

Intel® Active Management Technology

Step by Step Developers Users Guide for the Sample and Configuration Application

July 2008

Revision 4.0.5

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. Setup requires configuration by the purchaser and may require scripting with the management console or further integration into existing security frameworks to enable certain functionality. It may also require modifications of implementation of new business processes. 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 www.intel.com/technology/platform-technology/intel-amt/

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See www.intel.com/products/processor_number for details.

<CODENAME HERE> and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel, Microsoft* 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 © 2008, Intel Corporation. All rights reserved.



Contents

1 Introduction 9

1.1 What is Setup and Configuration? 9
1.2 Setup Types 10
1.3 Secure Communications and Authentication Options 11
1.4 Introduction: Provisioning Methods 12
1.5 Remote Configuration “flavors” 12
1.6 USB “provisioning” 12
1.7 Provisioning Methods – a Short Summary 13
1.8 Setup and Configuration Network Layout 14
1.8.1 Intel® AMT Setup and Configuration Application (SCA) 14
1.8.2 Intel® AMT Platform 15
1.8.3 DHCP Server 15
1.8.4 DNS Server 15
1.8.5 Optional Servers 15
1.8.6 Management Console 15

2 General Environment Setup 17

2.1 Install DHCP and DNS 17
2.2 Setup DHCP 20
2.3 Setup DNS 26

3 Configuring the SCA 33

3.1 Install the Configuration Server Application 33
3.2 Configuration Server Application Structure 37
3.3 Intel® AMT Device Configuration Parameters 39

4 PSK Provisioning Setup 43

4.1 Configuration Server Setup 43
4.2 SCA – Configuration File 45
4.3 Intel® AMT Platform Setup 46
4.3.1 Current Provisioning Mode 47
4.3.2 Provision Server IP 47
4.3.3 Provision Server FQDN 47
4.4 Using a USB Storage Device for Factory Mode Setup 47
4.4.1 Requirements 48
4.4.2 Preparation 48
4.4.3 Initializing a Platform 48

5 Remote Configuration 51

5.1 Overview of Remote Configuration Flow 51
5.1.1 Initial Conditions 51
5.1.2 Acquiring a Server Certificate 52
5.1.3 Steps leading to the start of Setup and Configuration 52



5.2 Remote Configuration Setup 53
5.3 ConfigurationServer Setup 54
5.4 Intel® AMT Platform Setup 56
5.4.1 Simplified One-Touch 58
5.4.2 Bare Metal Setup and Configuration 58
5.5 USB Key Support for Remote Configuration 58
5.5.1 Requirements 59
5.5.2 Preparation 59
5.5.3 Initializing a Platform 60
5.5.4 Moving to Setup Mode 60

6 Restoring Intel® AMT to Factory Mode 61

7 USBfile tool 63
7.1 Syntax 63
7.2 Optional/Additional parameter 63
7.2.1 General parameters 63
7.2.2 Version 1 parameters 64
7.2.3 Version 2 parameters 64
7.2.4 Version 2.1 parameters 65
7.3 Examples 67
7.4 USBTool Errors 68

8 Appendix A - Acquiring a Suitable Remote Configuration Certificate 71
8.1.1 Generating a CSR using Microsoft* Stand Alone CA 72
8.1.2 Acquiring Remote Configuration Certificate from a CA 78
8.1.3 Validating Certificate usability 80
8.1.4 Converting Certificate to a format recognized by the Configuration Server 83
8.1.5 Conversion from PFX to PEM Format 88
8.1.6 Add Certificate to the Configuration Server Configuration File 91

9 Appendix B – Generating a Remote Configuration Certificate – using IIS 93
9.1 Creating a server certificate using IIS 93
9.1.1 Creating a CSR 93
9.1.2 Exporting Certificate 94
9.2 Creating server certificate using Enterprise CA (running on Windows* 2003 Server) 95
9.2.1 Creating CSR 95
9.2.2 Issuing Certificate using the Internal CA 95
9.2.3 Exporting Certificate 96
9.3 Creating server certificate using Stand Alone CA (running on Windows Server* 2003) 97
9.3.1 Creating CSR 97
9.3.2 Issuing Certificate using the Internal CA 97
9.3.3 Exporting Certificate 98
9.4 Creating server certificate using OpenSSL 98
9.4.1 Creating CSR 98
9.4.2 Issuing CSR using OpenSSL CA 99
9.4.3 Creating PFX file from Certificate and private key 102
9.5 Converting PFX to PEM (to be used with Configuration Server) 102



10 Appendix C: Creating Certificate Template for Remote Configuration in Microsoft Server* CA 105

11 Appendix D: Fixing chain of trust in converted PEM files 107

12 Appendix E: *.CONF.xml File Format 109

13 Appendix F: PSK.REPOSITORY.XML File Format 125

Figures

Figure 1. Setup and Configuration Network Layout 14
Figure 2. Intel® AMT Platform Setup 46
Figure 3. Intel® Configuration Server Screen Capture 47
Figure 4. Steps 1- 9 of Setup and Configuration 53
Figure 5. <pki_configuration> script 54
Figure 6. Additional Tag descriptions 55
Figure 7. Intel® AMT Platform Setup - Screen Capture 56
Figure 8. XML Five Entity References 109

Tables

Table 1. Intel® AMT Generations 13
Table 2. Install DHCP and DNS 17
Table 3. Setup DHCP 20
Table 4. Setup DNS 26
Table 5. Install Configuration Server Application – Step by Step 33
Table 6. Configuration Parameters 39
Table 7. Configuration Server Setup – Step by Step 44
Table 8. Steps Running Microsoft* Stand Alone CA 72
Table 9. Steps for Acquiring Remote Configuration Certification 78
Table 10. Steps Validating Certification Usability 80
Table 11. Continued Process - Converting Certificate to a format recognized by the Configuration Server 84
Table 12. PFX to PEM Format Conversion 88
Table 13. Add Certificate to Configuration Server config File 91
Table 14. Variable Name Table / Allowed Settings / Usage 110
Table 15. Variable Name Table / Allowed Settings / Usage …cont. 125





Revision History

Document Number	Revision Number	Description	Revision Date
N/A	4.0.5	Merge with Step By Step document by FTL.	July 2008

§





1 Introduction

The sample Setup and Configuration Application (SCA) is a computer program that can be used by developers as a learning vehicle for creating setup and configuration solutions for platforms incorporating Intel® Active Management Technology (Intel® AMT). Setup and configuration in the past was referred to as provisioning. There are still software development kit (SDK) functions and structures that use this terminology.

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.

When an Intel® AMT platform is configured to Enterprise mode, it can be in one of 3 operational states:

- **PRE Provisioning,** also known as “Factory Mode” – in this state, Intel® AMT is not fully operational. The network interface is closed and the local interface is partially functional.
- **IN Provisioning**, also known as “Setup Mode” – in this state, network interface is opened for **Secure Connection** only, starting with Intel® AMT Release 2.0.[1] The Intel® AMT device periodically tries to contact a configuration server by sending “Hello” Packets.
  *Note: as provisioning process is always done in a secure manner it is advised to configure **all** sensitive while in this state.*
- **POST Provisioning**, also known as “Work mode” – Intel® AMT has completed its setup and is ready for daily use.

[1] Starting AMT 2.0

The provisioning process consists of two parts:

- **Changing the Intel® AMT platform state from PRE Provisioning to IN Provisioning** – this part can be done manually, using Intel® MEBX or USB "Provisioning" methods, or automatically by Intel® AMT platform as in "Remote Configuration – Bare Metal".
- **Changing Intel® AMT platform state from IN Provisioning to POST Provisioning** – this part is done by supplying all needed data using the configuration server and sending a CommitChange command.

Once the Intel® AMT platform enters the IN Provisioning state, it performs the following operations:

- Opens the network interface and acquires an IP address. By default, Intel® AMT platform uses DHCP to acquire an IP address. Along with the address, the Intel® AMT platform will also try to get the DNS Server address from DHCP server using DHCP option 6 or legacy option 5, the gateway address using DHCP option[2] and the domain suffix, using DHCP option 15[3].

- Tries to locate Configuration Server – in the most common use-case, the Intel ® AMT platform will issue DNS query to "provisionserver" in the domain it received from DHCP. For example, if DHCP supplied the domain "myDomain.myOrg" then Intel® AMT platform will look for the "provisionserver.myDomain.myOrg.

Once Configuration Server is located, Intel® AMT platform will send it "Hello" packets periodically, in a pre-defined scheme. The "Hello" packet contains relevant data for the current provisioning method.

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 User Guide* for a detailed description on how to perform a Small Business Setup.

[2] DHCP option 3.
[3] DHCP option 15.



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 and later releases support 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. 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 unsecured manner with a configuration server. Therefore, such Intel® AMT devices should be configured on an isolated network. Release 4.0 and later releases do not support Legacy Mode.

- 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.
- Releases 2.2, 2.6 and 3.0 and later releases 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.

1.4 Introduction: Provisioning Methods

When Intel® AMT platform is in IN Provisioning mode, it will allow only secure connection. Intel® AMT platform supports two methods for securing the connection, each one of them require different setup:

- **PSK Provisioning** – uses symmetric encryption with a Pre-shared Key known only to Intel® AMT and Configuration Server. It is considered as a "one-touch" method, as user must physically interact with the platform to supply needed data.
- **Remote Configuration** – uses a-symmetric encryption, based on PKI Certificates, to communicate with Intel® AMT. With a suitable environment, remote configuration is a "zero-touch" process. Once the platform is connected to power and to the network, there is no need to access the platform locally to initiate provisioning.

Note: If Remote Configuration is enabled and PSK data exist, Intel® AMT platform will perform PSK provisioning as it is considered more secured than Remote Configuration.

1.5 Remote Configuration "flavors"

There are several methods for remote configuration:

- **Unsecured DNS** vs. **Secure DNS – Unsecured DNS** means that Intel® AMT relies on network infra-structure to determine the domain in which it is deployed: it relies on DHCP option 15 and optionally on a DNS suffix passed from a software agent. **Secure DNS** means that Intel® AMT was pre-configured with the name of the domain where it will be deployed. Secure DNS is sometimes referred as **Simplified One Touch** as the domain could be written to the platform using a USB key at the end customer level.
- **Bare metal** vs. **Delayed remote configuration –** in **Bare metal**, the Intel® AMT platform initiates remote configuration automatically once it is powered on; therefore, there is no need for an active operating system to be present in the platform. In **Delayed remote configuration** the configuration process needs to be triggered using a Local Agent, which requires the operating system to be present.

1.6 USB "provisioning"

There is an option to supply PSK or Remote Configuration data to Intel® AMT using a USB flash drive that contains a unique setup file. This method is often called "USB Provisioning" – a misleading term, as the platform isn't provisioned at end of process (i.e. it stays I Provisioning state)



1.7 Provisioning Methods – a Short Summary

The following table shows Intel® AMT generations and their support of the above methods:

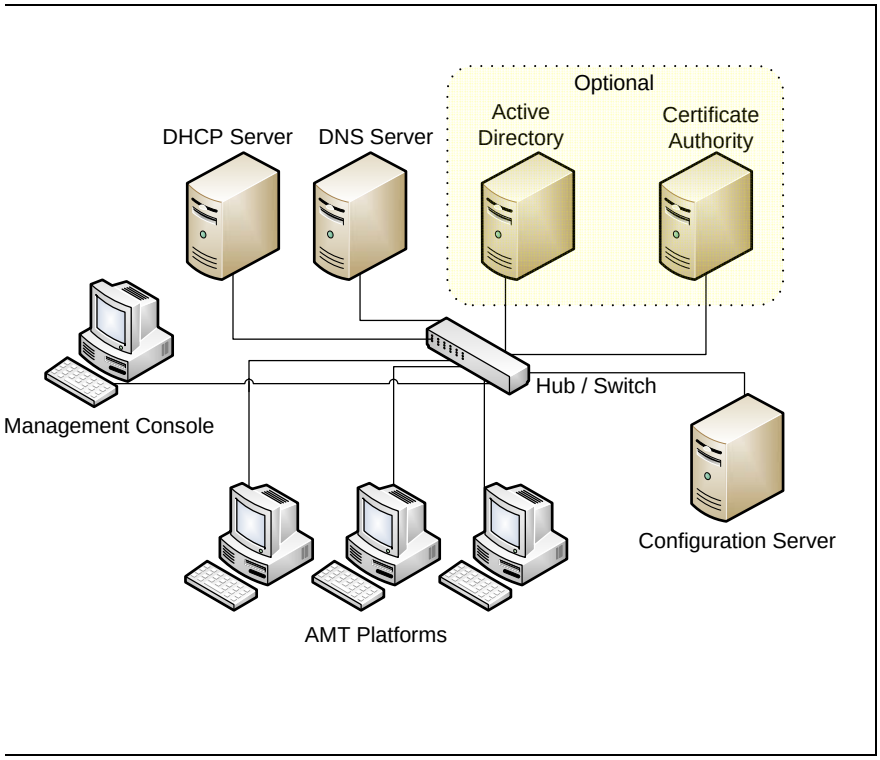
Table 1. Intel® AMT Generations

Setup Method	Description	Applicable Intel® AMT Releases	Initial Conditions	Operator Actions Required
Enterprise PSK	PID-PPS pair; type 2 “Hello” message.	2.0 and up	No pair provided	Operator enters pair manually via menu or with USB storage device
Enterprise PKI-CH (Remote Configuration) (Delay Provisioning)	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 and later releases	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.
Enterprise PKI-CH (Remote Configura-tion) (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 and later releases	No ISV local agent running on host and OEM sets Provisioning time period > 0	No actions required: Platform starts sending “Hello” message as soon as it is connected to the network.
Enterprise PKI-CH (Remote Configura-tion) using USB Key	User customizes 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.	3.0 and later releases	Depend on the selected method : Delay Provisioning/Bare Metal provisioning	

1.8 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:

Figure 1. Setup and Configuration Network Layout



1.8.1 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 platform that SCA software runs on is referred to as the Setup and Configuration Server, sometimes referred to as a provisioning server. When an Intel® AMT device enters Setup Mode, it attempts to establish a network connection with the setup and configuration server and waits for the software running on the server to deliver configuration settings.



1.8.2 Intel® AMT Platform

An Intel® AMT platform 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.

1.8.3 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.

1.8.4 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 hostname of "ProvisionServer". (Note that an OEM platform provider can change “ProvisionServer” to some other value.) If a DNS server is unavailable, then the SCA 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.

1.8.5 Optional Servers

Optionally, a setup environment may include and Active directory (AD) domain server and a Certification Authority (CA).

1.8.6 Management Console

A management console is a platform running an application that is used for managing Intel® AMT platforms.

§



2 General Environment Setup

The instructions in this section are both for PSK and Remote Configuration and are usually done once. This guide assumes that the Intel® AMT platform is deployed in the domain **ftl10.com**

2.1 Install DHCP and DNS

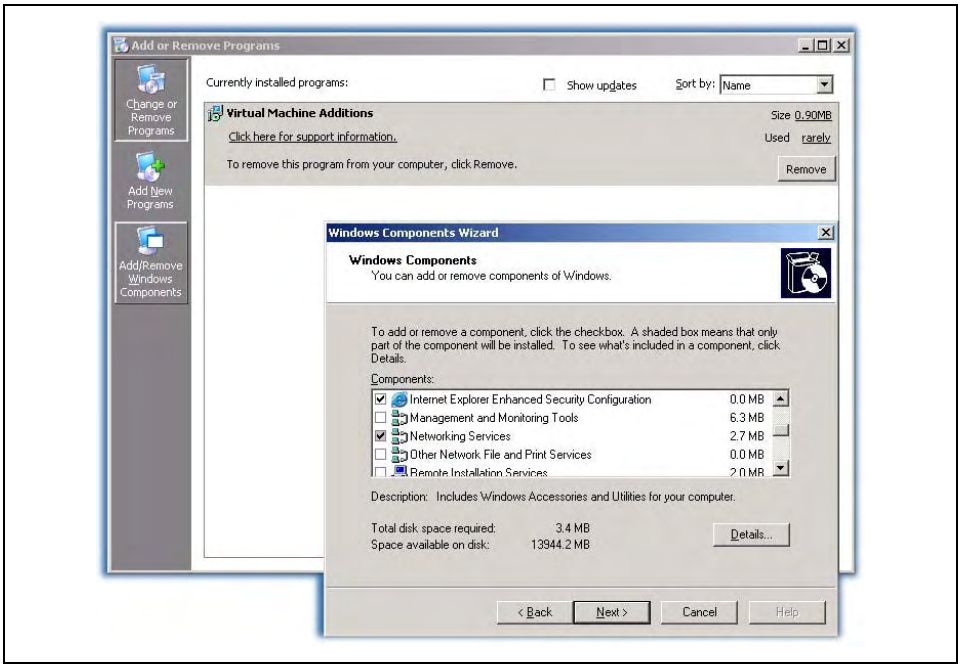
Table 2. Install DHCP and DNS

Open **Add or Remove Programs** in **Control Panel.**

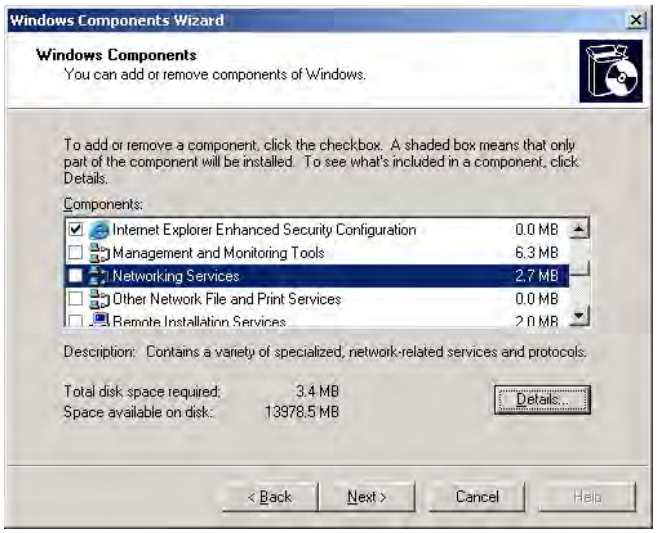


Choose **Add/Remove Windows Components** (last item in left pane).

The **Windows Components Wizard** will appear.

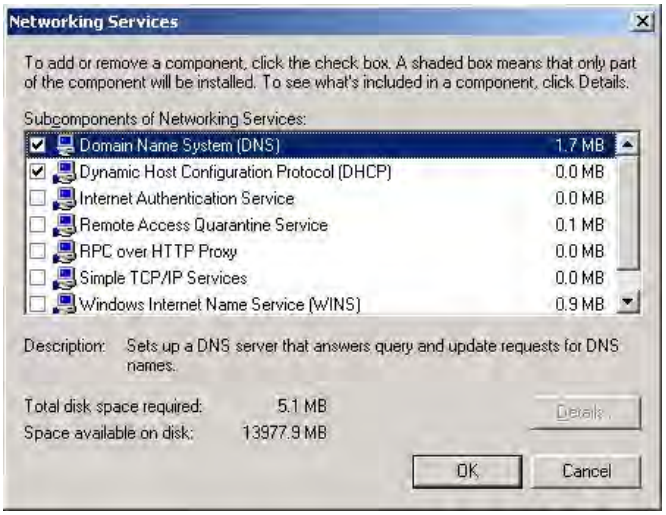


Navigate to **Networking Services** and click **Details…** on the right side of the window



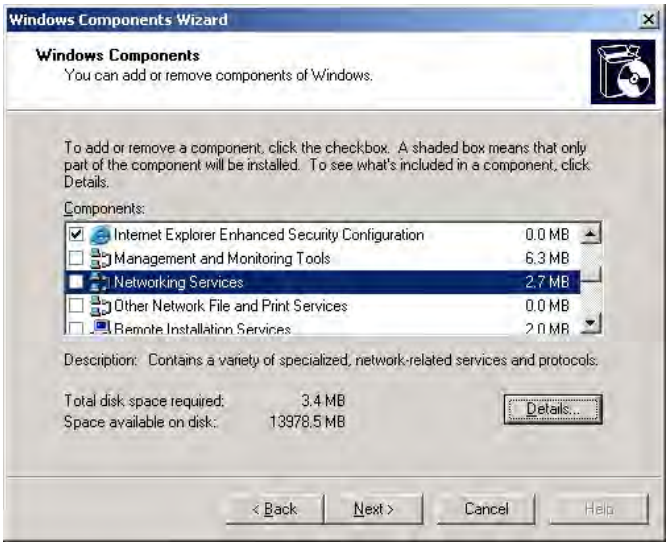
Select the checkboxes next to the **DNS** and **DHCP** services and click **OK**.

The Network Services window will close.



Intel Confidential

Click **Next >** to continue. Windows will start the installation process. You may be requested to load the setup CD.



Once installation is completed, click **Finish** to exit the wizard.

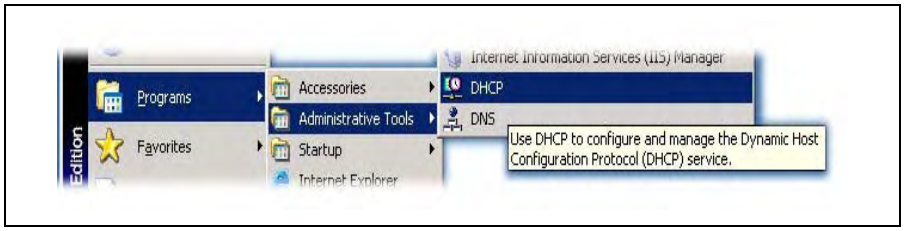
Close the **Add or Remove Programs** window.



2.2 Setup DHCP

Table 3. Setup DHCP

Open **DHCP** from **Administrative Tools** in **Programs** menu

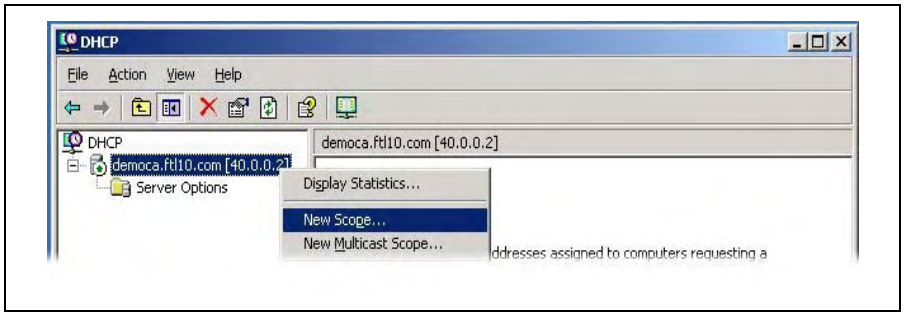


Adding New Scope

Right click on the **DHCP server** in the left pane.

A pop-up menu will appear

Click on **New Scope**… in the pop-up menu.



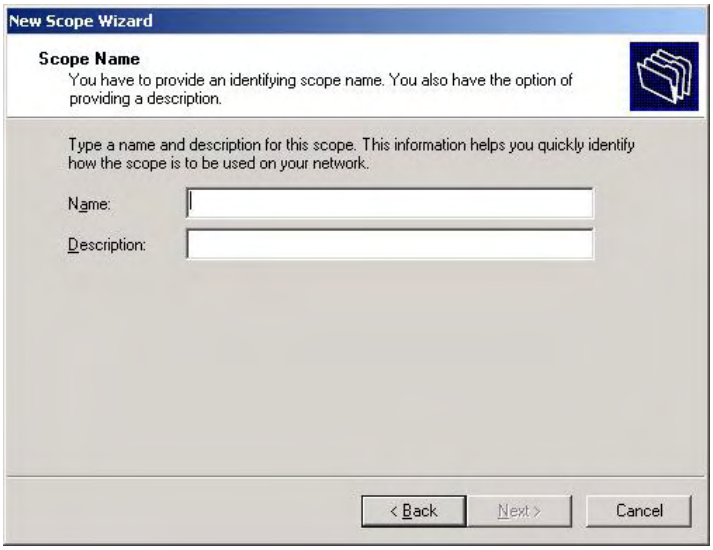
The **New Scope Wizard** will appear. Click **Next >** to continue.



Give a name and a description for your scope.

The name and description are used internally (i.e. they are not exposed by DHCP server), so you can choose anything you want.

Click **Next >** to continue.

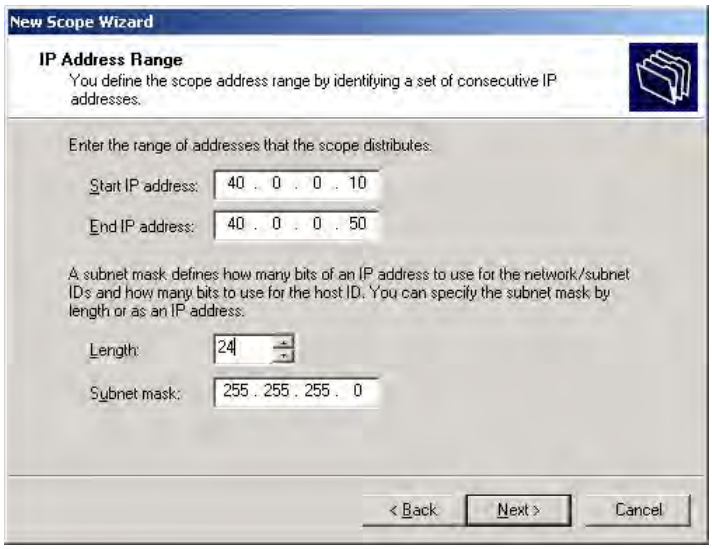


Set the range of IP addresses your DHCP will distribute to clients, as well as the network subnet mask

This example use the IP addresses from 40.0.0.10 to 40.0.0.50 and a subnet mask of 255.255.255.0

(You can specify 24 in the length field instead of typing the mask).

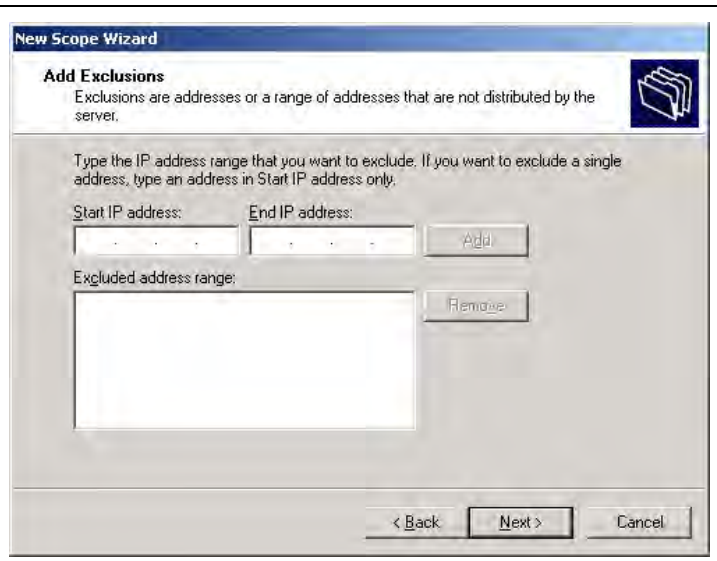
Click **Next >** to continue.



Exclude IP addresses from the above range if needed.

(e.g. some addresses in the above range are used as static IP addresses in your network)

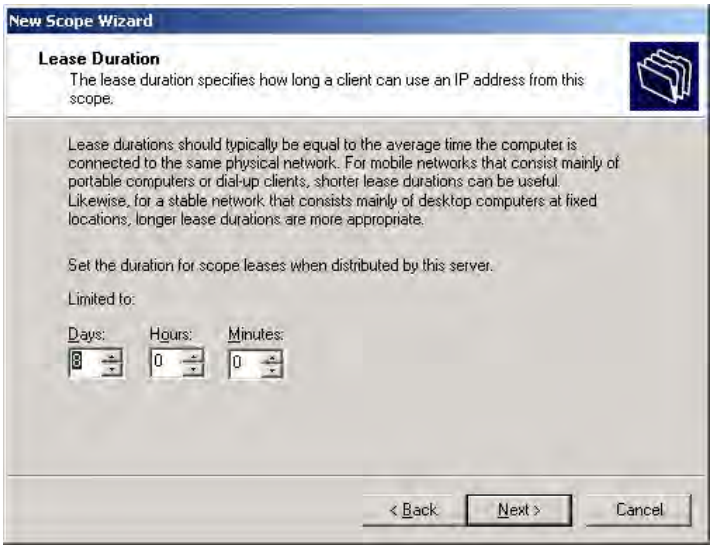
Click **Next >** to continue.



Intel Confidential

Set the DHCP lease duration if needed, or leave it at the default values (8 days).

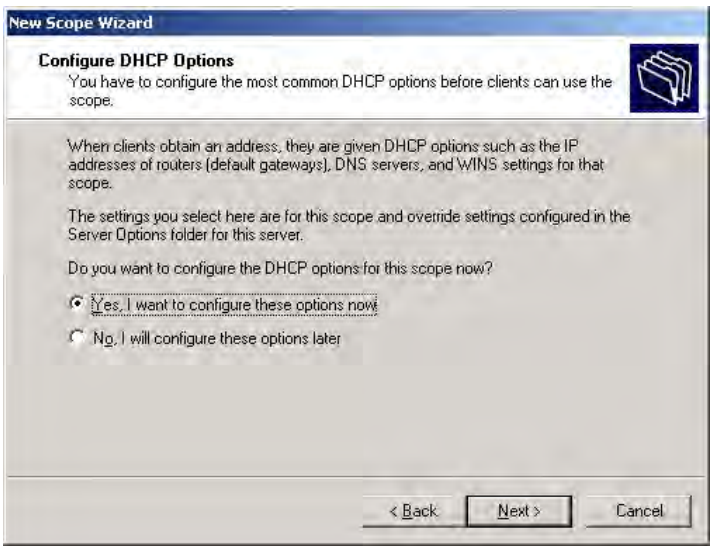
Click **Next >** to continue.



Configure the DHCP options of your scope.

Choose **Yes, I want to configure these options now**.

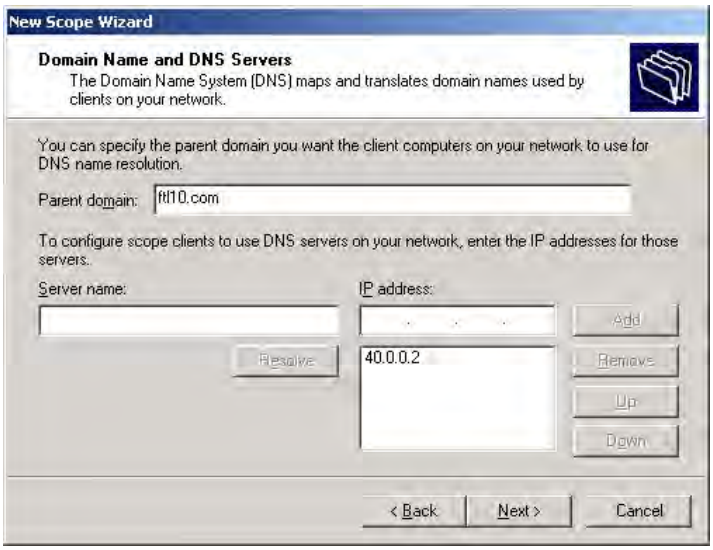
Click **Next >** to continue.



In **Parent domain** field set the name of your domain. (this field is known as DHCP option 15)

Set the IP of your DNS server in the **IP address** field and click **Add**.
(this field is known as DHCP option 6)

Click **Next >** to continue.



If you want to add a router IP address (gateway), you can write in the **IP Address** field and click **Add**.

(this field is known as DHCP option 3)

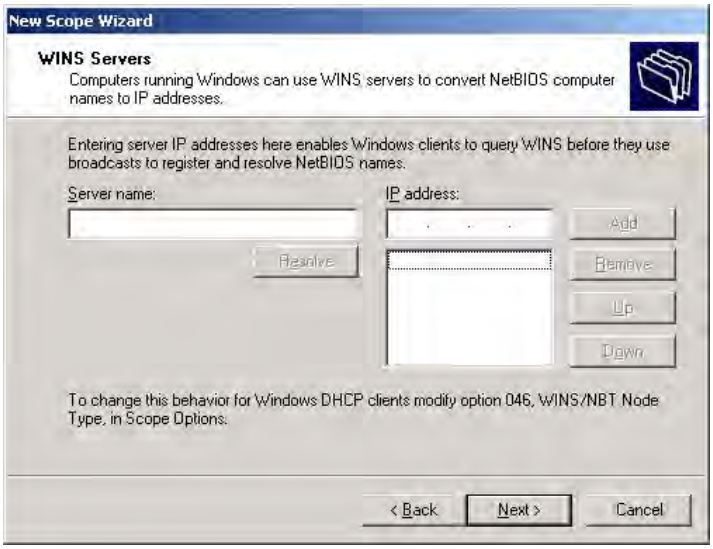
Click **Next >** to continue.



Intel Confidential

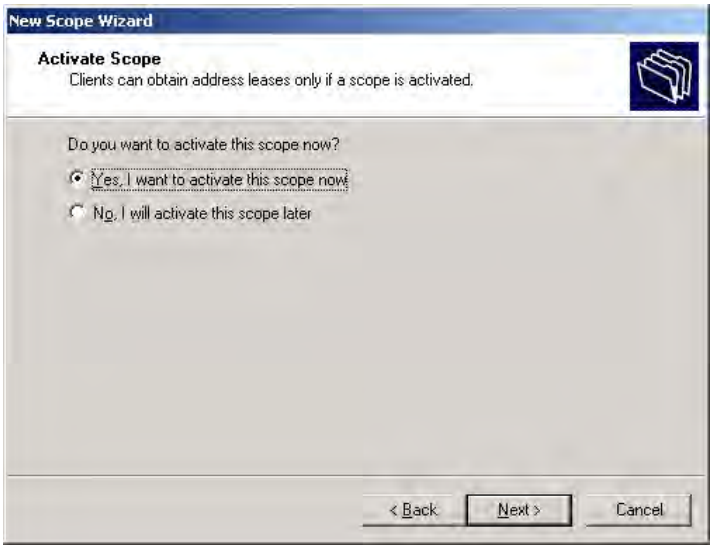
If you have a WINS server for NetBIOS name resolution, you can add it here.

Click **Next >** to continue.



Select **Yes**, I want to activate this scope now.

Click **Next** > to continue.



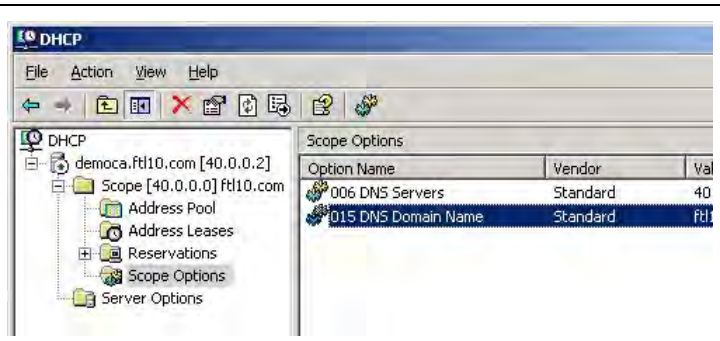
Click **Finish** to close the wizard.



Expand the newly created scope.

Select **Scope Options**.

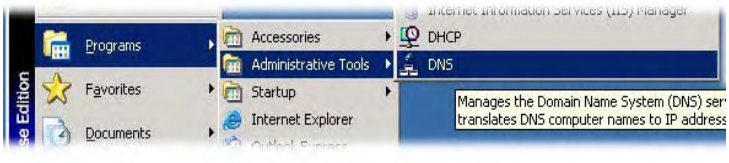
You should have at least option 6 and option 15 set.



2.3 Setup DNS

Table 4. Setup DNS

Open **DNS from Administrative Tools** in **Programs** menu

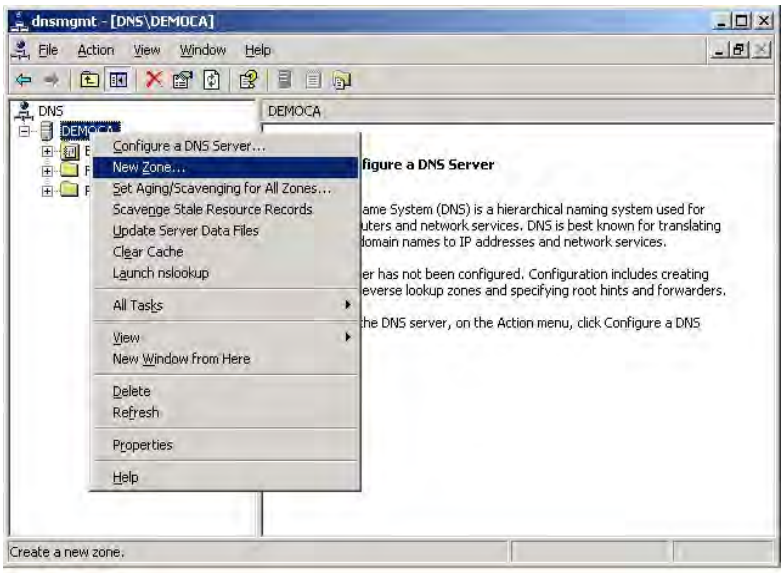


Create a new zone

Right click on your DNS Server name.

A pop-up menu will appear.

Click on **New Zone…** in the pop-up menu.



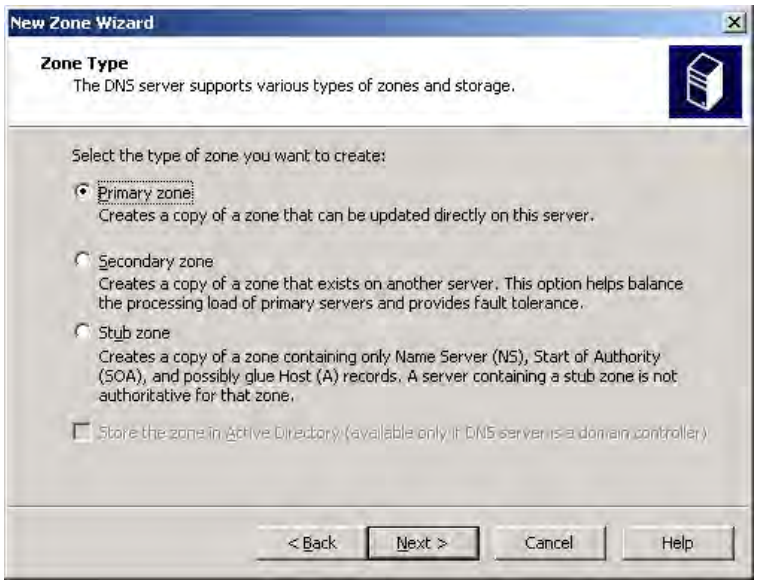
The **New Zone wizard** will appear.

Click **Next >** to continue.



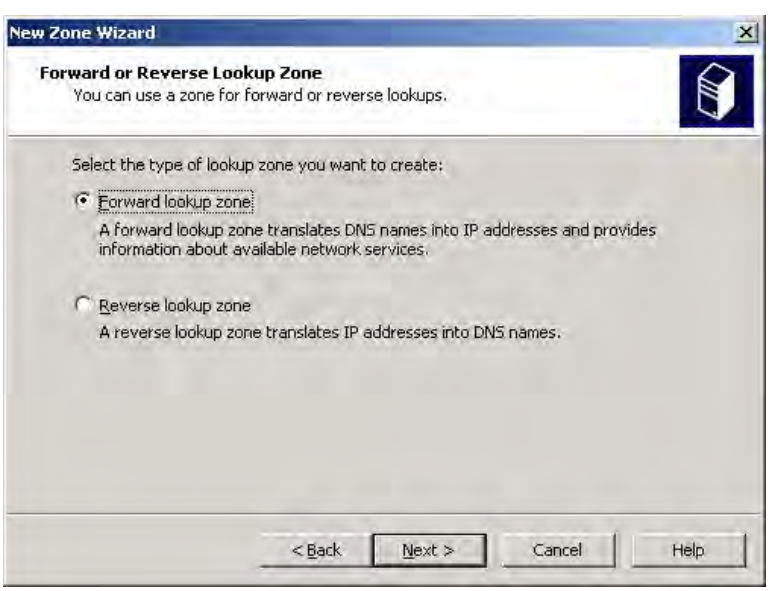
Select the zone type. The example uses the **Primary zone**

Click **Next >** to continue.



Select **Forward lookup zone**.

Click **Next** > to continue.

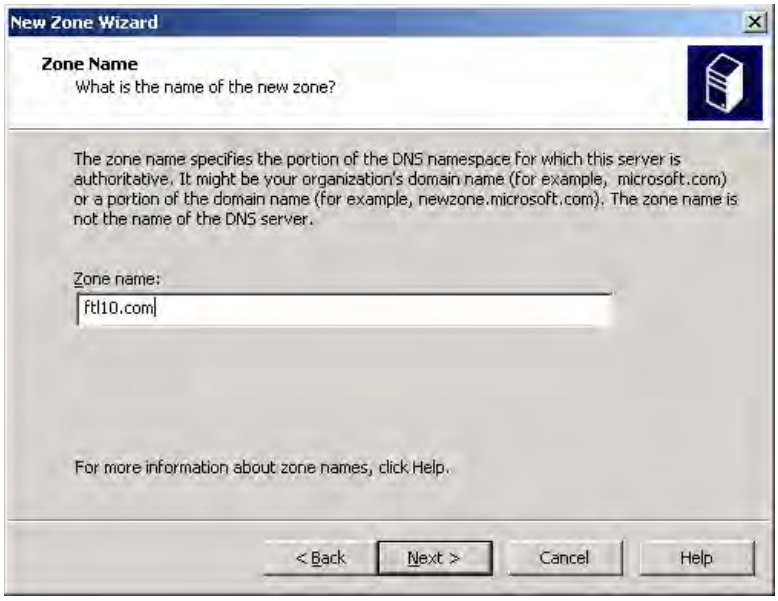


Intel Confidential

Enter the **FQDN suffix o**f your desired domain in the **Zone name field.**

This value should match the one entered for domain name during the DHCP setup.

Click **Next** > to continue.



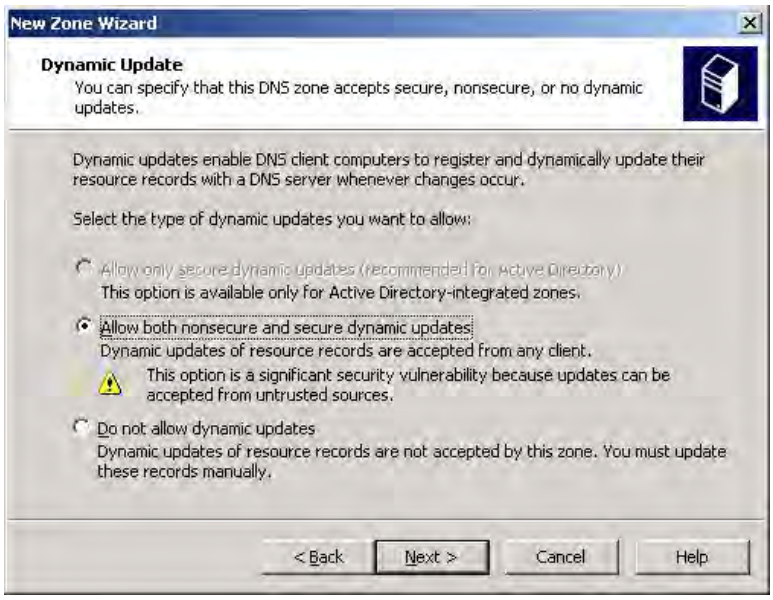
Click **Next >** to continue.



Click **Next >** to continue.

Select a dynamic update option.

(This example selects **Allow both nonsecure and secure dynamic updates**. This option does not affect PSK or Remote Configuration, but it will be easier to access the Intel® AMT platform later, when using Server/Mutual authentication).



Click **Finish** to close the wizard



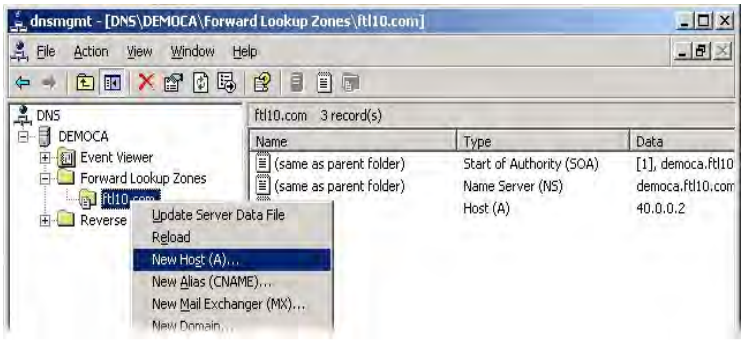
Add DNS Entry for Configuration Server

Expand the **Forward Lookup Zones** in your DNS, at the left pane.

Right click on your newly created domain,

A pop-up menu will appear.

Click on **New Host (A)…** in the pop-up menu.



The **New Host** window will appear.

Enter **provisionserver** in the **Name** field.

Enter the **configuration server** IP in the **IP address.** The example we uses 40.0.0.30.

If you already have an entry for the configuration server in your DNS, you can create an **Alias (CNAME)** record instead of a **New Host (A)** record, and point it to your configuration server record.



Click **OK** to finish.



§



3 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.

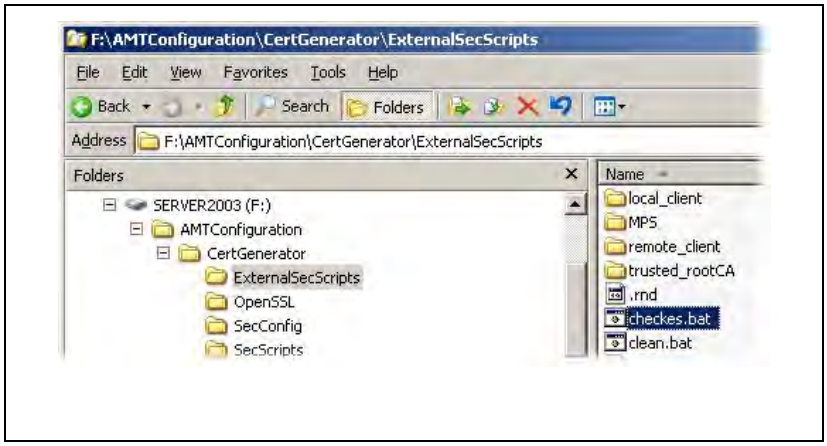
3.1 Install the Configuration Server Application

Locate the Configuration Server application. It can be obtained from:
Official SDK Software Development Kit (located under: Windows\Intel_Manageability_Configuration\Bin)
or from **firmware kits** (located under: Tools\AMT Tools\AMTConfiguration).

Before running the Configuration Server for the first time, a few small modifications are needed so the configuration server will work flawlessly in our environment.

Table 5. Install Configuration Server Application – Step by Step

Navigate to the **EternalSecScripts** sub folder and open **checkes.bat** for edit.



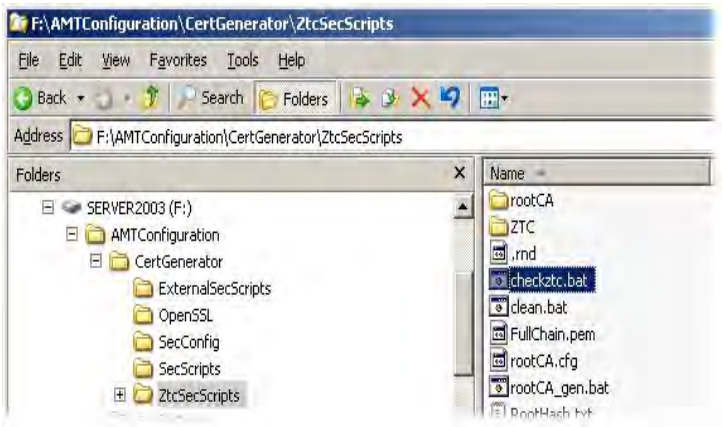
Modify the following lines with the domain suffix, **ftl10.com** in our example:

REMOTE_CLIENT_CN = management_console.**ftl10.com**

LOCAL_CLIENT_CN =
acme_app.**ftl10.com**

```
REM ------------------------------------------------------------------------
REM The following environment parameters can be customized.
REM Note that they need to be coordinated with the appropriate .conf.xml file
REM in order to achieve properly working environemnt.
REM ------------------------------------------------------------------------
set TRUSTED_ROOT_CRL_DISTRIBUTION_POINT=URI:http://trusted_root_crls.ftl10.com
set REMOTE_CLIENT_CN=management_console.ftl10.com
set LOCAL_CLIENT_CN=acme_app.ftl10.com
set MPS_CN=mps.ftl10.com
set PKCS12_PASSWORD=qwerty
REM ------------------------------------------------------------------------
```

Navigate to **ZtcSecScripts** sub folder and open **checkztc.bat** for edit.



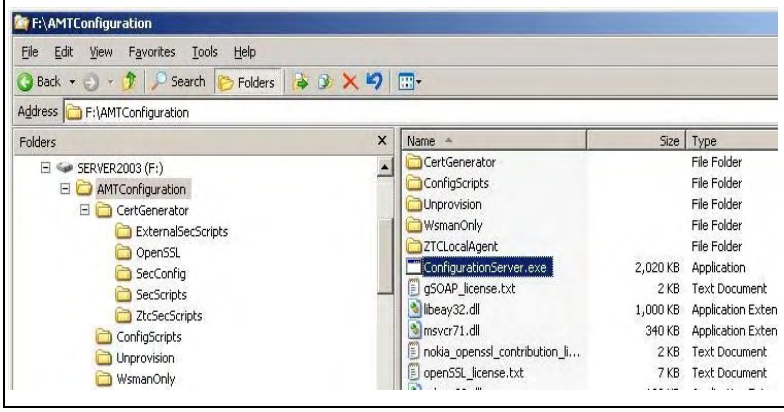
Edit the highlighted line. Change the domain suffix to match your domain. In our case,

ZTC_CLIENT_CN =
acme_app.ftl10.com

```
REM ------------------------------------------------------------------------
REM The following environment parameters can be customized.
REM Note that they need to be coordinated with the appropriate .conf.xml file
REM in order to achieve properly working environemnt.
REM ------------------------------------------------------------------------
set CA_CRL_DISTRIBUTION_POINT=URI:http://crl.demoCA.com
set ZTC_CLIENT_CN=acme_app.ftl10.com
set ZTC_CLIENT_OU=Intel(R) Client Setup Certificate
set PKCS12_PASSWORD=qwerty
```

Double click on
ConfigurationServer.exe

Note: you can determine the port number that the SCA will listen to by starting the configuration server using <-port #num_of_port> option,

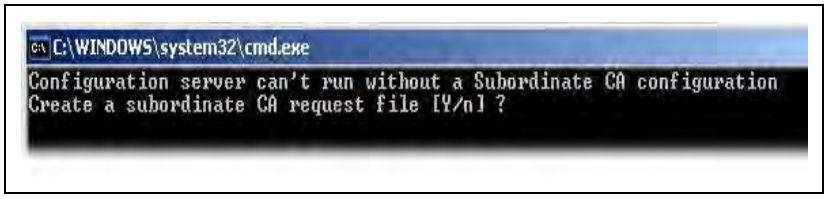


Intel Confidential

When running this application for the first time, the following steps will take place:

A window will pop up requesting to create a **Subordinate CA** request file.

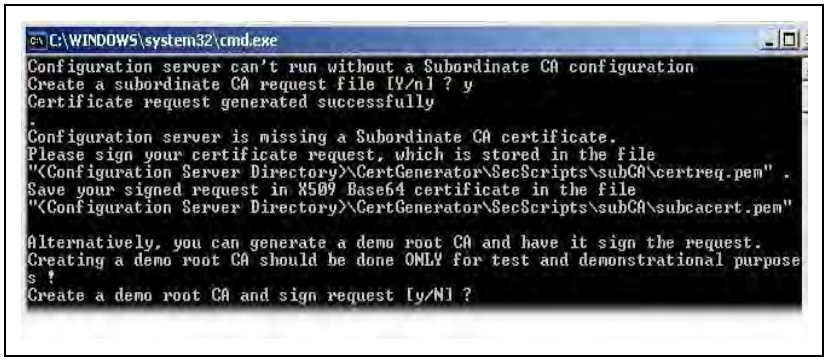
Answer **"y"** and press the **"Enter"** key to continue.



The Subordinate CA request needs to be signed by a CA.

The Configuration Server suggests creating a demo Root CA, and signing the above request.

Answer **"y"** and press **"Enter"** key to continue.

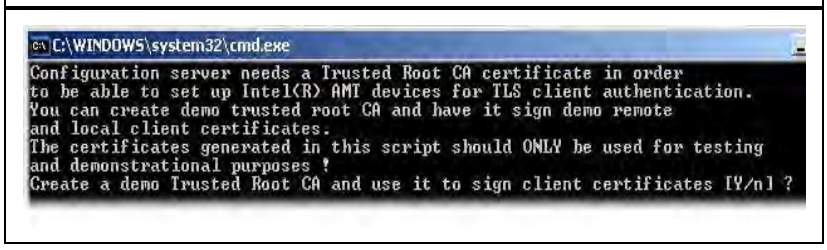


The Configuration Server suggests creating an Auditing Certificate to use the auditing feature introduced in Intel® AMT Release 4.0

Answer **"y"** and press **"Enter"** key to continue.

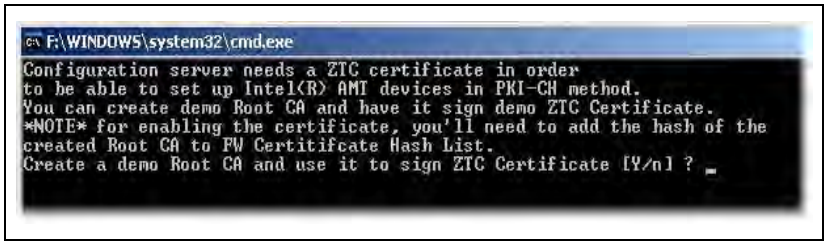


In order to be able to use mutual authentication after setup and configuration completes, Configuration Server will create another demo root CA and two client certificates, one for local access and one for remote access.



Answer **"y"** and press **"Enter"** key to continue.

In order to simplify Remote Configuration Setup, Configuration server will create a demo root CA and sign an appropriate ZTC Certificate.



Answer **"y"** and press **"Enter"** key to continue.

Configuration Server is now running. You can close it for now.



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 Appendix Efor 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.



3.2 Configuration Server Application Structure

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

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.

ExternalSecScripts – contains 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.

AuditCertGen.bat – generates RSA key and certificate for an auditing certificate for Intel® AMT used by the Configuration Server.

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.

ZtcSecScripts – contains scripts used to create certificates for remote configuration use.

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 three 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 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.

ZTCLocalAgent directory – contains the source code for the remote configuration sample agent.

3.3 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 Appendix E for more detailed information about each parameter.

Table 6. Configuration Parameters

Parameter Name	Description	Applicable Intel® AMT Release
host_name	The hostname 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.	All
cfg_password	Current admin password. Must be the same as the password entered during the factory mode setup.	All
tcpip_dhcp_enable	Set to “true” if using DHCP.	All

Parameter Name	Description	Applicable Intel® AMT Release
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
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 used during setup and configuration	2.0 and up
new_pps		2.0 and up
set_network_time	Set to true to synchronize the Intel® AMT internal clock with SCA’s clock. Required for	2.0 and up



Parameter Name	Description	Applicable Intel® AMT Release
	Kerberos and for TLS mutual authentication.	
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, and up
extend_provisioning_period	Restarts the “provisioning period” for the number of hours set in this parameter.	2.2, 2.6, and up
set_8021x_active_in_S0	Enables Intel® AMT 802.1x authentication in S0 power state when host authentication fails.	2.6 and up
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
Digest_acls	Allows creation of digest ACL entries using the SCA (Added to support adding an audit log user).	4.0 and up
audit	Defines auditable events and the credentials used to read the audit log.	4.0 and up
cira	Parameters for client initiated remote access. Defines management presence server address and port and credentials and policies	4.0 and up
environment_detection	A set of domains the define “inside the enterprise”. SCA enables environment detection the Intel® AMT device when this parameter is included.	2.5 and up

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 ISV Storage (Vendor Name, Application Name, Enterprise Name)
Event Filters
PET Packet Subscribers



For more details Command list please see Appendix E – default.conf.xml file format.
§



4 PSK Provisioning Setup

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.

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

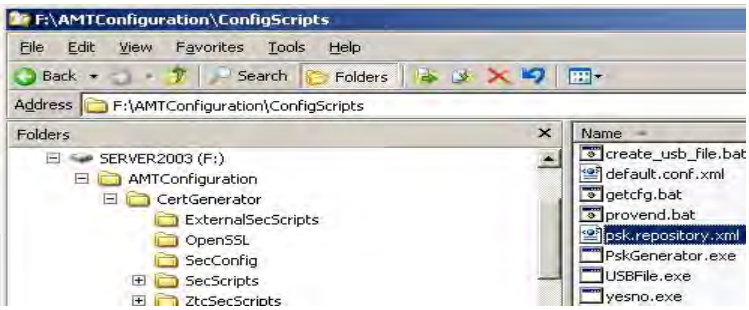
When the PID and PPS are entered via the INTEL® MEBX submenu manually, the firmware checks for checksum characters embedded in the values. The last character of the PID is expected to be a checksum of the previous seven characters, and the fourth character in each group of four characters in the PPS 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 SDK contains the source code for a function that generates PID/PPS pairs with checksums embedded in them. The sample values above were created with PskGenerator in the SDK and have the correct checksums.

4.1 Configuration Server Setup

The Configuration Server keeps a list of PID/PPS pairs in **psk.repository.xml** file . This example uses one of its sample pairs.

Table 7. Configuration Server Setup – Step by Step

'igate to ConfigScripts sub-folder



:n psk.repository.xml

: one of the available PSK pairs or add one of your own. Make sure to use capital letters and numbers only.

```
<?xml version="1.0" encoding="UTF-8"?>
<pairs>
    <!--
    The following element form is is used to store PID/PPS pairs.
    Use the supplied tool PskGenerator to create sample pairs.
    The PID should be unique (the first matching value is used).

    Note: each <pair></pair> tag must contain only one PID/PPS.
    -->
    <pair>
        <pid>XXXX-XXX4</pid>
        <pps>AAAF-AAAF-AAAF-AAAF-AAAF-AAAF-AAAF-AAAF</pps>
    </pair>
    <pair>
        <pid>YYYY-YYYB</pid>
        <pps>BBBI-BBBI-BBBI-BBBI-BBBI-BBBI-BBBI-BBBI</pps>
    </pair>
</pairs>
```



4.2 SCA – Configuration File

Open **default.conf.xml** (it is located in the same subfolder as psk.repository.xml) and modify any platform specific parameters, such as:

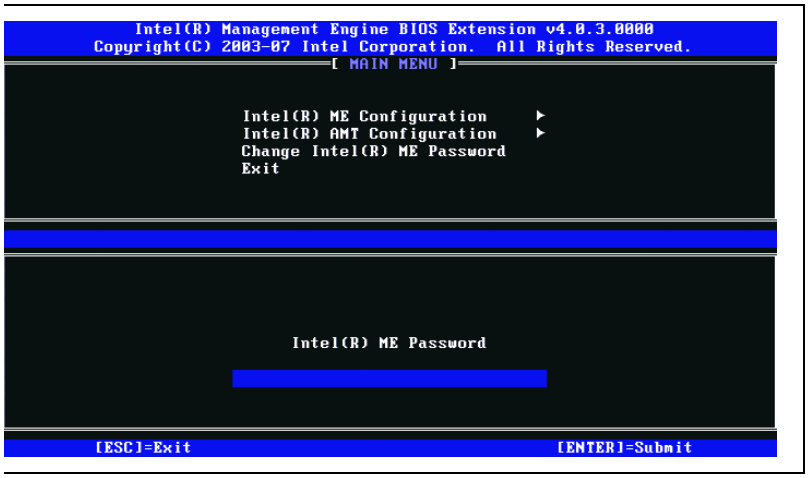
- **host_name –** Intel® AMT platform host name.
  Then the Intel® AMT device is configured to use DHCP it is advised to set the same name as the host platform.
- **domain_name –** the name of the domain where Intel®AMT device is deployed. ConfigurationServer will create SSL server certificate where the CN field value will be host_name.domain_name
- **cfg_password** – the current password of Intel® MEBX.
  If ConfigurationServer will fail to contact with current password, it will try to contact with "admin:admin" credentials.
- **tls_options –** this setting will determine the security level Intel® AMT will use once provisioning process is completed. It can be either "NoAuthentication" which means TLS is disabled on both local and network interfaces, or any combination of "ServerAuthentication" and "MutualAuthentication" which means TLS is Enable.
- **new_network_password –** usually, in IN Provisioning state, the network password is identical to the Intel® MEBX password. It is advised to change the network password to something else in order to increase total security.
- **new_pid** and **new_pps** – this will prepare the platform for future provisioning attempts (i.e. if the platform will perform partial-unprovision after it moved to POST Provisioning state). If this tag is omitted, Intel® AMT will use its current PSK data for future provisioning attempts as well. It is advised to supply new values.
- **set_enabled_interfaces –** by default, SOL, IDER and Web-UI interfaces are closed. This tag will determine what interfaces will be open after provisioning is completed.
- **trusted_root_certificates –** this tag will determine on what CA's Intel® AMT will trust when using Mutual Authentication or when opening Remote Access Connection (CIRA) with TLS.

Run ConfigurationServer application.

4.3 Intel® AMT Platform Setup

Turn on the Intel® AMT platform, press <Ctrl-P> when prompted and enter Intel® MEBX. You should see the following screen.

Figure 2. Intel® AMT Platform Setup



Enter the Intel® MEBX password. The default password is “admin”. If you logged in with the default password, you will be prompted to change it before you continue. You should supply a new strong password. This example we will uses **Admin!98**.

Note: If you don’t see the **“Intel® AMT Configuration”** sub menu, the manageability mode of your platform is probably not Intel® AMT. you should enter **“Intel® ME Configuration”** sub menu, then select **“Intel® ME Features Control”** and choose Intel® AMT mode.

Enter **“Intel® AMT Configuration”** sub menu, then select **“Setup and Configuration”**.

Note: if you can’t see “Setup and Configuration” menu, Intel® AMT provisioning mode is probably not set to Enterprise. Select **“Provisioning Model”** menu and choose **“Enterprise”**.

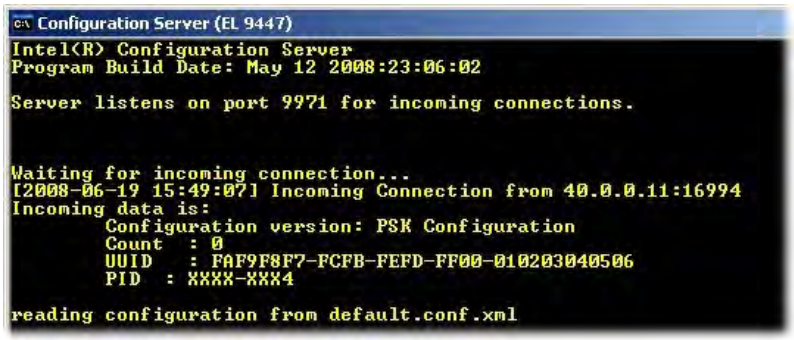
Select **“TLS PSK”** sub menu, then select “Set PID and PPD”. Intel® MEBX will prompt for PID then for PPS.

Exit Intel® MEBX.

The Intel® AMT platform will start sending “Hello” hello packets.



Figure 3. Intel® Configuration Server Screen Capture



4.3.1 Current Provisioning Mode

Return the Current Provisioning mode: PSK/PKI

Note: when both methods are enabled, Intel® AMT will choose the PSK method by default.

4.3.2 Provision Server IP

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 hostname, 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.

4.3.3 Provision Server FQDN

When Provision Server FQDN is given to the Intel® AMT platform, Intel® AMT will send “Hello” packets directly to the IP received as a response to a DNS query using the exact Provision Server FQDN.

4.4 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.

4.4.1 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 SDK includes a sample program and a supporting script. The program is USBFile.exe and the script is create_usb_file.bat)

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.

4.4.2 Preparation

All that is required is to execute the program, 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 USBFile sample program header files in the SDK 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 Intel® 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.

4.4.3 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.

§





5 Remote Configuration

Remote Configuration is a feature included with Intel® AMT Releases 2.2, 2.6 and 3.0 and later releases. 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.

5.1 Overview of Remote Configuration Flow

5.1.1 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 (DNS Name) 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 and later releases).



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® Client Setup Certificate**". This OU value is case-sensitive and must be entered exactly.

5.1.2 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.

See Appendix A for a detailed example of how to sign the certificate request using a Windows Server2003 Certification Authority.

5.1.3 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:

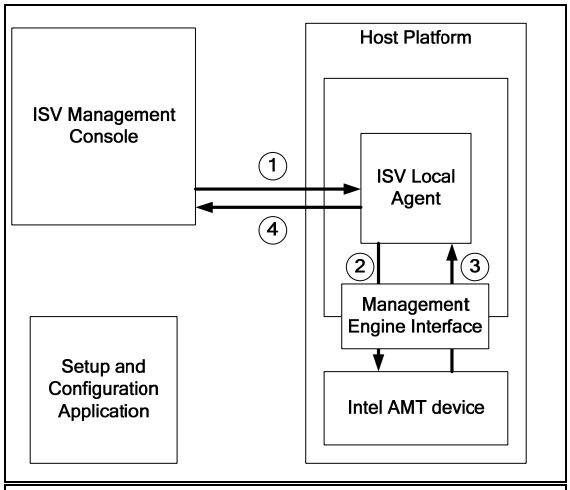
Figure 4. Steps 1- 9 of Setup and Configuration

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.



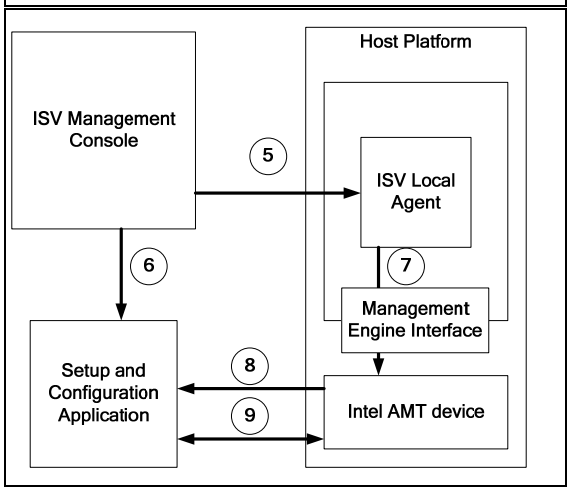
5. Management Console sends OTP to agent.

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.



5.2 Remote Configuration Setup

Remote configuration requires an environment with:

- DHCP and DNS servers that are correctly configured. – See General Setup section.
- An SSL Server Certificate qualified for remote configuration. “Qualified” means that the certificate is signed by one of the Certification Authorities (CAs) that has its root CA hash in Intel® AMT’s certificate hash (CH) list. Since the SCA will use the demo certificate created by ConfigurationServer, to the procedures below add the hash manually to Intel® AMT. For using an external ZTC remote configuration certificate, see appendix A.

5.3 ConfigurationServer Setup

Open **default.conf.xml** (it is located in the same subfolder as psk.repository.xml) and modify any platform specific parameters, such as:

- **host_name –** Intel® AMT platform host name.
  in case AMT is configured to use DHCP it is advised to set the same name as the host platform.
- **domain_name –** the name of the domain where Intel® AMT is deployed. ConfigurationServer will create SSL server certificate where the CN field value will be host_name.domain_name
- **cfg_password** – the current password of Intel® MEBX.
  If ConfigurationServer will fail to contact with current password, it will try to contact with "admin:admin" credentials.
- **tls_options –** this setting will determine the security level Intel® AMT will use once provisioning process is completed. It can be either "NoAuthentication" which means TLS is disabled on both local and network interfaces, or any combination of "ServerAuthentication" and "MutualAuthentication" which means TLS is active.
- **new_network_password –** usually, in IN Provisioning state, the network password is identical to the Intel® MEBX password. It is advised to change the network password to something else in order to increase total security.
- **new_pid** and **new_pps** – in case you wish the next provisioning attempt will be in PSK, this will prepare the platform for future provisioning attempts (i.e. if the platform will perform partial-unprovision after it moved to POST Provisioning state). If this tag is omitted, Intel® AMT will use its current PSK data for future provisioning attempts as well. It is advised to supply new values.
- **set_enabled_interfaces –** by default, SOL, IDER and Web-UI interfaces are closed. This tag will determine what interfaces will be open after provisioning is completed.
- **trusted_root_certificates –** this tag will determine which CA's Intel® AMT will trust when using Mutual Authentication or when opening Remote Access Connection (CIRA) with TLS.

In addition, go to the **pki_configuration** tag and make sure to remove the **otp** tag or put it in a remark.

Figure 5. <pki_configuration> script

```
<pki_configuration>
        <full_cert_chain_file>FullChain.pem</full_cert_chain_file>
        <root_cert_file>rootCert.pem</root_cert_file>
        <otp>password</otp>
        <new_mebx_password>Admin@98</new_mebx_password>
</pki_configuration>
```



Figure 6. Additional Tag descriptions

_configuration	ollection of parameters that support Remote Configuration
_cert_chain_file	n 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)
t_cert_file	nts to a root certificate that saves the SCA from having to extract the root certificate from the certificate chain
	e-time password used to validate the value returned by the Intel® AMT device
r_mebx_password	lacement strong password required to enable the Commit Changes command

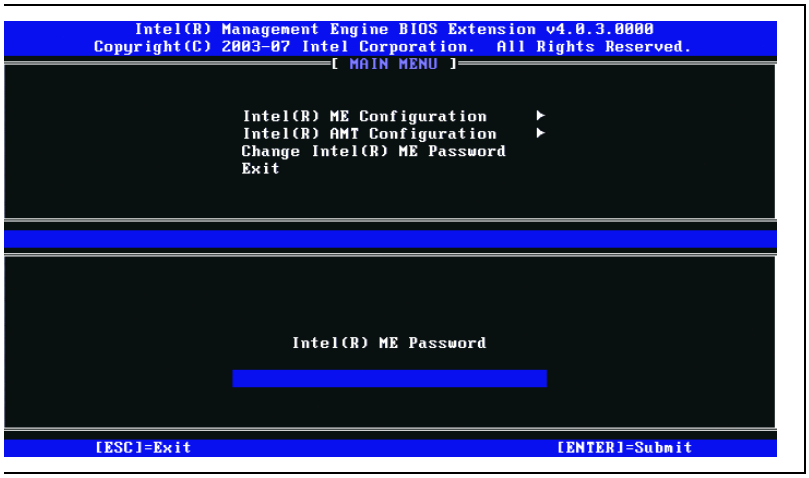
Consult ***Developers Guide to the Sample Setup and Configuration Application*** on available parameters and their values. You can also use the comments in **default.conf.xml**.

Double click on ConfigurationServer.exe to run the application.

5.4 Intel® AMT Platform Setup

Turn on Intel® AMT platform, press <Ctrl-P> when prompted and enter Intel® MEBX. You should see the following screen.

Figure 7. Intel® AMT Platform Setup - Screen Capture



Supply the Intel® MEBX password. Default password is “admin”. If you logged in with default password, you’ll be prompted to change it before you continue. You should supply a new strong password. This example uses **Admin!98**.

Note: in case you don’t see **“Intel® AMT Configuration”** sub menu, the manageability mode of your platform is probably not Intel® AMT. you should enter **“Intel® ME Configuration”** sub menu, then select **“Intel® ME Features Control”** and choose AMT mode.

Enter **“Intel® AMT Configuration”** sub menu, then select **“Setup and Configuration”.**

Note: if you can’t see “Setup and Configuration” menu, Intel® AMT provisioning mode is probably not set to Enterprise. Select **“Provisioning Model”** menu and choose “Enterprise”.

Select “TLS PKI” sub menu, then select “Manage Certificate Hashes”. In case you can’t see such an option, Remote Configuration is disabled. Select “enable/disable Remote Configuration” and choose “enable”.



In “Manage Certificate hashes” press “INS” key to add a certificate hash. Give it a name and then add the customized hash:

On ConfigurationServer platform , navigate to CertGenerator\ZtcSecScripts and open RootHash.txt. You should see something like this:

Replace ':' with spaces and insert the hash into the FW
SHA1 Fingerprint=8A:D3:6B:51:81:06:60:51:9B:69:51:9C:E4:CD:35:32:48:4D:A3:DA

Add the “SHA1 fingerprint” data to the FW in the following manner:

8AD3-6B51-8106-6051-9B69-519C-E4CD-3532-484D-A3DA

Exit Intel® MEBX and boot to OS.

Remote Configuration process should start unless Delayed ZTC Remote Confguration is performed. In this case do the following:

Install **HECI driver** on Intel® AMT platform. (Located in Drivers\HECI, in FW kit)

Double click on **setup.exe** and follow installation instructions.

Copy **ZTCLocalAgent** folder to Intel® AMT platform.

- In the FW kit it is under **iAMT Tools\iAMTConfiguration**
- In the SDK, open **Windows\Intel AMT SDK\Bin\Configuration**. Copy **ZTCLocalAgent.exe** and **StatusStrings.dll** to a new folder

Open command line, navigate to ZTCLocalAgent folder.

- To start remote configuration, run:
  ZTCLocalAgent –activate
- To also specify an OTP, run:
  ZTCLocalAgent –activate –otp otp_value
- To also specify PKI DNS suffix, run:
  ZTCLocalAgent –activate –dns pki_dns_suffix



You can specify both OTP and PKI DNS Suffix.

Specifying PKI DNS Suffix using SW agent is still considered **Unsecured DNS**, it will raise the security level somewhat, as the Intel® AMT device will now perform triple comparison between DHCP option 15, the DNS Suffix in the remote configuration certificate and the value entered via the ZTC agent.

The remote configuration process should start now, and if everything was set correctly, it should also end with a configured platform.

5.4.1 Simplified One-Touch

Intel AMT Release 3.0 and later releases support 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 **Error! Reference source not found.**, 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.

5.4.2 Bare Metal Setup and Configuration

With Intel AMT Release 3.0 and later releases, 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

5.5 USB Key Support for Remote Configuration

The Intel® ME BIOS extension for Intel® AMT Release 3.0 and later releases supports an added format (Version 2.0) for USB keys that aids in preparing Intel® AMT platforms for remote configuration. This automates the simplified one-touch process. (See USB Key with Version 2.1 Format for a description of extensions to the USB record format added with Intel® AMT Release 4.0.) 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 Intel® MEBX password
- Optional DNS suffix or the setup and configuration application FQDN
- Option to start Remote Configuration
- Up to three certificate hashes

5.5.1 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 type 2 file with all or a subset of the above parameters. The SDK includes a sample program and a supporting script. The program is USBFile.exe and the script is create_usb_file.bat. See the readme in **<SDKRoot>\Windows\Intel_Manageability_Configuration\Configuration\USBFile** for the usage of this function.

5.5.2 Preparation

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

1. Identify the parameters
2. Create a file named “setup.bin” in the proper format (see the USBFile sample program header files in the SDK for the exact format). The file will include:
   a. A header that notes file format.
   b. A record that includes:
      i. A flag that enables Intel® AMT
      ii. The default Intel® 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.)



3. Format the USB key to FAT16.

4. Write the file to the USB key.

5.5.3 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.

5.5.4 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.

See Issuing Certificates and Certification Authority for more information about certificate operations performed by the SCA.

§



6 Restoring Intel® AMT to Factory Mode

Intel® AMT is returned to Factory Mode by selecting the Un-provision 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 Un-provision 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 Intel® INTEL® 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. User customized certificates hashes are deleted from the FW.
11. Default pre define certificates hashes move back to active state.
12. 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.



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 hostname is not erased.
- The provisioning server IP and port are not erased.
- The domain name is not erased.
- Certificates Hashes state are reserved.

Note: 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.
- The OEM-configured Remote Configuration enabled/disabled state cannot be cleared or changed by any means, including clearing the CMOS, as described below.
- Restoring Intel® AMT to Factory Mode is not a supported feature of Sample SCA.

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



7 USBfile tool

USBfile is a command line tool used to create an Intel(R) AMT USB file. The tool allows the user to create two types of USB key files for the purpose of preparing an AMT machine for provisioning using a USB device.
For PSK provisioning, the tool allows for automatically generating PSK pairs.
For testing PKI provisioning, the tool can be used to create a single (non-consumable) PKI record file.

7.1 Syntax

To view the valid records of a USB file:
USBfile -view <usb file name>

To view a summary of a USB file:
USBfile -summary <usb file name>

To create a USB file:
USBfile -create <usb output file name> <current MEBx password>
<new MEBx password> < Optional/Additional parameters>

7.2 Optional/Additional parameter

7.2.1 General parameters

-v 1|2|2.1: the setup file version, 2.1 by default.
-v1file <version 1 outfile>: creates an additional version 1 setup file.

7.2.2 Version 1 parameters

-pid <pid> -pps <pps>: a psk pair.

-rpsk : this will generate a random psk pair.

-nrec <num of records>: create the requested number of records.

-gen <num of records>: create the requested number of records and generate a random psk pair for each one of the records,
Note: this option is deprecated, use -nrec and -rpsk options to generate multiple records with random psk pair.

-xml <xml file name>: if psk configuration is chosen the PSK records that are created will be dumped to the given file.

7.2.3 Version 2 parameters

-amt: this will set the manageability selection value to AMT.

-consume 0|1: generate inconsumable record or consumable record(s).
0 (inconsumable) by default.

-dns <DNS suffix>: sets the PKI dns suffix name (up to length 255).

-fqdn <prov server fqdn>: string up to length 255.

-ztc 0|1: disable/enable PKI Configuration, 0 (disable) by default.

-hash <certificate file name> <friendly name>: to compute and add the hash of the given root certificate file. Up to three certficate hashes may be specified
Note: each hash requires this usage separately,
for example: for two hashes use the following usage:
-hash <cert. file name> <friendly name> -hash <cert.> <friendly name>.

-redir <n>:
This is an integer that is calculated as follows:
bit 0 : 1 (Enable) or 0 (Disable) - SOL feature
bit 1 : 1 (Enable) or 0 (Disable) - IDER feature
bit 2 : 1 (Enable) or 0 (Disable) - Username/password authentication type of the SOL/IDER in the ME FW.



7.2.4 Version 2.1 parameters

-hostname <hostname>: host name max length is 63.

Note: this option is not valid when generating an inconsumable record.

-domname <domain name>: domain name max length is 255.

-dhcp 0|1 : disable/enable DHCP.

-pm 0|1: enterprise provisioning / small business(SMB), 0 (enterprise) by default.

-fwu 0|1: disable/enable Firmware Local Update.

-fwuq 0|1|2: Always|Never|Restricted Firmware Update Qualifier.

-sfwu 0|1 : disable/enable Secure Firmware Update.

-ito <4 byte of idle time out> : 4 char of idle time out (valid values: 1-65535).

Note: this setting may not be applicable under some power packege definitions.

-pspo <port number> provision server port number.

-psadd <ip addr> :ip address for provision server e.g 123.222.222.121

-s4p <localHost:SubnetMask:GatewayAddr:DNSaddr:SecondaryDNSaddr>

:e.g 10.0.0.1:255.255.255.0:10.0.0.2:10.0.0.3:10.0.0.4

Notes: This option is not valid when generating an inconsumable record.

DHCP flag must be disabled.

-passPolicyFlag<0|1|2> : Default/block in post/always open.

-vlan <VlanStatus(0|1)-VlanTag(1-4096)> : VlanStatus disable/enable, e.g 0-4011.

-pp <GUID>: set the power packege ,GUID length should be 16.

the GUID should be in network order just as it returns from the enumarate power package soap command.

Notes:

1. In order to create a PSK file (by using the -pid and -pps, or alternatively by -rpsk),

the user needs to provide the old and new Mebx passwords. The user must select the format

version of the generated file, according to the version of AMT that will be provisioned.

Format version 2 provides the ability to enable AMT (i.e. set the manageability mode to AMT)

from the usb key (by using the –amt flag), as well as to configure the

sol/ider settings (by using the -redir). However, this version is currently

only supported by AMT 2.6(+). If format version 2 is chosen, the user may optionally generate an associated XML file with the generated PSK pairs as well as a format version 1 file with identical PSK pairs. These files may be used to import the PSK pairs to the configuration server.

2. To create a PKI file, the user needs to provide the old and new Mebx passwords and enable PKI configuration (using the -ztc flag).
   In addition the user may optionally set the PKI dns suffix, set the provisioning server FQDN and add user defined certificate hashes. Note that if a user adds a user defined hash, the default hashes will be disabled and the previously added user defined hashes will be deleted. Note that a PKI file is only supported by the version 2(+) file format.
3. The tool is intended to support only the two use cases described above. There may be additional USB key record configurations that are “legal” (according to the USB format file specifications) that are allowed or blocked by the tool, however users are advised to follow one of the two configurations described.
4. If -nrec option is not selected a single record is created.
5. If -consume option is not selected an inconsumable record is created.
6. If -pid option was selected the -pps option must come with it and vice versa.
7. If -rpsk or -gen option was selected along with -pid and -pps options, the psk pair that will be used is the one supplied using -pid and -pps.
8. If -pspo option was selected the -psadd option must come with it and vice versa.
9. The BIOS requires a binary file with the name "setup.bin".
10. If a certificate hash is added, all default hashes will be disabled and all existing user defined hashes will be deleted.

Usage Concept:

1. Use USBFile to create a USB file. Use the command line options to determine the type of file that will be created (PSK or PKI), as well as the fields that will be included (dns suffix, provisioning server fqdn...).
2. If PSK records are generated, integrate the corresponding XML file (created using the -xml flag) into the Configuration Server.
3. Copy the generated file to a cleanly formatted USB storage device (FAT formatted).



4. Insert the USB storage device into an Intel AMT system. For version 1 files, AMT must be enabled. For version 2 files, AMT can be enabled using the USB device by configuring the records with the -amt flag.
5. Boot the AMT system.
6. Intel AMT reads the USB storage device, and reads the next available record. If the record is consumable, the record is marked as "used" on the USB storage device.
7. Intel AMT validates its password against the password in the record and then saves the parameters in the record.
8. Intel AMT can begin network provisioning.

7.3 Examples

1. Create a version 1.0 10-record USB file and a corresponding XML file with pseudo-randomly generated PID/PPS pairs, where all records have the same current MEBx password and new MEBx password.

USBfile -create setup.bin admin Admin22@ -rpsk -v 1 -nrec 10 -xml setup.xml -consume 1

2. Create a version 2.1 10-record USB file and a corresponding version 1.0 file with pseudo-randomly generated PID/PPS pairs, where all records have the same current MEBx password and new MEBx password.

USBfile -create setup.bin admin Admin22@ -rpsk -v1file setup1.bin -nrec 10 -consume 1

3. Create a single (version 2.1, inconsumable) PKI record file. Set the manageability mode, enable ZTC and Generate a hash from cert.pem.

USBfile -create setup.bin admin Admin22@ -amt -ztc 1
-hash cert.pem friendlyName

4. Dump contents of USB file to the screen

USBFile -view setup.bin

7.4 USBTool Errors

If an invalid argument was given to one of the parameters, an error message describing the related error will be presented

followed by the usage screen.

An error code will be presented to the user if an error occurs while the tool is creating or viewing the file.

1 - The setup file header has an illegal UUID.

2 - The setup file version is unsupported.

3 - A record entry that does not contain a current MEBx Password was encountered.

4 - The given buffer length is invalid.

5 - The header chunk count cannot contain all of the setup file header data.

6 - The record chunk count cannot contain all of the setup file record data.

7 - The requested index is invalid.

8 - The setup file header indicates that there are no valid records

(RecordsConsumed >= RecordCount).

9 - The given buffer is invalid

10 - A record entry with an invalid Module ID was encountered.

11 - A record entry with an invalid record number was encountered.

12 - The setup file header contains an invalid module ID list.

13 - The setup file header contains an invalid byte count.

14 - The setup file record id is not RECORD_IDENTIFIER_DATA_RECORD.

15 - The list of data record entries is invalid.

16 - The setup file is corrupted.

17 - The setup record has already been used.

18 - A record entry that does not contain a new MEBx password was encountered.

19 - A record with an invalid manageability feature selection was found.

20 - Invalid input was received.

21 - A record with invalid certificate hash settings was encountered.



101 - Failed to write to the given file.

102 - Failed to read from the given file.

103 - Failed to create random numbers.

104 - The CurrentMEBx password is invalid.

105 - The NewMEBx password is invalid.

106 - The PID is invalid.

107 - The PPS is invalid.

108 - The data record is missing a CurrentMEBx password entry.

109 - The data record is missing a NewMEBx password entry.

110 - The data record is missing a PID entry.

111 - The data record is missing a PPS entry.

112 - Invalid DnsSuffix.

113 - Invalid fqdn.

114 - Invalid ZTC setting.

115 - Invalid sol ide redirection config.

§





8 Appendix A - Acquiring a Suitable Remote Configuration Certificate

An application that wishes to configure an Intel® AMT platform using the remote configuration method, must introduce a valid certificate for remote configuration. By design, a certificate is valid for remote configuration if the following conditions are met:

- It is a server certificate. i.e. it has the following OID **1.3.6.1.5.5.7.3.1**
- It has at least one of the following:
  - A designated string in OU field: **Intel(R) Client Setup Certificate**
  - A designated OID in EKU (Extended Key Usage)

    : **2.16.840.1.113741.1.2.3** – see Appendix C to see how to create MS CA certificate template.

It is signed by a CA whose trusted root certificate hash is present and active in Intel® AMT's certificate hash list.

The DNS Suffix of the Certificate's CN field must match the domain name reported by DHCP (option 15).

The following steps will show you how to use Microsoft* Stand Alone CA for this purpose.

Note: There are several more methods to create a certificate for remote configuration. **Appendix A** will discuss each method in detail.

You can also use the following instructions from common certificate providers:

Verisign*:

http://www.verisign.com/support/mpki-for-ssl-support/flash/How%20to%20Generate%20a%20CSR%20using%20IIS%206_v1.htm

Comodo*:

http://www.instantssl.com/ssl-certificate-support/csr_generation/ssl-certificate-index.html

GoDaddy* / Starfield*:

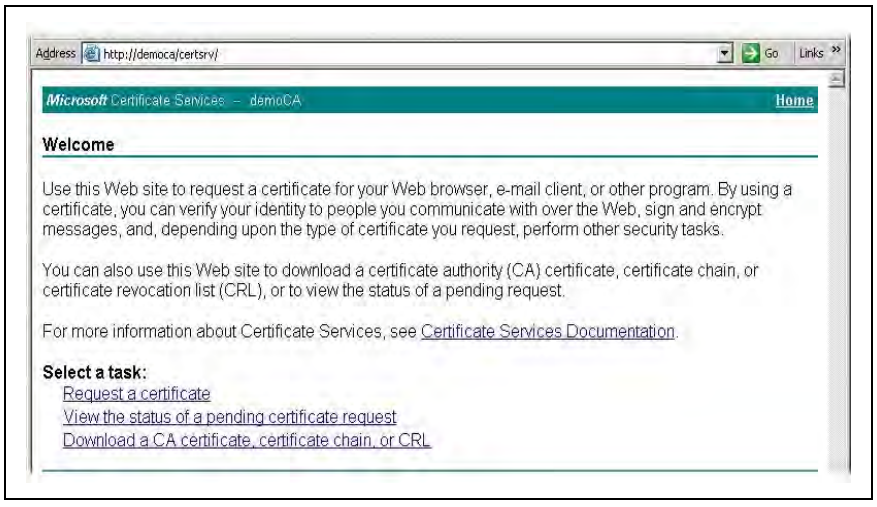
https://certificates.starfieldtech.com/CSRgeneration.go

Once you have obtained a certificate, you should convert it to a format recognized by the Configuration Server.

8.1.1 Generating a CSR using Microsoft* Stand Alone CA

The following procedure generates a Certificate Signing Request (CSR). Note: the following steps assume you are running a Microsoft Stand Alone CA in an environment with Microsoft* IIS.

Table 8. Steps Running Microsoft* Stand Alone CA

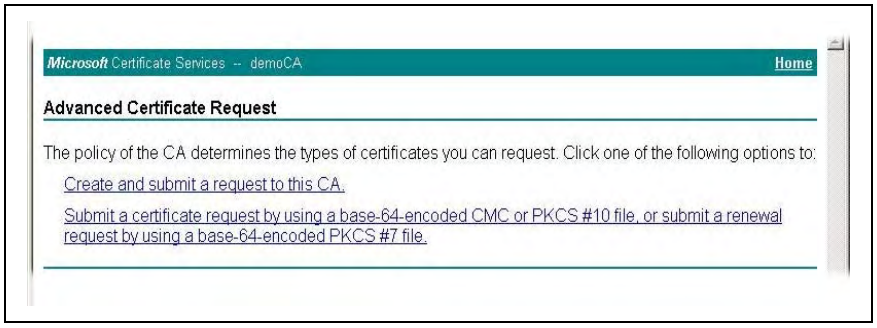
Open your web browser	
Enter in the address bar **http://CA_Name/certsrv**	
(in this example //demoCA/certsrv)	
The **Certificate Services** page will be open.	
Click on **Request a certificate** under the title **Select a task**.	

The **Request a Certificate** page will open.

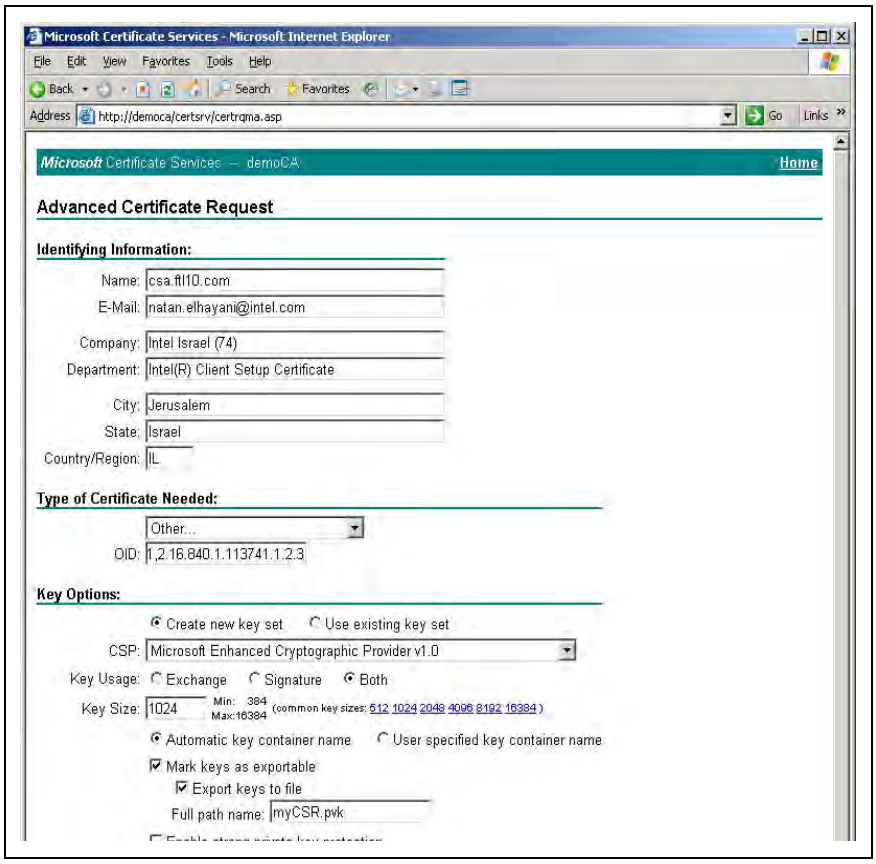
Click on **advanced certificate request**



On the **Advanced Certificate Request** page, choose **Create and Submit a request to this CA**



The **Advanced Certificate Request** page will open.



The following steps show how to fill in the necessary information on this page.

Note::

When sending a CSR to a commercial CA (e.g. Verisign*, Comodo*, etc.) the data entered must be accurate.

You must be an owner of a domain in order to get a commercial SSL Server Certificate.

Also note that CAs tend to override some of the data entered, therefore it is recommended to enter both designated OU and OID where the supplier allows them as often only one of them will be left in the purchased certificate.

The following example is for a domain called ftl10.com owned by Intel Israel Functional Test Lab (FTL).

In **Identifying Information** enter the following:

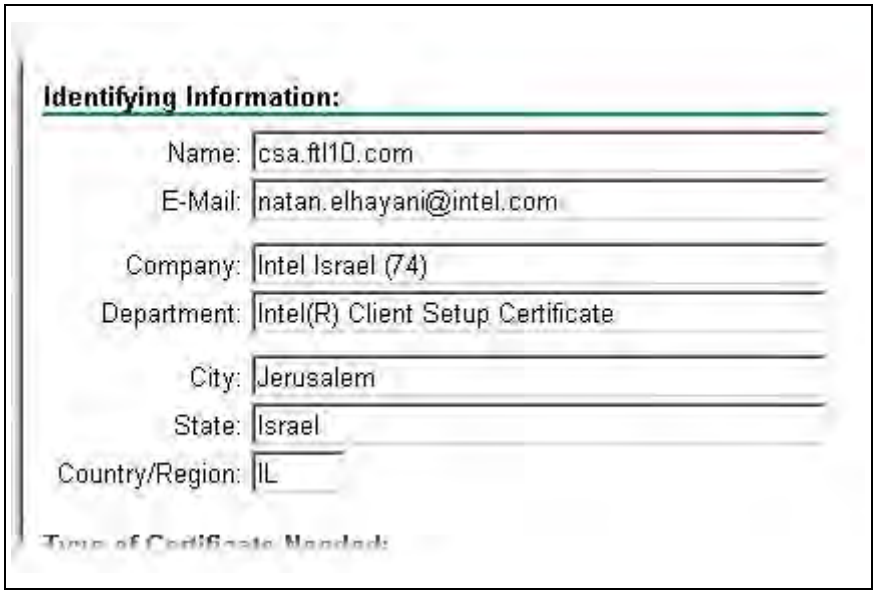
Name: the name of the entity this certificate intend for. In official documents it is called the CN field. Its suffix must match the one you configured in DHCP option 15.

E-Mail: the mail you gave when registering the domain (optional).

Company: the name of your company as registered in your country records.

Department: this is the OU field. For remote configuration it should be **"Intel® Client Setup Certificate".** String is case sensitive.

Complete rest of fields according to your locality.

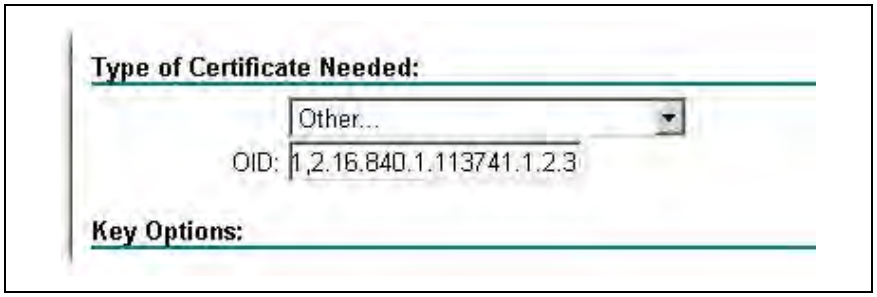


In **Type of Certificate Needed** section, choose **Other**...

In the OID field that will appear, enter: **1.3.6.1.5.5.7.3.1,**

2.16.840.1.113741.1.2.3

Both OIDs should be written one after the other with only a comma between them (and no spaces)



In the **Key Options** section, choose **Create new key set**. The example uses the default **CSP**.

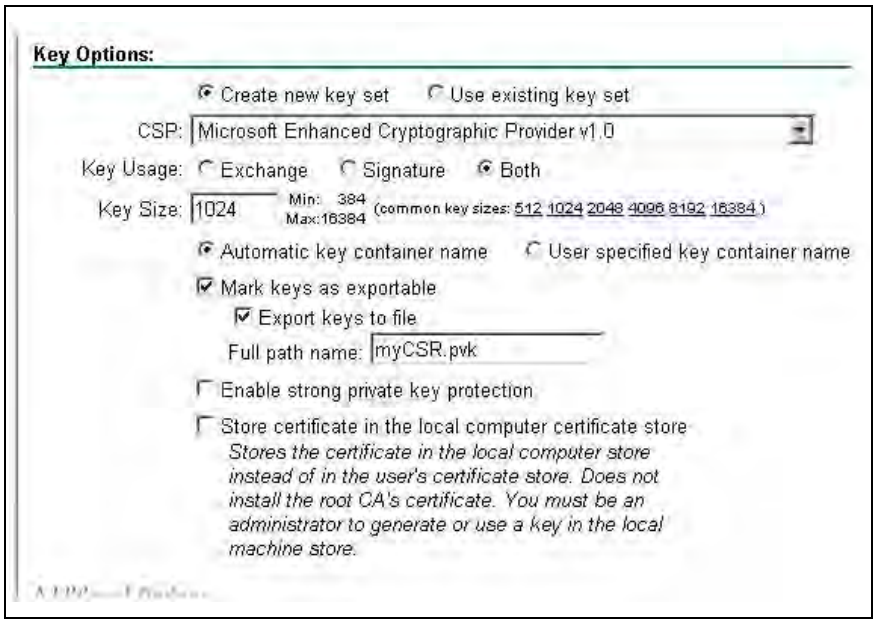
Select **Both** for the **Key Usage**.

In **Key Size,** choose the size of your keys. Intel® AMT supports 1024, 1536 and 2048.

Check the **Mark keys as exportable** checkbox. Check the **Export Keys to file** checkbox and enter a file name.

The procedure uses this file later for creating a .pfx file (a file that contains both private key and public certificate).

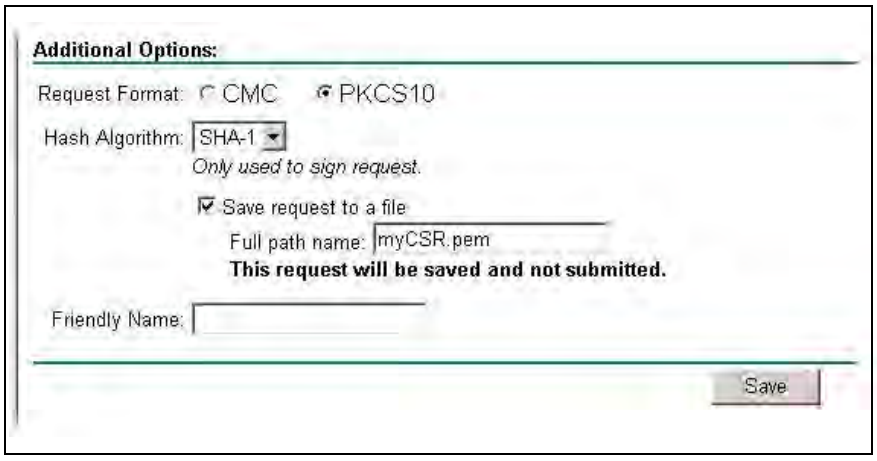
Note: saving your private key in an external file is a potential security hole. This should be done only for testing purposes, and not in a real environment.



In **Additional Options** choose PKCS10 as your Request Format.

Check the **Save request to a file** checkbox and give a name to your CSR.

Click **Save** at lower right corner of the page.



Intel Confidential

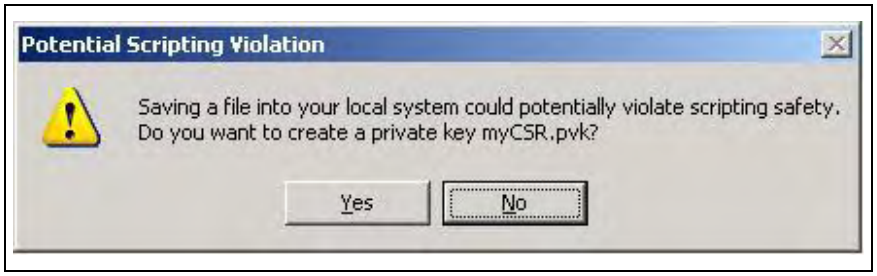
A warning pop-up will appear to ensure that you intend to create a certificate request.

Click **Yes** to continue.



Another warning will appear saying saving files to your local system is a scripting safety violation.

Click **Yes** to continue.



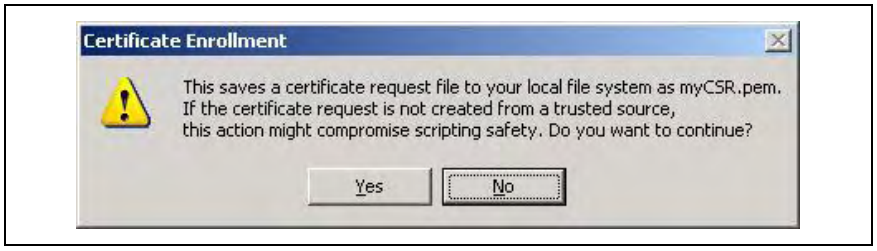
A pop-up will appear asking for a password to encrypt the private key. Supply a password and click **OK** to continue.

(It is advisable to supply a password, as it is more secure than just clicking the **None** to indicate no password is supplied.)



Another warning pop-up will appear, saying saving file to your local system compromises scripting safety.

Click **Yes** to continue.



A pop-up will appear to indicate the process completed.



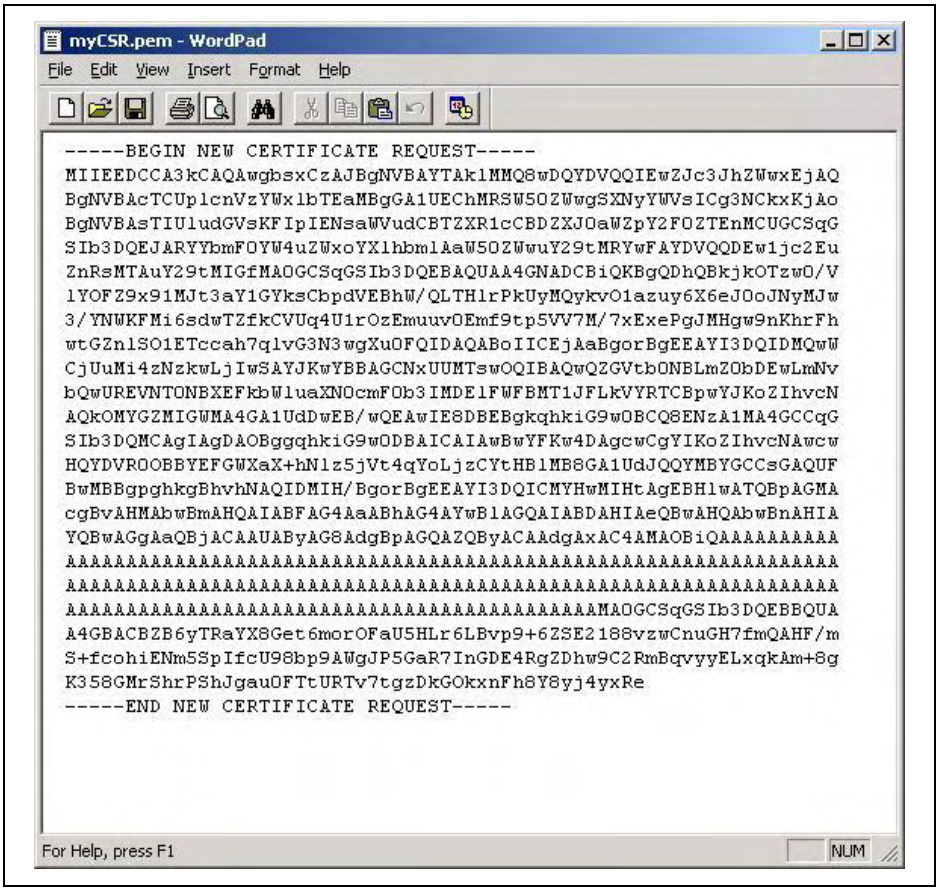
8.1.2 Acquiring Remote Configuration Certificate from a CA

Table 9. Steps for Acquiring Remote Configuration Certification

After the CSR was created it needs to be signed by a CA. The signing process might vary between the different CAs, but as part of signing process, user should submit its CSR.

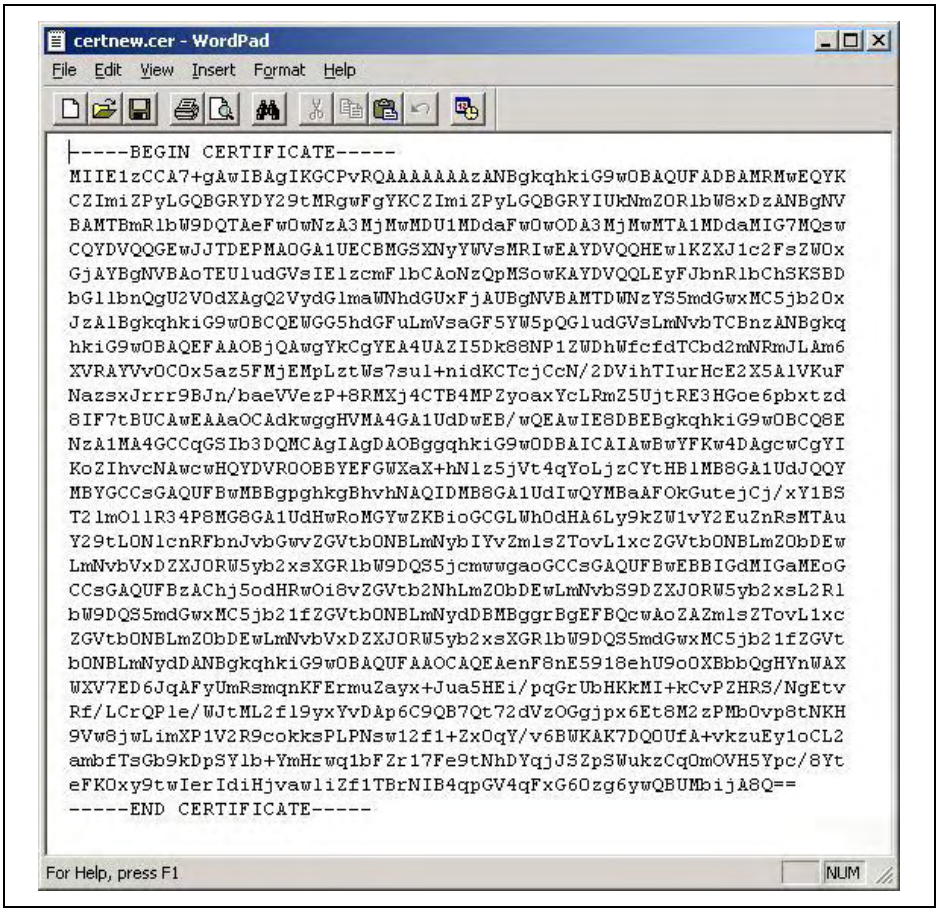
This is usually done by opening the CSR file created in the previous step, copying its contents and pasting it in the CA website.

The CA will process the CSR, verify needed data and, if everything is correct, it will sign the CSR.



Most CAs send the certificate by e-mail to the customer. In some cases, the certificate can be downloaded from the web-site of the CA.

IF the certificate is sent by e-mail, you should copy its contents from the e-mail and paste it in a file. Give it a .cer suffix.



8.1.3 Validating Certificate usability

Table 10. Steps Validating Certification Usability

Double click on the certificate file. Make sure the chain of trust is recognized by your platform before continuing to the next step. This is an example of a certificate with verification problems (root CA is not installed).	
The reason is the chain of trust is missing the signer of the certificate.	

The problem above was fixed by installing the Root CA certificate.



Now the chain of trust is presented correctly.



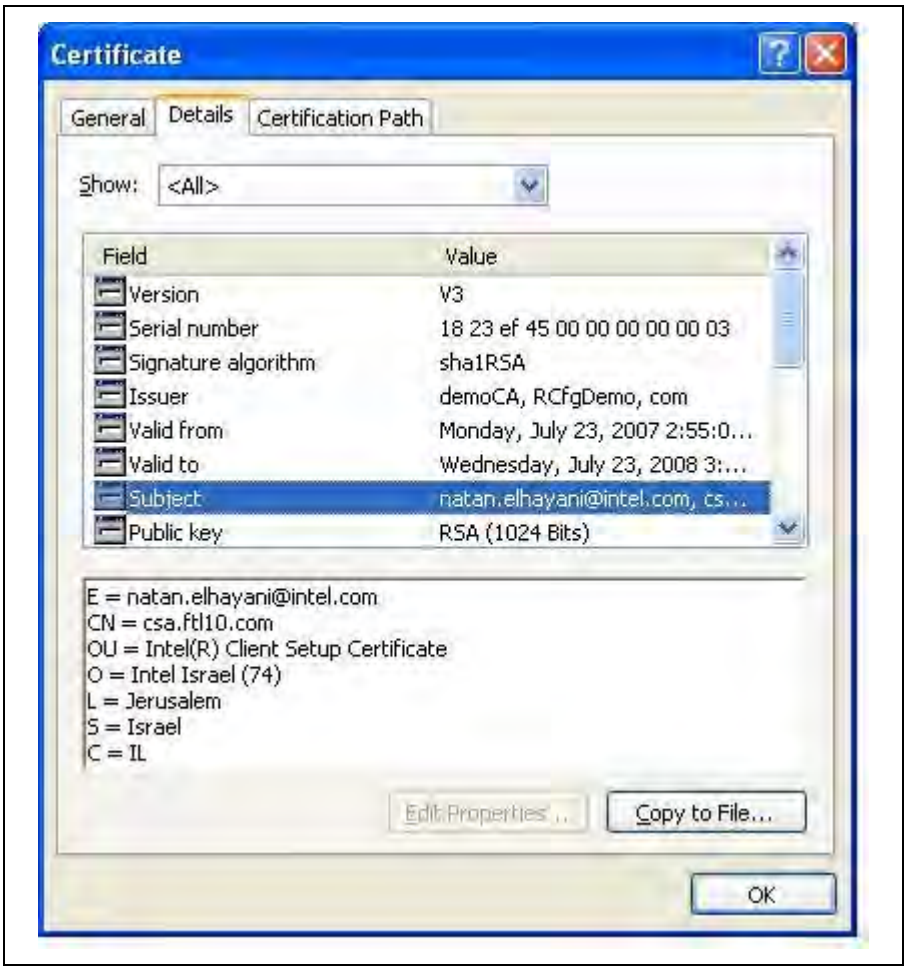
Validate that the required fields are set correctly:

Select **Details** tab.

Select **Subject** field.

Make sure the suffix of **CN** field matches your DHCP option 15 value.

Make sure the **OU** field is set to **Intel(R) Client Setup Certificate.**



Select the **Enhanced Key Usage** field.

Make sure **Server Authentication** is specified.

The **Intel® AMT Provisioning** is optional if **OU** was set correctly above and mandatory otherwise.

Make sure its OID is set correctly to **2.16.840.1.113741.1.2.3**

(Note: In most cases you will see only the OID value without the label.)



8.1.4 Converting Certificate to a format recognized by the Configuration Server

When a CSR was created using a Microsoft CA using the above procedure, the private key created with the CSR needs to be appended to the signed certificate. Follow the procedure using a Microsoft tool. The procedure assumes you have the pvk2pfx.exe tool. It is available as part of the Microsoft* Platform SDK; the latest version of the PSDK can be downloaded from this URL:
http://www.microsoft.com/downloads/details.aspx?FamilyId=484269E2-3B89-47E3-8EB7-1F2BE6D7123A&displaylang=en

The tool is also included with **Microsoft Visual Studio 2005.** It can be found at **"Microsoft Visual Studio 8\Common7\Tools\bin"**, under **Program Files** (unless Studio was installed at a different location).

OpenSSL is used later for conversion from PFX to PEM. It can be found as part of the Configuration Server application at **"CertGenerator\OpenSSL"** under the Configuration Server directory.

Table 11. Continued Process - Converting Certificate to a format recognized by the Configuration Server

Copy the .pvk file created in the previous procedure and the signed certificate to a folder that contains pvk2pfx tool. Open command line and run: **pvk2pfx** –pvk **pvk_file** –spc **cert_file** This will pop-up the following window.	
Enter the password you gave in the key creation step and click OK.	

The **Certificate Export Wizard** will appear.

Click **Next** > to continue.

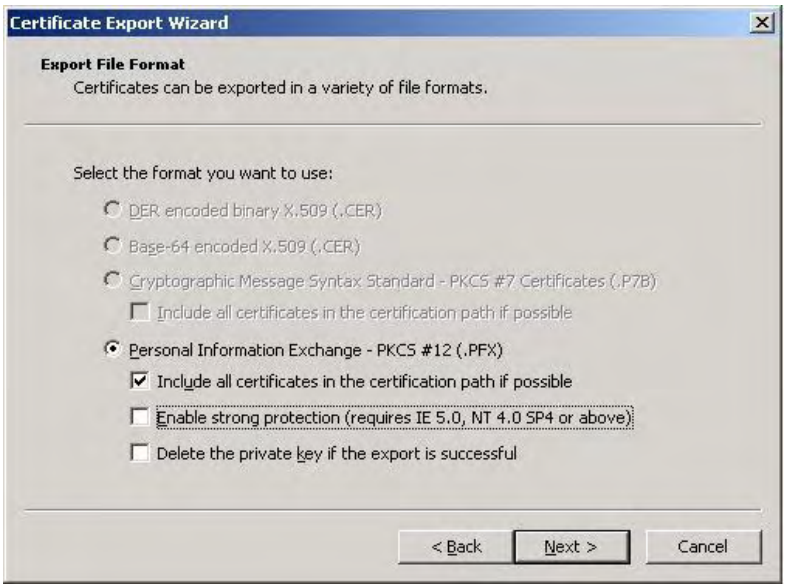


Check the **Yes, export the private key** and click **Next >** to continue.



Check the I**nclude all certificates in the certification path if possible.** Uncheck the **Enable strong protection.**

Click **Next** > to continue.



Enter a password for the pfx file.

Click **Next >** to continue.



Give a file name to your pfx.

Click **Next >** to continue.



Click **Finish** to continue.



A pop-up will indicate the process was completed. Click **OK** to close the pop-up.



8.1.5 Conversion from PFX to PEM Format

Table 12. PFX to PEM Format Conversion

Copy the pfx file to a folder with OpenSSL and run:

OpenSSL pkcs12 –in **pfx_file** –out **pem_file**
-nodes

You will be requested to enter the pfx password.

Once the process completes, you will get a message:

MAC verified OK



Open the PEM file in a text editor.

Make sure the chain of trust is in the right order.

Look at the **Bag Attributes** sections. (marked in yellow)

The private key bag should be first. Then the **Leaf** certificate (the certificate used for remote configuration).

These should be followed by all certificates up to the root in decreasing order.

The root certificate should come last. It is the only certificate where the **subject** value is identical to **issuer** value.



Appendix C describes an algorithm that verifies a chain of trust and supplies a Perl implementation.

Open a new text document. Copy the root certificate section (from the **Bag attributes** down to **-----END CERTIFICATE-----**, inclusive)

Paste it to the new document and save it as **RootCert.pem** (or any other name indicating this is the root certificate).

```
9VW8jWL1mAF1V2R9COKKSPLPNSW1ZI1+2XUQY/V6BWKAK/DQUUIA+VKZuLY1CL2
ambfTsGb9kDpSYlb+YmHrwq1bFZr17Fe9tNhDYqjJSZpSWukzCqOmOVH5Ypc/8Yt
eFKOxy9twIerIdiHjvawliZf1TBrNIB4qpGV4qFxG6Ozg6ywQBUMbijA8Q==
-----END CERTIFICATE-----
Bag Attributes: <Empty Attributes>
subject=/DC=com/DC=RCfgDemo/CN=demoCA
issuer=/DC=com/DC=RCfgDemo/CN=demoCA
-----BEGIN CERTIFICATE-----
MIIDzjCCAragAwIBAgIQUa8YYL+EdJpBiZVSoVd0jTANBgkqhkiG9w0BAQUFADBA
MRMwEQYKCZImiZPyLGQBGRYDY29tMRgwFgYKCZImiZPyLGQBGRYIUkNmZOR1bW8x
DzANBgNVBAMTBmR1bW9DQTAeFw0wNzA0MjkyMDIwNTRaFw0xMjA0MjkyMDI5ND1a
MEAxEzARBgoJkiaJk/IsZAEZFgNjb20xGDAWBgoJkiaJk/IsZAEZFghSQ2ZnRGVt
bzEPMA0GA1UEAxMGZGVtbONBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEA0ThB5OvAPfKbH+E5iI4/2Ywov2JYwHbFyeO+cOPv7DEisgOH/RzxBtXZhv05
VhhkfBYuvOQAQthXfyBWXxs1xoAE8wM4VIS2YYjuB7EDyHL3YBYSf2CRRn1Bf/rV
W6AMtLhsiaVcqHhj3C+oApDDmxtyT5ZZxwEsWM7hiWFjZPz3QiSQbCKOfE6anaHn
NnXRHgLwI4FA+6I8N31cVn16bnTpsewOVvzfc178xbWoxeujMOjjv8hXNmKu/b71
voJoVOi/sDiOJQXzes8r4CtVOB89RqSbE9W/Ydfp1gnxhB/uoCGkRxKGQxtYFNUa
XMZUIk13iLEZwbH3hQBNyuNOnQIDAQABo4HDMIHAMAsGA1UdDwQEAwIBhjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBTpBrrXowo/8WNQUk9pZjpZUd+D/DBvBgNV
HR8EaDBmMGSgYqBghi1odHRwOi8vZGVtb2NhLmZ0bDEwLmNvbS9DZXJ0RW5yb2xs
L2R1bW9DQS5jcmyGL2ZpbGU6Ly9cXGR1bW9DQS5mdGwxMC5jb21cQ2VydEVucm9s
bFxkZW1vQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEBBQUAA4IB
AQBRhez+s76FOZBjmNI/Un53SILEFzLqwu1WWDrxtcn3RwaCO2wbfB+NmF2pNWvo
OTF2ESyTFyvCovYvVYUNix/Mlnf/Mb6aiBtU1t2wv7Pcmb/eCO1h+Vhx5ZuByxCA
Is6M1B4ZMY3MZOQhtovMmFVIDYrS8YTmBAsmHKqBYVKutw+LJXvaI2rox6QWBc19
pUYeEuWeW6AOoeCva+NtE8uzuq1h9CHJp5IYSspxSFDBL3jKH9xo5gSFwn5RIYo/
/EV3R0t5gInUgASc2vSkQgVN6dJwET8WfkbIdnQFUhKjv5DkVg7r7fDgA7vMpxRP
qYf46oG8yO6hmiJTQS9uOtWa
-----END CERTIFICATE-----
```



8.1.6 Add Certificate to the Configuration Server Configuration File

Table 13. Add Certificate to Configuration Server config File

Open **default.conf.xml** (from **ConfigScripts** directory) in XML editor.

Go to **pki_configuration** section (it is at end of the file).

Edit **full_cert_chain_file** value to the full path of the PEM file, created in the conversion step.

Edit **root_cert_file** value to the full path of root certificate created in previous step.

```
<!-- The pki_configuration element is used when
     the AMT configuration mode is PKI.
     - full_cert_chain_file is the name of a file (full path) containing th
       client certificate chain including the root certificate
     - root_cert_file is the name of a file (full path) containing the root
       certificate
     - otp (optional) is the one time password. If provided the provisionin
       sample will verify the AMT devices otp.
     - new_mebx_password must be changed in order to complete the PKI provi
       process. The new password must meet the complex password requirement
       more details on strong password requirements, please refer to the
       "Network Interface Guide".
  -->
  <pki_configuration>
      <full_cert_chain_file>C:\RCFG cert\cert.pem</full_cert_chain_file>
      <root_cert_file>C:\RCFG cert\RootCert.pem</root_cert_file>
      <otp>password</otp>
      <new_mebx_password>Admin@98</new_mebx_password>
  </pki_configuration>
</config>
```

Set **new_mebx_password** value to a strong password. (This will be the local password after the configuration process completes successfully.)

If you intend to use OTP during configuration set the OTP value in **otp** field, otherwise, put this field in a XML remark.

Save changes.

§





9 Appendix B – Generating a Remote Configuration Certificate – using IIS

9.1 Creating a server certificate using IIS

This procedure describes the simplest method for creating a certificate; however, there are some limitations when issuing certificates using IIS.

There is no option to use the remote configuration special OID; therefore the predefined OU **must** be used.

There is no way to produce a certificate with multiple values for the CN.

Note also that IIS is not a CA, so you must issue the CSR externally.

9.1.1 Creating a CSR

1. Open IIS.
(found at Start→ Programs → Administrative Tools →
Internet Information Services (IIS) Manager)

2. In the left pane, expand **Web Sites**

3. Right click on **Default Web Site** and choose **Properties**.

4. Select **Directory Security** tab.

5. In the bottom there is a section called Secure Communications. Select Server Certificate. This will pop up the Web Server Certificate Wizard.

(Within the following steps, (N) indicates you should select Next to proceed.)

6. Choose **Create a new certificate. (N)**

7. Choose **Prepare the request now, but send it later. (N)**

8. The **Name** field is the “Friendly name” of the certificate (it is not the CN). Enter any value that will help you to identify this certificate later.

9. The **Bit Length** is the length of the encryption key. You can leave it at the default value (1024). **(N)**

10. Put any value in **Organization** field or leave it empty.

11. Put **"Intel® Client Setup Certificate"** in the Organizational unit field. (in some versions is it called **Department**.) **(N)**

12. The Common name is the CN field. You can enter the FQDN of your Configuration Server platform (e.g. csa.ftl10.com) or a string with a wildcard (e.g. *.ftl10.com). Make sure you use the correct DNS Suffix. **(N)**

13. Fill in additional fields in order to proceed. **(N)**

14. Give a file name and location for your request. **(N)**

15. Verify again that all needed fields have the correct values. Select Next and Finish.

9.1.2 Exporting Certificate

Issue the certificate by using either an Internal or External CA. To issue using an Internal CA, see below.

Once you have the issued certificate, repeat steps (1-5)

1. After entering the wizard again, choose **Process the pending request and install the certificate. (N)**

2. Locate the signed certificate. **(N)**

3. In the next screens select **(N)**, Finish.

4. In **Secure communications**, choose **View Certificate.**

5. Select **Detail** tab.

6. Click **Copy to file** at right bottom of window, the **Certificate Export** wizard will pop up. **(N)**

7. Choose Yes, export the private key. **(N)**

8. Mark **Include all certificates in the certification path if possible. (N)**

9. Enter a password (can be weak password) and confirm. **(N)**

10. Give location and file name for the resulting PFX. **(N) –** Finish - **OK**.

11. Close all windows.

See the earlier section on how to convert a PFX to PEM.



9.2 Creating server certificate using Enterprise CA (running on Windows* 2003 Server)

In order to issue certificates using an Enterprise CA, a corresponding Certificate Template needs to be defined. The user cannot use a regular server certificate since its private key is non exportable. Furthermore, in order to use a designated OID, one must define a template that does this. Consult Appendix A for creating needed data.

9.2.1 Creating CSR

1. On console platform, contact **//CAname/certsrv**.
2. Choose **request a certificate.**
3. Choose **advanced certificate request.**
4. Choose **Create and Submit a request to this CA.**
5. Choose the appropriate certificate template.
6. Fill in needed fields -
   a. For specified OU, fill the Department field with the pre-defined string **Intel® Client Setup Certificate.**
7. Check **mark keys as exportable.**
8. Choose **request format** to be **PKCS10.**
9. Check **save request to file**, give file name.
10. Click **submit -**
    A few warning messages will appear saying that by doing so you might compromise on security.

Note: A CSR generated by an Enterprise CA, will contain the certificate template that will be used for issuance, hence it must be issued by a CA familiar with the template that was used. (Some CAs might give an error for such a CSR.)

9.2.2 Issuing Certificate using the Internal CA

1. Do steps (1-3)
2. If you don't have a CSR

a. Do steps (4-7)

If you already have a CSR

a. Choose
Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file.

b. Copy / Paste your CSR content to Saved Request window

3. Click - submit.

4. If the CA is not configured to issue certificates automatically, do the following on the CA platform:

   a. Open **Certification Authority** application.

   b. Go to **Pending Request.**

   c. Right click on your request and choose **issue.**

5. Go to CA start page (//CAname/certsrv).

6. Choose **View the status of a pending certificate request.**

7. Choose the appropriate certificate.

8. Select **Base 64 Encoded.**

9. Choose **Download certificate.**

10. Download the certificate to your computer.

9.2.3 Exporting Certificate

1. Install the signed certificate. (Note: if you created the CSR using CertSrv make sure you install the certificate in the platform where you issued the request.)

2. Go to the certificate store
   (in Internet Explorer, go to Tools → Internet options → Content → Certificates)

3. Open the certificate for viewing.

4. At bottom of window a key symbol should appear saying you have the private key of this certificate.

5. Go to **Details** tab, click **Copy to file** and export the certificate including private key and all certificates in path to a .pfx file.



9.3 Creating server certificate using Stand Alone CA (running on Windows Server* 2003)

9.3.1 Creating CSR

1. On console platform, contact **//CAname/certsrv**
2. Choose **request a certificate.**
3. Choose **advanced certificate request.**
4. Choose **Create and Submit a request to this CA.**
5. Fill in needed fields
    a. For specified OU, fill the **Department** field with pre-defined string: **Intel® Client Setup Certificate.**
    b. For specified OID, choose **Type of certificate** needed to be **Other** - then fill in both **Server Certificate** OID and **Intel® AMT configuration** OID: **1.3.6.1.5.5.7.3.1,2.16.840.1.113741.1.2.3**
6. Check **mark keys as exportable.**
7. Choose **request format** to be **PKCS10**.
8. Check **save request to file**, give file name.
9. Click – **submit -** A few warning messages will appear saying that by doing so you might compromise on security.

9.3.2 Issuing Certificate using the Internal CA

1. Do steps (1-7)
2. Click - **submit.**
3. On the CA platform:
    a. Open Certification Authority application.
    b. Go to Pending Request
    c. Right click on your request and choose issue.
4. On console platform:
    a. Go to CA start page **(//CAname/certsrv)**
    b. Choose **View the status of a pending certificate request**

c. Choose the appropriate certificate

d. Select Base 64 Encoded

e. Choose **Download certificate**

f. Download the certificate to your computer

9.3.3 Exporting Certificate

1. Install the signed certificate. (Note: if you created the CSR using CertSrv make sure you install the certificate in the platform where you issued the request.)
2. Go to the certificate store
   (in Internet Explorer, go to Tools → Internet options → Content → Certificates)
3. Open the certificate for viewing
4. At bottom of window there should appear a key symbol saying you have the private key of this certificate.
5. Go to Details tab, click Copy to file and export the certificate including private key and all certificates in path to a .pfx file.

9.4 Creating server certificate using OpenSSL

OpenSSL is a free tool that supports full PKI functionality. Configuration Server sample code in SDK and the FW kit, uses this tool for crypto services. One should use this tool with great care as OpenSSL offer no means of security for its sensitive data. (For example: Private keys are saved in the clear and not in a secured store.

9.4.1 Creating CSR

Prepare a configuration file for your certificate request. Here is an example:

```
# SSLeay example configuration file.
# This request is used for generation of
# Remote Cfg certificate requests.

RANDFILE= ./.rnd

#################################################################
###
[ req ]
default_bits       = 1024
default_keyfile    = keySS.pem
```



```
distinguished_name  = req_distinguished_name
oid_section         = Amt_OID
encrypt_rsa_key     = no
default_md          = sha1
prompt                          = no
req_extensions                  = v3_ztc

[ req_distinguished_name ]
C = IL
ST = Israel
L = Israel
O = Intel Israel 74
OU = Intel(R) Client Setup Certificate
CN = csa.ftl10.com

[ v3_ztc ]
basicConstraints=CA:FALSE
keyUsage=digitalSignature
extendedKeyUsage=critical,serverAuth,2.16.840.1.113741.1.2.
3
subjectKeyIdentifier=hash

[ Amt_OID ]
amtConfiguration=2.16.840.1.113741.1.2.3
```

Issue the following command (it assumes the above file was named ztc.cfg):

OpenSSL req -config ztc.cfg -**new -keyout** ztckey.pem **–out** ztcreq.pem **–days** 365

This will create a CSR in Base64 PEM file format and its corresponding private key. The CSR will be valid for 365 days.

You can now issue this CSR using an external CA or create a CA using OpenSSL and sign the CSR with it.

9.4.2 Issuing CSR using OpenSSL CA

- **Create a CA using OpenSSL**
  Prepare a configuration file for your CA, see the following example:

```
#
# SSLeay example configuration file.
#

RANDFILE            = ./.rnd

####################################################################
##
[ req ]
default_bits        = 2048
```

```
default_keyfile     = keySS.pem
distinguished_name  = req_distinguished_name
encrypt_rsa_key     = no
default_md          = sha1
x509_extensions     = ext_ca

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = IL
countryName_value           = IL

organizationName            = Organization Name (eg,
company)
organizationName_value      = Intel Corporation

commonName                  = Common Name (eg, YOUR name)
commonName_value            = Demo Root CA

[ ext_ca ]
basicConstraints=CA:TRUE
keyUsage=digitalSignature,keyCertSign,cRLSign,keyEncipherme
nt,dataEncipherment
```

Assuming the above file is named **rootCA.cfg**, issue the following commands (can be done in a batch file):

```
set CATOP=.\rootCA

mkdir %CATOP%
mkdir %CATOP%\certs
mkdir %CATOP%\crl
mkdir %CATOP%\newcerts
mkdir %CATOP%\private
echo 01 > %CATOP%\serial
copy nul %CATOP%\index.txt

openssl req –config rootCA.cfg -new -x509 -keyout
%CATOP%\private\cakey.pem -out %CATOP%\cacert.pem -days 365
```

- **Issue the CSR using the created CA**

Create a configuration file for issuance with the above CA

```
#
# OpenSSL example configuration file.
#

HOME= .
RANDFILE     = $ENV::HOME/.rnd
CADIR        = ./rootCA

# Extra OBJECT IDENTIFIER info:
oid_section = new_oids
```



```
extensions = ext_ztc

[ new_oids ]
ztc_Configuration = 2.16.840.1.113741.1.2.3

###################################################################
##
[ ca ]
default_ca = CA_default     # The default ca section

###################################################################
##
[ CA_default ]

dir = $CADIR        # Where everything is kept
certs = $dir/certs # Where the issued certs are kept
crl_dir = $dir/crl # Where the issued crl are kept
database     = $dir/index.txt                # database
index file.
new_certs_dir       = $dir/newcerts           # default
place for new
                                               # certs.

certificate  = $dir/cacert.pem                # The CA
certificate
serial              = $dir/serial             # The current
serial number
crl          = $dir/crl.pem # The current CRL
private_key  = $dir/private/cakey.pem # The private key
RANDFILE     = $dir/private/.rand             # private
random number file

x509_extensions = ext_ztc   # The extentions to add to the
cert

name_opt     = ca_default   # Subject Name options
cert_opt     = ca_default   # Certificate field options

default_days = 365          # how long to certify for
default_crl_days= 30        # how long before next CRL
default_md   = sha1 # which md to use.
preserve     = no           # keep passed DN ordering

policy       = policy_anything

[ policy_anything ]
countryName         = optional
stateOrProvinceName= optional
localityName        = optional
organizationName    = optional
organizationalUnitName      = optional
commonName          = supplied
emailAddress        = optional

###################################################################
###
[ ext_ztc ]
```

```
basicConstraints=CA:FALSE
keyUsage=digitalSignature
extendedKeyUsage=critical,serverAuth,2.16.840.1.113741.1.2.
3
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
crlDistributionPoints = URI:http://crl.meca.me.co.il
```

Assuming the above example is named **ztcSign.cfg**, issue the following command:

OpenSSL ca -config ztcSign.cfg **-policy** policy_anything **-out** ztccert.pem **-infiles** ztcreq.pem

Answer Y when prompted.

9.4.3 Creating PFX file from Certificate and private key

After issuing the CSR, one needs to create a PFX from the private key and the issued certificate. Run the following command:

OpenSSL pkcs12 -export -in ztccert.pem -inkey ztckey.pem **-out** ztccert.p12 **-name** "Intel ZTC certificate" **-password "pass**:q"

Note: This PFX has only the client certificate. For remote configuration to work correctly all certificates in the chain of trust should be included as well.

You can include other certificates in PFX creation using "**-certfile** certificate_to_add" for every certificate in the above command.

There is a "simple" trick though. Assuming all other certificates are installed in the Certificate Store, we just install the PFX, then export it back from the certificate Store, marking the **"Yes, export the private key"** and checking the **"Include all certificates in the certification path if possible"** checkbox. Then save the file to a new PFX.

9.5 Converting PFX to PEM (to be used with Configuration Server)

1. Copy PFX file to a folder contains OpenSSL.
2. Open Command Prompt (Start → Run → cmd).



3. navigate to directory in step (1)

4. execute the following:
   OpenSSL pkcs12 –in <pfx file> -out <result pem file> -nodes

5. You will be prompted to supply the password you gave when exporting the certificate to PFX.

§





10 Appendix C: Creating Certificate Template for Remote Configuration in Microsoft Server* CA

Note: customized certificates are supported in Microsoft Server* 2003 Enterprise Edition. In Standard edition you will be able to create a template, but you won't be able to issue it.

1. Open **Certification Authority** application
   (Start → Programs → Administrative Tools → Certification Authority).
2. Expand your CA on left pane.
3. Right click **Certificate Templates** and select **Manage**. This will open the **certtmpl** application.
4. On right pane, highlight **Web Server** certificate template.
5. Right click the selected template and choose **Duplicate Template.**
6. In **General** tab, give a display name to your new template (e.g. RCfg Certificate Template).
7. In **Request Handling** tab, mark **Allow private key to be exported** checkbox.
8. In **Subject Name** tab, make sure that **Supply in the request** is selected.
9. In **Extension** tab, mark **Application Policies** and click on **Edit.**
10. A new window has opened, click on **Add**.
11. A new window has opened, click on **New**.
12. Give a name to the new Policy, e.g. **RCfg special OID**
13. In Object identifier, write the special OID:
    2.16.840.1.113741.1.2.3

14. Click **OK** to close this window.
15. Select the newly created **Application Policy** (e.g. remote configuration special OID) and click **OK.**
16. Now you will have two Application Policies listed; Server Authentication and the one you just created. **Click OK**.
17. In **Security** tab, select **Authenticated Users**. Check the **Enroll** and **Auto-enroll** checkboxes.
18. Click **OK**. Close **certtmpl** application.
19. Back in **Certification Authority** application, right click on **Certificate Templates**, choose **New → Certificate Template to Issue.**
20. Choose the certificate template you just created (e.g. RCfg Certificate Template) and click **OK.**
21. The new template should appear in the right pane.
22. Restart your CA.

After restarting your CA, you will be able to issue RCfg certificates with **Certsrv** web applet as described in **Creating server certificate using Enterprise CA** by selecting the template you have created in the **Certificate Template** drop down list.

§



11 Appendix D: Fixing chain of trust in converted PEM files

There is a bug in OpenSSL that causes the chain of trust in PFX files converted to PEM files to be disordered when using chains that are size 3 and above.

The following algorithm describes how to fix this problem.

Open the result PEM file, and validate that Chain of trust is in the right order.

Every section in the file begins with Bag attribute and followed by a **-----BEGIN <SOMETHING>-----** and ends by the corresponding **-----END <SAME THING>-----**

Sections in file should be as follows:

a. First there is a section of private key

b. Next there is the client certificate section

c. Then there are zero or more sections of subordinate CA. Every additional certificate in the file, has signed the one that precedes it, and is signed by the one after it.

d. The last certificate in the file is the CA root certificate (it is the only certificate that signed itself)

§





12 Appendix E: *.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:

Figure 8. XML 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: there must be a space before the command value.

Table 14. Variable Name Table / Allowed Settings / Usage

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 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 hostname 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)



Variable Name	Allowed Settings	Usage
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	<tls_options> <local>ServerAuthentication </local> <remote>MutualAuthentication </remote> </tls_options> 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	<tls_cert> <mode>GenerateCertificate </mode> </tls_cert> <tls_cert>	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
	<mode>FileCertificate </mode> <cert_chain_file>bla.raw </cert_chain_file> <key_file>bla.key</key_file> </tls_cert> <tls_cert> <mode>NoCertificate </mode> </tls_cert>	
cert_store	<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>	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
tls_cert_name	<tls_cert_name>newcert.pem </tls_cert_name>	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



Variable Name	Allowed Settings	Usage
		certificate.
wired_8021x_ profile	<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>	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 *Network Interface Guide* and the default.conf.xml file for detailed profile parameter descriptions. The following parameters are included in one or more of the profile types: server_id_cert_issuer server_cert_name server_cert_option username password domain_name protected_access_cred protected_access_cred_pwd client_certificate
wireless_8021x_ profile	<wireless_profiles> <profile> <profile_priority>N </profile_priority> <security_type> ProfileSecuritySettingWPAType or ProfileSecuritySettingRSNType </security_type> <encryption_type> DataEncryptionTKIPType or	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 *Network Interface Guide*. 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.

Variable Name	Allowed Settings	Usage
	DataEncryptionCCMPType </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>	
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	<power_package> <guid>XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX </guid> </power_package>	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 *Network Interface Guide* for power package details.
pki_configuration	<pki_configuration> <full_cert_chain_file> path to PEM file </full_cert_chain_file> <root_cert_file> path to root certificate pem file	The PKI configuration parameters are required when the Intel® AMT device to be set up is 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



Variable Name	Allowed Settings	Usage
	</root_cert_file> <otp>password</otp> <new_mebx_password> replacement mebx password </new_mebx_password > </ pki_configuration>	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 SCA 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. 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. new_mebx_password is a strong password to replace the Intel® 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.
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	<trusted_root_certificates> <file>trusted_cert.pem</file> </trusted_root_certificates>	One or more trusted root certificate files in pem format. They must reside in "..\Bin\CertGenerator\ExternalSecScripts\trusted_rootCA" The supplied default configuration points to the certificate generated automatically the first time that the SCA is executed.
crls	<crls> <crl> <url>---url of CDP distribution	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

Variable Name	Allowed Settings	Usage
	point --</url> <serials <serial>--certificate serial number--</serial> <serial>>--certificate serial number--</serial> </serials> </crl> </crls>	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. Each crl entry has a url and a serials section. Each serials section has one or more serial numbers. This is an optional entry.
trusted_fqdn_cn	<trusted_fqdn_cn> <fqdnsuffix>intel.com </fqdnsuffix> </trusted_fqdn_cn>	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	<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>	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).



Variable Name	Allowed Settings	Usage
	</acl> </acls> </kerberos>	
set_enabled_ interfaces	<set_enabled_interfaces> <interface>WebUI</interface> <interface>SerialOverLAN </interface> <interface>IdeRedirection </interface> </set_enabled_interfaces>	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	<power_options> <power_state>S-state </power_state> <wake_on_net_access_ sleep_timer>t </wake_on_net_access_ sleep_timer> </power_options>	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.
digest_acls	<digest_acls> <acl> <access>network</access> <user>username</user> <password>pw</password> <realms> <realm>10</realm> </realm>	Allows creation of Digest ACL entries. Intended for creation of ACL entries that will have auditor access, but this construct can be used for any digest ACL entry. Each entry lists the access option (local, network or any), the username and password (the password should be a strong password) and the realms to which the user has permission. The realm numbers are based on the UserAclRealmType as defined in the Network Interface Guide.

Variable Name	Allowed Settings	Usage
	</acl> <digest_acls>	
audit	<audit> <enable>yes</enable> <key_file>.\CertGenerator\ SecScripts\AuditCert\Auditkey.pem </key_file> <certs> <cert_file> path_to_Auditcert.pem </cert_file> <cert_file> Path_to_ subcacert.pem </cert_file> </certs> <sign_mech>SHA1</sign_mech> <policies> <policy> <appid>16</appid> <eventid>2</eventid> <critical>yes</critical> </policy> <policy> <appid>16</appid> <eventid>4</eventid> <critical>no</critical> </policy>	Enables the audit log mechanism. Identifies the key file and certificates used for credentials to access the audit log. Identifies the signing mechanism (hash algorithm). The policy entries identify auditable events, as defined in the table in the Network Interface Guide. Each event has an application ID, an event ID within the application and a criticality indication (either critical or not critical).



Variable Name	Allowed Settings	Usage
	</policies> </audit>	

Variable Name	Allowed Settings	Usage
Client Initiated Remote Access	<cira> <mp_servers> <mp_server> <address> <ip_address> <ip> <ipv4>10.0.0.100</ipv4> </ip> <cn>MP Server 1</cn> </ip_address> Or <fqdn>pltfrm.loc.com</fqdn> </address> <port>12345</port> <authentication> <username_password> <username>mpsuser</username> <password>Admin!98</password> </username_password> OR <mutual_auth> <tls_cert_handle>n</tls_cert_handle> </mutual_auth> </authentication> </mp_server>	The cira parameters are divided into three groups: management presence server (MPS) definitions, policy definitions, and interface enablement rules. mp server definitions include: server address: an IP address and trusted common Name (CN) or an fqdn port used to establish tunnels with the MPS Authentication credentials Username and password or A handle to a certificate already stored in the Intel® AMT device certificate store (used for mutual authentication) Policies determine when the Intel® AMT device connects to an MP server. There is a selection of a trigger (due to an alert, a periodic connection, or user initiated), extended data (the time in seconds between periodic connections), the tunnel lifetime (how long in seconds that the tunnel is maintained with the MPS), and up to two MP server address and port pairs. In Intel® AMT Release 5.0, a periodic policy can be based on the time of day in hours (0 to 24) and minutes in the hour> enable_interfaces enables or disables interfaces that can be used for a user-initiated connection. The possible “interfaces” are an agent in the BIOS and an agent running on the host under the OS.



Variable Name	Allowed Settings	Usage
	</mp_servers> <cira_policies> <cira_policy> <trigger>PolicyTriggerUserInitiated </trigger> <extended_data>n</extended_data> <tunnel_life_time>300 </tunnel_life_time> <mps_address_port>10.0.0.100:12345</mps_address_port> </cira_policy> <cira_policy> <trigger>PolicyTriggerPeriodic </trigger> <extended_data> <hour_of_day>10</hour_of_day> <minutes_of_hour>11</minutes_of_hour> </extended_data> </cira_policy> </cira_policies> <enable_interfaces> <enable_interface> <source>UserInitSourceOsAgent </source>	

Variable Name	Allowed Settings	Usage
	<is_enabled>yes</is_enabled> </enable_interface> </enable_interfaces> </cira>	



Variable Name	Allowed Settings	Usage
environment_ detection	<environment_detection> <local_domain>firstdomain.com </local_domain> <local_domain>nextdomain.com </local_domain> </environment_detection>	List of domains that define “inside the enterprise” to the Intel® AMT device. Required by CIRA to determine if a remote connection should be attempted. The SCA enables environment detection only when this parameter is defined.

§





13 Appendix F: PSK.REPOSITORY.XML File Format

The file is located under the Bin\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.

Table 15. Variable Name Table / Allowed Settings / Usage …cont.

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

flexbuf.cpp source files, described above.

§