



# **Intel® VPro™ Technology Platform**

## ***Setup and Configuration Application User Guide***

---

***August, 2006***

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. Intel products are not intended for use in medical, life saving, or life sustaining applications.

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 Intel® Active Management Technology 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.

I<sup>2</sup>C is a two-wire communications bus/protocol developed by Philips. SMBus is a subset of the I<sup>2</sup>C bus/protocol and was developed by Intel. Implementations of the I<sup>2</sup>C bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Alert on LAN is a result of the Intel-IBM Advanced Manageability Alliance and a trademark of IBM

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 200x, Intel Corporation

# Contents

---

1	Introduction.....	6
1.1	What is Setup and Configuration? .....	6
1.2	Setup Types .....	6
1.3	Secure Communications and Authentication Options .....	6
2	Intel AMT Enterprise Setup and Configuration Flow .....	7
2.1	Setup and Configuration Network Layout .....	7
2.1.1	Intel AMT Setup and Configuration Application (SCA) .....	8
2.1.2	Intel AMT Machine .....	8
2.1.3	DHCP Server .....	8
2.1.4	DNS Server .....	8
2.2	Configuring the SCA .....	9
2.3	Factory Mode Setup .....	9
2.3.1	Host Name .....	10
2.3.2	TCP/IP Settings .....	10
2.3.3	SCA Server Address ("Provisioning Server") .....	11
2.3.4	Setup Type ("Provision Model") .....	11
2.3.5	Virtual Local Area Network (VLAN) Settings .....	11
2.3.6	PID-PPS .....	11
2.3.7	Other Settings .....	12
2.3.8	Exit Intel AMT Configuration .....	12
2.4	Using a USB Storage Device for Factory Mode Setup .....	12
2.4.1	Requirements .....	13
2.4.2	Preparation .....	13
2.4.3	Initializing a Platform .....	13
2.4.4	Moving to Setup Mode .....	13
2.5	Preparing Intel AMT for Future Configuration .....	14
2.6	Setup and Configuration Application Flow .....	14
3	Restoring Intel AMT to Factory Mode .....	16
4	Installing and Running the Sample SCA .....	16
4.1	Sample SCA Installation Folders Layout .....	16
4.2	Obtaining a Certificate for the Sample SCA .....	17
4.3	Issuing a Management Console (Client) Certificate .....	18
4.4	Changing Certificate Properties .....	19
4.5	GETCFG.BAT .....	20
4.6	Intel AMT Device Configuration Parameters .....	20
4.7	Required Setup Parameters .....	21
4.8	SCA Command Usage .....	22
4.9	Unprovisioning Usage .....	22
4.10	Known Issues .....	22
5	Configuration Server Components .....	23
5.1	ConfigurationServer.exe Application .....	24
5.2	Unprovision.exe Application .....	24
6	Configuration Server Batch Scripts .....	24
6.1	Certificate Management Scripts .....	24
6.1.1	CHECKCA.BAT .....	24
6.1.2	ROOTCA_GEN.BAT .....	24
6.1.3	SUBCA_REQ.BAT .....	24
6.1.4	SUBCA_SIGN.BAT .....	25
6.1.5	CLEAN.BAT .....	25
6.1.6	CERTGEN.BAT .....	25
6.1.7	GENCERTCHAIN.BAT .....	25
6.1.8	CHECKCS.BAT .....	25
6.2	Configuration and Management Scripts .....	26
6.2.1	GETCFG.BAT .....	26
6.2.2	PROVEND.BAT .....	26

6.2.3	create_usb_file.bat.....	26
6.3	*.CONF.xml File Format.....	26
6.4	PSK.REPOSITORY.XML File Format.....	29
7	Issuing Certificates and Certificate Authority .....	30
7.1	CA Trust Relations .....	30
7.2	Certificate Enrollment.....	31
7.3	Certificate and Key Format .....	31
7.4	Certificate Chain Format .....	31

## ***Revision History***

---

<b>Document Number</b>	<b>Revision Number</b>	<b>Description</b>	<b>Revision Date</b>
<XXXX>	0.852	2.1 Release	08/11/2006

# 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.

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.

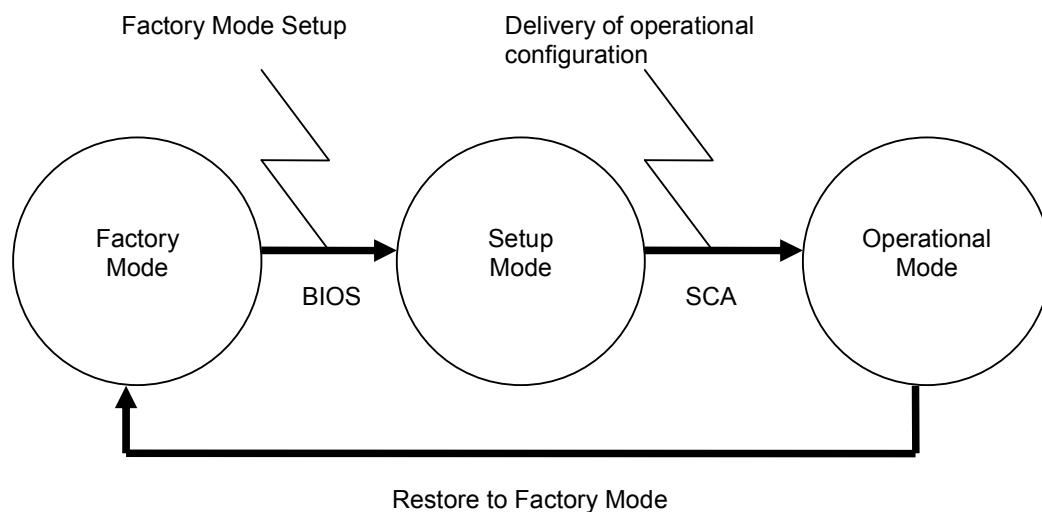
## 1.3 Secure Communications and Authentication Options

Intel AMT supports Transport Layer Security (TLS), and, 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 Intel AMT Release 2.0 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 Release 2.0 device is 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 an Intel AMT Release 2.0 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 Intel AMT. 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.

## 2 Intel AMT Enterprise Setup and Configuration Flow

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. Before an Intel AMT device can receive its configuration setting over the network, it first must be prepared with some initial setup information. 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

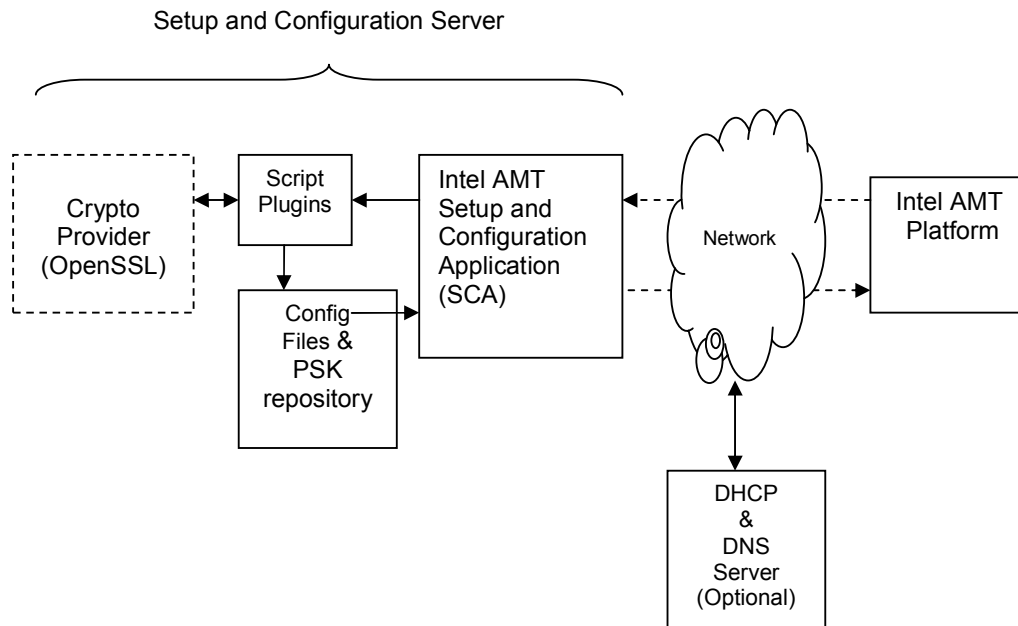
When an Intel AMT device enters Setup Mode it waits for delivery of its configuration settings from an SCS. After it enters setup mode, the Intel AMT device periodically sends messages to the SCS. When the SCS receives messages from the Intel AMT device, it responds by delivering the configuration settings 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.

## 2.1 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:



### 2.1.1 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.

### 2.1.2 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.

### 2.1.3 DHCP Server

Intel AMT devices by default automatically 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.

### 2.1.4 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.



## 2.2 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 in Intel AMT Release 2.0 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 for Intel AMT Release 2.0 platforms, and setup of the .CONF.XML file(s).

The first time the sample SCA is started the user is prompted to supply the SCA with a subordinate certificate. 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 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.*

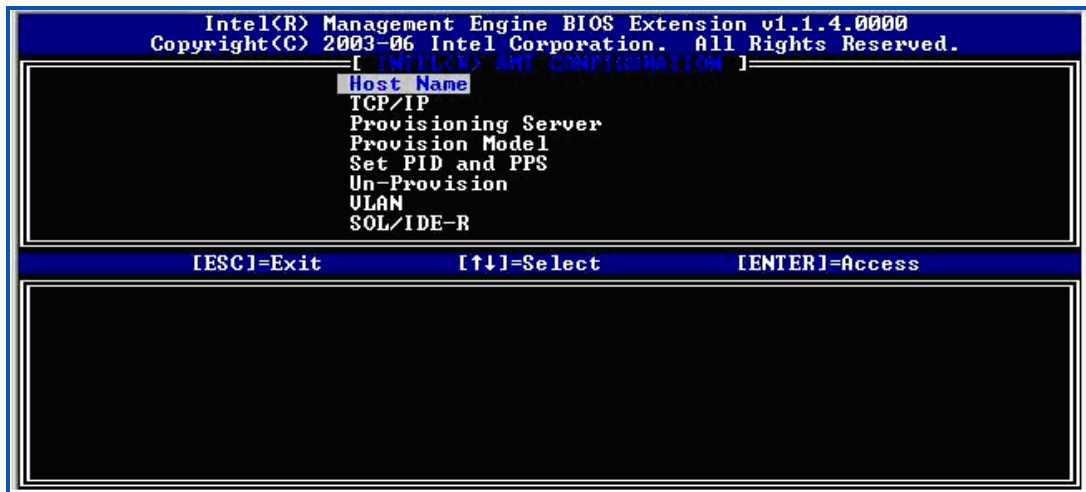
## 2.3 Factory Mode Setup

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. 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. Select the following power control settings
  - a. Intel(R) ME State upon Initial Power-On = ON
  - b. Intel(R) ME ON in Host Sleep States = Always
  - c. Intel(R) ME Visual LED Indicator = ON
9. Return to the previous menu.
10. Exit all menus. The computer will restart. Press <Ctrl+P> and enter the Main Menu.
11. Now select "Intel(R) AMT Configuration" and press "Enter". The Intel AMT Release 2.0 BIOS extension screen will be displayed as shown below:



### 2.3.1 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.

### 2.3.2 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). 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)
<b>*Note*</b> -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.

### 2.3.3 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.

### 2.3.4 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, Intel AMT Release 2.0 has the capabilities of Intel AMT Release 1.0. This allows use of ISV products developed to run with Intel AMT Release 1.0.

### 2.3.5 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.

### 2.3.6 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 Release 2.0 platforms. These settings are not available for Intel AMT Release 1.0 platforms and for Intel AMT Release 2.0 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 sample values above were created with PskGenerator and have the correct checksums.

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 Release 2.0 installations, 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.

### 2.3.7 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.

### 2.3.8 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](#).

## 2.4 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.

## 2.4.1 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. (The sample includes a function and a supporting script. The function is USBFile.exe and the script is create\_usb\_file.bat)
- Good security procedures for controlling the USB key.

## 2.4.2 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 (see the *Intel® Management Engine USB Local Provisioning Architecture* document or the sample program header files for the exact 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:
    - i. The PID-PPS pair
    - ii. The default MEBx password (usually “admin”)
    - iii. 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.

## 2.4.3 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.

## 2.4.4 Moving to Setup Mode

The platform may now be ready for moving to Setup mode, if the default parameters are appropriate for the specific enterprise. The critical 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)

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.

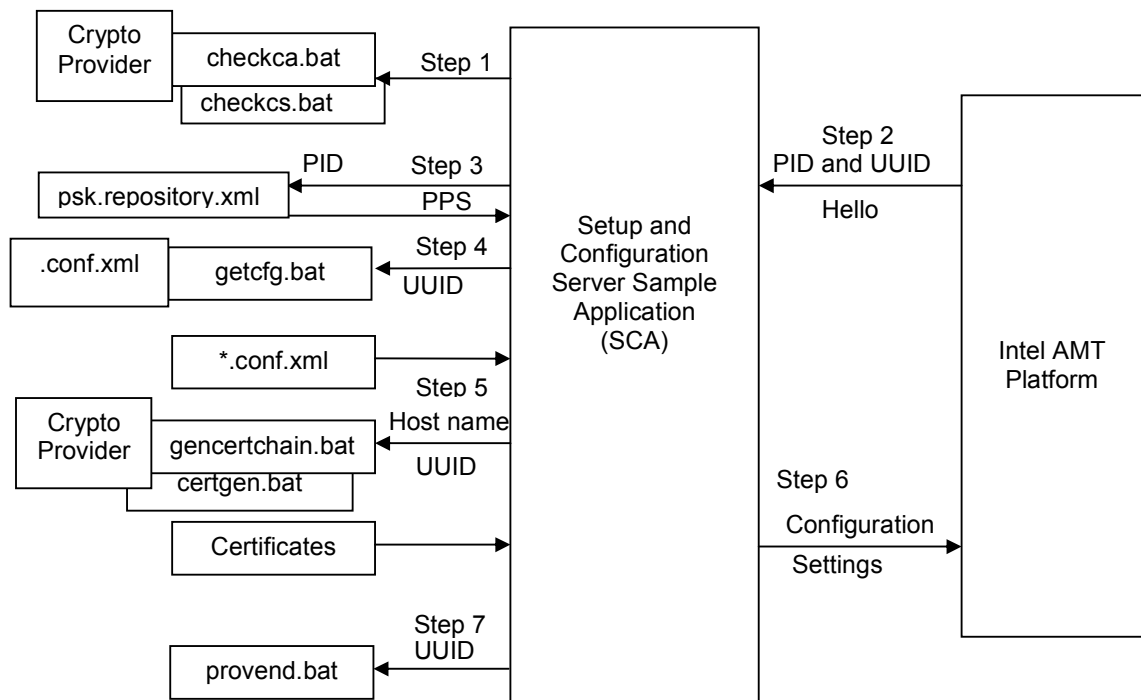
## 2.5 Preparing Intel AMT for Future Configuration

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 2.3, 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.

## 2.6 Setup and Configuration Application Flow

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 an 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), then Intel AMT 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. The three required certificates are a trusted root certificate, which is used to sign local\_client and remote\_client certificates.

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 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 Intel AMT Release 2.0) 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 (Intel AMT Release 2.0 only) 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.



## 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.

A utility for sending the Unprovision command remotely is supplied (unprovision.exe). This utility supports full & partial unprovisioning.

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 two items:

- The PID/PPS pair is not erased.
- The Admin Access Control List, containing the administrator username and password, is not erased.

**\*Note\*** - 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.

## 4 Installing and Running the Sample SCA

### 4.1 Sample SCA Installation Folders Layout

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

The **Configuration** subdirectory 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 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.  
**getcfig.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 of PID/PPS pairs in the proper format.  
**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.  
**Unprovision** subdirectory – contains application for unprovisioning (**unprovision.exe**) and a supporting library files (**StatusStrings.dll**).

## 4.2 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 subcert.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 demo root CA issues a X.509 certificate encoded in BASE64 format and the certificate is stored in the Configuration\CertGenerator\SecScripts\subCA directory with the file name **subcacert.pem**.

If the user answers "N", then the next two steps are 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 Configuration\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.

There is an SCA option to create a demonstration root CA certificate and use it to sign the subordinate CA request, instead of a real enterprise CA authority. This option exists for demonstration purposes only.

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

### 4.3 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 root CA on Windows 2003 (e.g. corporate CA).

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. 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 configuration\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>.

## 4.4 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](http://www.openssl.org) for additional information.

## 4.5 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.

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)
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.

## 4.6 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
host_name	The host name of the Intel AMT device.
Domain_name	The network domain of the Intel AMT device.
Provisioning_mode	The setup type used. Should be set to "enterprise."
Cfg_username	The current admin user name. Typically would be set to "admin" during setup operations.
Cfg_password	Current admin password. Must be the same as the password entered during the factory mode setup.
Tcpip_dhcp_enable	Set to "true" if using DHCP.
Tcpip_address	IP address
tcpip_subnet	IP subnet mask
tcpip_default_gateway	IP gateway address.

Parameter Name	Description
Primary_dns	Primary DNS server address.
Secondary_dns	Secondary DNS server address.
Tls_enable	Set to "true" if using TLS (Intel AMT Release 1.0 only).
Tls_options	Possible values are "ServerAuthentication", "MutualAuthentication", or "NoAuthentication". Can be set for local and remote interfaces (Intel AMT Release 2.0 only).
Tls_cert	Defines how server certificates are obtained. Server certificates can be generated or loaded from a pre-existing file.
Trusted_root_certificates	Specifies the trusted root certificate files used for mutual authentication.
Trusted_fqdn_cn	Sets the trusted fqdn suffix used for mutual authentication. Clients must present certificates containing this domain suffix.
Crls	Used to define certificate revocation lists. CRLs consist of certificate serial numbers and the URL of the issuer.
New_network_username	The new admin user name for remote digest connections.
New_network_password	The new admin password for remote digest connections.
New_pid	Replacement values for the parameters used during setup and configuration
new_pps	
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.
set_enabled_interfaces	Determines whether certain interfaces are enabled after configuration completes (they are disabled by default).
ping_response	If set to true, Intel AMT will respond to pings when the host OS is down.
Kerberos	Sets Kerberos domain security information.
power_options	Sets the power options

**\*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

## 4.7 Required Setup Parameters

An Intel AMT device can be made operational once its TCP/IP parameters 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 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, 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.

## 4.8 SCA Command Usage

Usage:

```
ConfigurationServer.exe <-port #listening_port>
```

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

## 4.9 Unprovisioning Usage

Usage:

```
Unprovision.exe <opt> [-verbose] [-user <username> -pass <password>] [-certName <name>] http[s]://<Hostname>:<Port>/<SecurityAdministrationUri>
```

Where <opt> is:

- fe: Full unprovisioning and set enterprise mode
- fs: Full unprovisioning and set small business mode
- p: Partial unprovisioning

Examples:

```
Unprovision.exe -fe http://hostname:16992/SecurityAdministrationService
```

```
Unprovision.exe -fe -certName MyCert https://hostname:16993/SecurityAdministrationService
```

## 4.10 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 browser 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 this prompts. The user should also use the hostname.domainName in the URL instead of the IP address to avoid the pop-up warning.



### Internet Explorer Alert:



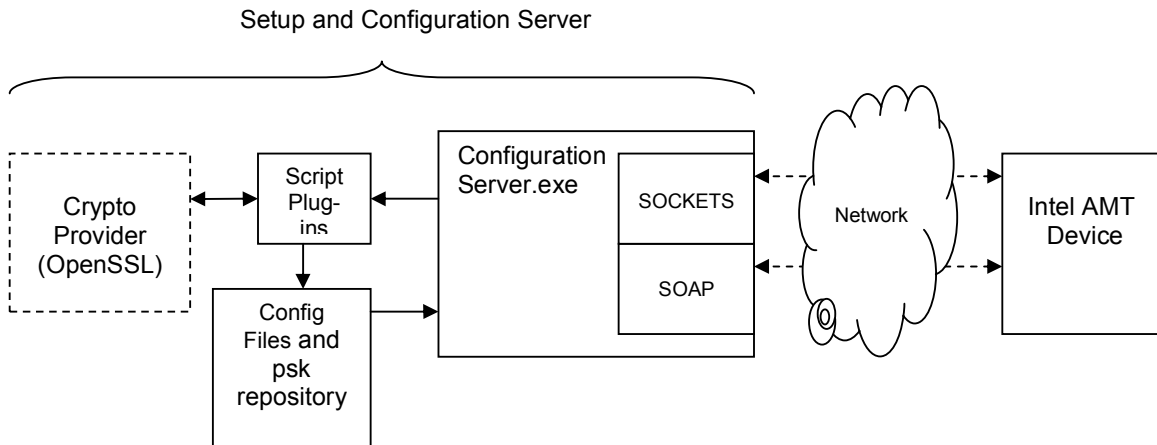
### Mozilla Alert:



## 5 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 an Intel AMT Release 2.0 device 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:



## 5.1 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.

## 5.2 Unprovision.exe Application

The unprovision application is a utility to restore the MEBX parameters to their default values on a previously provisioned Intel AMT system.

# 6 Configuration Server Batch Scripts

## 6.1 Certificate Management Scripts

The certificate management scripts can be found in the "Configuration\CertGenerator\SecScripts" subdirectory.

### 6.1.1 CHECKCA.BAT

This batch script checks for the presence of a subordinate Certificate Authority. If there is no subordinate CA present, the script calls ROOTCA\_GEN.BAT, SUBCA\_REQ.BAT, and SUBCA\_SIGN.BAT.

### 6.1.2 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.

### 6.1.3 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.



#### 6.1.4 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.

#### 6.1.5 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.*

#### 6.1.6 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 Release 2.0 supports 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.

#### 6.1.7 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.

#### 6.1.8 CHECKCS.BAT

This file is located under the 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.

## 6.2 Configuration and Management Scripts

The configuration and management scripts are located in the Configuration\ConfigScripts subdirectory.

### 6.2.1 GETCFG.BAT

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

### 6.2.2 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.

### 6.2.3 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.

## 6.3 \*.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.

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\*** - There must be a space before the command value.

Variable name	Allowed settings	Usage
<!-- ... -->	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

Variable name	Allowed settings	Usage
		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)
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 (Intel AMT Release 1.0)	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 username 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>&lt;tls_options&gt; &lt;local&gt;ServerAuthentication &lt;/local&gt; &lt;remote&gt;MutualAuthentication &lt;/remote&gt; &lt;/tls_options&gt;</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>&lt;tls_cert&gt; &lt;mode&gt;GenerateCertificate &lt;/mode&gt; &lt;/tls_cert&gt;  &lt;tls_cert&gt; &lt;mode&gt;FileCertificate &lt;/mode&gt; &lt;cert_chain_file&gt;bla.raw &lt;/cert_chain_file&gt; &lt;key_file&gt;bla.key&lt;/key_file&gt; &lt;/tls_cert&gt;</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
	<pre>&lt;tls_cert&gt;   &lt;mode&gt;NoCertificate &lt;/mode&gt; &lt;/tls_cert&gt;</pre>	
set_network_time	true/false	Required for Kerberos and TLS mutual authentication
trusted_root_certificates	<pre>&lt;trusted_root_certificates&gt;   &lt;file&gt;trusted_cert.pem&lt;/file&gt; &lt;/trusted_root_certificates&gt;</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>&lt;crls&gt;   &lt;crl&gt;     &lt;url&gt;---url of CDP distribution     point --&lt;/url&gt;     &lt;serials&gt;       &lt;serial&gt;--certificate serial       number--&lt;/serial&gt;       &lt;serial&gt;--certificate serial       number--&lt;/serial&gt;     &lt;/serials&gt;   &lt;/crl&gt; &lt;/crls&gt;</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>
trusted_fqdn_cn	<pre>&lt;trusted_fqdn_cn&gt;   &lt;fqdnsuffix&gt;intel.com &lt;/fqdnsuffix&gt; &lt;/trusted_fqdn_cn&gt;</pre>	<p>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.</p>
Kerberos	<pre>&lt;kerberos&gt;   &lt;containerDN&gt;CN=users,DC=cs   ,DC=com&lt;/containerDN&gt;   &lt;clock_tolerance&gt;5 &lt;/clock_tolerance&gt;   &lt;acls&gt;     &lt;acl&gt;       &lt;access&gt;local/network/any       &lt;/access&gt;       &lt;user_group_dn&gt;       CN=Domain       Users,CN=users,DC=cs,       DC=com       &lt;/user_group_dn&gt;     &lt;/acl&gt;     &lt;realms&gt;       &lt;realm&gt;n&lt;/realm&gt;       &lt;realm&gt;m&lt;/realm&gt;     &lt;/realms&gt;   &lt;/acl&gt; &lt;/acls&gt; &lt;/kerberos&gt;</pre>	<p>This tag enables configuring Intel AMT for Kerberos authentication. The host_name parameter must be defined for Kerberos setup to complete successfully.</p> <p>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.</p> <p>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).</p>
set_enabled_interfaces	<pre>&lt;set_enabled_interfaces&gt;   &lt;interface&gt;WebUI&lt;/interface&gt;</pre>	The redirection interface and Web user interface are disabled by default. The

Variable name	Allowed settings	Usage
	<pre> &lt;interface&gt;SerialOverLAN &lt;/interface&gt; &lt;interface&gt;IdeRedirection &lt;/interface&gt; &lt;/set_enabled_interfaces&gt; </pre>	<p>set_enabled_interfaces option is used to enable one or more of these interfaces. The possible options are:</p> <p>WebUI SerialOverLAN IdeRedirection.</p>
power_options	<pre> &lt;power_options&gt; &lt;power_state&gt;S5&lt;/power_state&gt; &gt; &lt;wake_on_net_access_sleep_timer&gt;5&lt;/wake_on_net_access_sleep_timer&gt; &lt;/power_options&gt; </pre>	<p>power_options option is use to set the power options. power_state is the lowest state in which AMT is working.</p> <p>wake_on_net_access_sleep_timer is the idle time until ME goes to sleep in low system states (0 for off).</p>

## 6.4 PSK.REPOSITORY.XML File Format

The file is located under the 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>	<pre> &lt;pair&gt;   &lt;pid&gt;xxxx-xxxx&lt;/pid&gt;   &lt;pps&gt;xxxx-xxxx-xxxx-xxxx-xxxx- xxxx-xxxx-xxxx&lt;/pps&gt; &lt;/pair&gt; </pre>	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

## 7 Issuing Certificates and Certificate Authority

### 7.1 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 ← SCA certificate ← 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 ← 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.

## 7.2 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 (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 subcert.pem in the same directory. Once this process is completed, the Configuration Server can issue certificates to Intel AMT devices.

## 7.3 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.

## 7.4 Certificate Chain Format

For the certificate chain format accepted by the Intel AMT device please refer to the Network Interface Guide => Data structures => Security Administration Interface data Type => CertificateChainType data structure.

# **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?".
3. The directory Configuration\CertGenerator\SecScripts\subCA will be created, and a request file "certreq.pem" will be placed in it.
4. Answer no ("n") to the question "Create a demo root CA and sign the request?".
5. 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".
10. A file by the name "subcacert.pem.cer" will be created.
11. Rename the file to "subcacert.pem".
12. Transfer the file "subcacert.pem" to the computer running the SCA to the Configuration\CertGenerator\SecScripts\subCA directory.
13. Restart the Configuration Server.
14. The SCA will now use the certificate signed by the Windows Server2003 root CA.