intel AMT-based platform, and, optionally, other servers to support such functions as generation of certificates and keys and allocation of IP addresses. The SCA is used throughout the following discussion as an example of a Setup and Configuration server (SCS). An ISV-developed SCS could perform the equivalent functionality of the SCA.

Before an Intel AMT device can receive its configuration setting over the network, it first must be prepared with some initial setup information. The initial information will be different, depending on the available options in Intel AMT release and the settings performed by the platform OEM. The table below shows the possible initial conditions and approaches for Enterprise setup and configuration.

Setup Method	Description	Applicable Intel AMT Releases	Initial Conditions	Operator Actions Required
Legacy Enterprise	Unsecure; uses username and password in the clear to start; type 1 "Hello" message.	1.0; later releases in legacy mode	Platform configured for enterprise mode.	Operator changes MEBx password
Enterprise PSK	PID-PSK pair; type 2 "Hello" message.	2.0 and up	No pair provided	Operator enters pair manually via menu or with USB storage device
			OEM pre-installs pair	No operator action required*
Enterprise PKI-CH (Remote Configuration)	Built-in root certificate hashes; self-signed certificate; type 3 "Hello" message. SCA has a client certificate that matches one of the certificate hashes. Depends on DHCP active. Delayed start of configuration.	2.2, 2.6, 3.0	OS running with ISV agent installed	No actions required: ISV console tells agent to open Intel AMT network interface (and may perform other settings). Console supplies OTP to Agent to pass to Intel AMT device and also sends it to SCA.

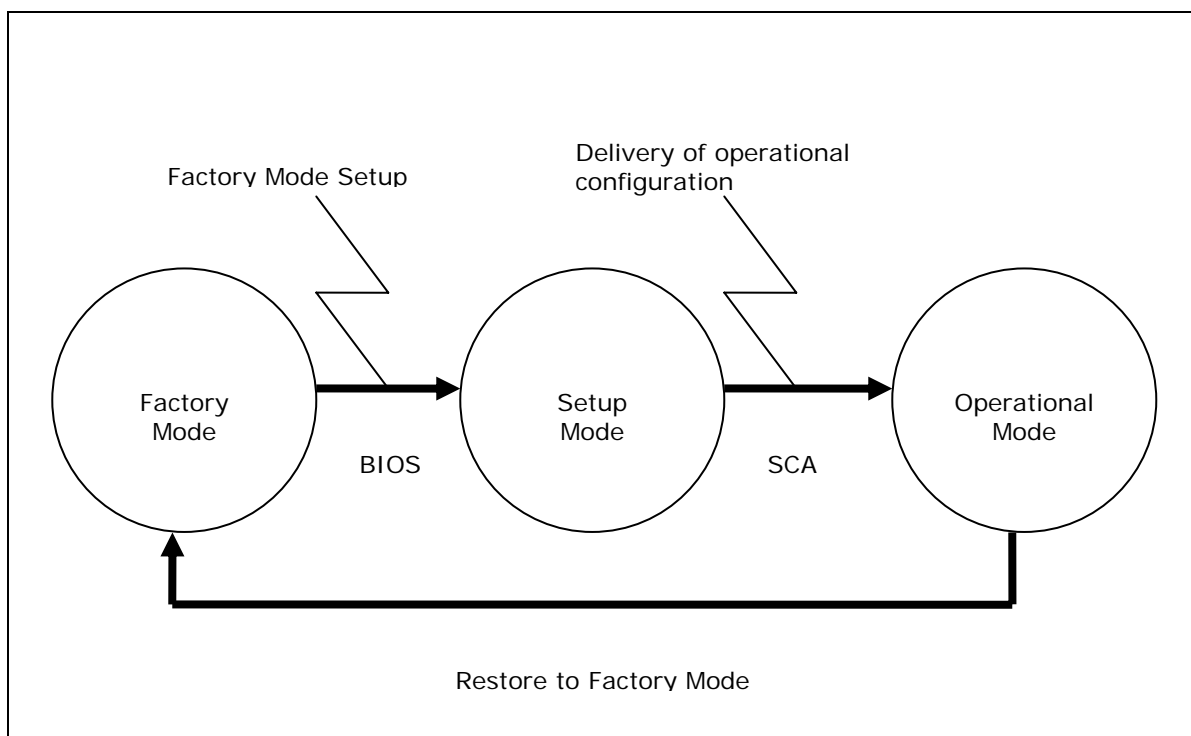


Setup Method	Description	Applicable Intel AMT Releases	Initial Conditions	Operator Actions Required
Enterprise PKI-CH (Remote Configuration) (Bare Metal)	Built-in root certificate hashes; self-signed certificate; type 3 "Hello" message. SCA has a client certificate that matches one of the certificate hashes. Depends on DHCP active. Setup starts as soon as platform is connected to the network.	3.0	No ISV local agent running on host and OEM sets Bare Metal as a default.	No actions required: Platform starts sending "Hello" message as soon as it is connected to the network.

The following sections describe the Enterprise TLS-PSK setup method. See [Remote Configuration](#) for a description of the PKI-CH-based Remote Configuration setup process.

2.2 TLS-PSK Modes

The following diagram shows the modes or stages that an Intel AMT device passes through before it becomes operational. The device arrives from the OEM's factory in "Factory Mode". It transitions to "Setup Mode" and, after setup, moves to "Operational Mode". With the proper commands, the device can return to "Factory Mode".





Factory Mode

Intel AMT comes from the factory in Factory Mode. In this mode Intel AMT is unconfigured and not available for use by management applications. When an operator enters information via the Intel AMT BIOS extension manually or with the aid of a USB storage device, Intel AMT makes the transition into setup mode. See [Factory Mode Setup](#) for instructions on how to prepare an Intel AMT device to receive its configuration settings from an SCA.

Setup Mode

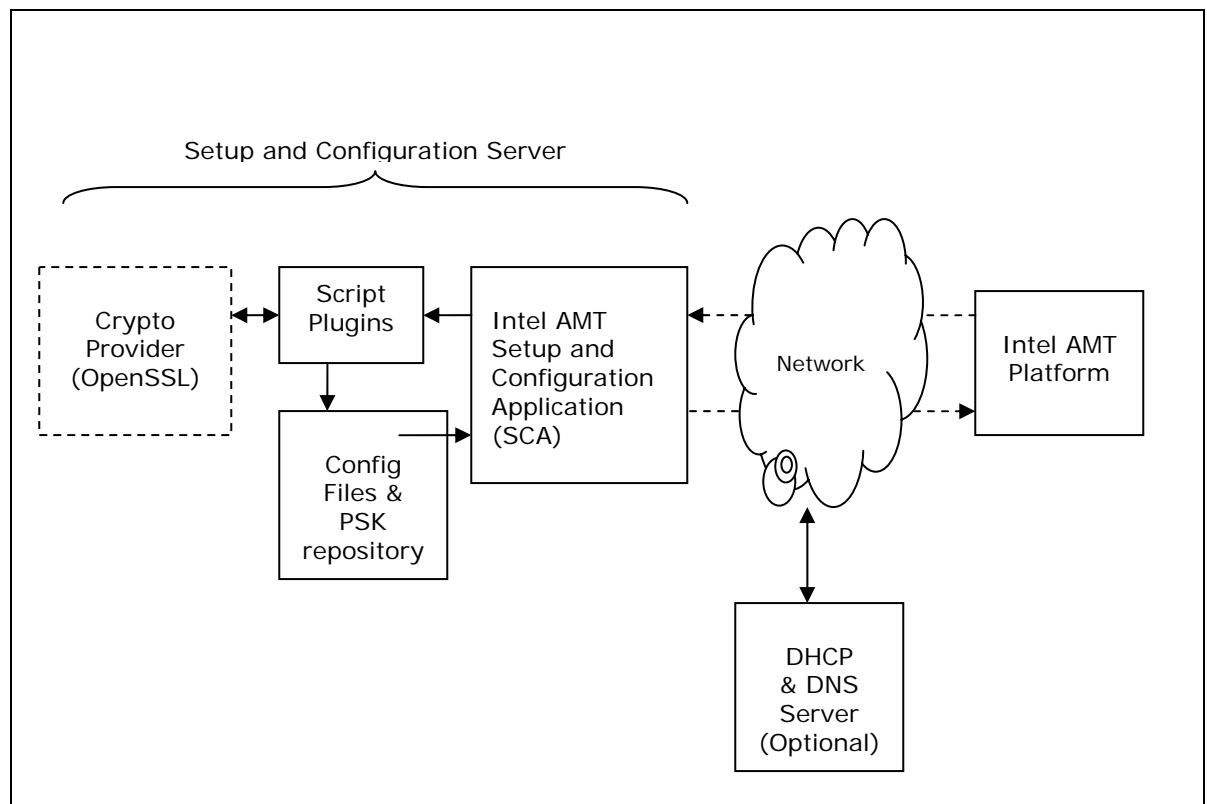
When an Intel AMT device enters Setup Mode it waits for delivery of its configuration settings from an SCS. After it enters setup mode, the Intel AMT device periodically sends messages to the SCS. When the SCS receives messages from the Intel AMT device, it responds by delivering the configuration settings and placing the device in Operational Mode. See [Setup and Configuration Application Flow](#) as well as subsequent sections of this guide.

Operational Mode

Intel AMT enters Operational Mode once its configuration settings have been supplied and committed. At this point Intel AMT is ready to interact with management applications.

2.3 Setup and Configuration Network Layout

The following sections describe the components involved in the Setup and Configuration process. The diagram below shows the components and their interactions:





2.3.1 Intel® AMT Setup and Configuration Application (SCA)

The Setup and Configuration Application (SCA) is a computer program used to deliver operational settings to Intel AMT devices over the network. The SCA completes the setup and configuration process by supplying the Intel AMT device with customized parameters. The machine that SCA software runs on is referred to as the Setup and Configuration Server (SCS), sometimes referred to as a provisioning server. When an Intel AMT device enters Setup Mode, it attempts to establish a network connection with the SCS and waits for the SCA software running on the server to deliver configuration settings.

2.3.2 Intel® AMT Machine

An Intel AMT Machine cannot receive its configuration settings from an SCA until it is brought out of its default factory state and placed into Setup Mode. Once they are in Setup Mode, Intel AMT devices periodically send messages to the SCA. These messages allow the SCA to identify the individual device needing to be configured. See [Factory Mode Setup](#) for instructions on how to place an Intel AMT device into Setup Mode.

2.3.3 DHCP Server

Intel AMT devices, by default, obtain their network settings from a DHCP server. If DHCP services are not available then the Intel AMT device must be configured to use static IP network settings. [TCP/IP Settings](#) describes configuring the network settings during Factory Mode setup.

2.3.4 DNS Server

When an Intel AMT device enters Setup Mode, by default it attempts to obtain the IP address of the SCA automatically by performing a DNS query for a host name of "ProvisionServer". (Note that an OEM platform provider can change "ProvisionServer" to some other value.) If a DNS is unavailable, then the SCS IP address must be explicitly set during Factory Mode setup. See [SCA Server Address](#) for the steps required to set the SCA Server IP address.

2.4 Configuring the SCA

The SCA must be configured so that all communications with Intel AMT devices under its control are secure. The optional mutual authentication capability available from Intel AMT Release 2.0 and onward requires additional support from the SCA to configure the appropriate root certificate.

Configuring the SCA software includes setting up the application to conduct certificate operations, defining the Pre-shared Key (PSK) Repository and setup of the .CONF.XML file(s).

The first time the sample SCA is started it creates a certificate request. The user is prompted to sign this request. See [Obtaining a Certificate for the Sample SCA](#) for details on configuring the SCA for certificate operations.

The `psk.repository.xml` should be modified to contain a set of PID-PPS key pairs for each Intel AMT platform. These keys match PID-PPS key pairs entered during Factory Mode setup and are tied to a specific Intel AMT device. PID-PPS key pairs are used to establish a secure communication channel with Intel AMT Release 2.0 and later devices during setup and configuration. See section [PID-PPS](#) for more information regarding PID-PPS key pairs.



Note: The `psk.repository.xml` is not used when setting up Intel AMT Release 1.0 devices. The default `conf.xml` file should be modified to contain the desired configuration settings for any Intel AMT devices to be configured by the SCA. These settings will be applied to all instances of Intel AMT unless the user creates a separate file for each device. An instance-unique file has the name `<UUID>.conf.xml` (where `<UUID>` is the actual UUID of the Intel AMT device). Such a file will contain the full set of configuration parameters including those that are unique for the device. See [Intel AMT Device Configuration Parameters](#) for the parameter options.

Note: Use the default `conf.xml` file to configure one device, then change the device-unique parameters (such as hostname), then configure the next device. This method assumes that the user knows which device will be connecting to the SCA next. By using UUID-specific xml files, the SCA can configure Intel AMT devices whenever they connect to the SCA in no particular order.

2.5 Factory Mode Setup

This section describes the steps required prepare an Intel AMT device to receive its configuration settings from an SCA. The Intel AMT BIOS extension screens are used to conduct the Factory Mode setup. During power up, the Intel AMT machine will first check for the presence of a USB storage device. If the device is present then the setup will proceed as described in [Using a USB Storage Device for Factory Mode Setup](#). The PID/PPS pair will be installed and, optionally, the Intel® Management Engine BIOS extension password will be changed. If there is no USB device, the platform will display the BIOS startup screen, and then the BIOS Extensions will be processed. Entry into the Intel AMT BIOS Extension is BIOS vendor dependent. The BIOS implementation may require that the user enable the BIOS extension from the BIOS.

Intel AMT reference platforms display a screen prompting the user to press `<Ctrl+P>`. Pressing `<Ctrl+P>` passes control to the Intel® Management Engine BIOS extension (MEBx) Main Menu. Perform the following steps:

1. Enter the MEBx default password ("admin")
2. Change the default password to a new value (this step is required to proceed.) The password should be a strong password (i.e., it should contain at least one upper case letter, one lower case letter, one digit and one special character, and be at least eight characters). Intel AMT uses this password for authentication during Setup and Configuration. (In Release 3.0, a setup and configuration application can use the original password "admin" until setup and configuration completes). The password may already have been changed using a USB storage device. See [Using a USB Storage Device for Factory Mode Setup](#). Once Setup mode has begun, a management console application can change the Intel AMT password without modifying the MEBx password.
3. Select Intel(R) ME Platform Configuration
4. A warning message is displayed saying that a reset will occur after configuration is complete. Enter "Y".
5. Select Intel(R) ME Features Control, and then select Manageability Feature Selection.
6. Select Intel(R) AMT, and return to previous menu.
7. Select the Intel(R) ME Power Control menu
8. (Intel AMT Release 2.0/2.1) Select the following power control settings
Section .01 Intel(R) ME State upon Initial Power-On = ON
Section .02 Intel(R) ME ON in Host Sleep States = Always
Section .03 Intel(R) ME Visual LED Indicator = ON
9. (Intel AMT Release 2.5)

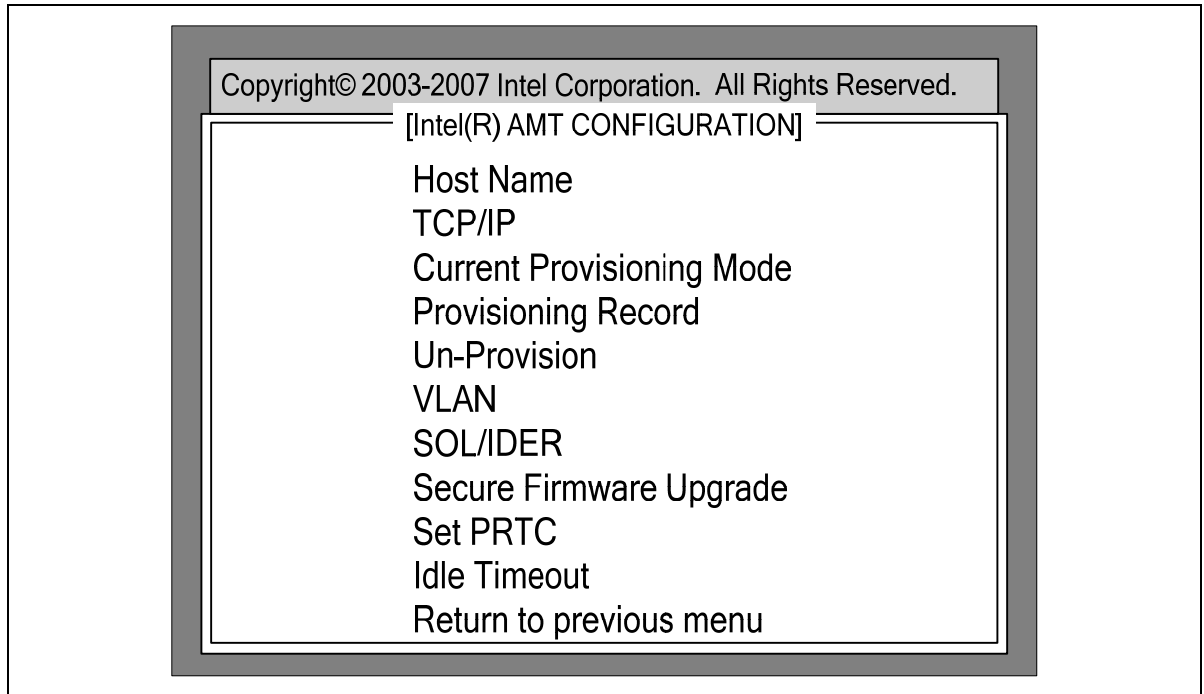


Select from one of the following choices (OEMs can select which options will be available in this list):

Mobile: On in S0	(The Intel ME and Intel AMT are on only when the host is on—this is the default setting.)
Mobile: On in S0, S3/AC	(The Intel ME and Intel AMT are on when the host is on or when the host is in S3, as long as the platform is connected to AC power.)
Mobile: On in S0, S3/AC, S4-5/AC	(The Intel ME and Intel AMT are on when the host is on or when the host is in S3 to S5, as long as the platform is connected to AC power.)
Mobile: On in S0, ME WoL in S3/AC	(The Intel ME and Intel AMT are on when the host is on. When the host is in S3 and the platform is connected to AC power, the ME will shut down after a defined period of time, but will awaken when it receives a network message –Wake on LAN.)
Mobile: On in S0, WoL in S3/AC S4-5/AC	(The Intel ME and Intel AMT are on when the host is on. When the host is in S3 to S5 and the platform is connected to AC power, the ME will shut down after a defined period of time, but will awaken when it receives a network message – Wake on LAN.)

(Intel AMT Release 3.0) Select Desktop: ON in S0, S3, S4 -5.

10. Return to the previous menu.
11. Exit all menus. The computer will restart. Press <Ctrl+P> and enter the Main Menu.
12. Now select “Intel(R) AMT Configuration” and press “Enter”. On Intel AMT Release 2.0/2.1 platforms, the following BIOS extension screen will be displayed:



2.5.1 Host Name

The host name is entered here optionally. The SCA needs to know the hostname independently, as the “Hello” message from Intel AMT does not include it.

2.5.2 TCP/IP Settings

By default, the Intel AMT network is enabled. After selecting the TCP/IP option, the BIOS extension prompts “Disable Network Interface Y/N”. Select 'N' to leave Intel AMT network capabilities enabled. Select 'Y' to disable Intel AMT network capabilities.

Disabling Intel AMT network interface automatically configures the TCP/IP settings to disable DHCP. Return to the TCP/IP option and select Y to enable the network interface. Enabling the network interface will restore any previous settings or the default settings if there were no previous settings.

If the user chooses to leave the networking capabilities enabled then the BIOS extension displays “Disable DHCP Y/N”. By default, Intel AMT is configured to depend on a DHCP server for an IP address. Select 'N' to continue to use DHCP. Optionally enter the platform domain name. The domain name, combined with the host name and the DHCP-supplied IP address will be used by the DHCP server to register the platform on the DNS (if this capability is enabled in the DHCP server). Intel AMT uses the domain name combined with “ProvisionServer” to query the DNS for the configuration server IP. If the request fails, Intel AMT will retry using DHCP option 15. See section [DHCP Server](#) for DHCP server requirements. If no DHCP server is available, select 'Y' to disable DHCP and enter the following parameters:



IP address	(Required) Note that the platform IP address and the Intel AMT IP address must be different.
IP mask	(Required)
Gateway IP address	(Optional)
Primary DNS IP address	(Optional, see note)
Secondary DNS IP address	(Optional)
Domain name	(Optional - When working with localized BIOS, this value cannot be changed from the BIOS extension screen.)

Note: Although the DNS IP address is optional, it is required if Intel AMT needs to query the DNS to locate the SCA IP address.

2.5.3 SCA Server Address (“Provisioning Server”)

By default, the SCA Server address is set to 0.0.0.0. A value of 0.0.0.0 means that Intel AMT will attempt to obtain the actual IP address of the SCA by performing a DNS lookup for a host named “ProvisionServer”. If the DNS is unable to resolve the host name, the IP address of the SCA must be supplied manually. The name ProvisionServer can be configured by an OEM to a different value.

By default, port 9971 is used to establish a connection to the SCA. This default may be changed by an OEM. If the SCA has been configured to listen on a different port, then the actual port the SCA is listening on should be supplied.

2.5.4 Setup Type (“Provision Model”)

The default setup type of Intel AMT is Enterprise. The Small Business Setup option is used in environments where infrastructure required for TLS is not available, and configuration can be completed from the BIOS menu. For more information regarding Small Business Setup, please see the Small Business User Manual.

The Setup Type menu also allows selection of Legacy Mode. In Legacy Mode, an Intel AMT Release 2.0 or later device has the capabilities of Intel AMT Release 1.0. This allows use of ISV products developed to run with Intel AMT Release 1.0.

2.5.5 Virtual Local Area Network (VLAN) Settings

The VLAN setting allows the out-of-band traffic targeted to and transmitted from Intel AMT to be assigned to a logical, virtual LAN separate from the in-band communication on the Local Area Network. Enable or Disable this setting depending on the enterprise network configuration. Intel AMT can use a VLAN that is different from the host processor, or the host processor can be configured to operate without a VLAN definition.

When the Intel AMT device is configured to share IP addresses with the host processor, using a DHCP-assigned address, both the host and Intel AMT must be configured to use the same VLAN. The DHCP server should be enabled on this VLAN as well.



2.5.6 PID-PPS

The Provisioning ID (PID) and the Provisioning Pre-Shared Key (PPS) settings are required for establishing secure communication during the Setup and Configuration of Intel AMT platforms. These settings are not available for Intel AMT Release 1.0 platforms and for Intel AMT platforms configured in Legacy Mode.

The PID-PPS pair may have been preloaded by a platform OEM or loaded using a USB storage device. See [Using a USB Storage Device for Factory Mode Setup](#).

The PID and PPS are 64-bit quantities made up of ASCII codes of some combination of characters – capital alphabet characters (A–Z), and numbers (0–9).

The PID is an eight character entry of the form: XXXX-XXXX and is sent in the open.

The PPS is a thirty-two character quantity of the form:

AAAA-BBBB-CCCC-DDDD-EEEE-FFFF-GGGG-HHHH and is a secret shared between the Intel AMT device and the SCA.

Here is an example pair:

PID: 0000-037M

PPS: NKLD-G5DC-RRNQ-E9YZ-ZIJL-7LFL-VJED-69XJ

When the PID and PPS are entered via the BIOS submenu manually, the firmware checks for checksum characters embedded in the values. The last character of the PID is expected to be a checksum of the previous seven characters, and the fourth character in each group of four characters in the PPS is expected to be a checksum of the previous three characters. This check is made to reduce the possibility of operator error when entering these values.

The checksum calculation is the sum of the characters modulo 0x24 + 0x30 if the result is 0x0 to 0x9 or + 0x37 if the result is 0xA to 0x23. Using the PID example above,

$$0x30+0x30+0x30+0x30+0x30+0x33+0x37 = 0x15A.$$

0x15A modulo 0x24 = 0x16. This is greater than 0x9, so add 0x37 to get 0x4D, which is an ASCII M.

Using the first group from the PPS example,

$$0x4E+0x4B+0x4C = 0xE5.$$

0xE5 modulo 0x24 = 0xD. Adding 0x37 yields 0x44, which is an ASCII D.

In Intel AMT installations with Release 2.0 or greater, the same PID-PPS pair must be entered in the PSK repository of the configuration server. For the sample SCA the repository is located in the `psk.repository.xml`. These values must be maintained in a secure database as they could be used for gaining access to Intel AMT devices during the setup and configuration process by a malicious party.

Intel strongly recommends that Intel AMT Release 1.0 platforms be configured on an isolated network to minimize the opportunities for exposing security information, since Setup and Configuration traffic is sent without encryption for these types of platforms. If the Setup and Configuration Server requires access to both a production network and a private isolated network, then equip the server with more than one network interface. One network device can be used to establish isolated network connections to Intel AMT systems to be configured, and the second network device can be used to connect to the production network.



2.5.7 Other Settings

The SOL/IDER, Remote Firmware Update and Set PRTC menu options are not required for setup and configuration. The SOL/IDE-R option enables the Intel AMT redirection capabilities. The Remote Firmware Update option enables the ability to perform remote updates to the firmware. The Set PRTC allows an IT technician to set the programmable real-time clock to a correct value if the clock lost its value inadvertently in a situation where it could not be reset remotely.

2.5.8 Exit Intel® AMT Configuration

Highlight the Return to Previous Menu option and press Enter. Upon exiting the Intel AMT BIOS extension, the Intel AMT device will enter Setup Mode and begin sending "Hello" messages to the SCA, as described at [Setup Mode "Hello" Messages](#).

2.6 Using a USB Storage Device for Factory Mode Setup

The Factory mode setup process can be simplified by using a USB key containing a file of PID/PPS pairs and replacement passwords. This method can be used for one-touch configuration if all the defaults listed below are suitable for an enterprise installation. Even if additional parameters need to be changed, the USB key can install the PID and PPS without the problem of operator error. Use this method also for preparing platforms for future Intel AMT configuration.

2.6.1 Requirements

The following items are required to be able to use a USB key for Intel AMT configuration:

- A dedicated USB key with no data on it.
- A function within a setup and configuration server that generates a file of PID/PPS pairs in the proper format. The function must generate secure PPS values using a strong random number generator.

Due to the sensitivity of the data on the USB key, it is recommended that good security procedures be established for controlling the key and the information on it.

2.6.2 Preparation

All that is required is to execute the function, which will do the following:

1. Create a list of PID/PPS pairs.
2. Create a file named "setup.bin" in the proper format. The file will include:
 - a. A header that notes the number of entries and the number of used entries (initially zero)
 - b. An entry per platform to be configured that includes:
 1. The PID-PPS pair
 2. The default MEBx password (usually "admin")
 3. Optionally, a replacement password (usually the same password for all platforms)
3. Format the USB key to FAT16.
4. Write the file to the USB key.
5. Save the generated PID-PPS data in the Setup and Configuration secure store.



2.6.3 Initializing a Platform

To install the PID/PPS information on an Intel AMT platform an IT technician will:

1. Take the platform out of the box and connect cables, a monitor, and a keyboard.
2. Connect the USB key to a USB port.
3. Turn on the platform.

The BIOS on the platform will detect the presence of the USB key, read the next available entry in the file, authenticate the password, save the PID/PPS values, optionally update with the replacement password, and mark the entry on the USB key as "used". A message displayed on the monitor informs the technician that the process is complete. The technician powers down the platform.

2.6.4 Moving to Setup Mode

The platform is now in Setup Mode. If it is connected to the network, Intel AMT will start to send "Hello" messages. If there are parameters that still need setting, such as the platform IP, this should be done before connecting to the network. The "standard" defaults are:

- DHCP mode with no domain defined
- Setup and Configuration Server with the default host name and port
- No DNS IP defined (The DHCP server must be configured to provide a DNS IP, which will be required to discover the IP of the Setup and Configuration Server)

The defaults may vary depending on OEM settings. If these defaults are acceptable, the platform can now be connected to the network and powered on. Otherwise, the technician can power on the platform, enter the MEBx sub-menu and configure additional parameters. Note that with Intel AMT Release 3.0, several of the MEBx parameters cannot be set once the device is in Setup Mode.

2.7 Preparing Intel® AMT for Future Configuration

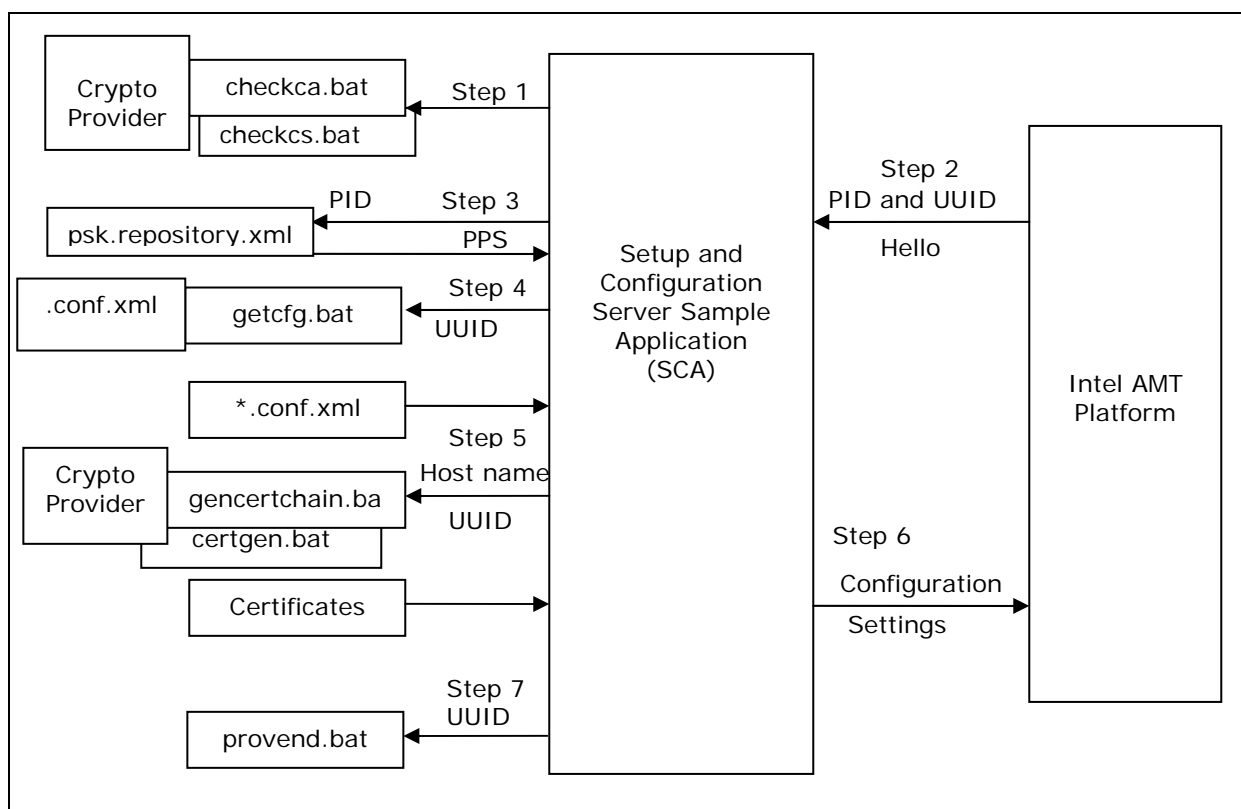
A user may wish to postpone Intel AMT setup and configuration until a later date. An OEM may supply platforms with a PID-PPS pair already written to the Intel AMT Flash memory. In this case, the platform may be already prepared for configuration, as described in the previous paragraph. The OEM will have to securely deliver a file of the PID-PPS pairs to the customer IT organization for use in the setup and configuration process. It is also possible to prepare the Intel AMT-based platform for configuration without entering Setup Mode. Either use a USB storage device, as described in [Using a USB Storage Device for Factory Mode Setup](#) or follow the steps in section 2.5, Factory Mode Setup, but under the TCP/IP menu item, select Y at the "Disable Network Interface?" option. Enter a PID-PPS pair as well. When the time comes to configure and enable Intel AMT, re-enter the BIOS sub-menu and change the TCP/IP settings to make the network interface operational by responding Y to "Enable Network Interface" and either changing to DHCP or setting the other TCP/IP parameters to valid values.

2.8 Setup and Configuration Application Flow (PSK)

The following sequence is followed when an Intel AMT device enters setup mode and the sample SCA is running. The steps are described below, with each part spelled out in detail later in the document.



Note: If a serious operational error occurs during the setup and configuration process (for example, TLS is configured incorrectly because a certificate or private key was installed inadvertently, or a certificate replacement was performed that does not align with current keys), and the platform is then transitioned to Operational Mode, the Intel AMT device may not be accessible remotely. The Intel AMT device needs to be returned to the Factory Mode by using the BIOS sub-menu Unprovision option. See [“Restoring Intel AMT to Factory Mode.”](#)



Step 1:

When the sample SCA starts, it runs two initialization scripts named “CHECKCA.BAT” and “CHECKCS.BAT”.

CHECKCA.BAT ensures that there is a subordinate CA certificate file named `subcacert.pem`. If the file is not found, the SCA performs the steps described in [Obtaining a Certificate for the Sample SCA](#).

CHECKCS.BAT ensures that there are certificates for TLS mutual authentication. Three certificates are required:

- a trusted root certificate, which is used to sign `local_client` and `remote_client` certificates
- a local client certificate, and
- a remote client certificate.



These certificates are for demonstration purposes only, and not for a production environment.

The trusted root certificate (not to be confused with the root CA created by CHECKCA.BAT) is sent to Intel AMT devices, where it will be used for client authentication in an Intel AMT device configured for mutual authentication.

Step 2:

An Intel AMT device in Setup Mode tries periodically to connect to the SCA using the settings defined during the Factory Mode setup. The platform sends setup ("Hello") messages to the SCA via a TCP/IP socket connection to the SCA listening port. The default destination port is 9971 or a value set by the platform OEM, but this value can be configured when the SCA is started. The version 2 "Hello" message contains the UUID and PID of the Intel AMT device.

Step 3:

The SCA searches `psk.repository.xml` for the PID and locates the corresponding PPS.

Step 4:

Responding to the "Hello" message, the SCA executes an external script named "GETCFG.BAT". Based on parameters received in the "Hello" message, "GETCFG.BAT" chooses an appropriate configuration file and saves its name to `conf.choice`. The sample SCA reads the name from `conf.choice`.

Step 5:

If TLS is enabled in "`..CONF.XML`", (the commands to configure TLS are different for Intel AMT Release 1.0 and for later releases) the SCA will additionally invoke "GENCERTCHAIN.BAT" to create the RSA key and certificate for the Intel AMT device. The SCA sends the RSA key and certificate to the Intel AMT platform using the SOAP protocol. If mutual authentication is enabled, then a trusted root certificate is sent to the Intel AMT device, along with optional Certificate Revocation List (CRL) and fully qualified domain name (FQDN) settings.

Step 6:

The SCA sends various configurations settings to the Intel AMT device using the SOAP protocol. The SCA finishes by sending a "CommitChanges" command which commits the settings to the Intel AMT platform.

Step 7:

The Intel AMT platform now enters Operational Mode and the SCA calls the "PROVEND.BAT" batch file to clean up files created during the setup and configuration process. A system administrator may customize this script to send email or update databases regarding the machine deployment. It is possible to make changes in the Intel AMT device configuration after it is in operational mode by using the SOAP interface.

2.9 Setup Mode "Hello" Messages

When an Intel AMT device transitions from Factory Mode to Setup Mode, it attempts to create a TCP/IP connection with the SCA on the default port. This is a four-step process:

1. Intel AMT connects to the SCA either by using the IP address entered via the BIOS extension or by looking up the address on the domain name server (DNS), using the SCA hostname. See [SCA Server Address \("Provisioning Server"\)](#). If an IP address was entered, then the connection is direct. Continue with step 2.



Intel AMT does a DNS lookup using the hostname ProvisionServer and the optional domain name entered via the BIOS sub-menu as one of the TCP/IP parameters or the default domain name, if no domain name was entered (this is an OEM option and may be blank). Intel AMT sends this lookup request even if no domain name was entered. If this lookup fails (and it will if there is no FQDN or domain suffix), Intel AMT tries a DNS lookup using a DNS suffix returned by the DHCP server, if the DHCP server is configured to return domain names (DHCP option 15). If the DNS server does not have a record for the SCS FQDN, the device will not be able to look up the FQDN of the SCA server. The user will need to either manually enter the SCA IP address via the BIOS extension or add a static alias to the DNS server, where the SCA hostname, combined with Intel AMT local domain, resolves to the SCA IP address.

2. When the device successfully connects to the SCA, it sends a "Hello" Message, which has the following format:

Byte Offset	Type	Content
0	Unsigned Short	Admin Credentials Set
2	Unsigned Short	Interface Version (2 for PSK applications)
4	Unsigned Long	Retry Count (0-14)
8	Byte 16	Device UUID
24	Byte 8	PID

3. The first two bytes will usually be 0x0001 unless the device has a localized BIOS. If there is a localized BIOS, the value will be 0x0000, indicating that new administrator credentials must be set in the Intel AMT device for Setup and Configuration to complete successfully.
4. After the "Hello" message is sent, the Intel AMT device closes the TCP/IP connection.

Intel AMT sends the "Hello" message in Host order, not in network order. To compensate for this, the sample setup and configuration application processes the message in host order. An ISV-created setup and configuration application must do the same.

The Intel AMT device will retry the three steps until configuration is complete, which is defined as having all mandatory parameters set and the Commit Changes command issued by the Configuration Server. The Intel AMT device performs retries according to the following algorithm:

- 5 retries on 1 minute intervals
- 5 retries on 10 minute intervals.
- 5 retries on 1 hour intervals.
- Releases 2.2, 2.6, and 3.0 continue with hourly "Hello" messages until the network interface closes.

The retry algorithm restarts after a firmware reset, which can happen due to a power-cycle of the Intel AMT device (i.e., disconnecting AC power from the platform).



Intel AMT Releases 2.2, 2.6, and 3.0 open the network interface for only six hours. If a PID-PPS based setup and configuration does not complete in six hours, the Intel AMT device will close the network interface. To re-open the network interface, either a local agent must command the device to open the interface or the partial unprovisioning option must be manually selected from the MEBx menu.

While Intel AMT is in Setup Mode, the device cannot be used for other application purposes. Configuration must be completed before trying to use the capabilities of the Intel AMT device.

§





3 *Restoring Intel® AMT to Factory Mode*

Intel AMT is returned to Factory Mode by selecting the Unprovision option on the BIOS Extension menu or by disabling Intel AMT from the BIOS extension Manageability Feature Selection.

Alternatively, a remote application can send an Unprovision command over the network using the SOAP interface.

The following takes place when Intel AMT is restored to Factory Mode:

1. Certificates are erased from the Non-Volatile Memory (NVM).
2. The NVM storage area is cleared.
3. The PID/PPS pair is erased.
4. The event log is cleared and all transient filters are removed from the NVM.
5. All Access Control Lists (ACL) assigned by the security administration interface are cleared and the administrator username is set to the default ("admin") and the Intel AMT password is set to the current MEBx password value.
6. The storage Factory Partner ACL (FPACL) list is restored to its factory condition.
7. The storage Enterprise ACL (EACL) list is deleted and restored to its factory state.
8. If the global storage parameters were modified, they will be restored to their default values. This applies to the default values of MaxPartnerStorage and MaxNonPartnerTotalAllocationSize.
9. Hardware asset information is erased.
10. The firmware is reset.

Once Intel AMT is restored to Factory Mode the device will no longer be available for use by management applications. The Setup and Configuration process must be performed again to restore the device operational state (see Factory Mode Setup section).

An Intel AMT device can also be partially unprovisioned. This can be done from the BIOS menu or via a remote command. The result is the same as the process described above except for the following:

- The PID/PPS pair is not erased.
- The Admin Access Control List, containing the administrator username and password, is not erased.
- The Host name is not erased.
- The provisioning server IP and port are not erased.
- The domain name is not erased.



Note the following:

Restoring Intel AMT to Factory Mode is sometimes referred to as "Un-Provisioning".

The setup type (Enterprise or Small Business) can only be changed when the device is in Factory Mode.

Restoring Intel AMT to Factory Mode is not a supported feature of Sample SCA.

What happens if the Intel AMT Admin password was lost or forgotten? This or other reasons may require unconfiguring the ME. Unconfiguring returns the ME to the factory-delivered state. It clears the internal real-time clock and all saved values, including all Intel AMT user-entered information, and returns the ME password to the default value. This can be done in one of several ways, but it is manufacturer dependent. The last two options require access to the platform motherboard. **This should be done only by qualified IT personnel:**

BIOS Command for ME reset or ME Unprovision

This option depends on an OEM including an **Unconfigure ME** (or equivalent) option in the BIOS. Local IT policy may require a password to access the BIOS on user platforms. The BIOS Chipset page may have an ME Subsystem Configuration option, which opens a page with the Unconfigure ME option. Selecting this option causes a reboot. The user will have an option to "Unconfigure ME Yes/No". Selecting "Yes" resets the ME to the state it was in when the platform was delivered.

Removing the CMOS backup battery

This requires opening the platform, locating and removing the CMOS battery, waiting about one minute, then returning the battery and restarting the platform. The platform will display a message: "ME unconfiguration in progress".

CMOS reset jumper

Many motherboard implementations include a jumper used to clear the CMOS containing local settings. See manufacturer documentation for the location and use of such a jumper. The result will be the same as removing the battery.



4 Remote Configuration

Remote Configuration is a feature added with Intel AMT Releases 2.2, 2.6 and 3.0. It eliminates the need for IT personnel to manually install a PID/PPS pair to enable setup. The Remote Configuration process depends on several Intel AMT enhancements:

- **Embedded hashed root certificates**
The Intel AMT device firmware image contains one or more root certificate hashes from recognized vendors. As part of the “Hello” message, the Intel AMT device sends all of the active hashes to the SCA. When the SCA authenticates to the Intel AMT device, it must do so with a certificate compatible with one of the hashed root certificates.
- **Self-signed certificate**
The Intel AMT device produces a self-signed certificate that it uses to establish a secure connection with the SCA. The SCA must be configured to accept such a certificate.
- **One-time password (OTP)**
Security policy may require use of a one-time password to improve security. An ISV-created agent running on the local host supplies the OTP to the Intel AMT device. The agent receives the OTP from a management console that also sends the OTP to the SCA.
- **Limited network access**
The network interface opens for a limited period of time to send “Hello” messages and to complete the setup and configuration process. After 24 hours (an OEM can change this default to up to 255 hours), the interface will close if the setup and configuration time was not extended by a network command from the SCA.

4.1 Overview of Remote Configuration Flow

4.1.1 Initial Conditions

Before Remote Configuration begins, the following initial conditions must be met:

1. The Intel AMT device is configured to receive its IP address from a DHCP server. The DHCP server must be configured to support option 15 to acquire the local domain suffix (Unsecure DNS mode) or the MEBx menu or a USB key must be used to supply the domain suffix or the FQDN of the setup and configuration application (available with Release 3.0 only).
2. The Intel AMT device is pre-programmed with at least one active root certificate hash.
3. For the delayed installation sequence described below (“delayed” meaning that the Intel AMT device was not setup immediately upon being connected to the network), an ISV-created local agent must be installed on the host platform.
4. The SCA is registered with a DNS server accessible to the Intel AMT device with the name “Provisionserver” (or the name defined by the OEM) and is in either the same domain as the device or it is in a domain with the same suffix.



5. The SCA has a server certificate, used only for setup and configuration, with the appropriate **OID** or **OU** that traces to a CA which has a root certificate hash stored in the Intel AMT device.

The **OID** in the **Extended Key Usage** field must be **2.16.840.1.113741.1.2.3** (this is the unique Intel AMT OID)

or

the **OU** value in the **Subject** field must be **"Intel(R) Client Setup Certificate"**. This OU value is case-sensitive and must be entered exactly.

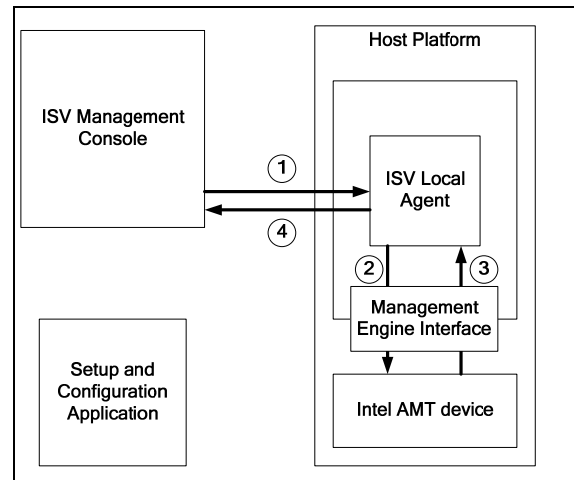
4.1.2 Acquiring a Server Certificate

Contact one of the vendors whose root certificate hashes are built into the Intel AMT firmware. A list of the hashes should be provided by the platform vendor. Go to the vendor's website site and purchase an "SSL certificate" For example, the following link to Verisign's* site <http://www.verisign.com/ssl/buy-ssl-certificates/index.html> shows how to purchase an appropriate certificate. Use the OID or the OU values above (or both) when defining the certificate.

4.1.3 Steps leading to the start of Setup and Configuration

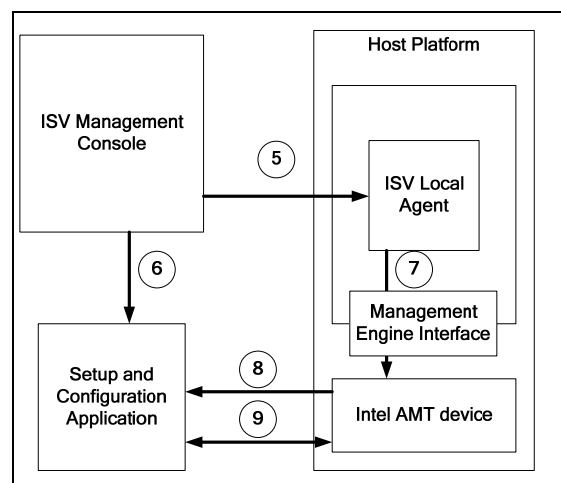
Once the above preparations are complete, the following steps are performed in support of delayed configuration. See [Bare Metal Setup and Configuration](#) for the simplified process there:

1. The Management Console requests the Local Agent to check for Intel AMT capability on the platform and to return key parameters.
2. The agent detects Intel AMT and requests the UUID and Intel AMT firmware version.
3. Intel AMT device returns the values to the agent.
4. The agent returns the information to the Management Console.





5. Management Console sends OTP to agent.
6. Management Console sends the identifying information and optionally an OTP to SCA.
7. Agent optionally sends OTP to Intel AMT device and commands it to open the network interface. The Intel AMT device generates a self-signed certificate. This process may take up to seven minutes to generate the necessary keys.
8. The Intel AMT device starts sending version 3 "Hello" messages.
9. Setup and configuration begins using the PKI-CH protocol.



4.2 The Remote Configuration "Hello" Message

The following message format is used when an Intel AMT device is configured to use Remote Configuration.

Byte Offset	Type	Content
0	Unsigned Short	Admin Credentials Set
2	Unsigned Short	Interface Version (3 for PKI-CH)
4	Unsigned Long	Retry Count (0-264)
8	Byte 16	Device UUID
24	Unsigned Char	Number of certificate hashes (maximum value is 23)
25 and on		Certificate hashes

The message continues with the certificate hashes, starting at byte 25. Each hash entry consists of a header and the hash itself. The format is:

Header: 2 bytes:

Byte 0: hash algorithm: 0 = MD5 (16 byte hash); 1 = SHA1 (20 byte hash)

Byte 1: hash length in bytes (16 or 20)

Hash: 16 or 20 bytes.

Each hash corresponds to a root certificate from a certificate authority.

4.3 Remote Configuration Setup and Configuration Process

After the local agent commands the Intel AMT device to start configuration, the device opens its network interface for 24 hours, and starts sending "Hello" messages according to the [retry algorithm](#) described above, with the following difference: The Intel AMT device will send one message per hour until the interface closes.



Note: The interface is open for 24 hours (configurable by the OEM) only the first time that it is enabled. If the time runs out before setup and configuration completes or the Intel AMT device is unprovisioned or partially unprovisioned, any subsequent calls to start configuration will open the interface for only six hours.

1. The SCA extracts the hashes from the "Hello" message.
2. The SCA sends a certificate chain that includes a root certificate matching one of the received hashes.
3. The Intel AMT device validates the SCA certificate: It checks that the OID or the OU is correct, that it is derived from a Certificate authority that matches one of the root certificate hashes and that it is a Server certificate.
4. The Intel AMT device verifies that the domain suffix matches the DNS suffix in the SCA certificate.
5. The SCA and the Intel AMT device perform a complete mutual authentication session key exchange:
 - a. The Intel AMT device uses a self-signed certificate, sending its public key.
 - b. The SCA creates a TLS session master key, encrypts it with the Intel AMT device public key, and sends it to the Intel AMT device.
 - c. The device decrypts the master key with its private key. The key is the shared secret used to establish the setup and configuration TLS session.
6. One Time Password verification: The SCA optionally requests the OTP from the Intel AMT device. The device sends the OTP securely. The SCA verifies the OTP for correctness.
7. Setup and configuration continues as described in [Setup and Configuration Application Flow Step 6](#). At some point before the SCA sends a CommitChanges command to complete the setup and configuration process, it sends a SetMEBx password command to change the password from its default, if it was not already changed.
8. Since the Intel AMT device network interface is open only for a maximum of 24 hours after sending the first "Hello" message, the SCA can command the device to reset the period to a new value of 1 to 24 hours.

4.4 Intel® AMT Release 3.0 Additional Features

4.4.1 Simplified One-Touch

Intel AMT Release 3.0 supports a one-touch configuration mechanism that avoids the possibility of a malicious user masquerading as a setup and configuration server. If an IT administrator enters the FQDN of the SCA via the MEBx menu or with a USB key (see below), then in Step 4 above, the Intel AMT device verifies that the FQDN in the SCA certificate matches the entered value. An OEM can optionally preset platforms to have an SCA FQDN. Providing an SCA FQDN in either case is more secure than depending on DHCP option 15.



4.4.2 Bare Metal Setup and Configuration

With Intel AMT Release 3.0, a platform containing Intel AMT can be configured by the manufacturer to start sending "Hello" messages as soon as the platform is connected to AC power and to the network. There may be no operating system up and running on the host, or there may be no Remote Configuration local agent, thus the name "bare metal". With no agent, there is no way to install a One Time Password.

This mode also allows entering an optional FQDN for the SCA. Either the OEM adds it before delivery or an IT administrator adds it, as described in [Simplified One-Touch](#). The Intel AMT device will acquire an IP address from a DHCP server, and then start sending "Hello" messages. There is no OTP to exchange in this case; otherwise, the setup and configuration flow is the same.

4.4.3 USB Key Support for Remote Configuration

The ME BIOS extension for Intel AMT Release 3.0 supports an added format for USB keys that aids in preparing Intel AMT platforms for remote configuration. This automates the simplified one-touch process. The record on the USB key contains the following information:

- Option to enable the Intel AMT capability on the platform if it is not already enabled
- Current and replacement MEBx password
- Optional DNS suffix or the setup and configuration application FQDN
- Option to start Remote Configuration
- Up to three certificate hashes

4.4.4 Requirements

The following items are required to be able to use a USB key for Intel AMT configuration:

- A dedicated USB key with no data on it.
- A function within a setup and configuration server that generates a type 2 file with all or a subset of the above parameters.



4.4.5 Preparation

All that is required is to execute the function, which will do the following:

1. Identify the parameters
2. Create a file named "setup.bin" in the proper format. The file will include:
 - a. A header that notes file format.
 - b. An record that includes:
 - i. A flag that enables Intel AMT
 - ii. The default MEBx password (usually "admin")
 - iii. A replacement password
 - iv. The DNS suffix or SCA FQDN
 - v. A request to start configuration
 - vi. Optional user-supplied certificate hashes (The readme usage notes that the hashes are provided as .pem files.)

Note: Only add hashes for root certificates that meet the following qualification: If the root certificate is a version V3 certificate and it includes **any** of the optional fields that are part of the V3 format, then it must include a **KeyUsage** field. If this field is absent, then Intel AMT will return an "unsupported certificate" error when a setup and configuration attempts to use a certificate based on this root certificate.

3. Format the USB key to FAT16.
4. Write the file to the USB key.

4.4.6 Initializing a Platform

To install the information from the key on an Intel AMT platform an IT technician will:

1. Take the platform out of the box and connect cables, a monitor, and a keyboard. The technician **should not** connect the platform to a network port, as a platform configured for Bare Metal configuration will start sending "Hello" messages immediately.
2. Connect the USB key to a USB port.
3. Turn on the platform.

The BIOS on the platform will detect the presence of the USB key, read the record in the file, authenticate the password, save the entered values, and update with the replacement password. A message displayed on the monitor informs the technician that the process is complete. The technician powers down the platform.

4.4.7 Moving to Setup Mode

The platform is now in Setup Mode. When it is connected to the network, Intel AMT will start to send "Hello" messages.



4.5 Remote Configuration Local Agent

The Remote Configuration process includes an ISV-developed Remote Configuration agent that runs on the host. The software development kit provides a sample agent that demonstrates how to interact with the Intel AMT device and perform the necessary steps.

A production agent starts by receiving a request from an external console to return identifying information from the Intel AMT device on the platform.

1. The agent connects to the Intel ME Interface (MEI) driver client by sending a CONNECT IOCTL command with a specific GUID.
 - a. If the connection fails, but the MEI driver is present, then the ME is not configured to support Intel AMT. The agent connects to the WD client with a different GUID.
 - b. The agent queries the WD client if it can switch to Intel AMT mode. The answer will be "yes" only if Remote Configuration will be enabled once Intel AMT is enabled.
 - c. If the answer is "yes", command via the WD client to switch to Intel AMT mode. The device will go through a restart cycle, while the agent waits for the cycle to complete. Control returns to the driver client. The agent must also restart the host so that the firmware can read the platform UUID.
2. The agent queries the Intel AMT device if it supports Remote Configuration.
3. The agent requests the Intel AMT device firmware version from the Intel AMT device.
4. The agent requests the Platform UUID with a Windows call.
5. The agent optionally retrieves the certificate hashes from the Intel AMT device.
6. The agent returns the information to the console.
7. Optionally, the console creates an OTP and sends the OTP to the agent.
8. Agent sends the OTP to Intel AMT device.
9. The agent optionally sends a domain name suffix to the device.
10. The agent requests Intel AMT device to open the network interface and start sending "Hello" messages.

The sample agent does not depend on an external console to start it. It accepts an OTP and a domain name suffix as input parameters and performs its functions without the presence of an external console send an initiating request. The sample agent requests that the Intel AMT device generate a random number generator seed. This is instantaneous with Release 3.0, but it may take as much as 30 seconds with Releases 2.2 and 2.6. Note that the sample agent uses delays to manage this interval. Refer to the sample agent code to apply the delays correctly.

Since the sample does not communicate externally, the SCA must be set up initially with the necessary parameters before the agent commands the Intel AMT device to start sending "Hello" messages.

See [Error! Reference source not found. Error! Reference source not found.](#) for a description of the functions used to implement a Remote Configuration agent. Also, see [Connecting to the ME Interface Driver](#).



The sample agent performs the following steps, using the commands listed in the appendix:

1. Connects to the Intel MEI driver.
2. Checks that the Intel MEI driver is installed.
3. If driver is present and the connection fails,
 - a. The agent connects to the WD client.
 - b. Calls the **STATE_INDEPENDENCE_COMMAND** to see if the ME can be put into Intel AMT mode.
 - c. Invokes **STATE_INDEPENDENCE_CHANGE_TO_AMT** to change to Intel AMT.
 - d. Waits for a firmware reset.
 - e. Reconnects to the MEI driver client.
4. Checks the provisioning state by calling **CFG_GetProvisioningState**.
5. Enumerates and obtains certificate hashes by calling **CFG_EnumerateHashHandles** and **CFG_GetCertificateHashEntry**.
6. Checks the provisioning mode to determine if a legacy provisioning mode is enabled by calling **CFG_GetProvisioningMode**.
7. Checks that Remote Configuration is enabled by calling **CFG_GetZTCEnabledStatus**.
8. Checks provisioning TLS mode by calling **CFG_GetProvisioningTLSMode**.
9. Generates the random number generator seed by calling **CFG_GenerateRngSeed**.
10. Waits for RNG seed generation and firmware reset to complete (Releases 2.2 and 2.6; Release 3.0 completes immediately).
11. Initializes a connection to the Intel ME Interface again.
12. Checks the status of the RNG seed generation by calling **CFG_GetRngSeedStatus**.
13. Optionally, sets the domain suffix in the ME by calling **CFG_SetPkiFQDNSuffix**.
14. Optionally, sets one-time password by calling **CFG_SetProvisioningServerOTP**.
15. Starts configuration by calling **CFG_StartConfiguration**.

4.6 Parameters in CONF.XML that support Remote Configuration

The following parameters support Remote Configuration. See [*.CONF.xml File Format](#) for details.

- **pki_configuration** A collection of parameters that support Remote Configuration
- **full_cert_chain_file** Path to a pem file containing a certificate chain that starts with the SCA certificate and includes the private key and contains the full chain of trust (including the root) in ascending order (from the leaf to the root)
- **root_cert_file** Points to a root certificate that saves the SCA from having to extract the root certificate from the certificate chain
- **otp** One-time password used to validate the value returned by the Intel AMT device
- **new_mebx_password** Replacement strong password required to enable the CommitChanges command

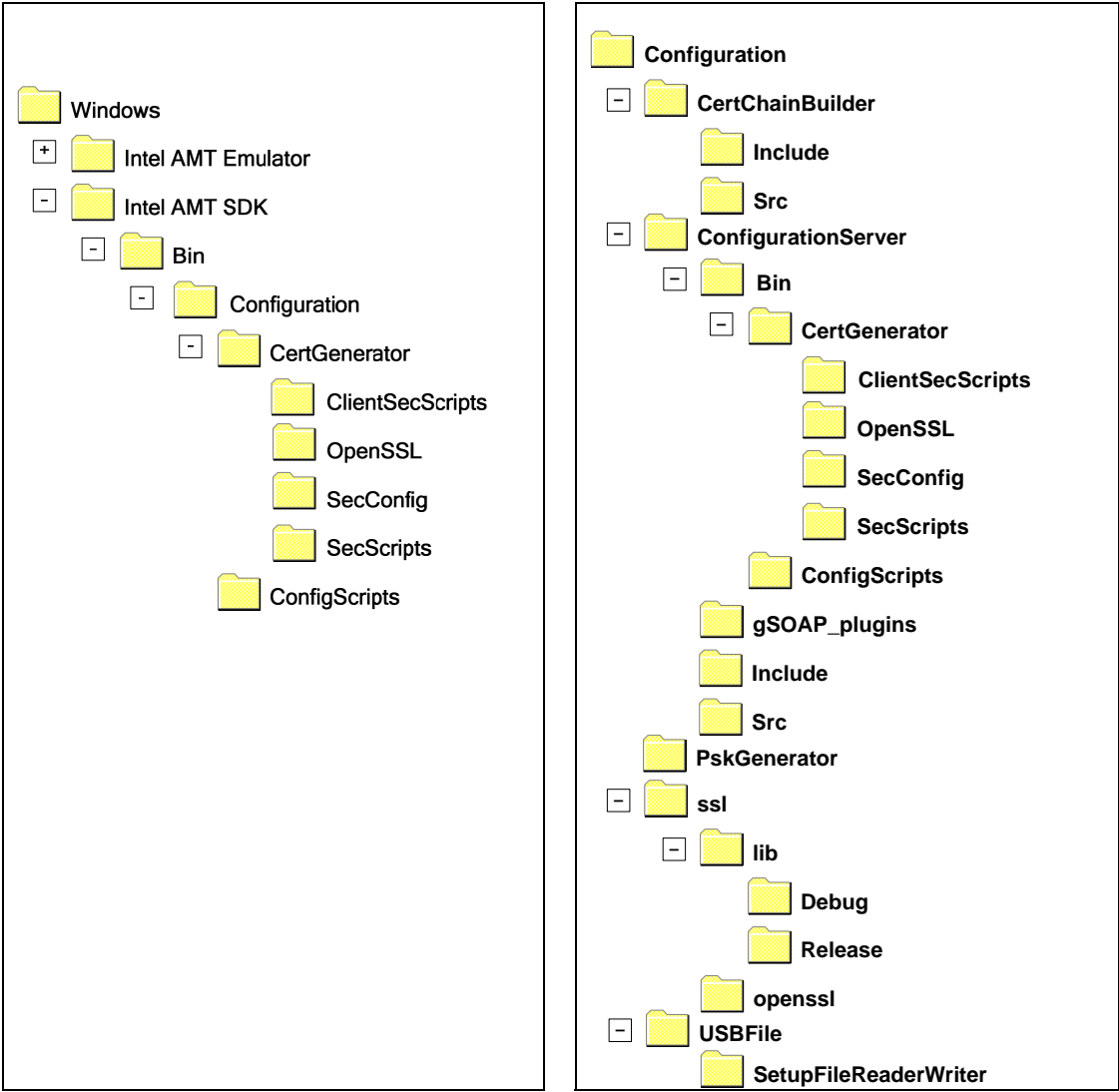
The `<new_network_password>` tag is also required for remote configuration.



5 *Installing and Running the Sample SCA*

5.1 Sample SCA Installation Folders Layout

The following figure shows the directory tree for the sample SCA installation. There are two parts: The sample source and the sample executable with its associated files.



Bin Folder

Configuration directory in Samples Folder



The following elements are included in the folders of the directory tree.

The **Configuration** subdirectory under the **Bin** directory contains the executable image of the Setup and Configuration Sample and all the necessary supporting files.

ConfigurationServer.exe – application executable

The directory also contains supporting DLLs.

CertGenerator subdirectory – contains scripts and utilities used to produce certificates.

ClientSecScripts – scripts and configuration files used to create trusted root and client certificates for use with mutual authentication

OpenSSL – Contains the **openssl** utility. Refer to Open SSL documentation for a description of these utilities

ssleay32.dll – DLL used by the openssl utility.

libeay32.dll – DLL used by the openssl utility.

yesno.exe – tool prompting for yes/no user input. Used by the configuration batch scripts.

openssl_root.cfg – is the demo root CA parameters file

openssl_sub.cfg – is the subordinate CA parameters file

SecConfig subdirectory

Uss.cfg – is the Intel AMT device certificate request parameters file

rootCA.cfg – is the demo root CA certificate request parameters file

subCA.cfg – is the subordinate CA certificate request parameters file

SecScripts subdirectory – contains various security scripts

CertChainBuilder.exe – Cert Chain Builder utility.

certgen.bat – generates RSA key and certificate for Intel AMT, used by the Configuration Server.

checkca.bat – checks if the subordinate CA is ready for use.

clean.bat – cleans all subordinate and root CA configurations.

gencertchain.bat – make a certificate chain for an Intel AMT device.

rootCA_gen.bat - generates a demo root CA certificate.

subCA_req.bat – generates a subordinate CA certificate request.

subCA_sign.bat – signs the subordinate certificate request with the demo root CA certificate.



yy.txt – text file used as input to batch scripts.

ConfigScripts subdirectory – contains scripts used to produce the Intel AMT device configuration.

getcfg.bat – retrieves a recommended configuration for the device to be configured.

provend.bat – reports back to the batch script of a successful operation, deletes device-specific configuration and security files.

create_usb_file.bat – initializes a USB storage device, creates a file of ten random PID-PPS pairs, writes them to the USB device, and optionally replaces the **psk.repository.xml** file in the same directory.

USBFile.exe – generates .bin and .XML files in a choice of two formats. The files can contain PID/PPS pairs in the proper format or they can contain the parameters required to prepare a platform for remote configuration.

PSKGenerator.exe – Sample program that generates an XML file containing PID-PPS pairs.

yesno.exe – tool prompting for yes/no user input.

default.conf.xml – default parameters used by the SCA.

psk.repository.xml – structure with PID/PPS pairs showing the format expected by the SCA.

The **Configuration** folder under **Samples** contains the source for the configuration server, as well as source for all supporting functions.

CertChainBuilder subdirectory – includes source code of the **CertChainBuilder** used by the Configuration Server to create the certificate chain file (**cchain.raw**) during the configuration process. Use this tool to support Intel AMT Releases 2.0 and 2.1. **CertChainBuilder** is deprecated for Intel AMT Releases 2.5 and greater in favor of the certificate store capability.

ConfigurationServer subdirectory contains source code of the Configuration Server application.

ConfigurationServer.vcproj – Microsoft Visual Studio .NET 2003 project file.

Include subdirectory– contains Configuration Server application header files.

Src subdirectory – contains Configuration Server application source files

gSOAP_plugins subdirectory – contains source that supports SOAP over HTTP communications.

PskGenerator subdirectory – contains sample source code for a program that generates PID/PPS pairs. The generated values have CRC digits built into them that are validated by the Intel AMT BIOS sub-menu.

Ssl subdirectory – Contains source code and compiled libraries for the secure sockets layer based on the Open SSL implementation.

USBFile subdirectory – contains sample source code for a program that generates a setup.bin file to write to a USB storage device and an XML file for the Sample SCA to use as a **PSK.REPOSITORY.XML** file.

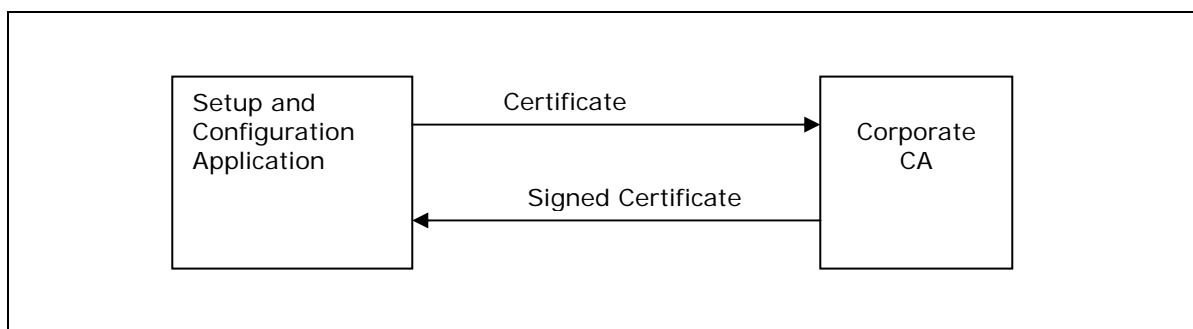


SetupFileReaderWriter subdirectory – contains sample functions that generate files of PID/PPS pairs to be written to a USB storage device.

5.2 Obtaining a Certificate for the Sample SCA

One of the roles of the SCA is to issue certificates for Intel AMT platforms. To do this, it needs to act as a subordinate certificate authority that receives an issued certificate from the enterprise root CA (or another enterprise trusted CA).

The SCA performs the following sequence of steps to obtain a signed subordinate CA certificate. This sequence is performed only once: When the SCA has a signed subordinate CA certificate there is no need to perform the steps again.



1. When the sample SCA executes for the first time it looks for a file named subcacert.pem containing a subordinate certificate. If this file is not present, the program will display a message asking the user if it should create the subordinate CA request file:

"Configuration Server can't run without a Subordinate CA configuration.

Create a subordinate CA request file [Y/n]?"

2. If "y" is selected, the sample SCA generates a certificate request and places it in the certreq.pem file under the applications current working directory, in ..\CertGenerator\secScripts\SubCa. The file is a PKCS#10 request file in BASE64 format.
3. The user is asked if a DemoRootCA should be created to sign the certificate request.

If the user answers "Y", the SCA creates a demonstration root CA that issues a X.509 certificate encoded in BASE64 format and the certificate is stored in the ConfigurationServer\Bin\CertGenerator\SecScripts\subCA directory with the file name **subcacert.pem**.

If the user answers "N", then the next three steps should be performed.



4. The enterprise root CA authority signs the request.

See [Appendix A](#) for a detailed example of how to sign the certificate request using a Windows Server2003 Certificate Authority.

5. The Enterprise CA issues a X.509 certificate encoded in BASE64 format. The user places the resulting file in the ConfigurationServer\Bin\CertGenerator\SecScripts\subCA directory with the file name **subcacert.pem**.
6. The user restarts the sample SCA.

The sample SCA can now start issuing certificates for Intel AMT devices.

See [Issuing Certificates and Certificate Authority](#) for more information about certificate operations performed by the SCA.

5.3 Issuing a Management Console (Client) Certificate

The SCA must provide root certificates to an Intel AMT device as a step in configuring mutual authentication. The device uses the root certificates so it can trust client certificates presented by management applications. A management application is any application attempting to access Intel AMT features through either the network interface or the host interface.

This section contains detailed instructions on how to create a new Intel AMT-compatible client certificate using an existing standalone root CA on Windows 2003.

1. Go to the Windows 2003 machine containing the corporate CA.
2. Open Internet Explorer on the URL <http://localhost/certsrv>
3. Select "Request a certificate".
4. Select "Advanced certificate request".
5. Select "Create and submit a request to this CA".
6. In the "Name" field, type the fully qualified name of the host (host and domain name).
7. Fill the relevant fields of the certificate request (company, department, etc.).
8. Type of Certificate Needed: choose "Other..."
9. OID: the complete OID value must be

"1.3.6.1.5.5.7.3.2,2.16.840.1.113741.1.2.1" (Remote application).or

"1.3.6.1.5.5.7.3.2,2.16.840.1.113741.1.2.2" (Local application)

The OID must be entered without spaces.

10. Set a Key Size (the valid values are 1024, 1536, or 2048).
Select "Mark keys as exportable"
11. Press the submit button.
12. Sign the certificate request.
13. Open Control Panel → Administrative Tools → Certificate Authority
14. Go to "Pending Requests" under the corporate root CA.
15. Right click the request and choose All Tasks → Issue.
16. Go to "Issued Certificates" under the enterprise corporate root CA.



17. Double click on the new certificate.
18. Choose Details → Copy to File... and save the certificate on the hard drive.
19. Double click on the saved certificate file and choose "Install certificate".
20. From the start menu choose run and enter "mmc".
21. Choose File → Add/Remove Snap-in...
22. Press Add...
23. Choose Certificates and press Add and then Finish.
24. Press Close and then OK.
25. Expand the Personal folder under "Certificates – Current User".
26. Click on Certificates.
27. Right click the new client certificate and choose "All Tasks → Export..."
28. Choose to export the private key.
29. Enter a passphrase to protect the private key in the exported file.
30. Choose a filename (.pfx) and finish exporting the file.

The output of this section is a file with a .pfx extension that contains a TLS client certificate in the PKCS#12 format, and a private key. The pkcs12 OpenSSL utility can be used to convert the .pfx file to the .pem format. This is done using the command:

```
openssl pkcs12 -in <the .pfx certificate> -out ClientCertOut.pem
```

The command prompts for the passphrase which protects the .pfx file. It then prompts for a new passphrase that will protect the .pem file.

After the sample SCA receives a certificate from the enterprise root CA, as described above, the SCA prompts the user to generate a trusted root CA certificate.

To avoid additional prompts from the SCA (for example, if the response is "no" to the previous request, the SCA continues to prompt for a "yes" answer), create the following directories:

```
...\Bin\CertGenerator\ClientSecScripts\trusted_rootCA
```

```
...\Bin\CertGenerator\ClientSecScripts\remote_client
```

```
...\Bin\CertGenerator\ClientSecScripts\local_client
```

The SCA sends the trusted root CA certificate to Intel AMT devices if the configuration file so specifies. Alternatively, an enterprise root CA authority's certificate can be used directly for this task.

The resulting file should be placed in the ...\\Bin\\CertGenerator\\ClientSecScripts\\trusted_rootCA directory. After placing the file in the named directory, the user will need to re-run the ConfigurationServer application.

A Windows Server2003 Certificate Authority can be used to issue client certificates. Intel AMT will accept such a client certificate only if the CA certificate is installed in the Intel AMT root certificates. To do this, copy the CA certificate in base64 format to the trusted_root directory and add a <file>cert_name</file> entry to .conf.xml under <trusted_root_certificates>.



5.4 Changing Certificate Properties

The OpenSSL infrastructure allows modifying most of the fields which appear in a certificate.

Note that the CN field must match the hostname.domainName of the platform where the certificate resides. If it does not match, a TLS connection may fail when using a SOAP request because of a name mismatch (a pop-up warning will appear when using a web browser and the connection will not fail automatically).

Below is a list of file names and paths that may be changed for customization.

Configuration\CertGenerator\OpenSSL\

Openssl_root.cfg – is the demo root CA parameters file

openssl_sub.cfg – is the subordinate CA parameters file

Configuration\CertGenerator\SecConfig\

Uss.cfg – is the Intel AMT device certificate parameters file

rootCA.cfg – is the demo root CA certificate request parameters file

subCA.cfg – is the subordinate CA certificate request parameters file

Please refer to www.openssl.org for additional information.

5.5 GETCFG.BAT

The Configuration Server executable calls GETCFG.BAT to select which .CONF.XML file is used to configure an Intel AMT device. When a "Hello" message is received from an Intel AMT device with a UUID [X], GETCFG.BAT will first search for a file called [X].CONF.XML. If the file is not found, it will return the default file, DEFAULT.CONF.XML.

The chosen .conf.xml file contains all the parameters that can be configured in the Setup and Configuration process. Some of the parameters specified in the CONF.XML can also be configured manually through the BIOS Extensions. If some of the parameters were configured from the BIOS Extensions screens, they can be omitted from the .CONF.XML by commenting them out. However, the host name and domain name parameters are required.

The SCA invokes the GETCFG.BAT with the following environment variables defined:

Variable name	Purpose
PROVISIONING_UUID	The UUID of the platform which is to be configured (sent in the "Hello" message from the platform)
PROVISIONING_VERSION	The version of Intel AMT on the platform being configured (for logging purposes only)



Variable name	Purpose
PROVISIONING_ORIGINATING_IP	The IP of the platform being configured (for logging purposes only)

GETCFG.BAT creates a log file named getcfg.log. It contains a record of all platforms that connected with the SCA, including their UUID and, optionally, their Intel AMT version and originating IP. The script also logs the configuration file used for each platform.

5.6 Intel® AMT Device Configuration Parameters

The following table lists the configuration parameters that can be included in a configuration file in CONF.XML format. See [*.CONF.XML file format](#) for more detailed information about each parameter.

Parameter Name	Description	Applicable Intel AMT Release
host_name	The host name of the Intel AMT device.	All
domain_name	The network domain of the Intel AMT device.	All
provisioning_mode	The setup type used. Should be set to "enterprise."	All
cfg_username	The current admin user name. Typically would be set to "admin" during setup operations. If Intel AMT is in factory mode, then the SCA will connect with the default username if cfg_username fails.	All
cfg_password	Current admin password. Must be the same as the password entered during the factory mode setup. If Intel AMT is in factory mode, then the SCA will connect with the default password if cfg_password fails.	All
tcpip_dhcp_enable	Set to "true" if using DHCP.	All
tcpip_address	IP address	All
tcpip_subnet	IP subnet mask	All
tcpip_default_gateway	IP gateway address.	All
primary_dns	Primary DNS server address.	All



Parameter Name	Description	Applicable Intel AMT Release
secondary_dns	Secondary DNS server address.	All
tls_enable	Set to "true" if using TLS	1.0 only
tls_options	Possible values are "ServerAuthentication", "MutualAuthentication", or "NoAuthentication". Can be set for local and remote interfaces.	2.0 and up
tls_cert	Defines how server certificates are obtained. Server certificates can be generated or loaded from a pre-existing file.	Up to, but not including, 2.5
cert_store	Defines one or more certificates to be added to the Intel AMT certificate store.	2.5 and up
tls_cert_name	Selects a certificate from the certificate store to use as the TLS server certificate	2.5 and up
wired_8021x_profile	Provides a set of parameters that define an 802.1x connection on the wired LAN network interface	2.5 and up
wireless_profiles	Provides one or more sets of parameters that define 802.1x connections on the wireless LAN network interface	2.5 and up
eac_settings	Used to enable the NAC feature and to define an associated certificate	2.5 and up
trusted_root_certificates	Specifies the trusted root certificate files used for mutual authentication.	2.0 and up
trusted_fqdn_cn	Sets the trusted fqdn suffix used for mutual authentication. Clients must present certificates containing this domain suffix.	2.0 and up
crls	Used to define certificate revocation lists. CRLs consist of certificate serial numbers and the URL of the issuer.	2.0 and up
new_network_username	The new admin user name for remote digest connections.	All
new_network_password	The new admin password for remote digest connections.	All
new_pid	Replacement values for the parameters	2.0 and up



Parameter Name	Description	Applicable Intel AMT Release
new_pps	used during setup and configuration	2.0 and up
set_network_time	Set to true to synchronize the Intel AMT internal clock with SCA's clock. Required for Kerberos and for TLS mutual authentication.	2.0 and up
set_enabled_interfaces	Determines whether certain interfaces are enabled after configuration completes (they are disabled by default).	2.0 and up
ping_response	If set to true, Intel AMT will respond to pings when the host OS is down.	All
kerberos	Sets Kerberos domain security information.	2.0 and up
power_options	Sets highest power state that the ME will operate at and the amount of time that the ME will be idle before it shuts down.	2.0, 2.1 and up (deprecated in 2.5 and up)
power_package	Selects one of the predefined power packages	2.5 and up
pki_configuration	defines parameters needed when setup and configuration will be done using Remote Configuration	2.2, 2.6, 3.0
extend_provisioning_period	Restarts the "provisioning period" for the number of hours set in this parameter.	2.2, 2.6, 3.0
set_8021x_active_in_S0	Enables Intel AMT 802.1x authentication in S0 power state when host authentication fails.	2.6, 3.0
PXE_8021x_timeout	Sets the amount of time that a PXE boot is allowed to complete, and Intel AMT maintains the 802.1x port authentication.	2.6

Note: There are additional parameters that must be configured for Intel AMT features to work correctly. These parameters are configured after Intel AMT is made operational and are not covered in this document. The following are examples of additional parameters:

- Access Control Lists for ISVS Storage (Vendor Name, Application Name, Enterprise Name)
- Event Filters
- PET Packet Subscribers



5.7 Required Setup Parameters

An Intel AMT device can be made operational once its TCP/IP parameters are set correctly. If TLS mode is enabled, then the device needs to have security-related parameters configured as well. All other parameters are optional and could be set at a later time.

TCP/IP Configuration – When a DHCP server assigns IP addresses, and the server is configured to update DNS entries, Intel recommends that the host name for the Intel AMT device be configured to the same hostname as the operating system hostname. Otherwise, when the host operating system transitions to a suspended or standby state, the DHCP server will change the DNS hostname to the Intel AMT hostname. This also applies when working with Active Directory.

In static mode, the IP address must be configured to a value different from the operating system IP address. It is recommended that Intel AMT and the host have different hostnames if Intel AMT is to be addressed by its name, rather than by its IP address. This is the case when TLS is enabled. The IP subnet mask must also be configured. The default gateway, DNS servers, and Domain Name are configured optionally.

The Intel AMT and host TCP/IP settings must be compatible with each other:

- If the host is configured for DHCP, then Intel AMT must also be configured for DHCP.
- If the host is configured with a static IP address, then Intel AMT must also be configured with a different static IP address.

TLS Configuration – RSA keys, certificate, and RNG seed (Intel AMT Release 2.0/2.1 only), and host name must be set.

Note: When working with localized BIOS, Domain Name, Host Name and New Administrator credentials must be set by the SCA. The Domain Name and Host Name parameters have to be set in the *.conf.xml file used by the SCA if TLS is enabled. This is true for both DHCP and Static IP modes. Even though the parameters may not be used by the Intel AMT device, they are needed for the certificate operations performed by SCA.

5.8 SCA Command Usage

Usage:

ConfigurationServer.exe <-port #listening_port>

The sample SCA can be started without any parameters. In this case the default port 9971 will be used for listening. The listening port should match the one configured on the Intel AMT device during Factory Mode setup.

5.9 Known Issues

Domain Name and Host Name

If TLS is enabled, Domain Name and Host Name must be specified in *.CONF.XML even if Intel AMT is running in DHCP mode and both values are provided by the DHCP server. The sample SCA requires these parameters for constructing certificates.

Handling Root Certificates

After a successful completion of the Setup and Configuration process, a user can access the Intel AMT device via a web browser tool. The Web UI must be enabled during setup and configuration (see the `set_enabled_interfaces` parameter) or with a network SOAP command. The user needs to install the root certificate on the web browser machine to avoid prompts for an unknown certificate issuer. The figures below show examples of these prompts. The user should also use the `hostname.domainName` in the URL instead of the IP address to avoid the pop-up warning if using TLS.

Figure 5-1. Internet Explorer* Alert:





Figure 5-2. Mozilla* Alert:



S

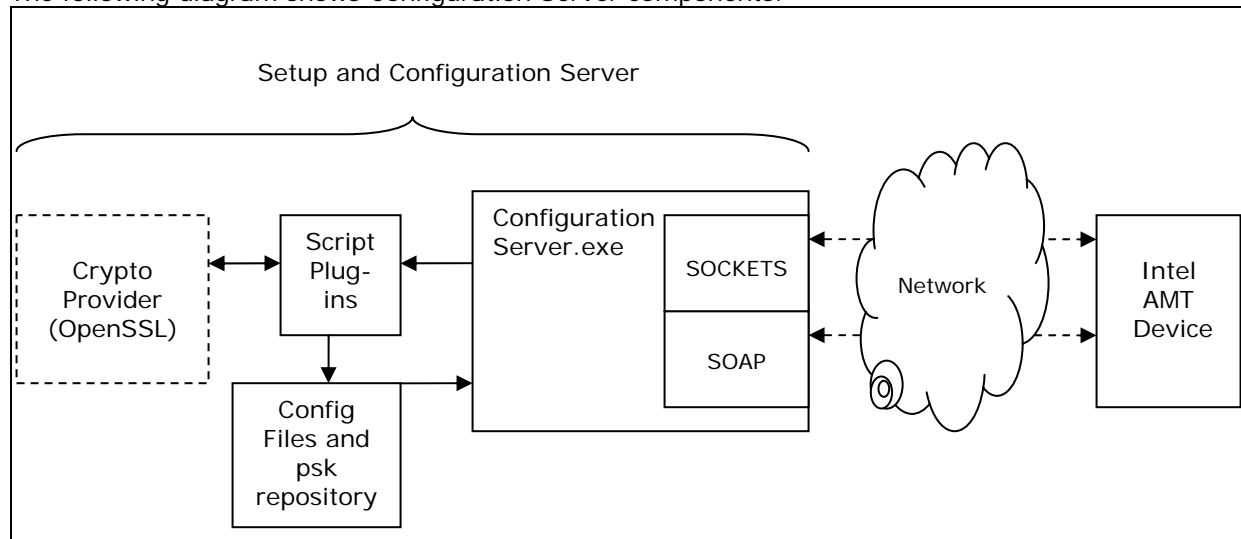




6 Configuration Server Components

The Configuration Server is a machine running the ConfigurationServer.exe application. The Configuration Server communicates with the Intel AMT device via a network interface card (NIC) attached to a network containing the Intel AMT device. If the device is an Intel AMT Release 1.0 device or a later release running in Legacy Mode, the network should be isolated to avoid potential security problems. The purpose of the Configuration Server is to complete the Setup and Configuration process by configuring the Intel AMT device using parameters customized by the Information Technology (IT) organization of the enterprise.

The following diagram shows Configuration Server components:



6.1 ConfigurationServer.exe Application

The ConfigurationServer application is the main executable that listens for incoming connections and configures the Intel AMT device based on information contained in the configuration files. These configuration files are created after related batch scripts are called. The enterprise IT administrator may modify these batch scripts to customize the configuration parameters.

6.2 The SOAP Module

This module is a part of the SCA. It uses the WSDL interface definitions which are supported by Intel AMT to configure the device.



6.3 The Socket Module

This module is a part of the SCA. It listens for incoming Intel AMT "Hello" messages on TCP port 9971 or the port value entered when starting the SCA. When an incoming request arrives, the Intel AMT Setup and Configuration application will read the request parameters (UUID, Version, count, and possibly the PID), close the connection and configure the device using the SOAP module.

6.4 Script Plug-in and Configuration Files

This section is composed of several batch scripts which are called by the SCA program. The SCA invokes the scripts, passing environment variables that describe what needs to be done. The batch scripts create output files which are read by the SCA. System administrators or services may customize these scripts to achieve custom behaviors.

6.5 Crypto Provider

This module offers certificate issuance and signing services. Other services may vary based on the provider type. The Crypto Provider functionality is invoked by batch scripts and therefore is customizable. The SCA uses OpenSSL as a reference implementation. Further information on OpenSSL can be obtained from this web site: <http://www.openssl.org>.

§



7 *Configuration Server Batch Scripts*

7.1 Certificate Management Scripts

The certificate management scripts can be found in the ConfigurationServer\Bin\CertGenerator\SecScripts subdirectory in the Configuration Server installation folder.

7.1.1 CHECKCA.BAT

This batch script checks for the presence of a subordinate Certificate Authority. If there is no subordinate CA present, the script calls SUBCA_REQ.BAT, ROOTCA_GEN.BAT, and SUBCA_SIGN.BAT.

7.1.2 ROOTCA_GEN.BAT

This script is used to create a demo root CA certificate. It uses configuration data stored in the rootCA.cfg file to set the certificate values. The script demonstrates how the certificate could be created using the crypto provider.

7.1.3 SUBCA_REQ.BAT

This script is used to create subordinate CA certificate request. It uses configuration data stored in the subCA.cfg file to set the certificate request values. The script demonstrates how the certificate request could be created using the crypto provider, creates the subCA directory and the file certreq.pem that contains the certificate that needs to be signed by the enterprise root CA.

7.1.4 SUBCA_SIGN.BAT

This script is used to sign the subordinate CA certificate request with the demo root CA. It uses configuration data stored in the openssl_root.cfg to sign the request. The script demonstrates how the certificate request could be signed by the root CA using the crypto provider, and creates the file subcacert.pem which contains the subordinate certificate signed by the demo root CA.



7.1.5 CLEAN.BAT

This batch script deletes all the certificates, keys and certificate request files previously created.

When cleaning the certificate database by activating the clean.bat script, all manually-installed root certificates should be also manually removed from the system and replaced by new ones (regenerated by the application) to get newly generated key pairs.

Deleting certificates may make Intel AMT devices unreachable. They will then require a return to factory mode and reconfiguration. This script should be used with this in mind.

Note: CLEAN.BAT is not called from inside the Configuration Server code.

7.1.6 CERTGEN.BAT

This batch script creates certificate and RSA key files for the configured Intel AMT device using the openssl tool. First it creates a request using the information stored in the uss.cfg configuration file, then the newcert.pem file is created using the configuration data stored in the openssl_sub.cfg file, and finally the certificate in the X509 format is created

The Intel AMT Configuration Server supports RSA keys of certain fixed sizes: Intel AMT Release 1.0 supports only keys of size 1536 bits. Intel AMT Releases 2.0 and greater support a key of 1024 bits, 1536 bits, or 2048 bits (the default value). The size of the key is embedded in the SCA code and is set depending on the Intel AMT release version. The key size should not be changed by the user if the Configuration Server is to parse the keys correctly.

7.1.7 GENCERTCHAIN.BAT

This batch script creates certificate and RSA keys files for the configured Intel AMT device, using the GENCERTCHAIN.BAT script file. When the certificate is ready it calls the CertChainBuilder utility to create a certificate chain data file recognized by the Intel AMT device. The chain includes the Intel AMT device certificate and the subordinate CA certificate. This file does not apply to Release 2.5 and 3.0 and subsequent releases.

7.1.8 CHECKCS.BAT

This file is located under the Bin\Configuration\CertGenerator\ClientSecScripts subdirectory.

This batch script creates a trusted root certificate, a remote client certificate and a local certificate – with both client certificates signed by the trusted root – along with corresponding RSA keys. The trusted root certificates are sent to an Intel AMT device during the setup and configuration process if mutual authentication is configured. The client certificates and private keys are exported to pkcs12 format, which is the standard way to transfer a certificate and key from one machine to another.

These files can be installed into the Microsoft certificate store on machines that need to be authenticated to Intel AMT devices that are either local or remote.



The order of events is as follows:

First the script creates a self signed certificate for the trusted root using the information stored in the trusted_rootCA.cfg configuration file, by invoking trusted_rootCA_gen.bat. Then a certificate request is made for a remote client using remote_client_req.bat. Next, the remote client certificate is signed by the trusted root using remote_client_sign.bat. Finally, the remote client certificate and private key are packaged in a pkcs12 file by calling remote_client_export.bat. The process is repeated for a local client certificate by calling local_client_req.bat, local_client_sign.bat, and local_client_export.bat. The batch files create and store the certificates and associated files in new subdirectories: trusted_rootCA, remote_client, and local_client.

These operations are carried out in separate batch files for clarity, but, unlike the server certificates created for the individual Intel AMT devices, the trusted root certificate is shared by all the devices. Large organizations might use a more elaborate PKI setup, but in general it holds that the trusted root certificate should be shared among devices managed by a particular management console application instance.

7.2 Configuration and Management Scripts

The configuration and management scripts are located in the ConfigurationServer\Bin\ConfigScripts subdirectory in the SCA installation folder.

7.2.1 GETCFG.BAT

This script identifies the file used as a configuration file for a particular Intel AMT device. See [GETCFG.BAT](#)

7.2.2 PROVEND.BAT

This script is used to perform tasks required when the SCA exits. It deletes the Intel AMT device certificate and key files created during the setup and configuration process.

7.2.3 create_usb_file.bat

This script initializes a USB storage device with a FAT file system, then calls USBFile.exe to generate a setup.bin file, which contains ten sets of four parameters: a randomly generated PID/PPS pair, a default MEBx password ("admin") and a replacement password (see the script file for the value of this password). USBFile.exe also creates setup.xml, which contains the ten PID/PPS pairs in PSK.REPOSITORY.XML format. The script writes setup.bin to the USB storage device and then asks whether the user wants to replace the existing PSK.REPOSITORY.XML with the newly created xml file.

USBFile.exe either creates a setup.bin file and a corresponding XML file or it displays the contents of a setup.bin file. See the readme in the USBFile directory for the function usage parameters.

The functions in the **SetupFileReaderWriter** subdirectory create the setup.bin file in the proper format. The file has a file header record and a series of data records. SetupFileDefinitions.h describes the file header and data record formats. The function can generate both version 1 and version 2 file formats. See the readme and the code for detailed information.



7.3 *.CONF.xml File Format

The ConfigurationServer.exe application uses the information in the *.CONF.XML selected by GETCFG.BAT to configure an Intel AMT device. The file is read using an XML interpreter. As a result, any use of special characters must be according to standard XML rules. For example, the characters "<" and "&" are invalid in an XML element. Use an entity reference instead for these characters (for example, if they are embedded in a strong password).

XML defines the following five entity references:

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

For example, `<cfg_password>Admin1'<>"&</cfg_password>`

translates to Admin1'<>"&

Below are the lists of the supported keywords which are recognized by the SCA.

Note: an unsupported keyword will be ignored (for future forward compatibility considerations).

The format of a setting in a configuration xml file is

`<command>value</command>`

Note that there must be a space before the command value.

Variable name	Allowed settings	Usage
<code><!-- ... --></code>	Any	XML comment. Used to bracket any remarks.
cfg_username	String	A username string used when logging into the Intel AMT device
cfg_password	String	A password string used when logging into the Intel AMT device
provisioning_mode	enterprise / smallbusiness	Determines the Intel AMT setup type; although an Intel AMT device must be in enterprise mode for remote configuration to start, a configuration server can configure the device to be in small business mode when the setup is complete.
host_name	String	The host name of the Intel AMT device
tcpip_dhcp_enable	true / false	Enables or disables DHCP usage on the Intel AMT device
tcpip_address	x.x.x.x	Static TCP/IP address for the Intel AMT device (used only when DHCP mode is disabled)
tcpip_subnet	x.x.x.x	TCP/IP subnet mask for the Intel AMT device (used only when DHCP mode is disabled)
tcpip_default_gateway	x.x.x.x	TCP/IP default gateway address for the Intel AMT device (used only when DHCP mode is disabled)



Variable name	Allowed settings	Usage
domain_name	String	Domain Name for the Intel AMT device (mandatory field)
primary_dns	x.x.x.x	Primary DNS address for the Intel AMT device (used only when DHCP mode is disabled)
secondary_dns	x.x.x.x	Secondary DNS address for the Intel AMT device (used only when DHCP mode is disabled)
tls_enable	true / false	Enables or disables TLS on the Intel AMT device
ping_response	true / false	Configures Intel AMT device response to ICMP Ping requests.
new_network_username	String	A username string specifying the new administrator username for Intel AMT device.
new_network_password	String	A string specifying the new administrator password for Intel AMT device.
new_pid	PID as described above	Optional replacement parameters. If a PartialUnprovision is performed, the new values will not be erased and will be available for use the next time the platform is configured.
new_pps	PPS as described above	
tls_options	<pre><tls_options> <local>ServerAuthentication </local> <remote>MutualAuthentication </remote> </tls_options></pre> Valid values are: NoAuthentication ServerAuthentication, MutualAuthentication	Determines the authentication scheme required for each interface (local and remote). NoAuthentication: TLS is not configured for the selected interface. ServerAuthentication: Intel AMT is configured with a private key and certificate. MutualAuthentication: ServerAuthentication plus at least one trusted root certificate installed.
tls_cert	<pre><tls_cert> <mode>GenerateCertificate </mode> </tls_cert></pre> <pre><tls_cert> <mode>FileCertificate </mode> <cert_chain_file>bla.raw </cert_chain_file> <key_file>bla.key</key_file> </tls_cert></pre> <pre><tls_cert> <mode>NoCertificate </mode> </tls_cert></pre>	Determines how the SCA obtains server certificates. They are either generated by the SCA ("GenerateCertificate") or loaded from pre-existing files ("FileCertificate"). When the mode is "FileCertificate", then the parameters provide the location of the certificate and key files. "NoCertificate" indicates that no certificate is required since TLS is not enabled.



Variable name	Allowed settings	Usage
cert_store	<pre> <cert_store> <mode><NoCertificate></mode> <mode><GenerateCertificate> </mode> <mode><FileCertificate> </mode> <cert> <cert_file>newcert.pem </cert_file> <key_file>newkey.pem </key_file> </cert> <cert> <cert_file>subcacert.pem </cert_file> </cert> </cert_store> </pre>	<p>This option adds certificates and keys to the certificate store in the Intel AMT device. The NoCertificate option skips sending any certificates. The GenerateCertificate option generates a certificate file and a key file and sends them to the certificate store. Any configurable certificate settings will be set to the generated certificate. The FileCertificate option loads certificate files with or without key files. Certificate files should be in the format created by certgen.bat. Key files should be in .pem format. The files are assumed to be in ...\\CertGenerator\\SecConfig</p>
tls_cert_name	<pre> <tls_cert_name>newcert.pem </tls_cert_name> </pre>	<p>When the certificate store has one or more certificates added using the FileCertificate option, this parameter selects which certificate should be used as the TLS certificate.</p>
wired_8021x_profile	<pre> <profile_type>NoProfile </profile_type> <profile_type>TLSType </profile_type> <profile_type> TTLS_MSCHAPv2Type </profile_type> <profile_type> PEAP_MSCHAPv2Type </profile_type> <profile_type>EAP_GTCType </profile_type> <profile_type> EAPFAST_MSCHAPv2Type< / profile_type> <profile_type> EAPFAST_GTCType </profile_type> </pre>	<p>Determines the 802.1x wired profile. If 802.1x is used on the wired interface, then the type of profile must be selected. See the <i>Network Interface Guide</i> and the default.conf.xml file for detailed profile parameter descriptions. The following parameters are included in one or more of the profile types:</p> <ul style="list-style-type: none"> server_id_cert_issuer server_cert_name server_cert_option username password domain_name protected_access_cred protected_access_cred_pwd client_certificate



Variable name	Allowed settings	Usage
wireless_8021x_profile	<pre> <wireless_profiles> <profile> <profile_priority>N </profile_priority> <security_type> ProfileSecuritySettingWPAType or ProfileSecuritySettingRSNType </security_type> <encryption_type> DataEncryptionTKIPType or DataEncryptionCCMPTType </encryption_type> <passphrase>ApassPhrase\$01 </passphrase> or <raw_key_file_name>xxx.key </raw_key_file_name> or <wireless_8021x_profile> an 802.1x profile as defined in the wired profiles </wireless_8021x_profile> </profile> </wireless_profiles> </pre>	One or more wireless profiles. Each profile contains a priority and a set of security settings. The settings correspond to the wireless profile definitions in the <i>Network Interface Guide</i> . Security_type is the key management scheme. encryption_type is the selected encryption mechanism, passphrase, raw_key_file_name, and wireless_8021x_profile are authentication options.
set_8021x_active_in_S0	true or false	If true, Intel AMT will attempt to perform 802.1X authentication when host 802.1X authentication fails
PXE_8021x_timeout	integer seconds	Number of seconds that Intel AMT will authenticate 802.1X while a PXE boot is in progress. 0 indicates disabling this feature.
power_package	<pre> <power_package> <guid>XXXXXXXX-XXXX-XXXX- XXXX-XXXXXXXXXXXX </guid> </power_package> </pre>	The GUID selects one of the pre-defined power packages built into an Intel AMT device. An OEM defines which of the power packages are enabled on a platform. If there is no power package entry, the device assumes a default power package. See "Power Packages" in the <i>Network Interface Guide</i> for power package details.



Variable name	Allowed settings	Usage
pki_configuration	<pre> <pki_configuration> <full_cert_chain_file> path to PEM file </full_cert_chain_file> <root_cert_file> path to root certificate pem file </root_cert_file> <otp>password</otp> <new_mebx_password> replacement mebx password </new_mebx_password > </ pki_configuration> </pre>	<p>The PKI configuration parameters are required when the Intel AMT device to be set up using Remote Configuration mode. The SCA detects the mode from the "Hello" message. In this mode, communications with the Intel AMT device use TLS mutual authentication. The device has one or more root hashes pre-installed. The SCA starts the TLS session and the Intel AMT device responds with a self-signed certificate. The SCA responds with a certificate that has a path to a root CA that matches one of the Intel AMT device's hashes. The SCS builds this certificate from the pem file pointed to by the full_cert_chain_file parameter using SSL tools. The SCA also requires a root certificate to create a root hash to compare with the entries in the "Hello" message. Providing the root_cert_file saves the SCA from having to extract the root certificate from the certificate chain.</p> <p>otp is a one-time password optionally provided by a management console both to the Intel AMT device via a local agent and to the SCA. When this parameter is included in the Conf.xml file, the SCA requests that the Intel AMT device send an otp and verifies that it matches the value in the configuration file before proceeding with setup and configuration.</p> <p>new_mebx_password is a strong password to replace the MEBx password in the Intel AMT device. The password must be changed so that the CommitChanges() command sent at the end of setup will be effective.</p>
set_network_time	true/false	Required for Kerberos and TLS mutual authentication
extend_provisioning_period	0-24	Resets the provisioning period to the number of hours selected.
trusted_root_certificates	<pre> <trusted_root_certificates> <file>trusted_cert.pem</file> </trusted_root_certificates> </pre>	<p>One or more trusted root certificate files in pem format. They must reside in ".\\Configuration\\CertGenerator\\ClientSecScripts\\trusted_rootCA"</p> <p>The supplied default configuration points to the certificate generated automatically the first time that the SCA is executed.</p>
crls	<pre> <crls> <crl> <url>---url of CDP distribution point --</url> <serials> <serial>--certificate serial number--</serial> <serial>---certificate serial number--</serial> </serials> </crl> </crls> </pre>	<p>The crls section defines a certificate revocation list (CRL). The CRL mechanism as implemented in Intel AMT does not contact a CRL distribution point. Rather, it uses the URL in a CRL entry and the certificate serial numbers to identify certificates in its certificate store that should be revoked. Intel AMT extracts the URL from the CRL distribution point (CDP) in a client certificate and matches it with the URL in the CRL, then compares certificate serial numbers.</p> <p>Each crl entry has a url and a serials section. Each serials section has one or more serial numbers.</p> <p>This is an optional entry.</p>



Variable name	Allowed settings	Usage
trusted_fqdn_cn	<pre><trusted_fqdn_cn> <fqdnsuffix>intel.com </fqdnsuffix> </trusted_fqdn_cn></pre>	A list of one or more fqdn suffixes. If a client certificate is configured, it must have its DNS name in the CN fields of the DN, and it must have one of the given fqdn suffixes as a proper suffix (preceded by a dot). For example, CN=demo.mc.intel.com matches the fqdn suffix "intel.com", but demo_mc_intel.com does not.
Kerberos	<pre><kerberos> <containerDN>CN=users,DC=cs,DC= com</containerDN> <clock_tolerance>5 </clock_tolerance> <acls> <acl> <access>local/network/any </access> <user_group_dn> CN=Domain Users,CN=users,DC=cs, DC=com </user_group_dn> <realms> <realm>n</realm> <realm>m</realm> </realms> </acl> </acls> </kerberos></pre>	This tag enables configuring Intel AMT for Kerberos authentication. The host_name parameter must be defined for Kerberos setup to complete successfully. containerDN, combined with the host_name, is used to create an Active Directory user entry. The clock tolerance, which is measured in minutes, is used by Intel AMT in conjunction with the replay cache to prevent replay attacks. The acls tag is used to define one or more access control list entries. Each entry has a pointer to a valid user or group object in a reachable Active Directory domain, a tag showing if the user or group can access Intel AMT remotely, locally or both, and a list of realms (see default.conf.xml for a list of realms and their corresponding numbers).
set_enabled_interfaces	<pre><set_enabled_interfaces> <interface>WebUI</interface> <interface>SerialOverLAN </interface> <interface>IdeRedirection </interface> </set_enabled_interfaces></pre>	The redirection interface and Web user interface are disabled by default. The set_enabled_interfaces option is used to enable one or more of these interfaces. The possible options are: WebUI SerialOverLAN IdeRedirection.
Power options	<pre><power_options> <power_state>S-state </power_state> <wake_on__net_access_ sleep_timer>t </wake_on__net_access_ sleep_timer> </power_options></pre>	The power state S-state defines the highest power state at which Intel AMT will operate while the device is connected to AC power. This includes operation in higher power states. For example, if the platform is in S3 and this parameter is set to S0, Intel AMT will not operate. Note that Intel AMT treats S0 and S1 as equivalent states. The S-state can be S0 through S5. t is an integer that determines the minimum time (in minutes) that the Intel AMT device will remain operable when there is no activity.



7.4 PSK.REPOSITORY.XML File Format

The file is located under the Bin\Configuration\ConfigScripts subdirectory in the SCA installation folder. It is the structure that the SCA expects when searching for a PID/PPS pair.

PskGenerator.exe is used to generate PID-PPS key pairs in PSK.REPOSITORY.XML format, which are used to establish secure connections to Intel AMT devices when delivering configuration settings over the network.

CreateUSBFile.bat also generates a PSK file in PSK.REPOSITORY.XML format and stores it in the appropriate directory.

Platform OEMs may pre-configure Intel AMT devices with PID/PPS pairs. The repository will be based on a file delivered by the OEM.

Variable name	Allowed settings	Usage
<pairs>	<pair> <pid>xxxx-xxxx</pid> <pps>xxxx-xxxx-xxxx-xxxx- xxxx-xxxx-xxxx-xxxx</pps> </pair>	Used to define a PID-PPS key pair. This same PID-PPS should be used during the Factory Mode setup of each Intel AMT device

§



8 *Issuing Certificates and Certificate Authority*

8.1 CA Trust Relations

A Certificate Authority (CA) is an entity used by an enterprise IT to issue certificates for network nodes or individuals. The IT administrator often installs at least one trusted enterprise root CA certificate in every enterprise machine. Any certificate that has a trust chain that ends in the enterprise root CA certificate is assumed to belong to the organization and is therefore a **trusted certificate**.

A Subordinate CA is a certificate authority which is often used to issue certificates for network nodes or other purposes, but it is trusted only after a trusted enterprise root CA has issued a certificate for it.

For the Setup and Configuration process, there are several possible usage models for certificate entrustment:

Option 1:

The sample SCA creates/receives an RSA key-pair and certificate that are signed by the enterprise CA. The SCA creates an RSA key-pair and certificate request for each Intel AMT device it configures. The SCA will then sign this request and create the following trust chain "Intel AMT device certificate \leftarrow SCA certificate \leftarrow Corporate CA". As the enterprise CA certificate is installed on each machine, no further actions need to take place for computers in the enterprise to trust the configured Intel AMT device

Option 2:

The sample SCA creates an RSA key-pair and certificate request, and self-signs this request. The sample SCA creates an RSA key-pair and certificate request for each Intel AMT device it configures. The sample SCA will then sign this request and create the following trust chain "Intel AMT device certificate \leftarrow SCA certificate". This means that the sample SCA certificate will have to be installed on each machine that wishes to open TLS connections with an Intel AMT device: Since the chain of trust has to end in a trusted entity, in this case the SCA is the trusted entity.



Option 3:

The sample SCA creates a self-signed RSA key-pair and certificate for each Intel AMT device it configures. In this method there is no trust chain as the certificate of the Intel AMT device is self-signed. With this approach, each computer that wishes to open TLS connections with a specific Intel AMT device will have to install the certificate of that device. For example, an application that wishes to work securely with three different Intel AMT devices will have to have the three self-signed certificates the devices installed on the computer where the application runs.

The sample SCA is designed by default to act as a subordinate CA (Option 1). This is the common usage model which is assumed to be deployed in enterprise environments; however the default may be customized: IT departments may modify the supported scripts to follow deployment options 2 and 3.

8.2 Certificate Enrollment

A System Administrator has to issue a certificate request according to organizational procedures to manually issue a certificate for the SCA. A certificate request is created when the sample SCA is run for the first time and is placed in the `certreq.pem` file in the generated directory (`..\Bin\Configuration\SecScripts\subCA.`) This certificate request must be signed by the enterprise CA and the resulting signed certificate must be placed in a file called `subcacert.pem` in the same directory. Once this process is completed, the Configuration Server can issue certificates to Intel AMT devices.

8.3 Certificate and Key Format

The Configuration Server parses BASE64 X509 certificates. Every certificate which is issued externally must be converted to that format for the sample SCA to process it.

For more information, please refer to the Network Interface Guide => Data structures => Security Administration Service => CertificateType data structure.

8.4 VeriSign Certificate Chain Format

The VeriSign certificate chain is reversed in PEM file used by the SCA. Detailed description and workaround is outlined below.

Description:

In order to supply the certificate to the configuration server, the certificate is exported from Microsoft Certificate store to a .PFX file – this file is password-protected. Intel uses OpenSSL to convert the .PFX file containing the certificate into .PEM format which the SCA expects. When converting the VeriSign certificate, the resulting chain is not in the order which the Intel® AMT FW expects (the chain contains Client-Root-Intermediate instead of Client-Intermediate-Root).



Workaround:

In order to be able to configure a vPro system with the SCA, the following steps should be followed:

- 1. Convert the .PFX file to .PEM file using OpenSSL:

```
Openssl pkcs12 -in pfx_file -out pem_file -nodes
```

Note: You will be requested to supply a password for the .PFX file – the password which was used to export the certificate from the Microsoft's Certificate store to the .PFX file

- 2. Open the .PEM file in a text editor
- 3. Each certificate in the chain has "Bag Attributes" that contain (at least) the issuer of the certificate and the subject to whom it was issued.
- 4. For the VeriSign certificate, reverse the chain so that the root is located at the end – note that the root is the certificate where the subject is equal to the issuer ("A" in the example below).

Example:

Before	After
<div>Bag Attributes</div> <div>Some data on the key and the Crypto provider</div> <div>Key Attributes</div> <div>More data on the Key</div> <div>-----BEGIN RSA PRIVATE KEY-----</div> <div>Ffgjlkfjg.....</div> <div>-----END RSA PRIVATE KEY-----</div> <div>Bag Attributes</div> <div>Data on C</div> <div>Subject = C</div> <div>Issuer = B</div>	<div>Bag Attributes</div> <div>Some data on the key and the Crypto provider</div> <div>Key Attributes</div> <div>More data on the Key</div> <div>-----BEGIN RSA PRIVATE KEY-----</div> <div>Ffgjlkfjg.....</div> <div>-----END RSA PRIVATE KEY-----</div> <div>Bag Attributes</div> <div>Data on C</div> <div>Subject = C</div> <div>Issuer = B</div> <div>-----BEGIN CERTIFICATE-----</div>



Before	After
<p>-----BEGIN CERTIFICATE-----</p> <p>...</p> <p>-----END CERTIFICATE-----</p> <p>Bag Attributes</p> <p><i>Data on A</i></p> <p>Subject = A</p> <p>Issuer = A</p> <p>-----BEGIN CERTIFICATE-----</p> <p>...</p> <p>-----END CERTIFICATE-----</p> <p>Bag Attributes</p> <p><i>Data on B</i></p> <p>Subject = B</p> <p>Issuer = A</p> <p>-----BEGIN CERTIFICATE-----</p> <p>...</p> <p>-----END CERTIFICATE-----</p>	<p>...</p> <p>-----END CERTIFICATE-----</p> <p>Bag Attributes</p> <p><i>Data on B</i></p> <p>Subject = B</p> <p>Issuer = A</p> <p>-----BEGIN CERTIFICATE-----</p> <p>...</p> <p>-----END CERTIFICATE-----</p> <p>Bag Attributes</p> <p><i>Data on A</i></p> <p>Subject = A</p> <p>Issuer = A</p> <p>-----BEGIN CERTIFICATE-----</p> <p>...</p> <p>-----END CERTIFICATE-----</p>



Appendix A : Using an Enterprise CA to Sign the Sample SCA Certificate

The following procedure demonstrates how to sign the Sample SCA subordinate CA certificate using the Microsoft Certificate Authority in Windows 2003.

Create a Certificate Authority

On a processor running Windows 2003:

1. Enter "Start → Settings → Control Panel → Add or Remove Programs".
2. Choose "Add/Remove Windows Components".
3. Mark "Certificate Services" check box and choose "Next".
4. Choose "Stand-alone root CA", then choose "Next".
5. Choose a name for the CA and choose "Next".
6. Choose location of the folder and files for the CA and choose "Next".

Launch Certificate Authority Services

In Windows 2003 computer.

Choose "Start → Programs → Administrative Tools → Certification Authority".

Create subordinate certificate request for the sample SCA

1. Run ConfigurationServer.exe for the first time.
2. Reply yes ("y") to the question "Create a subordinate CA request file?".

The directory \Bin\Configuration\CertGenerator\SecScripts\subCA will be created, and a request file "certreq.pem" will be placed in it.

3. Answer no ("n") to the question "Create a demo root CA and sign the request?".
4. Transfer the file "certreq.pem" to Windows 2003.

Sign the certificate request



In Windows 2003 computer.

1. In Certificate Authority service, R-click the root CA and choose "All Tasks → Submit new request".
2. Select the "certreq.pem" file.
3. Open the "Pending Requests" folder under the root CA.
4. R-click on the submitted request and choose "All Tasks → Issue".
5. Open the "Issued Certificates" folder under the root CA.
6. R-click on the certificate and choose "Open".
7. Enter the "Details" tab and choose "Copy to File".
8. Select the "Base-64 encoded X.509" option and choose "Next".
9. Enter "subcacert.pem" as the filename and choose "Next".

A file by the name "subcacert.pem.cer" will be created.

10. Rename the file to "subcacert.pem".
11. Transfer the file "subcacert.pem" to the computer running the SCA to the \Bin\Configuration\CertGenerator\SecScripts\subCA directory.
12. Restart the Configuration Server.

The SCA will now use the certificate signed by the Windows Server2003 root CA.

§