Cisco Secure Services Client Administrator Guide

Software Release 4.2.0

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-13519-01

Cisco Secure Services Client Administrator Guide, Release 4.2.0

CONTENTS

Preface

The preface provides an overview of the *Cisco Secure Services Client Administrator Guide*, references related publications, and explains how to obtain other documentation and technical assistance.

The following topics are covered in this section:

Audience and Scope

This publication is for system and IT administrators responsible for configuring and deploying a derived, end-user version of Cisco Secure Services Clients (SSCs) in multiple end-user machines used by your various enterprise departments/organizations. By using the information supplied in this document, you will be able to fully define and customize the following for the end-user machines that you support:

- Policy—Defines the capabilities and user experience of the deployed SSC.
- Networks—Defines the configuration of all enterprise network connections that you control.

Organization

This guide contains the following sections:

Chapter 1, “Enterprise Deployment,” provides instructions for deploying a preconfigured end-user SSC.

Chapter 2, “Schema Elements,” describes the usage and properties of each of the configurable entities found in the SSC’s distribution package XML schema.

Appendix A, “Network Decision Tree Flow Diagram,” contains a high level flow diagram of the decision tree for configuring a network.

Appendix B, “Distribution Package Examples,” contains examples of valid distribution package (configuration) .xml files for different enterprise environments.

Appendix C, “Postprocessing Verification Errors,” contains a listing of error types and error messages used with the postprocessing utility.

Conventions

This publication uses the following conventions to convey instructions and information:

- For utility commands
  - Commands are in **boldface** type.
  - Variables are in *italic* type.
- For schema objects.
  - Element and attribute names when used in the text are in *italic* type.
- Notes use the following conventions and symbols:

Note Means *reader take note*. Notes contain addition information for the subject at hand or references to materials not contained in this manual.

Tip Tips contain helpful suggestions.

Related Publications

For more information about Cisco Secure Services Client, refer to these publications:

- *Cisco Secure Services Client User Guide*—Provides detailed information on operating SSC and configuring it from the local user interface.
- *Cisco Secure Services Client Release Notes*—Describes new features and the open and resolved caveats in each SSC release.

You can find these Cisco SSC technical documents at this URL:

http://www.cisco.com/en/US/products/ps7034/tsd_products_support_series_home.html

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New* in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

CHAPTER 1

Enterprise Deployment

This chapter contains the following sections:

Introduction

The Cisco Secure Services Client (SSC) is an 802.1X authentication supplicant for creating secure wired and wireless connections. SSC also has a user interface for displaying status and accepting commands from a user. It allows your computer to connect to a network that is protected by the IEEE 802.1X security protocol. Only after successful client-server authentication will the port access control on the 802.1X-enabled access device (the wireless access point or the wired Ethernet switch) allow end-user connectivity to the network.

SSC has two basic versions:

- The out-of-the-box version

  SSC as downloaded from cisco.com is not configured. It is intended for use by an IT organization that is responsible for configuring and deploying a derived, end-user version. This deployed version is appropriate for use by the various enterprise departments and organizations that you support. As the IT Administrator you have control over the user experience and the end-user's allowed choices and configuration options. The out-of-the-box version has a fully open policy that allows access to most features and requires configuring a network when initially started. However, only through a deployed distribution package file, that is, a SSC configuration file, does the IT Administrator have full access to all settings and network configurations.

  There are two configurations for the out-of-the-box version:

  - Default download package—configured with a nonexpiring, wired-only license.
  - Re-licensed package—adds a trial, wireless license.
    (Also available for download from cisco.com on the SSC page.)
    See "Activating the Client" in the companion *Cisco Secure Services Client User Guide*.

    Once activated with the wireless trial license, you are able to:

(1) Evaluate wireless functionality for 90 days, via the temporary license.

(2) Permanently license the product for both wired and wireless functionality.

- The deployed end-user version

  The deployed end-user version is pre-configured with a distribution package description, possibly with a restricted feature set, and deployed by you the IT/System Administrator. It most likely contains one or more pre-defined enterprise networks that allow instant connection to your enterprise networks. Two types of end-user interfaces are available as follows:

  – The configurable end-user version

    This version allows your end-users to create new network profiles within the scope of your policy. It is an excellent choice for end-stations that will move out of the enterprise network to home or travel networks.

  – The preset end-user version

    This version contains only your pre-defined network profiles that allow instant connection to your enterprise networks. It is an excellent choice for end-stations that will encounter only enterprise networks that you control.

Note The out-of-the-box default wired SSC supports:

- Wired (802.3) network adapters
- EAP methods: EAP-FAST, EAP-MSCHAPv2, EAP-GTC, EAP-TLS
- Smartcard provided credentials
- Cisco Trust Agent (CTA) processing when CTA is also installed

The trial license adds support for:

- Wireless (802.11) network adapters
- WPA2/802.11i protocols
- Additional EAP methods: LEAP, EAP-PEAP, EAP-TTLS, EAP-MD5

Supported Operating System Environments

The supported operating system environments are:

- XP Professional (SP1, SP2), 2K (SP4), Win2K Servers (SP4), Win2003 Server
- Novell Client version 4.91 SP1 with Hotfix TID2972711

Note By omission, other editions of Windows XP such as Home, Media Center, Tablet PC, Professional x64 and so forth, are not supported.

Distribution Package

The distribution package defines how an individual end-user SSC operates and creates connections. A distribution package contains the following functional blocks:

- License

  The deployed end-user SSC may initially require the enterprise license that you obtained from Cisco Systems. This will replace the wired-only license built into the out-of-the-box version.

- Policy
  - User control policy

    Sets the deployed type and network media support.
  - Network policy

    Sets the limitations on the types and capabilities of all supported networks.

- Station Settings

  Configures the global operational aspects of making network connections.

- Networks

  Contains a single or a set of network profile descriptions. A network profile defines the specific properties and operational behavior of a single network. This profile includes the following characteristics:
  - The user-friendly name of the network.
  - Network access media (wired, Wi-Fi) and adapter details used for the network connection.
  - Definition of the security class (open, shared key, authenticating) of the network.
  - Definition of the connection context (machine only, user only, machine and user) for the network.
  - Wi-Fi Association and Encryption method (Wi-Fi network).
  - Authentication methods supported and properties (authenticating network).
  - Static keys, if applicable (non-authenticating network).
  - Definition of types and source of credentials (authenticating network).
  - Definition of trusted servers (authenticating network) and support for deploying Certificate Authority (CA) certificates and manual provisioning of EAP-FAST Protected Access Credentials (PACs).

  Networks defined as part of the distribution package are locked; that is, the end-user is not able to edit the configuration settings.

The major steps that must take place to tailor the SSC to the desired enterprise environment are:

1. Creation—The administrator creates a distribution package file. A single distribution package file may contain configuration descriptions for more than one network. See "Distribution Package Creation" for complete details on the format, structure and contents of the distribution package.
2. Deployment—The administrator packages the application and/or the distribution package file and deploys to the end station. See section "Distribution Package Deployment" for details on deployment options and instructions.

3. Introduction—The SSC detects and uses the distribution package file. This step is automatic and does not require any administrator intervention. Shortly after the deployment step, the existence of the new distribution package file is detected. It is then processed for validity and, if valid, the SSC reconfigures itself accordingly.

Distribution Package Utilities

All of the utility tools and support files needed for the creation and deployment of a distribution package are contained in a single packaged file, SSCMgmtToolkit_{release}.zip. The individual items are introduced and described in the remainder of this chapter.

You can download the utility package online at the Cisco SSC download page. Go to SSC product support at:

http://www.cisco.com/en/US/products/ps7034/tsd_products_support_series_home.html

and click on **Download Software** and follow the **wireless software** links to the SSC download page.

Distribution Package Creation

Distribution Package Schema

SSC utilizes the XML format for the distribution package file. The overall structure of a specific .xml distribution package (configuration) file is defined by the SSC distribution package schema, distributionPackage.xsd.

The SSC distribution package schema is a standard W3C XML Schema compliant document used for describing and constraining the content of any .xml configuration file. It is assumed that the user of this document is familiar with the syntax of the W3C XML Schema specification and an instantiated XML output.

Schema Properties

The schema has the following aspects:

- Any distribution package instance XML file is a readable text file that helps the reader to fully understand the end-user configuration. To support user readability the schema has the following characteristics:
  - Each configuration setting is represented by a specific schema element.
  - Configuration settings are conveyed by the existence of an optional element or a value of an element.
  - The use of schema attributes is reserved for clarifying a configuration setting.
- The definition of a network is a hierarchical decision tree structure. The schema walks you through the tree based on your choices as you proceed. Traversing the tree automatically narrows down the set of configurable parameters to those that are of concern for your particular type of network. Additionally, this automatically refines the set of values allowed for a given configuration parameter. For example, in a wireless network one needs to configure an association mode for the connection. But the set of allowed values if you choose an authenticating network is different than if you choose a shared network. The basic order in which decisions are made is as follows:

For all networks:

1. Choosing connection media (wired or wireless) for the network.
2. Choosing security class (open, shared key, authenticating) for the network.
3. Choosing connection context (machine only, user only, machine and user) for the network.

For an authenticating network the decision tree continues:

4. Choosing credential type and collection method.
5. Choosing authentication method(s).

Schema Validation

Although the schema includes enumeration values it does not explicitly specify all of the allowed uses and combinations of elements, nor requirements for non-enumerated strings. Those details are covered by a set of Business Rules (which are detailed later in this document in Chapter 2, "Schema Elements").

A generated .xml distribution package file must, therefore, satisfy the following criteria in order to be accepted by the SSC.

- The .xml file must be valid with respect to the syntactical requirements of the SSC distribution package schema.
- The .xml file must be valid with respect to the element relationship requirements of the schema Business Rules.

Distribution Package Creation Steps

Cisco supports two basic methods for creating your distribution package xml instance file:

- Methods Based on the Language of the Schema—the manual process supported in releases earlier than Release 4.2
- Methods Based on Descriptive English—a new wizard utility

Methods Based on the Language of the Schema

Follow these steps to create a distribution package file.

Step 1 Generate the descriptive .xml distribution package file as specified by the SSC schema and this document (Chapter 2, "Schema Elements"). Alternative methods for accomplishing this include:

- Use a commercially available XML editor that supports direct creation of an XML instance file from a schema. These tools provide some contextual help during the XML editing and helps you validate the instance file. Examples of such applications are:
  - XMLSpy by Altova
  - Stylus Studio by DataDirect Technologies
- Use any text editor and the detailed description of the schema structure and elements given in Chapter 2, "Schema Elements" to create an XML instance file either from scratch or by cut-and-paste from the included examples.
  - From scratch—Chapter 2, "Schema Elements" walks you through the schema and contains XML examples of element structure for reference.

- Cut and paste—Appendix B, "Distribution Package Examples" contains complete distribution package examples. Pick one from the list that best matches your network environment and edit it by referring to the details in Chapter 2, "Schema Elements". The file, sscAdminGuideExXml.zip, also distributed in the SSCAdminUtils zip file, contains all of the examples as individual .xml files, for a convenient starting point and easy text editing.

Tip Text editing is greatly simplified by using a programming text editor that recognizes the syntax of the text language (in this case, XML). There are many such editors available commercially. Some support additional features such as automatic tag closing and element indentation cleanup.

Tip XML Syntax:

The syntax rules of XML are very simple. A few basic concepts are listed here:

- Each .xml file has a root element, in our case *configuration*, which servers as the container for the descriptive elements.
- All XML elements must have a closing tag.
- XML elements must be properly nested.
- XML tags are case sensitive.
- An element may contain child elements, content (text values) or attributes, in any combination.
- All attribute values must be quoted.
- Illegal XML characters must be replaced by the following entity references. Entity references always start with the '&' character and end with the ';' character.

  less than—use < for the character <
  greater than— use > for the character >
  ampersand—use & for the character &
  apostrophe—use ' for the character '
  quotation mark—use " for the character "
- White space is preserved. (This is important, for example, when entering specified enumerated content values. Avoid leading and trailing white space for enumerated and boolean values.)
- A comment is surrounded by the following syntax: <!-- your comment -->.

A specific .xml distribution package file (also known as an instance of the distribution package schema) is therefore constructed from the following building blocks:

```
<configuration>
  <childElement>with content</childElement>
  <elementWithAttr attr="{value}">
    <anotherChild>
      <!-- more hierachical elements -->
    </anotherChild>
  </elementWithAttr> <!--properly nested closing tag-->
  <emptyElement1></emptyElement1> <!--an empty element has no children or content-->
  <emptyElement2/> <!-- a shorthand notation for an empty element, used in this document-->
```

```
</configuration>
```

Note Distribution package file name:
The only restriction on the naming of your distribution package is it must have the .xml file extension.

Step 2 Pass the generated package distribution .xml file through the SSC postprocess command line utility, sscManagementUtility.exe. The sscManagementUtility utility performs the following required operations:

- Validates the preprocessed distribution package for both schema and business rule violations.
- Encrypts all credentials and secrets from their original clear text.
- Retrieves and packages any optional files referred to in the input file.
- Digitally signs the distribution package file to help prevent any tampering with its contents while it resides in the end station.

See "Postprocessing Utility" for a command-line description of this utility.

Methods Based on Descriptive English

Cisco provides a wizard that walks you through the distribution package file creation process. The GUI version of the sscManagementUtility allows you to:

- Create a distribution package from scratch
- Import an existing file to use as a starting point for making changes
- Postprocess an existing distribution package

Note This is a major upgrade of the built-in deployment wizard found in the earlier 4.0 releases. It is now a standalone utility. The dialog flow used in the wizard tracks that used in this document; such as the textual flow used in Chapter 2, "Schema Elements," and the graphical flow described in Chapter A, "Network Decision Tree Flow Diagram."

The GUI version of the sscManagementUtility supports creating and processing distribution package xml files for all versions of SSC, Release 4.1 and later.

Execute sscManagementUtility to open the utility.

Postprocessing Utility

The syntax of the command-line version of the postprocessing utility is:

sscManagementUtility {**help** | **validate** | **sign**} [command specific arguments]

sscManagementUtility help

sscManagementUtility validate {**-i** *input-file* | **--in**=*input-file*}

sscManagementUtility sign {**-i** *input-file* | **--in**=*input-file*} {**-o** *output-file* | **--out**=*output-file*}

Table 1-1 sscManagementUtility Command Elements

Command Elements	Meaning
validate	Validate a distribution package xml file only.
sign	Postprocess (validate, encrypt, sign) a distribution package xml file.
help	Displays utility release and command usage information.
-i *input-file* **--in=***input-file*	Path, absolute or relative, to the distribution package xml file to be processed.
-o *output-file* **--out=***output-file*	Path, absolute or relative, to the processed distribution package xml file ready for deployment.

Errors sent to the standard error output (stderr) include:

- usage errors (incorrect command)
- file I/O errors
- unknown distribution package XML file version
- XML schema validation errors
- XML encryption errors
- XML signing errors
- Business rule violations

See Appendix C, "Postprocessing Verification Errors" for an overview of errors produced during postprocessing.

Note The sscManagementUtility utility requires that the following support files. These files are provided in the SSCAdminUtils_{release}.zip file in a data folder that is structured by SSC version. This folder structure must be left intact when extracting the contents of the zip file.

- distributionPackage.xsd, schema file

  Release numbering is defined in Configuring the Distribution Package, page 2-2. Each instantiated distribution package xml file retains the release numbering scheme of its associated schema file.

- validateRules.xsl, business rules file

  Release numbering is controlled by a namespace for the file, as follows:

  xmlns:validateRules="http://www.cisco.com/2007/CSSCValidationRules/A.B.C", where A, B and C correspond to major, minor and maintenance, respectively.

Note The management utility uses the Microsoft msvcp71.dll and msvcr71.dll files. These files are normally loaded into the system area when installing SSC. To allow for the use of these deployment tools in a non-SSC machine, these files are supplied in the SSCAdminUtils_{release}.zip file and should be left in the same folder as the utility.

Additionally, the GUI version of the utility uses several supplied QT dll files. These should also be left in the same folder as the utility.

Distribution Package - SSC Release Compatibility

The SSC and the distribution package .xml file have readily viewable release numbers. It is important to understand their relationship and use the correct combination when deploying a particular release of SSC.

Release Numbering for SSC

The out-of-the-box installation file (.msi) obtained from Cisco has the following format:

Cisco_SSC-{OS}-A_B_C_xxxx.msi

For the Windows 2000/XP release of SSC, this becomes:

Cisco_SSC-XP2K-A_B_C_xxxx.msi, where

- Digits used for release compatibility tracking are:

  A—Indicates major release change
  B—Indicates minor release change
  C—Indicates maintenance release change

- Digits not used for release compatibility tracking are:

  xxxx—A Cisco software build identifier

Compatibility Between SSCMgmtToolkit and SSC

The following table lists the release of the management utility package that may be used to produce a full-featured distribution package for the designated release of SSC.

Table 1-2 Management Utility vs. SSC

This Release of Management Utility Package	Supports These SSC Releases
SSCMgmtToolkit_4.2.0.xxxx.zip	Cisco_SSC-XP2K-4_1_0_xxxx.msi
	Cisco_SSC-XP2K-4_1_1_xxxx.msi
	Cisco_SSC-XP2K-4_1_2_xxxx.msi
	Cisco_SSC-XP2K-4_2_0_xxxx.msi

Compatibility Between Distribution Package and SSC

The following table lists the release of a distribution package file (.xml) that is compatible with the designated release of SSC. In addition to full-featured compatibility, SSC, Release 4.1.2 and later, also supports distribution package files (.xml) from an earlier release. This allows upgrading the SSC application without the necessity of also deploying an updated distribution package file. Networks created with the earlier release will function the same.

Table 1-3 Distribution Package vs. SSC

Release of SSC	Compatible Release of XML File	Comment
Cisco_SSC-XP2K-4_1_0_xxxx.msi	4.1	Full functionality
Cisco_SSC-XP2K-4_1_1_xxxx.msi	4.1	Full functionality
Cisco_SSC-XP2K-4_1_2_xxxx.msi	4.1.2	Full functionality
Cisco_SSC-XP2K-4_1_2_xxxx.msi	4.1	Backwards compatible
Cisco_SSC-XP2K-4_2_0_xxxx.msi	4.2	Full functionality
Cisco_SSC-XP2K-4_2_0_xxxx.msi	4.1.2	Backwards compatible
Cisco_SSC-XP2K-4_2_0_xxxx.msi	4.1	Backwards compatible

Distribution Package Deployment

Cisco assumes that the IT Administrators already have a preferred method of moving files to end-user stations (for example, Microsoft's SMS method).

Cisco provides a separate command line utility, sscPackageGen.exe, to facilitate the following enterprise deployment operations:

- Windows Installer single-step installation of a pre-configured SSC
- Windows Installer update of an initially deployed and installed SSC

Note Deployment by means of remote desktop is not supported.

Enterprise Deployment Utility

The syntax of the enterprise deployment utility is:

sscPackageGen {**insert** | **patch**} *source dest file*

Table 1-4 sscPackageGen Command Elements

Command Elements	Meaning
insert	Command to create a msi file.
patch	Command to create a msp file.
source	The full, absolute path for the input msi file.
dest	The full, absolute path for the output msi or msp file.
file	The full, absolute path for the input distribution package xml file.

Note The Cisco distributed (out-of-the-box) SSC installation file has the following generalized format:

Cisco_SSC-<os version>-<release>.msi

In particular, this translates to Cisco_SSC-XP2K-4_1_0_xxxx for the Windows XP/2000 release of the Cisco Secure Services Client.

The sscPackageGen utility uses PatchWiz.dll and mspatchc.dll files. These files are loaded at run-time. As a consequence, sscPackageGen.exe will run even if these dll files are not present. These files are required to create patches (the **patch** command), but are not required to configure an original package (the **insert** command). These two Microsoft files are part of the Windows Software Development Kit (SDK). These files may not be redistributed but can be freely obtained from the Microsoft web site as follows:

1. To search for the latest version, go to www.microsoft.com/downloads/.
2. From the Download Center window, choose **Developer Tools** in the Browse for Downloads list.
3. From the Developer Tools window, choose **Platform SDK** from the Show downloads for: drop-down list. Click **Go**.
4. From the Platform SDK window search for and choose the latest **Microsoft Windows Server 2003 Platform SDK Web Install**.
   At the time of the writing of this document, this was: Windows Server 2003 R2 Platform SDK Web Install. A direct link to this download is:

   http://www.microsoft.com/downloads/details.aspx?FamilyID=0baf2b35-c656-4969-ace8-e4c0c0716adb&DisplayLang=en
5. Download and install the PSDK-x86.exe version of the Windows Server 2003 Platform SDK Web Install.
6. On the Installation Type window, choose **Custom installation**.
7. On the Custom Installation window, choose **Will not be available** for all features but **Microsoft Windows Installer SDK**.
8. Once installed, obtain the dll files from the following default install location:

   C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Samples\SysMgmt\Msi\Patching
9. Copy the dll files to the folder containing the sscPackageGen.exe utility.

Note The sscPackageGen utility checks the version of these two dll files before loading them. Only the following versions are acceptable. An error message is displayed when attempting to run the utility if this version check fails or if these required dll files are not properly installed.

- PatchWiz.dll must be major version 3.
- mspatchc.dll must be major version 5.

End-User Initial Installation

Choose one of the following methods to initially install an end-user SSC.

- Enterprise deployment installation method
- Legacy installation method

Enterprise Deployment Installation Method (recommended)

SSC and its companion distribution package are deployed as a single file and installed in a single operation. (Recall that any required support files, optional CA certificates and optional FAST PACs, have already been added to the distribution package itself.) The sscPackageGen utility takes as input the out-of-the-box installation file (.msi) and the distribution package file (.xml) and creates a new pre-configured installation file (.msi).

Example 1-1 Initial Installation File

Create a pre-configured installation file, called *yourSSCInstallPkg.msi*, from the installation file obtained from Cisco (Cisco_SSC-XP2K-4_2_0_xxxx) and your validated and postprocessed distribution package file (distributionPackage.xml).

```
sscPackageGen insert C:\Cisco_SSC-XP2K-4_2_0_xxxx.msi C:\yourSSCInstallPkg.msi
C:\distributionPackage.xml
```

Deploying and executing yourSSCInstallPkg.msi on the end station will install SSC with your predefined distribution package configuration.

SSC supports a single-step, silent install by the standard Microsoft Installer mechanism. For this example, execute

msiexec /i yourSSCInstallPkg.msi /quiet /norestart.

Legacy Installation Method

A multistep operation (similar to releases earlier than Release 4.1) can also be used.

1. Deploy and install the installation file obtained from Cisco (Cisco_SSC-XP2K-4_2_0_xxxx).
2. Update the end-user configuration as outlined in the next section.

Updating End-User Configurations

Choose one of the following methods to update an end-user configuration.

- Enterprise deployment update method
- Legacy update method

Enterprise Deployment Update Method (recommended)

To update an initially deployed and installed SSC, the sscPackageGen utility takes as input your pre-configured installation file (.msi) and the distribution package file (.xml) and creates a patch file (.msp).

Example 1-2 Update Based on Preconfigured File

Create an update patch file, called *yourSSCUpdatePkg.msp*, from your previously deployed preconfigured file and an updated distribution package file (.xml):

```
sscPackageGen patch C:\yourSSCInstallPkg.msi C:\yourSSCUpdatePkg.msp
C:\distributionPackage.xml
```

Note The updated distribution package file must have the same name as the original distribution package file, and the two must have different content.

Legacy Update Method

The deployment of a postprocessed distribution package .xml file (similar to releases earlier than Release 4.1) can also be performed.

Deploy the new/updated postprocessed distribution package .xml file into the following folder created by the SSC installer:

<install folder>\distribution, where the default <install folder> is: Program Files\Cisco Systems\Cisco Secure Services Client.

Upgrading End-User Installations

There are two scenarios for updating an end-user installation with Release 4.2:

- updating Release 4.1 (any maintenance release)
- updating Release 4.0 (any maintenance release)

Release 4.1

Upgrading the Cisco SSC Release 4.1.x to 4.2.x is the same process as the initial installation described above.

All previously deployed (locked) networks will be replaced by those in the updated distribution package file. Therefore when adding new networks or modifying an existing network you must also include in the updated configuration file any unaltered network that you want to keep. Deleting any previously existing network in the updated distribution package will delete that network.

Release 4.0

Upgrading an earlier Cisco SSC Release 4.0.x to Release 4.2.x is the same process as the initial installation described above.

When you update an administrator version (all networks are user-defined), the user-configured networks are migrated to the upgraded version. However if there is a user created network for a wired network or for one of your enterprise SSIDs, and the distribution package also configures one or more of these networks, then the original network profiles are replaced with the new distribution package (locked) versions.

When you update a deployed end-user version, all existing administrator deployed (locked) networks are replaced by the set of new (locked) networks in the distribution package. The user-configured networks are migrated to the upgraded version. However if there is a user created network for a wired network or for one of your enterprise SSIDs, and the distribution package also configures one or more of these networks, then the original network profiles are replaced with the new distribution package (locked) versions.

Pre-Installation of Client Certificates

If the end-user SSC is using a client-certificate-based EAP method, then the client certificate used to supply the user's credentials must be independently deployed and placed in the proper Windows Certificate Store (User-Personal Store). The distribution package file does not include deploying client certificates.

CHAPTER 2

Schema Elements

Introduction

This chapter contains detailed specifications for naming conventions, allowed element and attribute values, element structure and element combinations required to create the distribution package file.

This chapter contains the following sections:

Note Throughout this chapter, a full schema path is given for each occurrence of an element. There are two common instances of multiple paths for which the following abbreviation is used:

The path configuration/networks/[*wifiNetwork* | *wiredNetwork*]/ is an abbreviation which expands to two separate paths:

configuration/networks/wifiNetwork/

configuration/networks/wiredNetwork/

The path configuration/networks/[*wifiNetwork* | *wiredNetwork]/authenticationNetwork/ [machineAuthentication* | *userAuthentication* | *machineUserAuthentication/machine* | *machineUserAuthentication/user*]/ is an abbreviation which expands to eight separate paths:

configuration/networks/wifiNetwork/authenticationNetwork/machineAuthentication/

configuration/networks/wifiNetwork/authenticationNetwork/userAuthentication/

configuration/networks/wifiNetwork/authenticationNetwork/ machineUserAuthentication/machine/

configuration/networks/wifiNetwor/authenticationNetwork/ machineUserAuthentication/user/

configuration/networks/wiredNetwork/authenticationNetwork/machineAuthentication/

configuration/networks/wiredNetwork/authenticationNetwork/userAuthentication/

configuration/networks/wiredNetwork/authenticationNetwork/machineUserAuthentication/machine/

configuration/networks/wiredNetwork/authenticationNetwork/machineUserAuthentication/user/

Note Throughout this chapter, where an element has a relational restriction with another element, the requirement is captured in its business rule statement. The concept of a business rule is described in the "Schema Validation" section on page 1-5.

Configuring the Distribution Package

Start here to create your distribution package. Configure the following element:

configuration

Schema path:

configuration

The base element *configuration* forms the container for the distribution package. No element value is specified.

This element has the following required attributes:

- major_version—Required with value = 4.
- minor_version—Required with value = 2.
- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"—Copy as defined here.
- xsi:noNamespaceSchemaLocation="C:\yourPath\distributionPackage.xsd"—Contains the absolute or relative path to the schema used to instantiate a particular .xml distribution package file; in this case it must point to distributionPackage.xsd.

  The value is important only if you are using a commercial XML development tool. The sscManagementUtility utility does not use this attribute value, so use the following text in your distribution package .xml file:

  xsi:noNamespaceSchemaLocation="distributionPackage.xsd"

Note The first line of your distribution package.xml file contains the following text when the XML file is created by a commercial tool or from the examples in this document:
<?xml version="1.0" encoding="UTF-8"?>

The need to include this line depends on your choice of distribution package file creation tools. The postprocessing utility and the SSC do not require this statement in the XML file.

Step 1 Perform the tasks defined in the "Configuring Your License" section on page 2-3.

Step 2 Perform the tasks defined in the "Configuring Your Policy" section on page 2-4.

Step 3 Perform the tasks defined in the "Configuring Your Station Settings" section on page 2-14.

Step 4 Perform the tasks defined in the "Configuring Networks" section on page 2-16.

The following example illustrates the distribution package XML for the base element, *configuration*, and its child elements. The order of the child elements is restricted to that shown.

Example 2-1 Base Element

```
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\yourPath\distributionPackage.xsd" minor_version="1"
major_version="4">
    <license>your-license</license>
    <networkPolicy>
        {child elements}
    </networkPolicy>
    <networks>
        {child elements}
    </networks>
    <stationSettings>
        {child elements}
    </stationSettings>
    <userControlPolicy>
        {child elements}
    </userControlPolicy>
</configuration>
```

Configuring Your License

Configure the following element:

license

Schema path:

configuration/license

The value of the optional element *license* specifies the license for the deployed end-user SSC.

The following items are affected by the license:

- Individual authentication methods.
- Network adapter media types - wired, wireless.
- Credentials through a smartcard.
- Wi-Fi WPA2/802.11i (Wi-Fi WPA is standard with wireless media support.)
- Cisco Trust Agent (CTA) processing when CTA is also installed.

A companion User Control policy element, *allowLicensing*, will allow the end-user to enter any required license.

Note If you want to control licensing in the end-user SSC, the initial deployment of the end-user SSC requires the use of this element with your enterprise license. Subsequent distribution package updates do not have to include this optional element.

Example 2-2 license

```
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3
MEM-LGAA</license>
```

Configuring Your Policy

All distribution package files must contain a configuration definition for the policy of any deployed SSC.

Note The rights granted by the license pre-empt any policy configuration. For example, if you configure the policy for wireless media support but the license is wired media only, the deployed distribution package file will be accepted by the SSC, but it will only support wired networks. The relationship of the license to the policy is not verified by the postprocessing utility.

User Control Policy

Configure the following element:

userControlPolicy

Schema path:

configuration/userControlPolicy

The mandatory element *userControlPolicy* forms the container for specifying the policy for the user control of the SSC. No element values are specified.

Follow these steps to configure the following child elements of *userControlPolicy*. The order of the child elements is restricted as shown in these steps.

Step 1 Configure the following policy element for user interface:

clientUIType

Schema path:

configuration/userControlPolicy/clientUIType

The value of the mandatory element *clientUIType* specifies the user interface type.

The element has the following values:

- preset—Prevents the end-user from creating new networks and is an excellent choice for end-stations that will only encounter networks that you control. The Preset client has a limited user interface allowing the end-user to obtain status only for predefined networks.
- configurable—Allows your end-users to create new networks and is an excellent choice for end-stations that will move out of your enterprise networks to home or travel networks. The Configurable client has a robust user interface allowing the end-user to obtain status as well as define networks.

Step 2 Configure the following policy element for licensing methods:

allowLicensing

Schema path:

configuration/userControlPolicy/allowLicensing

The boolean value of the mandatory element *allowLicensing* specifies whether or not the end-user can directly license SSC from the user interface.

The element has the following values:

- true—Allows the end-user access to the Activate Product Features dialog where direct installation of a new license is available.
- false—Disallows licensing by the user interface. Use this setting if you intend to control licensing only from the distribution package.

Step 3 Configure the following policy element for media support:

allowedMedia

Schema path:

configuration/userControlPolicy/allowedMedia

The mandatory element *allowedMedia* forms the container for specifying which media types are supported. No element values are specified.

Note The allowed media types are also controlled by the license that has precedence. In other words if your license permits only wired media, then specifying Wi-Fi support here in the distribution package will have no effect.

Business rule: at least one child element must be specified.

Specify one or both of the following child elements. The order of the two child elements is not restricted.

wifi

Schema path:

configuration/userControlPolicy/allowedMedia/wifi

The presence of the optional element *wifi* specifies support for wireless (Wi-Fi) connections. It is an empty element with no value.

wired

Schema path:

configuration/userControlPolicy/allowedMedia/wired

The presence of the optional element *wired* specifies support for wired connections. It is an empty element with no value.

The following example illustrates the distribution package XML for the *userControlPolicy* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-3 userControlPolicy

```
<userControlPolicy>
    <clientUIType>configurable</clientUIType>
```

```
    <allowLicensing>false</allowLicensing>
    <allowedMedia>
        <wired/>
        <wifi/>
    </allowedMedia>
</userControlPolicy>
```

Network Policy

Configure the following element:

networkPolicy

Schema path:

configuration/networkPolicy

The mandatory element *networkPolicy* forms the container for specifying the policy for how networks can be configured and what settings are accessible to the end-user. No element values are specified.

Follow these steps to configure the following child elements of *networkPolicy.* The order of the child elements is restricted as shown in these steps.

Step 1 Configure the following policy element for association modes:

allowedAssociationModes

Schema path:

configuration/networkPolicy/allowedAssociationModes

The mandatory element *allowedAssociationModes* forms the container for specifying the wireless association modes allowed in any or your wireless network configurations. No element values are specified.

This policy specification applies to networks created by the administrator elsewhere in the distribution package file and to networks created by the end-user from the deployed SSC's user interface.

Business rule: At least one child element must be specified when you are also configuring a wireless network (element *wifiNetwork*).

Specify one or more of the following wireless association modes:

The order of the child elements is not restricted.

In a wired-only environment, only element *open* is necessary.

- Wi-Fi open association with no encryption or Wired open—Use element *open*.
- Wi-Fi WPA Personal—Use element *wpa-Personal*.
- Wi-Fi WPA Enterprise—Use element *wpa-Enterprise*.
- Wi-Fi WPA2 Personal—Use element *wpa2-Personal*.
- Wi-Fi WPA2 Enterprise—Use element *wpa2-Enterprise*.
- Legacy wireless open association with static WEP encryption (*staticWep*) or shared association with WEP shared keys (*shared*) or open association with 802.1X WEP encryption (*dynamicWep*)—Use element *wep*.

open

wpa-Personal

wpa-Enterprise

wpa2-Personal

wpa2-Enterprise

wep

Schema paths:

configuration/networkPolicy/allowedAssociationModes/open
configuration/networkPolicy/allowedAssociationModes/wpa-Personal
configuration/networkPolicy/allowedAssociationModes/wpa-Enterprise
configuration/networkPolicy/allowedAssociationModes/wpa2-Personal
configuration/networkPolicy/allowedAssociationModes/wap2-Enterprise
configuration/networkPolicy/allowedAssociationModes/wep

The presence of any of these elements specifies support for the association mode. All are empty elements with no values.

Step 2 Configure the following policy element for authentication methods:

allowedEapMethods

Schema path:

configuration/networkPolicy/allowedEapMethods

The mandatory element *allowedEapMethods* forms the container for specifying which EAP methods are allowed to be used for the primary (or outer tunnel) authentication protocol in any of your network configurations. (The set of EAP methods allowed for use in any inner tunnel of a tunneled EAP method is not affected by this policy.) No element values are specified.

This policy specification applies to networks created by the administrator elsewhere in the distribution package file and to networks created by the end-user from the deployed SSC's user interface.

Note The allowed EAP methods are also controlled by the license that has precedence. In other words if your license does not permit EAP-FAST, then specifying FAST support here in the distribution package will have no effect.

Business rule: At least one child element must be specified when also configuring an authenticating network (element *authenticationNetwork*).

Specify one or more of the following authentication methods:

The order of the child elements is not restricted.

- EAP-MD5—Use element *eapMd5*.
- EAP-MSCHAPv2D5—Use element *eapMschapv2*.
- EAP-GTC—Use element *eapGtc*.
- EAP-FAST—Use element *eapFast*.
- EAP-PEAP—Use element *eapPeap*.

- EAP-TTLS—Use element *eapTtls*.
- EAP-TLS—Use element *eapTls*.
- LEAP—Use element *leap*.

eapMd5

eapMschapv2

eapGtc

eapFast

eapPeap

eapTtls

eapTls

leap

Schema paths:

configuration/networkPolicy/allowedEapMethods/eapMd5
configuration/networkPolicy/allowedEapMethods/eapMschapv2
configuration/networkPolicy/allowedEapMethods/eapGtc
configuration/networkPolicy/allowedEapMethods/eapFast
configuration/networkPolicy/allowedEapMethods/eapPeap
configuration/networkPolicy/allowedEapMethods/eapTtls
configuration/networkPolicy/allowedEapMethods/eapTls
configuration/networkPolicy/allowedEapMethods/leap

The presence of any of these elements specifies support for the authentication method. All are empty elements with no values.

Step 3 Configure the following policy element for trusted servers:

serverValidationPolicy

Schema path:

configuration/networkPolicy/serverValidationPolicy

The mandatory element *serverValidationPolicy* forms the container for specifying how authenticating networks must process the validation of the associated authentication server. No element value is specified.

Specify one of the following policies:

- Force server validation for all networks—Use element *alwaysValidate*.
- Configure server validation on a per network basis—Use element *allowUserValidationControl*.

The chosen policy applies to networks created by the administrator elsewhere in the distribution package file and to networks created by the end-user from the deployed SSC's user interface.

allowUserValidationControl

Schema path:

configuration/networkPolicy/serverValidationPolicy/allowUserValidationControl

The existence of the *allowUserValidationControl* element allows for individualized configuring of server validation. The configuration of whether or not server validation is performed is made on a per EAP method basis within each network. It is an empty element with no value.

alwaysValidate

Schema path:

configuration/networkPolicy/serverValidationPolicy/alwaysValidate

The existence of the *alwaysValidate* element specifies that all authenticating networks using a mutually authenticating method must perform server validation as part of the authentication process. This applies for networks created by the IT administrator in the distribution package and by an end-user from the user interface.

Business rule: All network *validateServerIdentity* elements must have the value true.

Configure the following child element dealing with the policy for end-user creation of trusted server rules.

allowUserTrustedServers

Schema path:

configuration/networkPolicy/serverValidationPolicy/alwaysValidate/allowUserTrustedServers

The boolean value of the mandatory element *allowUserTrustedServers* specifies whether or not to allow the end users to define trusted servers for their own locally created private networks. (Trusted servers defined by the IT administrator, and deployed to the end-user can never be edited by the end-user and are not affected by this policy element.)

The element has the following values:

- true—Allows end-users to create trusted server rules.
- false—Disallows end-users from creating trusted server rules. The deployed user interface is modified accordingly.

Step 4 Configure the following policy element for multiple connection operation:

allowUserSimultaneousConnectionsControl

Schema path:

configuration/networkPolicy/allowUserSimultaneousConnectionsControl

The boolean value of the mandatory element *allowUserSimultaneousConnectionsControl* specifies whether or not to allow the end users to have control over changing the setting which specifies how SSC deals with multiple network adapters. (The companion station setting element, *simultaneousConnections*, sets the deployed mode.)

The element has the following values:

- true—Allows end-users to change the way connections are made.
- false—Disallows end-users from changing the way connections are made. The deployed user interface is modified accordingly.

  Business rule: If the deployed station setting element, *simultaneousConnections*, is set to singleHomed, then the end-user is not allowed to change to a less secure mode and this option is not allowed.

Step 5 Configure the following policy element for storing credentials:

allowedCredentialStorage

Schema path:

configuration/networkPolicy/allowedCredentialStorage

The mandatory element *allowedCredentialStorage* forms the container for specifying how long to store credentials that are obtained directly from the user through prompting. No element values are specified.

Business rule: At least one child element must be specified when you are also configuring an authenticating network with a credential collection method of prompting (element *prompt/credentialsStorage*).

Specify one or more of the following storage durations for user-prompted credentials:

The order of the child elements, when present, is restricted as listed here.

- Forever, that is, until changed—Use element *forever.*
- For the duration of the current login session—Use element *logonSession*.
- For a specified timed duration—Use element *duration*.

This policy specification applies to networks created by the administrator elsewhere in the distribution package file and to networks created by the end-user from the deployed SSC's user interface.

forever

Schema path:

configuration/networkPolicy/allowedCredentialStorage/forever

The presence of this optional element specifies support for permanently saving the user credentials. When either the credentials fail or the authentication server issues a password change request, the user will be re-prompted for the new credentials which will replace the previously saved values. After the initial prompt and save, this option acts like a static credential. It is an empty element with no value.

logonSession

Schema path:

configuration/networkPolicy/allowedCredentialStorage/logonSession

The presence of this optional element specifies support for saving the user credentials only during the current login session. When the user logs out, the credentials are deleted. It is an empty element with no value.

duration

Schema path:

configuration/networkPolicy/allowedCredentialStorage/duration

The presence of this optional element specifies support for saving the user credentials only for a specified time period. When the time period expires, the credentials are deleted. However the connection is maintained and there is no immediate re-prompt. A subsequent re-authentication request that is issued after the time-out will result in a re-prompt for the user's credentials. The value of the element specifies the global time-out period (in minutes) which applies to all networks defined to use this storage type.

Restriction: The specified time must be between 1 - 3600 (1 minute to approximately 2 1/2 days).

Step 6 Configure the following policy element for multiple connection operation:

allowUserWpaHandshakeValidationControl

Schema path:

configuration/networkPolicy/allowUserWpaHandshakeValidationControl

The boolean value of the mandatory element *allowUserWpaHandshakeValidationControl* specifies whether or not to allow the end users to have control over changing the setting which specifies how SSC deals with WPA handshake validation. (The companion station setting element, *validateWpaHandshake*, sets the deployed mode.)

The element has the following values:

- true—Allows end-users to change the way the WPA protocol is processed.
- false—Disallows end-users from changing the way the WPA protocol is processed. The deployed user interface is modified accordingly.
  Cisco recommends this setting since your user may not have sufficient knowledge of the capabilities or the network adapter in use. See the station setting element, *validateWpaHandshake*, for more information.

In a wired-only environment, this element is not used and can be given either value.

Step 7 Configure the following policy element for the scope of the network connection:

allowPublicProfileCreation

Schema path:

configuration/networkPolicy/allowPublicProfileCreation

The boolean value of the mandatory element *allowPublicProfileCreation* specifies the connection scope of networks created by the end-user through the SSC's user interface.

The element has the following values:

- true—the end-user is capable of defining a public network that allows for:
  - creating networks that will be shared among all users.
  - creating networks with a machine connection context.
- false—end-users are restricted to only creating private networks for themselves. The deployed user interface is modified accordingly.

Note All networks defined in the distribution package by the administrator are public.

Step 8 Configure the following policy element for client certificates:

allowedClientCertificates

Schema path:

configuration/networkPolicy/allowedClientCertificates

The mandatory element *allowedClientCertificates* forms the container for specifying whether or not filtering of client certificates based on their Extended Key Usage field is performed. No element value is specified.

Specify one of the following policies:

- No filtering required—Use element *noEkuFilter*.
- Filtering required—Use element *certificateEkuFilterExpression*.

The chosen policy applies to networks created by the administrator elsewhere in the distribution package file and to networks created by the end-user from the deployed SSC's user interface.

noEkuFilter

Schema path:

configuration/networkPolicy/allowedClientCertificates/noEkuFilter

The existence of the *noEkuFilter* element specifies that there are no usage restrictions based on the client certificate's Extended Key Usage field. It is an empty element with no value.

certificateEkuFilterExpression

Schema path:

configuration/networkPolicy/allowedClientCertificates/certificateEkuFilterExpression

The existence of the *certificateEkuFilterExpression* element specifies that client certificates are used only if their Extended Key Usage (EKU) is specified as allowed. The value of the element contains the EKU specification. The value is any (parenthesized) boolean (and, or, not) expression of the following EKU keywords.

- ServerAuth
- ClientAuth
- CodeSign
- SecureEmail
- IpsecEndSystem
- IpsecTunnel
- IpsecUser
- TimeStamp
- SmartCardLogon

Note The following lists the relationship between the above EKU keywords used in the distribution package and the actual certificate EKU strings:
ServerAuth—Server Authentication
ClientAuth—Client Authentication
CodeSign—Code Signing
SecureEmail—Secure Email
IpsecEndSystem—IP security end system
IpsecTunnel—IP security tunnel termination
IpsecUser—IP security user
TimeStamp—Time Stamping
SmartCardLogon—Smart Card Logon

Example 2-4 allowedClientCertificates

```
<allowedClientCertificates>
    <certificateEkuFilterExpression>ClientAuth or
SmartCardLogon</certificateEkuFilterExpression>
</allowedClientCertificates>

<allowedClientCertificates>
    <certificateEkuFilterExpression>(not ServerAuth and
ClientAuth)</certificateEkuFilterExpression>
</allowedClientCertificates>
```

The following example illustrates the distribution package XML for the *networkPolicy* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-5 networkPolicy

```
<networkPolicy>
    <allowedAssociationModes>
        <!--open network-->
        <open/>
        <!--shared key network-->
        <staticWep/>
        <shared/>
        <wpa-Personal/>
        <wpa2-Personal/>
        <!--authenticating network-->
        <dynamicWep/>
        <wpa-Enterprise/>
        <wpa2-Enterprise/>
    </allowedAssociationModes>
    <allowedEapMethods>
        <!--wired only-->
        <eapMd5/>
        <eapMschapv2/>
        <eapGtc/>
        <!--wired or wireless-->
        <eapFast/>
        <eapPeap/>
        <eapTls/>
        <eapTtls/>
        <leap/>
    </allowedEapMethods>
    <serverValidationPolicy>
        <alwaysValidate>
            <allowUserTrustedServers>true</allowUserTrustedServers>
        </alwaysValidate>
    </serverValidationPolicy>
    <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsContro
l>
    <allowedCredentialStorage>
        <forever/>
        <logonSession/>
        <duration>5</duration>
    </allowedCredentialStorage>
    <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
    <allowPublicProfileCreation>false</allowPublicProfileCreation>
    <allowedClientCertificates>
        <noEkuFilter/>
    </allowedClientCertificates>
</networkPolicy>
```

Configuring Your Station Settings

Configure the following element:

stationSettings

Schema path:

configuration/stationSettings

The mandatory element *stationSettings* forms the container for configuring the deployed settings for any global operational aspects of making network connections. No element values are specified.

Follow these steps to configure the following child elements of element *stationSettings*. The order of the child elements is restricted as shown in these steps.

Step 1 Configure the following station setting element:

simultaneousConnections

Schema path:

configuration/stationSettings/simultaneousConnections

The value of the mandatory element *simultaneousConnections* specifies the multiplicity of connections for all networks.

The element has the following values:

- singleHomed—restricted to creating only a single connection at a time (prevents multi-homed configurations).
- multiHomed—allows multiple simultaneous connections (allows multi-homed network connections). For the selected network, SSC will attempt to make a connection for all equipped and managed wired and wireless network adapters.

Allowing the end-user to override the deployed (initial) setting is controlled by its companion network policy element, *allowUserSimultaneousConnectionsControl.*

Business rule: If singleHomed is configured, then the companion network policy element, *allowUserSimultaneousConnectionsControl*, must be configured to false. End-users may not override the administrator's choice of the restricted mode of operation.

Step 2 Configure the following station setting element:

validateWpaHandshake

Schema path:

configuration/stationSettings/validateWpaHandshake

The boolean value of the mandatory element *validateWpaHandshake* specifies how SSC deals with WPA handshake validation. WPA's sophisticated key management requires driver capabilities that may not all be available in older embedded network adapters. In order to support situations where the environment contains a large base of older adapters, SSC provides a security bypass capability for WPA/WPA2 so that no RSN probe response/beacon IE verification is required in the 4-Way Handshake.

The element has the following values:

- true—enable WPA/WPA2 handshake validation. (recommended)
  Use this setting when your end-stations all have wireless adapters with fully compliant WPA/WPA2 drivers, as required by the standards.

- false—disable WPA/WPA2 handshake validation.
  Use this setting only for special cases in which your wireless adapter's driver is known to have this deficiency.

In a wired-only environment, this element is not used and can be given either value.

Allowing the end-user to override the deployed (initial) setting is controlled by its companion network policy element, *allowUserWpaHandshakeValidationControl.*

Step 3 Configure the following station setting element:

credentialLabels

Schema path:

configuration/stationSettings/credentialLabels

The optional element *credentialLabels* is used to configure alternate names for end-user text entry boxes in credential dialogs. The element forms the container for specifying the alternate names. No element value is specified.

Configure the following child elements:

userNameLabel

Schema path:

configuration/stationSettings/credentialLabels/userNameLabel

The value of the mandatory element *userNameLabel* is used to specify your replacement text for the default text of “Username:”.

Restriction: the value must be no more than eighty characters in length.

passwordLabel

Schema path:

configuration/stationSettings/credentialLabels/passwordLabel

The value of the mandatory element *passwordLabel* is used to specify your replacement text for the default text of “Password:”.

Restriction: the value must be no more than eighty characters in length.

The following examples illustrate the distribution package XML for the *stationSettings* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-6 stationSettings

```
<stationSettings>
    <simultaneousConnections>singleHomed</simultaneousConnections>
    <validateWpaHandshake>false</validateWpaHandshake>
</stationSettings>

<stationSettings>
    <simultaneousConnections>singleHomed</simultaneousConnections>
    <validateWpaHandshake>false</validateWpaHandshake>
    <credentialLabels>
        <userNameLabel>Student ID</userNameLabel>
        <passwordLabel>Student Code Word</passwordLabel>
    </credentialLabels>
</stationSettings>
```

Configuring Networks

Tip Configuring a network is the central part of the distribution package definition and is also the most complex. As an aid, Appendix A, "Network Decision Tree Flow Diagram," provides an overview of the XML schema decision tree for configuring a network connection and serves as a graphical index to the following sections.

Configure the following element:

networks

Schema path:

configuration/networks

The element *networks* forms the container for your predefined enterprise networks. No element values are specified. Each child and its contents represent the configuration of an individual network.

This is an optional element. Omitting it implies that there will be no administrator-defined networks in the deployed end-user client. In this case only the end-user is expected to create network definitions.

Note The client cannot make unilateral choices - the configuration of a network is primarily determined by the policy of the authentication server and its associated access devices. The client must be appropriately configured to conform to its overall environment.

The first choice required in defining a network is to select a network type based on the media type of the connection.

Next item: "Choosing a Network Media Type'.

Choosing a Network Media Type

Specify one of the following network media types:

- 802.3 wired (Ethernet)—Use element *wiredNetwork*.
- 802.11 wireless (Wi-Fi)—Use element *wifiNetwork*.

wiredNetwork

Schema path:

configuration/networks/wiredNetwork

The optional element *wiredNetwork* forms the container for configuring an Ethernet (802.3) network. No element values are specified.

Business rule: only one *wiredNetwork* element is allowed. All wired (Ethernet) adapters can only be applied to a single wired network.

Business rule: This is a valid choice only if the wired media type is supported by the policy. See element wired in section "User Control Policy'.

Next item: "Configuring a Wired Network'.

wifiNetwork

Schema path:

configuration/networks/wifiNetwork

The optional element *wifiNetwork* forms the container for configuring an individual Wi-Fi (802.11) network. No element values are specified. Multiple wifiNetwork elements may be defined.

Business rule: This is a valid choice only if the wireless media type is supported by the policy. See element wifi in section "User Control Policy".

Next item: "Wi-Fi Network Base Elements".

The following example illustrates the distribution package XML for the *networks* element and its child elements. The order of the two possible child elements is not restricted.

Example 2-7 networks

```
<networks>
    <wiredNetwork>
        {child elements}
    </wiredNetwork>
    <wifiNetwork>
        {child elements}
    </wifiNetwork>
    <wifiNetwork>
        {child elements}
    </wifiNetwork>
</networks>
```

Configuring a Wi-Fi Network

Follow the tasks in the following sections to configure a Wi-Fi network.

1. "Wi-Fi Network Base Elements"
2. "Choosing the Wi-Fi Network's Security Class"

Configuring a Wired Network

Follow the tasks in the following sections to configure a wired network.

1. "Wired Network Base Elements"
2. "Choosing the Wired Network's Security Class"

Wi-Fi Network Base Elements

Configure the following elements:

displayName

Schema path:

configuration/networks/wifiNetwork/displayName

The value of the mandatory element *displayName* specifies the user-friendly name that is used only for display purposes throughout the SSC's various dialogs.

ssid

Schema path:

configuration/networks/wifiNetwork/ssid

The value of the mandatory element *ssid* contains the configured name of the access point, that is, its Service Set Identifier (SSID). The SSID is a unique identifier that distinguishes between multiple wireless networks in the same vicinity.

Note The value must be as defined by the access point's configuration.

Restriction: SSIDs are limited to 32 ASCII characters.

Business rule: SSIDs are unique. The same value may not be applied to more than one *ssid* element.

associationRetries

Schema path:

configuration/networks/wifiNetwork/associationRetries

The value of the mandatory element *associationRetries* specifies the number of times SSC attempts to associate with the access point during a connection attempt. Due to the variability of radio transmissions, association attempts are typically retried a few times before the authentication session gives up so as to avoid being too sensitive to occasional lost bits in the transmission.

Additionally, even though *associationRetries* is configured on an individual network basis, only one global setting applies to all networks. After deployment, SSC extracts the maximum value entered for all configured networks and uses that for its global value.

When a distribution package file does not contain any wireless networks, end-user created wireless networks when allowed by license and policy use a default value of three.

Restriction: the number of retries is limited to 99.

Default: the recommended value is 3.

beaconing

Schema path:

configuration/networks/wifiNetwork/beaconing

The boolean value of the mandatory element *beaconing* specifies whether or not the access point advertises its name and therefore sets the criteria SSC uses for determining the physical existence of the access point.

The element has the following values:

- True—Issues beacons or responses to active probe and is detected through standard wireless radio scanning.
- False—Not configured to be detectable via a standard scan and therefore requires an active search process to detect its existence. A non-beaconing access point is referred to as a hidden access point.

The following example illustrates the distribution package XML for the *wifiNetwork* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-8 partial wifiNetwork

```
<wifiNetwork>
    <displayName>My Corporate Wi-Fi Network</displayName>
    <ssid>MyCorpNet</ssid>
    <associationRetries>3</associationRetries>
    <beaconing>true</beaconing>
    <!--{your choice of network security class goes here}-->
</wifiNetwork>
```

Choosing the Wi-Fi Network's Security Class

Specify one of the following security classes for the network:

- Open network—Use element *openNetworkUserConnection*. Most likely you will not be deploying an open network to your end-users because of the need to pre-specify the SSID. Your end-users, if allowed, can create a network profile for a connection to a specific open network.
- Shared-key Network—Use element *sharedKeyNetwork*. Most likely you will be deploying a shared-key network only if your mobile end-user has a home network access device for which the shared secret is controlled by you, the network administrator. Your end-users, if allowed, can create a network profile for a connection to their home access point.
- Authentication Network—Use element *authenticationNetwork*. This is the most likely choice because you will want to preconfigure your enterprise network consistent with your authentication server and its policies and with your credentials environment.

openNetworkUserConnection

Schema path:

configuration/networks/wifiNetwork/openNetworkUserConnection

The optional element *openNetworkUserConnection* forms the container for configuring an open wireless network. An open network in SSC is one which does not use any form of data encryption and therefore represents the least secure class of networks. No element value is specified.

Business rule: This is a valid choice only if the *open* association mode is supported by the policy. See element allowedAssociationModes in section "Network Policy'.

Next item: "Configuring an Open Wi-Fi Network'.

sharedKeyNetwork

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork

The optional element *sharedKeyNetwork* forms the container for configuring a wireless network that uses a static key which is pre-defined in both the client and the access point. The key is ultimately used to provide for encryption of the data. This network class primarily serves the home/small office environment. No element value is specified.

Next item: "Configuring a Shared-key Wi-Fi Network'.

authenticationNetwork

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork

The optional element *authenticationNetwork* forms the container for configuring an 802.1X wireless network. An authenticating/802.1X network adds two important aspects to wireless security, mutual authentication of the client and server and network provide keys for encryption. This network class represents the highest security level choice. No element value is specified.

Next item: "Configuring an Authenticating Wi-Fi Network'

The following example illustrates the distribution package XML for the three security classes of the *wifiNetwork* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-9 wifiNetwork

```
<wifiNetwork>
    <displayName>My Corporate Wi-Fi Network</displayName>
    <ssid>MyCorpNet</ssid>
    <associationRetries>3</associationRetries>
    <beaconing>true</beaconing>
    <openNetworkUserConnection>
        {child elements}
    </openNetworkUserConnection>
</wifiNetwork>

<wifiNetwork>
    <displayName>My Corporate Wi-Fi Network</displayName>
    <ssid>MyCorpNet</ssid>
    <associationRetries>3</associationRetries>
    <beaconing>true</beaconing>
    <sharedKeyNetwork>
        {child elements}
    </sharedKeyNetwork>
</wifiNetwork>

<wifiNetwork>
    <displayName>My Corporate Wi-Fi Network</displayName>
    <ssid>MyCorpNet</ssid>
    <associationRetries>3</associationRetries>
    <beaconing>true</beaconing>
    <authenticationNetwork>
        {child elements}
    </authenticationNetwork>
</wifiNetwork>
```

Configuring an Open Wi-Fi Network

Configure the following element:

autoConnect

Schema path:

configuration/networks/wifiNetwork/openNetworkUserConnection/autoConnect

The mandatory boolean element *autoConnect* specifies whether or not the user-context connection process includes this network in its network selection algorithm. In other words, when the user logs into the system this element determines whether or not an automatic connection is attempted. Only a user-context connection is enabled and processed.

The element has the following values:

- True—Auto-connection is enabled.
- False—Auto-connection is disabled. A connection can always be initiated manually.

The following example illustrates the distribution package XML for the *openNetworkUserConnection* element and its child element.

Example 2-10 openNetworkUserConnection

```
<openNetworkUserConnection>
    <autoConnect>true</autoConnect>
</openNetworkUserConnection>
```

Configuring a Shared-key Wi-Fi Network

Specify one of the following connection contexts for the shared key network:

- Machine-only connection — Use element *machineConnection*.
- User-only connection — Use element *userConnection*.

machineConnection

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection

The optional element *machineConnection* forms the container for configuring a network that supports only a machine context connection. A connection is made at system boot using the configured machine credentials and is maintained when users log into or log off of the system. No element values are specified.

Next item: “Configuring a Shared-key, Machine Network’

userConnection

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection

The optional element *userConnection* forms the container for configuring a network that supports only a user context connection. A connection is made when a user logs into the system using the configured user credentials and is maintained until the user logs off of the system. No element values are specified.

Next item: “Configuring a Shared-key, User Network’.

The following example illustrates the distribution package XML for the *sharedKeyNetwork* element and its child element.

Example 2-11 sharedKeyNetwork

```
<sharedKeyNetwork>
    <machineConnection>
        {child elements}
    </machineConnection>
</sharedKeyNetwork>

<sharedKeyNetwork>
    <userConnection>
        {child elements}
    </userConnection>
</sharedKeyNetwork>
```

Configuring a Shared-key, Machine Network

Configure the following element:

keySettings

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings

The mandatory element *keySettings* forms the container for specifying the type of shared-key protocol. No element values are specified.

Next item: "Choosing the Shared-key Type".

The following example illustrates the distribution package XML for the shared-key network's *machineConnection* element and its child element.

Example 2-12 machineConnection

```
<machineConnection>
    <keySettings>
        {child elements}
    </keySettings>
</machineConnection>
```

Configuring a Shared-key, User Network

Configure the following elements:

autoConnect

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/autoConnect

The mandatory boolean element *autoConnect* specifies whether or not the user-context connection process includes this network in its network selection algorithm. In other words, when the user logs into the system this element determines whether or not an automatic connection is attempted.

The element has the following values:

- True—Auto-connection is enabled.
- False—Auto-connection is disabled. A connection can always be initiated manually.

keySettings

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings

The mandatory element *keySettings* forms the container for specifying the type of shared-key protocol. No element values are specified.

Next item: "Choosing the Shared-key Type'.

The following example illustrates the distribution package XML for the shared-key network's *userConnection* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-13 userConnection

```
<userConnection>
    <keySettings>
        {child elements}
    </keySettings>
    <autoConnect>true</autoConnect>
</userConnection>
```

Choosing the Shared-key Type

Specify one of the following key types for the network:

- WEP—Use element wep.
- WPA—Use element wpa.
- WPA2—Use element wpa2.

wep

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep

The optional element *wep* forms the container for configuring a Wired Equivalent Privacy (WEP) shared key or static key. The existence of this element indicates WEP key type. No element values are specified.

These are legacy security solutions which provide a low-level mechanism for a basic, but easily breakable, data privacy capability between the client and network access device. These legacy methods are supported for backwards compatibility but are not an integral part of an enterprise level security solution.

Business rule: This is a valid choice only if the *wep* association mode is supported by the policy. See element allowedAssociationModes in section "Network Policy'.

Next item: "Configuring a WEP Shared-key'.

wpa

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wpa

The optional element *wpa* forms the container for configuring a WPA-Personal mode. The existence of this element indicates WPA key type. No element values are specified.

Wi-Fi Protected Access (WPA) is the security solution of the Wi-Fi Alliance and improves upon the legacy 802.11's encryption method, WEP. WPA-Personal uses a pass-phrase preshared key (PSK).

Business rule: This is a valid choice only if the *wpa-Personal* association mode is supported by the policy. See element allowedAssociationModes in section "Network Policy".

Next item: "Configuring a WPA/WPA2 Shared-key"

wpa2

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa2

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wpa2

The optional element *wpa2* forms the container for configuring a WPA2-Personal mode. The existence of this element indicates WPA2 key type. No element values are specified.

WPA2 is a recent upgrade based on the full 802.11i standard. WPA2 is Wi-Fi Alliance branding for 802.11i interoperability. WPA2 is not released to address any flaws in WPA. The major aspect of WPA2 is the mandating of a new and stronger encryption cipher (AES). WPA2 also introduces subtle improvements in the association request/response messaging and in the key exchange messaging. WPA2-Personal uses a pass-phrase preshared key (PSK).

Business rule: This is a valid choice only if the *wpa2-Personal* association mode is supported by the policy. See element allowedAssociationModes in section "Network Policy".

Next item: "Configuring a WPA/WPA2 Shared-key"

Configuring a WEP Shared-key

Follow these steps to configure a Wi-Fi, WEP shared-key network.

Step 1 Configure the following elements:

ieee80211Authentication

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/ieee80211Authentication

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/ieee80211Authentication

The mandatory element *ieee80211Authentication* forms the container for configuring the type of association used between SSC and the access point. No element values are specified.

Next item: "Choosing the WEP Association".

Step 2 Perform the tasks defined in section "Choosing the WEP Key Format'.

The following example illustrates the distribution package XML for the WEP *keySettings* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-14 WEP keySettings

```
<keySettings>
    <wep>
        <wepAscii40 encrypt="true">aaaaa</wepAscii40>
        <ieee80211Authentication>
            <shared/>
        </ieee80211Authentication>
    </wep>
</keySettings>
```

Choosing the WEP Association

Specify one of the following WEP association modes:

- Open—Use element *open*.
- Shared—Use element *shared*.

open

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/ieee80211Authentication/open

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/ieee80211Authentication/open

The existence of the optional element *open* specifies the 802.11 open association mode. In SSC a shared-key network using open association is a legacy "Static WEP" network. It is an empty element with no value.

This element has a required boolean attribute, *encrypt*, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

shared

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/ieee80211Authentication/shared

The existence of the optional element *shared* specifies the 802.11 shared association mode. In SSC a shared-key network using shared association is a legacy "Shared WEP" network. It is an empty element with no value.

The following example illustrates the distribution package XML for the two choices for the *ieee80211Authentication* element.

Example 2-15 Distribution Package XML

```
<ieee80211Authentication>
    <shared/>
</ieee80211Authentication>

<ieee80211Authentication>
    <open/>
</ieee80211Authentication>
```

Choosing the WEP Key Format

Specify one of the following key formats and lengths for the network:

- ASCII with a 40 bit key—Use element *wepAscii40*.
- ASCII with a 128 bit key—Use element *wepAscii128*.
- Hex with a 40 bit key—Use element *wepHex40*.
- Hex with a 128 bit key—Use element *wepHex128*.

wepAscii40

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/wepAscii40

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wep/wepAscii40

The optional element *wepAscii40* specifies ASCII format and a 40-bit key length.

This element has a required boolean attribute, *encrypt*, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: the value must be five printable, ASCII characters long.

wepAscii128

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/wepAscii128

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wep/wepAscii128

The optional element *wepAscii128* specifies ASCII format and a 128-bit key length.

This element has a required boolean attribute, *encrypt*, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: the value must be thirteen printable, ASCII characters long.

wepHex40

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/wepHex40

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wep/wepHex40

The optional element *wepHex40* specifies Hex format and a 40-bit key length.

This element has a required boolean attribute, *encrypt*, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: the value must be ten Hex characters long.

wepHex128

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wep/wepHex128

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wep/wepHex128

The optional element *wepHex128* specifies Hex format and a 128-bit key length.

This element has a required boolean attribute, *encrypt*, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: the value must be twenty six Hex characters long.

The following example illustrates the distribution package XML for the four choices for the *wep* child elements.

Example 2-16 WEP Choices

```
<wepAscii40 encrypt="true">aaaaa</wepAscii40>

<wepAscii128 encrypt="true">aaaaaaaaaaaaa</wepAscii128>

<wepHex40 encrypt="true">AAAAAAAAAA</wepHex40>

<wepHex128 encrypt="true">ABCDEFABCDEFABCDEFABCDEFAB</wepHex128>
```

Configuring a WPA/WPA2 Shared-key

Follow these steps to configure a Wi-Fi, WPA/WPA2 shared-key network.

Step 1 Configure the following element:

encryption

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa/encryption

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa/encryption

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa2/encryption

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa2/encryption

The mandatory element ***encryption*** specifies the data encryption scheme.

The element has the following values:

- TKIP—the standard method for WPA association. Also supported with WPA2 association for backwards compatibility.
- AES—normally linked to WPA2 association but may be available in some WPA compliant access devices. The highest data security mode that is currently standardized for Wi-Fi.

Step 2 Configure the following element:

key

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa/key

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa/key

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa2/key

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa2/key

The mandatory element ***key*** forms the container for configuring the format of the WPA or WPA2 key. No element values are specified.

Next item: "Choosing the WPA/WPA2 Key Format'.

The following example illustrates the distribution package XML for the WPA/WPA2 *keySettings* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-17 WPA KeySettings

```
<keySettings>
    <wpa>
        <key>
            <ascii encrypt="true">mySecret</ascii>
        </key>
        <encryption>TKIP</encryption>
    </wpa>
</keySettings>

<keySettings>
    <wpa2>
        <key>
            <ascii encrypt="true">mySecret</ascii>
        </key>
        <encryption>TKIP</encryption>
    </wpa2>
</keySettings>
```

Choosing the WPA/WPA2 Key Format

Specify one of the following key formats for the network:

- ASCII—Use element .
- Hex—Use element .

ascii

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa/key/ascii

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa/key/ascii

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa2/key/ascii

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa2/key/ascii

The optional element ***ascii*** specifies an ASCII format.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: the value must be 8 to 63 printable, ASCII characters in length.

hex

Schema paths:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa/key/hex

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa/key/hex

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings/wpa2/key/hex

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings/wep/wpa2/key/hex

The optional element ***hex*** specifies a Hex format.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: the value must be64 Hex characters in length.

The following example illustrates the distribution package XML for the two choices for the *key* element and its child element.

Example 2-18 Key

```
<key>
    <ascii encrypt="true">mySecret</ascii>
</key>

<key>
    <hex
encrypt="true">1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF</hex>
</key>
```

Configuring an Authenticating Wi-Fi Network

Follow these steps to configure an authenticating Wi-Fi network.

Step 1 Perform the tasks defined in "Configuring the Authentication Association Mode".

Step 2 Perform the tasks defined in "Configuring the Authenticating Network Base Elements".

Step 3 Perform the tasks defined in "Choosing the Authentication Network's Connection Context".

Step 4 Perform the tasks defined in "Choosing Wi-Fi EAP Methods".

The following example illustrates the distribution package XML for the Wi-Fi *authenticationNetwork* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-19 Partial authenticationNetwork

```
<authenticationNetwork>
    <!--{your choice of network connection context goes here}-->
    <serverValidation>
        {child elements}
    </serverValidation>
    <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
    <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
    <associationMode>
        {child element}
    </associationMode>
</authenticationNetwork>
```

Configuring the Authentication Association Mode

Configure the following element:

associationMode

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork/associationMode

The mandatory element *associationMode* forms the container for configuring the type of association used between SSC and the access point. No element values are specified.

Next item: "Choosing the Association Mode".

Choosing the Association Mode

Specify one of the following association modes for the network:

- Dynamic WEP—Use element dynamicWep.
- WPA-Enterprise—Use element wpa-Enterprise.
- WPA2-Enterprise—Use element wpa2-Enterprise.

Business rule: Only association modes supported by the policy are permitted. See element allowedAssociationModes in section "Network Policy'.

dynamicWep

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork/associationMode/dynamicWep

The optional element *dynamicWep* specifies the 802.11 open association mode used in conjunction with a network provided Wired Equivalent Privacy (WEP) dynamic key. The existence of this element indicates dynamic WEP. It is an empty element with no value.

This legacy security solution provides basic data privacy between the client and network access device. This legacy method is supported for backwards compatibility but is not an integral part of an enterprise level security solution.

wpa-Enterprise

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork/associationMode/wpa-Enterprise

The optional element *wpa-Enterprise* specifies the Wi-Fi WPA-Enterprise association mode. The existence of this element indicates WPA-Enterprise.

Wi-Fi Protected Access (WPA) is the security solution of the Wi-Fi Alliance and improves upon the legacy 802.11/802.1X's dynamic WEP. WPA-Enterprise mandates authentication before key exchange and uses 802.1X authentication to provide encryption seeds for key exchange.

The element has the following values:

- TKIP—the standard method for WPA association. Also supported with WPA2 association for backwards compatibility.
- AES—normally linked to WPA2 association but may be available in some WPA compliant access devices. The highest data security mode that is currently standardized for Wi-Fi.

wpa2-Enterprise

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork/associationMode/wpa2-Enterprise

The optional element *wpa2-Enterprise* specifies the Wi-Fi WPA2-Enterprise association mode. The existence of this element indicates WPA2-Enterprise.

WPA2 is a recent upgrade based on the full 802.11i standard. WPA2 is Wi-Fi Alliance branding for 802.11i interoperability. WPA2 is not released to address any flaws in WPA. The major aspect of WPA2 is the mandating of a new and stronger encryption cipher (AES). WPA2 also introduces subtle improvements in the association request/response messaging and in the key exchange messaging.

The element has the following values:

- TKIP—the standard method for WPA association. Also supported with WPA2 association for backwards compatibility.
- AES—normally linked to WPA2 association but may be available in some WPA compliant access devices. The highest data security mode that is currently standardized for Wi-Fi.

The following example illustrates the distribution package XML for the three choices for the *associationMode* element and its child element.

Example 2-20 associationMode

```
<associationMode>
    <dynamicWep/>
</associationMode>

<associationMode>
    <wpa-Enterprise>TKIP</wpa-Enterprise>
</associationMode>

<associationMode>
    <wpa2-Enterprise>AES</wpa2-Enterprise>
</associationMode>
```

Configuring the Authenticating Network Base Elements

Follow these steps to configure the base elements of an authenticating network.

Step 1 Configure the following authentication retry elements:

Some network access devices support special features for handling authentication failures, for example, the ability to open the port but switch the user into a special VLAN. In order to support these network access devices, SSC provides the administrator with configurable parameters. These parameters adjust the number of connection retries made before disconnecting, allowing the access device to make intelligent decisions based on multiple authentication failures.

Note Making changes to the default settings is not recommended without a through understanding of the impact on connection performance and knowledge of any special needs and features of your enterprise network access devices. Any changes should be comprehensively tested before general end-user deployment.

interactiveAuthenticationRetries

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/interactiveAuthenticationRetries

The value of the mandatory element *interactiveAuthenticiationRetries* specifies the number of times SSC retries after a failed authentication session. Interactive applies for cases in which a user intervention might correct the fault. In general, this applies to connection attempts involving user text entry or list selection associated with an Enter Your Credentials dialog that allow for user corrections.

For wireless networks, even though *interactiveAuthenticiationRetries* is configured on an individual network basis, only one global setting applies to all networks. After deployment, SSC extracts the maximum value entered for all configured networks and uses that for its global value.

For wired networks, the value used by SSC is based on the one configured in your single wired network and is independent of any configured wireless networks.

When a distribution package file does not contain a network for a particular media type, networks created for that particular network type by the end-user when allowed by license and policy are assigned the system default value of four.

Restriction: the number of retries is limited to 99.

Default: Cisco recommends the value 4. This supports the Failed Authentication VLAN feature of Cisco switches. Set the supplicant to be one more than what your switch is set to for retries. This is so that SSC tries one more time to get onto the restricted VLAN. Increasing the interactive retry count will result in more user dialog prompts.

nonInteractiveAuthenticationRetries

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/nonInteractiveAuthenticationRetries

The value of the mandatory element *nonInteractiveAuthenticiationRetries* specifies the number of times SSC retries after a failed authentication session. Noninteractive applies for cases in which a user intervention would not help to correct the fault. In general, this applies to connection attempts not involving an Enter Your Credentials dialog, such as, single-sign-on, a PSK mismatch, or all failures associated with a server certificate validation.

For wireless networks, even though *nonInteractiveAuthenticiationRetries* is configured on an individual network basis, only one global setting applies to all networks. After deployment, SSC extracts the maximum value entered for all configured networks and uses that for its global value.

For wired networks, the value used by SSC is based on the one configured in your single wired network and is independent of any configured wireless networks.

When a distribution package file does not contain a network for a particular media type, networks created for that particular network type by the end-user when allowed by license and policy are assigned the system default value of four.

Restriction: the number of retries is limited to 99.

Default: Cisco recommends the value 4. This supports the Failed Authentication VLAN feature of Cisco switches. Set the supplicant to be one more than what your switch is set to for retries. This is so that SSC tries one more time to get onto the restricted VLAN. Increasing the noninteractive retry count will add delay to the machine boot or user login connection to the system in cases where network connectivity fails.

Step 2 Configure the following server validation element:

serverValidation

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation

The optional element *serverValidation* forms the container for configuring validation of the server certificate used during authentication. No element values are specified.

- Add this element when requiring server certificate validation.
- Omit this element when not requiring server certificate validation.

Business rule: at least one of the optional child elements, *validationRules* or *trustedServerIds*, must be specified.

Next item: "Configuring Server Validation'

Configuring Server Validation

Step 1 Configure your trusted server list as follows:

a. When using a server certificate and requiring its validation, perform the tasks defined in section "Configuring Certificate Trusted Server Rules'.

b. When using EAP-FAST with anonymous (unauthenticated), autonomous PAC provisioning or manual PAC provisioning, perform the tasks defined in section "Configuring PAC Trusted Server Rules'.

All deployed trusted server rules are of type Machine / All Users (public profiles) as displayed in the SSC's user interface. Deployed rules are locked and can not be modified.

Additionally, even though *serverValidation* is configured on an individual network basis, all deployed trusted server rules apply to all networks. (In other words, their use in Release 4.1 is the same as earlier releases, that is, the Release 4.0.x series.)

Step 2 Configure the deployment method for your Trusted Certificate Authority (CA) certificates as follows:

Specifying one of the following methods for deploying any required CA certificates:

- Independently deploy—Use element *trustAnyRootCaFromOs*.
- Deploy as part of the distribution package—Use element *trustedRootCaCerts*.

trustAnyRootCaFromOs

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustAnyRootCaFromOs

The existence of the optional element *trustAnyRootCaFromOs* specifies that any Trusted Root CA and any Intermediate CA certificates used to trust the server certificate have been placed in the proper Windows Certificate Store by an independent deployment process. It is an empty element with no value.

This corresponds to CA certificate deployment for SSC earlier than Release 4.1.0.

trustedRootCaCerts

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedRootCaCerts

The existence of the optional element *trustedRootCaCerts* specifies a container for direct deployment of any Trusted Root CA and any Intermediate CA certificates used to trust the server certificate. No element value is specified.

With this option, after deployment, SSC automatically places the certificates in the Windows Trusted Authority (Trusted Root CA) Certificate Store. However, it maintains knowledge of these deployed certificates and uses that to filter the contents of the Windows store when authenticating a connection.

Even though *trustedRootCaCerts* is configured on an individual network basis, all deployed CA certificates apply to all networks. For example, if one deploys CA1 to network1 and CA2 to network2, when authenticating to network1 both CA1 and CA2 would validate network1's server certificate, and so forth.

Next item: "Adding CA Certificates".

Note Self-signed certificates:
A server certificate may be signed by itself (having a certificate chain length of 0). SSC will trust such certificates if they appear in the list of trusted root entities in the appropriate store.

The following example illustrates the distribution package XML for the two CA certificate source options for the *serverValidation* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-21 serverValidation

```
<serverValidation>
    <validationRules>
        {child element}
    </validationRules>
    <trustedServerIds>
        {child element}
    </trustedServerIds>
    <trustedRootCaCerts>
        {child element}
    </trustedRootCaCerts>
</serverValidation>

<serverValidation>
    <validationRules>
        {child element}
    </validationRules>
    <trustedServerIds>
        {child element}
    </trustedServerIds>
    <trustAnyRootCaFromOs/>
</serverValidation>
```

Configuring Certificate Trusted Server Rules

When using a server certificate and requiring its validation, configure the following elements:

This use includes EAP-FAST with authenticated, autonomous PAC provisioning. In this case validating a server certificate will transfer that trust to an automatically created PAC rule. In other words, you do not have to additionally specify the element *trustedServerId*.

validationRules

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/validationRules

The element *validationRules* forms the container for specifying certificate validation rules. Since certificates are allowed to use different sets of optional attributes, you may specify the specific certificate attribute(s) to use in the validation rule. Specify one of more of the following rule types:

- A rule based on the certificate field, Subject Alternative Name—Use element *matchSubjectAlternativeName*.

- A rule based on the certificate field, Subject—Use element *matchSubjectName*.

Business rule: at least one child element must be specified.

matchSubjectAlternativeName

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/validationRules/matchSubjectAlternativeName

The existence of this optional element implies that you are specifying to search the following certificate field:

- Subject Alternative Name: DNSName.

  This typically takes the form of a Fully Qualified Domain Name (FQDN) - a domain name including all higher level domain names up to the top-level (root) domain name, for example, engr.mycompany.com.

You may add as many *matchSubjectAlternativeName* elements as required.

The element value contains the text string that is used in a comparison check with the contents of the server certificate.

Values for required attributes:

- **name**—the value of this attribute specifies a display name for the rule.
- **match**—the value of this attribute specifies how the configured text is used in the comparison test.
  - exactly—the certificate field must contain the full value of the element *matchSubjectAlternativeName*
  - endsWith—the certificate field must end with the value of the element *matchSubjectAlternativeName*

    This value is typically used to quantify a FQDN - for example, if *matchSubjectAlternativeName* contains "mycompany.com", certificates with the Subject Alternative Names engr.mycompany.com or mkrt.mycompany.com would be considered trusted.

matchSubjectName

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/validationRules/matchSubjectName

The existence of this optional element specifies that you are searching the following certificate fields:

- Subject: CN (Common Name).

  This is typically a simple ASCII string. If multiple Common Names are specified, all those listed in the certificate are searched.

- Subject: DN (Domain Name) - a composite of a set of DC (Domain Component) attributes; for example, a DC set of DC=Mycompany, DC=com, results in a Domain Name of Mycompany.com.

  Therefore, this field typically represents a Fully Qualified Domain Name (FQDN) - a domain name including all higher level domain names up to the top-level (root) domain name, for example, engr.mycompany.com.

You may add as many *matchSubjectName* elements as required.

The element value contains the text string that is used in a comparison check with the contents of the server certificate.

Values for required attributes:

- **name**—The value of this attribute specifies a display name for the rule.
- **match**—The value of this attribute specifies how the configured text is used in the comparison test.
  - exactly—The certificate field must contain the full value of the element *matchSubjectName.*
  - endsWith—The certificate field must end with the value of the element *matchSubjectName.*

    This value is typically used to quantify a FQDN; for example, if *matchSubjectName* contains "mycompany.com", certificates with the Subject engr.mycompany.com or mkrt.mycompany.com would be considered trusted.

The following example illustrates the distribution package XML for the *validationRules* element and its child elements. The order or the number of the child elements is not restricted.

Example 2-22 validationRules

```
<validationRules>
    <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
    <matchSubjectName name="Cert Rule 2" match="exactly">My Corporation</matchSubjectName>
    <matchSubjectAlternativeName name="Cert Rule 3"
match="endsWith">myCorp2.net</matchSubjectAlternativeName>
</validationRules>
```

Configuring PAC Trusted Server Rules

When you are using EAP-FAST with anonymous (unauthenticated), autonomous PAC provisioning, configure the following elements:

Note When you are using EAP-FAST with authenticated, autonomous PAC provisioning, this element is not required because SSC will transfer trust from your server certificate (*validationRules*) to the PAC.

trustedServerIds

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds

The mandatory element *trustedServerIds* forms the container for specifying FAST PAC validation rules. No element value is specified.

Configure the following child element:

trustedServerId

Schema path:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds/trustedServerId

The mandatory element *trustedServerId* forms the container for an individual FAST PAC validation rule. No element value is specified.

You may add as many *trustedServerId* elements as required.

Values for required attributes:

- **name**—the value of this attribute specifies a display name for the rule in the user interface.

Specify one of the following sources for the PAC rule information:

- Let the postprocess utility enter the data—Use element *reference*. (recommended)
- Enter the data yourself—Use elements *aid* and *aidInfo*.

reference

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds/trustedServerId/reference

The element *reference* forms the container for a PAC file identification. No element value is specified.

Configure the child elements.

aidReference

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds/trustedServerId/reference/aidReference

The value of the element *aidReference* specifies the full path to your PAC. The file must be accessible by the postprocess sscPackageProcess utility. The postprocess tool for the XML distribution package file retrieves the PAC file, automatically extracts the rule information, and substitutes for the *aidReference* element the *aid* and *aidInfo* elements. Any optional *secretKey* element is removed from the processed distribution package file.

Tip It's sufficient for the referenced PAC file to be any Cisco ACS exported FAST PAC file with the correct server A-ID. The PAC file itself is not part of the distribution package.

secretKey

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds/trustedServerId/reference/secretKey

The option element *secretKey* is specified when the PAC identified in *aidReference* has been read-protected with a key. The value of the element contains the key value and allows the postprocess tool to access the desired information.

aid

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds/trustedServerId/aid

The value of the element *aid* specifies the Authority Identity (A-ID) for the PAC.

Restriction: HEX formatted.

aidInfo

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedServerIds/trustedServerId/aidInfo

The value of the element *aidInfo* specifies a friendly name for the A-ID for the PAC.

Restriction: ASCII formatted.

The following example illustrates the distribution package XML for the *trustedServerId* element and its child element. The order of the multiple child elements is restricted to that shown.

Example 2-23 trustedServerId

```
<trustedServerIds>
    <trustedServerId name="PAC AID Rule 1">
        <reference>
            <aIdReference>E:\path\pacFile1</aIdReference>
            <secretKey>1234</secretKey>
        </reference>
    </trustedServerId>
    <trustedServerId name="PAC AID Rule 2">
        <reference>
            <aIdReference>E:\path\pacFile2</aIdReference>
        </reference>
    </trustedServerId>
</trustedServerIds>

<trustedServerIds>
    <trustedServerId name="PAC AID Rule 1">
        <aid>9eb4674987654a4796f62abc6e403060</aid>
        <aidInfo>Corp ACS</aidInfo>
    </trustedServerId>
</trustedServerIds>
```

Adding CA Certificates

Configure the following elements:

certificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedRootCaCerts/certificate

The mandatory element *certificate* forms the container for the contents of a CA certificate. After deployment, SSC automatically places the certificate in the Windows Trusted Authority (Trusted Root CA) Certificate Store. No element value is specified.

You may add as many *certificate* elements as required.

Specify and configure the following element:

caReference

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/serverValidation/trustedRootCaCerts/certificate/caReference

The value of the element *caReference* specifies the full path to your CA certificate. The file must be accessible by the postprocess sscPackageProcess utility. The postprocess tool for the XML distribution package file retrieves the certificate file, automatically encodes (base64 string) the information, populates the *content* element and substitutes for the *caReference* element the *content* element.

The certificate file format supported is .pem. The *content* element's *format* attribute is added and configured with value pem.

The following example illustrates the distribution package XML for the *certificate* element and its child element.

Example 2-24 certificate

```
<certificate>
    <caReference>E:\path\CaCertFile.pem</caReference>
</certificate>
```

Choosing the Authentication Network's Connection Context

Specify one of the following connection contexts for the network:

- Machine-only connection — Use element *machineAuthentication.*
- User-only connection — Use element *userAuthentication*.
- Machine/User connection — Use element *machineUserAuthentication*.

machineAuthentication

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication

The optional element *machineAuthentication* forms the container for configuring a network that supports only a machine context connection. A connection is made at system boot using the configured machine credentials and is maintained when users log into or log off of the system. In the *Cisco Secure Services Client User Guide* this is referred to as an extended machine only context connection. No element values are specified.

Next item: "Configuring an Authenticating, Machine-only Network'.

userAuthentication

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication

The optional element *userAuthentication* forms the container for configuring a network that supports only a user context connection. A connection is made when a user logs into the system using the configured user credentials and is maintained until the user logs off of the system. In the *Cisco Secure Services Client User Guide* this is referred to as a user only context connection. No element values are specified.

Next item: "Configuring an Authenticating, User-Only Network'.

machineUserAuthentication

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication

The optional element *machineUserAuthentication* forms the container for configuring a network that supports both a machine context and a user context connection. A connection is made at system boot using the configured machine credentials and is maintained when a user logs into the system based on a successful reauthentication using the configured user credentials. When the user logs off of the system the machine connection is restored. In the *Cisco Secure Services Client User Guide* this is referred to as a machine and user context connection. No element values are specified.

Next item: "Configuring an Authenticating, Machine and User Network'.

The following example illustrates the distribution package XML for the three connection contexts for the *authenticationNetwork* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-25 authenticationNetwork

```
<authenticationNetwork>
    <machineAuthentication>
        {child elements}
    </machineAuthentication>
    <serverValidation>
        {child elements}
    </serverValidation>
    <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
    <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
    <associationMode>
        {child element}
    </associationMode>
</authenticationNetwork>

<authenticationNetwork>
    <userAuthentication>
        {child elements}
    </userAuthentication>
    <serverValidation>
        {child elements}
    </serverValidation>
    <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
    <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
    <associationMode>
        {child element}
    </associationMode>
</authenticationNetwork>

<authenticationNetwork>
    <machineUserAuthentication>
        {child elements}
    </machineUserAuthentication>
    <serverValidation>
        {child elements}
    </serverValidation>
    <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
    <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
    <associationMode>
        {child element}
    </associationMode>
</authenticationNetwork>
```

Configuring an Authenticating, Machine-only Network

Follow these steps to configure an authenticating, machine-only context Wi-Fi network.

Step 1 Perform the tasks defined in "Configuring the Authenticating, Machine Credential Source Elements".

Step 2 Perform the tasks defined in "Configuring the Authentication Static Credential Elements".

The following example illustrates the distribution package XML for *machineAuthentication* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-26 machineAuthentication

```
<machineAuthentication>
    <collectionMethod>
        {child elements}
    </collectionMethod>
    <useAnonymousId>true</useAnonymousId>
    <staticIdentity encrypt="true">machineName</staticIdentity>
    <staticPassword encrypt="true">machineSecret</staticPassword>
    <eapMethods>
        {child elements}
    </eapMethods>
</machineAuthentication>
```

Configuring the Authenticating, Machine Credential Source Elements

Configure the following element:

collectionMethod

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication/collectionMethod

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/machine/collectionMethod

The mandatory element *collectionMethod* forms the container for configuring the source and type of the machine credentials. It is an empty element with no value.

Specify one of the following credential types for the machine connection:

- Active Directory—Use element *auto*.
- Predefined—Use element *static*.

auto

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication/collectionMethod/auto

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/machine/collectionMethod/auto

The existence of the optional element *auto* specifies the use of the Microsoft Active Directory (AD) provided machine credentials. The following types are supported:

- Machine certificate—must be used with a TLS-based EAP method.

  Normally only a single certificate is stored here. However, for cases in which multiple certificates are present the first valid certificate found is used (for example, temporary overlap while provisioning a newer certificate to replace an old one or provisioning from different certificate authorities).

  The certificate must not require a PIN or have strong private key protection.

  Note The machine identity is provided from the machine certificate (the dnsName field of the Subject Alternative Name).

- Machine password—must be used with a password-based EAP method such as EAP-MSCHAPv2.

Note The domain controller containing the computer must be performing the machine authentication. The policy for the computer must automatically enroll the computer for a machine certificate or password.

It is an empty element with no value.

static

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication/collectionMethod/static

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/machine/collectionMethod/static

The existence of the optional element *static* specifies the use of predefined credentials that are deployed as part of this distribution package (configuration file). It is an empty element with no value.

Business rule: With this option, you must add and configure the elements discussed in "Configuring the Authentication Static Credential Elements".

Business rule: Static credentials for machine authentication require a password-based only EAP method, excluding the use of client certificates and EAP FAST PACs. Therefore the following methods are not allowed: EAP TLS, EAP FAST, and EAP PEAP with an inner method of EAP TLS.

The following example illustrates the distribution package XML for the two choices for the *collectionMethod* element and its child element choices.

Example 2-27 collectionMethod (machine)

```
<collectionMethod>
    <auto/>
</collectionMethod>

<collectionMethod>
    <static/>
</collectionMethod>
```

Configuring the Authenticating, Connection Independent Base Elements

Configure the following elements:

unprotectedIdentityPattern

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/
[machineAuthentication | userAuthentication | machineUserAuthentication/machine |
machineUserAuthentication/user]/
unprotectedIdentityPattern

The mandatory element *unprotectedIdentityPattern* specifies the content of the initial EAP Response/Identity message. For tunneled EAP methods this represents the phase 1, outer (unprotected) tunnel. An identity has a Network Access Identifier (NAI) format and typically takes the following generalized form: Username@Domain, where the use of the @Domain (also referred to as the realm) is optional. The actual syntax and data content of an EAP Identity is based on the requirements of your specific authentication server.

The value of *unprotectedIdentityPattern* is sent in the clear.

The value of the element consists of fixed text that you explicitly define and placeholder keywords that represent variable values for certain standard components of an NAI that are dynamically supplied by SSC at the time of authentication processing. This flexibility allows you to use any standard or special formats compatible with your credential storage environment and the requirements of your authentication server. The placeholder keywords are bracketed by the following characters: < >. Keywords are restricted by the connection context in which they are specified and the associated credential collection method. Allowed keyword values and their restrictions are as follows:

- <username>—the identity of the user currently attempting a login
  - Allowed connection contexts—userAuthentication and machineUserAuthentication/user
  - Allowed credential collection methods—prompt and singleSignOn
- <domain>—any realm entered with the user's identity
  - Allowed connection contexts—userAuthentication and machineUserAuthentication/user
  - Allowed credential collection methods—prompt and singleSignOn
- <fqhn>—fully qualified host name of the machine

  For example, a machine cat in domain mydomain.company.com is seen as cat.mydomain.company.com.
  - Allowed connection contexts—machineAuthentication and machineUserAuthentication/machine
  - Allowed credential collection methods—auto
- <mac>—MAC address of the associated network adapter
  - unrestricted usage

Note In order to avoid conflicts with the XML syntax, the keyword bracketing characters, which are illegal XML characters, require special handling. You may replace them with their entity reference. In your distribution package xml file the keyword bracketing characters may be substituted as follows:

- use < for the character <
- use > for the character >

Alternately, the entire element value may be placed within a CDATA construct as follows:

```
<unprotectedIdentityPattern>
   <![CDATA[
      <username>@<domain>
   ]]>
</unprotectedIdentityPattern>
```

Optional substitution of a placeholder keyword and any associated text is indicated when bracketed by the following characters: [ ]. If the value of the bracketed keyword is not known at the time of authentication, then the entire contents are omitted.

When using static credentials, as signified by choosing the child element *static* for the associated *collectionMethod* element, the value of *unprotectedIdentityPattern* represents the static identity.

Restriction: The identity specified may contain up to 63 ASCII characters and is case sensitive.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess utility.

protectedIdentityPattern

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/
[machineAuthentication | userAuthentication | machineUserAuthentication/machine |
machineUserAuthentication/user]/
protectedIdentityPattern

The optional element *protectedIdentityPattern* specifies the content of the EAP Identity response of any phase 2, inner tunnel (protected identity). Therefore its use is restricted to authentication methods that use tunneling.

Business rule: the corresponding *eapMethods* element must specify only EAP FAST, PEAP or TTLS.

The value of *protectedIdentityPattern* is sent encrypted within the EAP tunnel.

The value of the element has the same format specification as described above for the *unprotectedIdentityPattern* element.

When using static credentials, as signified by choosing the child element *static* for the associated *collectionMethod* element, the value of *protectedIdentityPattern* represents the static identity.

Restriction: The identity specified may contain up to 63 ASCII characters and is case sensitive.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess utility.

The following examples illustrate the distribution package XML for several different applications.

Example 2-28 userAuthentication-Tunneled Authentication Method

```
<userAuthentication>
    <autoConnect>
        <connectBeforeLogon>
    </autoConnect>
    <collectionMethod>
        <prompt/>
    </collectionMethod>
    <unprotectedIdentityPattern
encrypt="true">anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
    <protectedIdentityPattern encrypt="true">&lt;username&gt;</protectedIdentityPattern>
    <eapMethods>
        {child elements}
    </eapMethods>
</userAuthentication>
```

Example 2-29 userAuthentication-Tunneled Authentication Method, Legacy NT4 Format

```
<userAuthentication>
    <autoConnect>
        <connectBeforeLogon>
    </autoConnect>
    <collectionMethod>
        <prompt/>
    </collectionMethod>
    <unprotectedIdentityPattern
encrypt="true">&lt;domain&gt;\&lt;username&gt;</unprotectedIdentityPattern>
    <protectedIdentityPattern encrypt="true">&lt;username&gt;</protectedIdentityPattern>
    <eapMethods>
        {child elements}
    </eapMethods>
</userAuthentication>
```

Example 2-30 userAuthentication-Tunneled Authentication Method, Static Credentials

```
<userAuthentication>
    <autoConnect>
        <connectBeforeLogon>
    </autoConnect>
    <collectionMethod>
        <static/>
    </collectionMethod>
    <unprotectedIdentityPattern encrypt="true">anonymous</unprotectedIdentityPattern>
    <protectedIdentityPattern encrypt="true">employeeName</protectedIdentityPattern>
    <staticPassword encrypt="true">employeeSecret</staticPassword>
    <eapMethods>
        {child elements}
    </eapMethods>
</userAuthentication>
```

Example 2-31 machineAuthentication-Tunneled Authentication Method

```
<machineAuthentication>
    <collectionMethod>
        <auto/>
    </collectionMethod>
    <unprotectedIdentityPattern encrypt="true">host/anonymous</unprotectedIdentityPattern>
    <protectedIdentityPattern encrypt="true">host/&lt;fqhn&gt;</protectedIdentityPattern>
    <eapMethods>
        {child elements}
    </eapMethods>
</machineAuthentication>
```

Example 2-32 userAuthentication-Authenticated Local Login

The administrator has a local login on a machine (a non active directory account) and expects that account to have network connectivity authenticating against the 802.1X network. Under normal network login circumstances the value of <domain> is known. But during the local login its value is nondeterministic.

```
<userAuthentication>
    <autoConnect>
        <connectBeforeLogon>
    </autoConnect>
    <collectionMethod>
        <prompt/>
    </collectionMethod>
    <unprotectedIdentityPattern
encrypt="true">&lt;username&gt;[@&lt;domain&gt;]</unprotectedIdentityPattern>
    <protectedIdentityPattern encrypt="true">&lt;username&gt;</protectedIdentityPattern>
    <eapMethods>
        {child elements}
    </eapMethods>
</userAuthentication>
```

Configuring the Authentication Static Credential Elements

Configure the following element:

Business rule: This element is required only if the corresponding credentials *collectionMethod* element is configured as static.

staticPassword

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/[machineAuthentication | userAuthentication | machineUserAuthentication/machine | machineUserAuthentication/user]/staticPassword

The value of the optional element *staticPassword* specifies the password shared secret.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Restriction: The password specified may contain up to 80 ASCII characters and is case sensitive.

Note The use of static credentials in a user-context connection may require individualized values in the distribution package file. In this case, each end-user has a different file and global deployment of a common distribution package file is not applicable.

Configuring an Authenticating, User-Only Network

Follow these steps to configure an authenticating, user-only context Wi-Fi network.

Step 1 Perform the tasks defined inthe "Configuring the Authenticating, User-Only Connection Occurrence Elements" section on page 2-49.

Step 2 Perform the tasks defined in the "Configuring the Authenticating, User Credential Source (1) Elements" section on page 2-50.

Step 3 Perform the tasks defined inthe "Configuring the Authenticating, Connection Independent Base Elements" section on page 2-44.

Step 4 Perform the tasks defined in the "Configuring the Authentication Static Credential Elements" section on page 2-47.

Step 5 Perform the tasks defined in the "Configuring the FAST PAC Elements" section on page 2-55.

The following example illustrates the distribution package XML for the two connection occurrence options for the *userAuthentication* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-33 userAuthentication

```
<userAuthentication>
    <autoConnect>
        {child element}
    </autoConnect>
    <collectionMethod>
        {child element}
    </collectionMethod>
    <useAnonymousId>true</useAnonymousId>
    <staticIdentity encrypt="true">userName</staticIdentity>
    <staticPassword encrypt="true">userSecret</staticPassword>
    <pacs>
        {child elements}
    </pacs>
    <eapMethods>
        {child elements}
    </eapMethods>
</userAuthentication>

<userAuthentication>
    <manualConnect/>
    <collectionMethod>
        {child element}
    </collectionMethod>
    <useAnonymousId>true</useAnonymousId>
    <staticIdentity encrypt="true">userName</staticIdentity>
    <staticPassword encrypt="true">userSecret</staticPassword>
    <pacs>
```

```
        {child elements}
    </pacs>
    <eapMethods>
        {child elements}
    </eapMethods>
</userAuthentication>
```

Configuring the Authenticating, User-Only Connection Occurrence Elements

Specify one of the following connection initiation types for the user connection:

- Auto-connect—Use element *autoConnect*.
- Manual-connect—Use element *manualConnect*.

autoConnect

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/autoConnect

The existence of the optional element *autoConnect* specifies that an attempt to make a user-context network connection will be automatically initiated when the user logs into the system. No element value is specified.

Configuration of the following child element is required:

connectBeforeLogon

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/autoConnect/connectBeforeLogon

The mandatory boolean element *connectBeforeLogon* specifies when the connection attempt is made with respect to the login request.

The element has the following values:

- True—Attempt to connect to the network before the user logs into Windows.

  Use this option only if this network is going to be used in a domain login environment such as the Microsoft Active Directory or Novell domains, and an early network connection is required; for example, use it to support the use of specific Microsoft Group Policy Objects (GPO).

  Only password and smartcard credentials are supported with auto-connection in this type of network arrangement. (Specifically, a Windows User-Personal Certificate Store credential is not supported because it cannot be accessed prior to the user login completion.)
  Business rule: The corresponding *certificateSource* element, if present, must have the *smartCardOnlyCertificate* child element chosen.

- False—Attempt to connect to the network after the user successfully logs into Windows at the desktop.

  Use this option when an early network connection is not required. In this mode there are no restrictions on user credential types allowed.

The following example illustrates the distribution package XML for the *autoConnect* element and its child element.

Example 2-34 autoConnect

```
<autoConnect>
    <connectBeforeLogon>true</connectBeforeLogon>
</autoConnect>
```

manualConnect

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/manualConnect

The existence of the optional element *manualConnect* specifies that no attempt to make a user-context network connection will be automatically initiated when the user logs into the system. The user, at the desktop, must open SSC and manually select and connect to the network. It is an empty element with no value.

Configuring the Authenticating, User Credential Source (1) Elements

Configure the following element:

collectionMethod

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod

The mandatory element *collectionMethod* forms the container for configuring the source and type of the user credentials. It is an empty element with no value.

Specify one of the following credential types for the user connection:

- User on-demand provided—Use element *prompt*.
- Operating system credentials—Use element *singleSignOn*.
- Predefined—Use element *static*.
- Machine Active Directory—Use element *autoMachine*.

prompt

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod/prompt

The existence of the optional element *prompt* specifies that credentials will be requested from the user at the time of the connection attempt. No element value is specified.

Next item: "Choosing Prompted Credential Storage" section on page 2-53.

singleSignOn

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod/singleSignOn

The existence of the optional element *singleSignOn* specifies that the credentials, specifically username and password, entered by a user for the operating system login will also be used for authentication. No SSC provisioning is required. This is often referred to as single-signon. SSC supports single-signon authentication based on the user's Windows or Novell login name and password. It is an empty element with no value.

static

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod/static

The existence of the optional element *static* specifies the use of predefined credentials that are deployed as part of this distribution package (configuration file) and permanently saved (or at least until they are updated). It is an empty element with no value.

Business rule: With this option, you must add and configure the elements of the "Configuring the Authentication Static Credential Elements" section on page 2-47.

Business rule: Static credentials require a password-based EAP method, excluding the use of client certificates. Therefore the following methods are not allowed: EAP TLS, EAP PEAP or FAST with an inner method of EAP TLS.

autoMachine

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod/autoMachine

The existence of the optional element *autoMachine* specifies the use of any existing Active Directory machine credentials. In the *Cisco Secure Services Client User Guide* this is referred to as a user only machine context connection. It is an empty element with no value.

Note This option maintains compatibility with the earlier 4.0.x releases of SSC. The newly supported static option will replace it for most environments.

The following example illustrates the distribution package XML for the four choices for the *collectionMethod* element and its child element.

Example 2-35 collectionMethod (user)

```
<collectionMethod>
    <prompt>
        <credentialsStorage>
            {child elements}
        </credentialsStorage>
    </prompt>
</collectionMethod>

<collectionMethod>
    <singleSignOn/>
</collectionMethod>
```

```
<collectionMethod>
    <static/>
</collectionMethod>

<collectionMethod>
    <autoMachine/>
</collectionMethod>
```

Configuring the Authenticating, User Credential Source (2) Elements

Configure the following element:

collectionMethod

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod

The mandatory element collectionMethod forms the container for configuring the source and type of the user credentials. It is an empty element with no value.

Specify one of the following credential types for the user connection:

- User on-demand provided—Use element *prompt*.
- Operating system credentials—Use element *singleSignOn*.
- Predefined—Use element *static*.

prompt

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/prompt

The existence of the optional element *prompt* specifies that credentials will be requested from the user at the time of the connection attempt. No element value is specified.

Next item: "Choosing Prompted Credential Storage" section on page 2-53.

singleSignOn

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/singleSignOn

The existence of the optional element *singleSignOn* specifies that the credentials, specifically username and password, entered by a user for the operating system login will also be used for authentication. No SSC provisioning is required. This is often referred to as single-signon. SSC supports single-signon authentication based on the user's Windows or Novell login name and password. It is an empty element with no value.

static

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/static

The existence of the optional element *static* specifies the use of predefined credentials that are deployed as part of this distribution package (configuration file) and permanently saved (or at least until they are updated). It is an empty element with no value.

Business rule: With this option, you must add and configure the elements of section "Configuring the Authentication Static Credential Elements" section on page 2-47.

Business rule: Static credentials require a password-based EAP method. Therefore the following methods are not allowed: EAP TLS, EAP PEAP or FAST with an inner method of EAP TLS.

The following example illustrates the distribution package XML for the three choices for the *collectionMethod* element and its child element.

Example 2-36 collectionMethod (machine/user)

```
<collectionMethod>
    <prompt>
        <credentialsStorage>
            {child elements}
        </credentialsStorage>
    </prompt>
</collectionMethod>

<collectionMethod>
    <singleSignOn/>
</collectionMethod>

<collectionMethod>
    <static/>
</collectionMethod>
```

Choosing Prompted Credential Storage

Configure the following element:

credentialsStorage

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod/prompt/credentialsStorage

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/prompt/credentialsStorage

The mandatory element *credentialsStorage* forms the container for configuring the storage time of the user-prompted credentials. It is an empty element with no value.

Specify one of the following credential storage times for the prompted credentials:

- Save forever—Use element *forever*.
- Save while logged in—Use element *logonSession*.
- Save for timed duration—Use element *duration*.

Business rule: Only credential storage methods supported by the policy are permitted. See element allowedCredentialStorage in the "Network Policy" section on page 2-6.

forever

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/collectionMethod/prompt/credentialsStorage/forever

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/prompt/credentialsStorage/forever

The existence of the optional element *forever* specifies that the prompted credentials will be saved forever (or at least until they are updated). Once stored the usage is the same as if the credentials were statically deployed. It is an empty element with no value.

logonSession

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthenticationcollectionMethod/prompt/credentialsStorage/logonSession

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/prompt/credentialsStoragelogonSession

The existence of the optional element *logonSession* specifies that the prompted credentials will not be saved beyond the current login session. The user must re-enter credentials each time he or she logs into the system. It is an empty element with no value.

duration

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthenticationcollectionMethod/prompt/credentialsStorage/duration

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/collectionMethod/prompt/credentialsStorage/duration

The existence of the optional element *duration* specifies that the prompted credentials will be saved for a timed period configured in the network policy. While a user session is in progress expiration of the duration time does not itself cause any credential prompting. However, after the time-out, re-authentication requests will force the user to re-enter the credentials via the Request for Credentials dialog. Re-entry will restart the duration timer. It is an empty element with no value.

The following example illustrates the distribution package XML for the three configuration options for the *credentialStorage* element and its child element.

Example 2-37 credentialsStorage

```
<credentialsStorage>
    <forever/>
</credentialsStorage>

<credentialsStorage>
    <logonSession/>
</credentialsStorage>

<credentialsStorage>
    <duration/>
</credentialsStorage>
```

Configuring the FAST PAC Elements

Configure the following element:

pacs

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/pacs

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/pacs

The optional element *pacs* forms the container for supporting the manual provisioning by means of this deployment distribution package (configuration file) of the user's EAP-FAST Tunnel PAC. No element value is specified.

Business rule: Existence of this element implies that EAP-FAST must be configured for this network. Specifically, for the same wifiNetwork or wiredNetwork element, the corresponding *eapMethods*/*eapFast* element must exist.

Configuration of the following child element is required:

pac

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/pacs/pac

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/pacs/pac

The mandatory element *pac* forms the container for an individual Tunnel PAC's information. No element value is specified.

You may add as many *pac* elements as required.

Perform the following steps to configure the Tunnel PAC's information:

Step 1 Specify and configure the following element:

pacReference

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/pacs/pac/pacReference

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/pacs/pac/pacReference

The value of the element *pacReference* specifies the full path to the location of the PAC file. The file must be accessible by the postprocess sscPackageProcess utility. The postprocess tool for the XML distribution package file retrieves the PAC file, automatically encodes (base64 string) the information, populates the *content* element and substitutes for the *pacReference* element the *content* element.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Step 2 Configure the following element:

secretKey

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/userAuthentication/pacs/pac/secretKey

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/pacs/pac/secretKey

The value of the optional element *secretKey* specifies the secret (password) if the PAC was created as password-protected by the Cisco ACS.

This element has a required boolean attribute, encrypt, which has a fixed value of True. It indicates that this element needs to be (or has been) encrypted by the postprocess sscPackageProcess utility.

Note The use of the distribution package to perform manual provisioning of the user's tunnel PAC requires individualized values in the distribution package file. Each end-user has a different file and global deployment of a common distribution package file is not applicable.

The following example illustrates the distribution package XML for the *pacs* element and its child element. The order of the multiple child elements is restricted to that shown.

Example 2-38 pacs

```
<pacs>
    <pac>
        <pacReference encrypt="true">E:\path\pacFile</pacReference>
    </pac>
</pacs>

<pacs>
    <pac>
        <pacReference encrypt="true">E:\path\pacFile</pacReference>
        <secretKey encrypt="true">my pac secret</secretKey>
    </pac>
</pacs>
```

Configuring an Authenticating, Machine and User Network

Follow these steps to configure an authenticating, machine and user context Wi-Fi network.

Step 1 Configure the machine portion.

a. Perform the tasks defined in the "Configuring the Authenticating, Machine Credential Source Elements" section on page 2-42.

b. Perform the tasks defined in the "Configuring the Authenticating, Connection Independent Base Elements" section on page 2-44.

c. Perform the tasks defined in the "Configuring the Authentication Static Credential Elements" section on page 2-47.

Step 2 Configure the user portion.

a. Perform the tasks defined in the "Configuring the Authenticating, User Connection Occurrence Elements" section on page 2-58.

a. Perform the tasks defined in the "Configuring the Authenticating, User Credential Source (2) Elements" section on page 2-52.

b. Perform the tasks defined in the "Configuring the Authenticating, Connection Independent Base Elements" section on page 2-44.

c. Perform the tasks defined in the "Configuring the Authentication Static Credential Elements" section on page 2-47.

d. Perform the tasks defined in the "Configuring the FAST PAC Elements" section on page 2-55.

The following example illustrates the distribution package XML for the *machineUserAuthentication* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-39 userAuthentication

```
<machineUserAuthentication>
    <machine>
        <collectionMethod>
            {child element}
        </collectionMethod>
        <useAnonymousId>true</useAnonymousId>
        <staticIdentity encrypt="true">machineName</staticIdentity>
        <staticPassword encrypt="true">machineSecret</staticPassword>
    </machine>
    <user>
        <autoConnect>true</autoConnect>
        <collectionMethod>
            {child element}
        </collectionMethod>
        <useAnonymousId>true</useAnonymousId>
        <staticIdentity encrypt="true">userName</staticIdentity>
        <staticPassword encrypt="true">userSecret</staticPassword>
        <pacs>
            {child elements}
        </pacs>
    </user>
    <eapMethods>
        {child elements}
    </eapMethods>
</machineUserAuthentication>
```

Configuring the Authenticating, User Connection Occurrence Elements

Specify one of the following connection initiation types for the user connection:

- Auto-connect—Use element *autoConnect*.
- Manual-connect—Use element *manualConnect*.

autoConnect

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/autoConnect

The existence of the optional element *autoConnect* specifies that an attempt to make a user-context network connection will be automatically initiated when the user logs into the system. No element value is specified.

Configuration of the following child element is required:

connectBeforeLogon

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/autoConnect/connectBeforeLogon

The mandatory boolean element *connectBeforeLogon* specifies when the connection attempt is made with respect to the login request.

The element has the following values:

- True—Attempt to connect to the network before the user logs into Windows.

  Use this option if this network is going to be used in a domain login environment such as the Microsoft Active Directory domain, and an early network connection is required. This option allows for:

  - processing of specific Microsoft Group Policy Objects (GPO) over the user-context connection
  - accommodating an IP address change if required when transferring between the machine and user contexts, for example, if accompanied by a VLAN change.

  Only password and smartcard credentials are supported with auto-connection in this type of network arrangement. (Specifically, a Windows User-Personal Certificate Store credential is not supported because it cannot be accessed prior to the user login completion.)
  Business rule: The corresponding *certificateSource* element, if present, must have the *smartCardOnlyCertificate* child element chosen.

- False—Attempt to connect to the network after the user successfully logs into Windows at the desktop.

  Use this option when an early network connection is not required. If this network is going to be used in a domain login environment such as the Microsoft Active Directory domain, this option allows for:

  - processing of specific Microsoft Group Policy Objects (GPO) over the machine-context connection as long as no IP address change is required when transferring between the machine and user contexts.

  In this mode there are no restrictions on user credential types allowed.

The following example illustrates the distribution package XML for the *autoConnect* element and its child element.

Example 2-40 autoConnect

```
<autoConnect>
    <connectBeforeLogon>true</connectBeforeLogon>
</autoConnect>
```

manualConnect

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineUserAuthentication/user/manualConnect

The existence of the optional element *manualConnect* specifies that no attempt to make a user-context network connection will be automatically initiated when the user logs into the system. The user, at the desktop, must open SSC and manually select and connect to the network. It is an empty element with no value.

Wired Network Base Elements

Configure the following element:

displayName

Schema path:

configuration/networks/wiredNetwork/displayName

The value of the mandatory element *displayName* specifies the user-friendly name that is used only for display purposes throughout the SSC's various dialogs.

Choosing the Wired Network's Security Class

Specify one of the following security classes for the network:

- Open network—Use element *openNetworkMachineConnection*. Make this choice when your network consists of non-authenticating switches but you want to monitor and display connections.
- Authentication Network—Use element *authenticationNetwork*. Make this choice when you want to preconfigure your enterprise network consistent with your authentication server and its policies and with your credentials environment.

openNetworkMachineConnection

Schema path:

configuration/networks/wiredNetwork/openNetworkMachineConnection

The existence of the optional element *openNetworkMachineConnection* specifies an open wired network. An open network in SSC does not use any form of data encryption and therefore represents the least secure class of networks. It is an empty element with no value.

Business rule: This is a valid choice only if the *open* association mode is supported by the policy. See element allowedAssociationModes in the "Network Policy" section on page 2-6.

authenticationNetwork

Schema path:

configuration/networks/wiredNetwork/authenticationNetwork

The optional element *authenticationNetwork* forms the container for configuring an 802.1X wired network. An authenticating/802.1X network adds two important aspects to wired security, mutual authentication of the client and server and network provide keys for encryption. This network class represents the highest security level choice. No element value is specified.

Next item: "Configuring an Authenticating Wired Network" section on page 2-60.

The following example illustrates the distribution package XML for the two security classes of the *wiredNetwork* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-41 wiredNetwork

```
<wiredNetwork>
    <displayName>My Corporate Ethernet Network</displayName>
    <openNetworkMachineConnection/>
</wiredNetwork>

<wiredNetwork>
    <displayName>My Corporate Ethernet Network</displayName>
    <authenticationNetwork>
        {child elements}
    </authenticationNetwork>
</wiredNetwork>
```

Configuring an Authenticating Wired Network

Follow these steps to configure an authenticating Wired network.

Step 1 Perform the tasks defined in the "Configuring the Authenticating Network Base Elements" section on page 2-32.

Step 2 Perform the tasks defined in the "Choosing the Authentication Network's Connection Context" section on page 2-40.

Step 3 Perform the tasks defined in the "Choosing Wired EAP Methods" section on page 2-61.

The following example illustrates the distribution package XML for the wired *authenticationNetwork* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-42 partial authenticationNetwork

```
<authenticationNetwork>
    <!--{your choice of network connection context goes here}-->
    <serverValidation>
        {child elements}
    </serverValidation>
    <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
    <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
</authenticationNetwork>
```

Choosing Wi-Fi EAP Methods

Perform the tasks defined in the "Choosing Wi-Fi/Wired EAP Methods" section on page 2-62.

Choosing Wired EAP Methods

Configure the following element:

eapMethods

Schema paths:

configuration/networks/wiredNetwork/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods

The mandatory element *eapMethods* forms the container for listing the supported EAP methods for the network. During the EAP negotiation phase of authentication process, if SSC receives a request from the server for a particular EAP method that it is not configured to support, it will respond with an ordered list of alternate EAP methods. The server will process the list to search for an acceptable alternate. If it finds one, it will re-negotiate with the mutually agreed-to method, otherwise authentication will fail. The order of the list is determined by the order in which the chosen child elements are in the XML.

Business rule: at least one child element (at least one EAP method) must be specified.

Business rule: Only EAP methods supported by the policy are permitted. See element *allowedEapMethods* in the "Network Policy" section on page 2-6.

Specify one or more of the following wired-only EAP methods:

- EAP-MD5—Use element *eapMd5*.
- EAP-MSCHAPv2—Use element *eapMschapv2*.
- EAP-GTC—Use element *eapGtc*.

Or, specify one or more of the common EAP methods listed in the "Choosing Wi-Fi/Wired EAP Methods" section on page 2-62.

eapMd5

Schema paths:

configuration/networks/wiredNetwork/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapMd5

The existence of the optional element *eapMd5* specifies the support for the Message Digest 5 (EAP-MD5) authentication method for this network. No additional configuring is required. It is an empty element with no value.

eapMschapv2

Schema paths:

configuration/networks/wiredNetwork/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapMschapv2

The existence of the optional element *eapMschapv2* specifies the support for the Microsoft Challenge-Handshake Authentication Protocol v2 (EAP-MSCHAPv2) authentication method for this network. No additional configuring is required. It is an empty element with no value.

eapGtc

Schema paths:

configuration/networks/wiredNetwork/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapGtc

The existence of the optional element *eapGtc* specifies the support for the Generic Token Card (EAP-GTC) authentication method for this network. No additional configuring is required. It is an empty element with no value.

The following example illustrates the distribution package XML for the wired-only *eapMethods* element and its child elements. The order of the child elements is not restricted.

Example 2-43 eapMethods (wired)

```
<eapMethods>
    <eapMd5/>
    <eapMschapv2/>
    <eapGtc/>
</eapMethods>
```

Choosing Wi-Fi/Wired EAP Methods

Configure the following element:

eapMethods

Schema paths:

configuration/networks/wiredNetwork | wifiNetwork/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods

The mandatory element *eapMethods* forms the container for listing the supported EAP methods for the network. During the EAP negotiation phase of authentication process, if SSC receives a request from the server for a particular EAP method that it is not configured to support, it will respond with an ordered list of alternate EAP methods. The server will process the list to search for an acceptable alternate. If it finds one, it will re-negotiate with the mutually agreed to method, otherwise authentication will fail. The order of the list is determined by the order in which the chosen child elements are listed in the XML.

Business rule: at least one child element, in other words, at least one EAP method, must be specified.

Business rule: Only EAP methods supported by the policy are permitted. See element allowedEapMethods in the "Network Policy" section on page 2-6.

Specify one or more of the following common (Wi-Fi or Ethernet) EAP methods:

- EAP-FAST—Use element *eapFast*.
- EAP-PEAP—Use element *eapPeap*.
- EAP-TTLS—Use element *eapTtls*.
- EAP-TLS—Use element *eapTls*.
- EAP-LEAP—Use element *leap*.

eapFast

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast

The existence of the optional element *eapFast* specifies the support for the Flexible Authentication via Secure Tunneling (EAP-FAST), a Cisco initiative, authentication method for this network.

Next item: "Configuring EAP-FAST" section on page 2-64.

eapPeap

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapPeap

The existence of the optional element *eapPeap* specifies the support for the Protected Extensible Authentication Protocol (EAP-PEAP), both Microsoft & Cisco initiatives, authentication method for this network.

Next item: "Configuring EAP-PEAP" section on page 2-65.

eapTtls

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls

The existence of the optional element *eapTtls* specifies the support for the Tunneled Transport Layer Security (EAP-TTLS), a Funk initiative, authentication method for this network.

Next item: "Configuring EAP-TTLS" section on page 2-65.

eapTls

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTls

The existence of the optional element *eapTls* specifies the support for the Transport Layer Security (EAP-TLS) authentication method for this network.

Next item: "Configuring EAP-TLS" section on page 2-66.

leap

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/leap

The existence of the optional element *leap* specifies the support for the (Light Extensible Authentication Protocol (LEAP) authentication method for this network. No additional configuring is required. It is an empty element with no value.

Note LEAP, a Cisco initiative, is an example of a pre-standard (802.1X), proprietary authentication method that uses a shared secret between the client and server to provide mutual authentication. It is supported in SSC for legacy compatibility.

The following example illustrates the distribution package XML for the wired or wireless *eapMethods* element and its child elements. The order of the child elements is not restricted.

Example 2-44 eapMethods (wired or wireless)

```
<eapMethods>
    <leap/>
    <eapFast>
        {child elements}
    </eapFast>
    <eapPeap>
        {child elements}
    </eapPeap>
    <eapFast>
        {child elements}
    </eapTls>
    <eapFast>
        {child elements}
    </eapTls>
    <eapTtls>
        {child elements}
    </eapTtls>
</eapMethods>
```

Configuring EAP-FAST

Follow these steps to configure EAP-FAST.

Step 1 Perform the tasks defined in the "Configuring EAP Base Elements" section on page 2-66.

Step 2 Perform the tasks defined in the "Configuring FAST Client Certificates" section on page 2-67.

Step 3 Perform the tasks defined in the "Configuring Inner Methods" section on page 2-71.

Note EAP-FAST with only anonymous (unauthenticated), autonomous PAC provisioning corresponds to the initial version v1 of the EAP-FAST standard.
EAP-FAST with authenticated, autonomous PAC provisioning was introduced in version v1a. This version is also backwards compatible by supporting the earlier unauthenticated method.

The following example illustrates the distribution package XML for the *eapFast* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-45 eapFast

```
<eapFast>
    <validateServerIdentity>true</validateServerIdentity>
    <enableFastReconnect>true</enableFastReconnect>
    <protectClientCertificate>true</protectClientCertificate>
    <certificateSource>
        {child element}
    </certificateSource>
    <innerEapMethods>
        {child elements}
    </innerEapMethods>
</eapFast>
```

Configuring EAP-PEAP

Follow these steps to configure EAP-PEAP.

Step 1 Perform the tasks defined in the "Configuring EAP Base Elements" section on page 2-66.

Step 2 Perform the tasks defined in the "Configuring PEAP Client Certificates" section on page 2-68.

Step 3 Perform the tasks defined in the "Configuring Inner Methods" section on page 2-71.

The following example illustrates the distribution package XML for the *eapPeap* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-46 eapPeap

```
<eapPeap>
    <validateServerIdentity>true</validateServerIdentity>
    <enableFastReconnect>true</enableFastReconnect>
    <protectClientCertificate>false</protectClientCertificate>
    <certificateSource>
        {child element}
    </certificateSource>
    <innerEapMethods>
        {child elements}
    </innerEapMethods>
</eapPeap>
```

Configuring EAP-TTLS

Follow these steps to configure EAP-TTLS.

Step 1 Perform the tasks defined in the "Configuring EAP Base Elements" section on page 2-66.

Step 2 Perform the tasks defined in the "Configuring TTLS Inner Methods" section on page 2-73.

The following example illustrates the distribution package XML for the *eapTtls* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-47 eapTtls

```
<eapTtls>
    <validateServerIdentity>true</validateServerIdentity>
    <enableFastReconnect>true</enableFastReconnect>
    <innerMethods>
        {child element}
    </innerMethods>
</eapTtls>
```

Configuring EAP-TLS

Follow these steps to configure EAP-TLS.

Step 1 Perform the tasks defined in the "Configuring EAP Base Elements" section on page 2-66.

Step 2 Perform the tasks defined in the "Configuring the Client Certificate Source" section on page 2-69.

The following example illustrates the distribution package XML for the *eapTls* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-48 eapTls

```
<eapTls>
    <validateServerIdentity>true</validateServerIdentity>
    <enableFastReconnect>true</enableFastReconnect>
    <certificateSource>
        {child element}
    </certificateSource>
</eapTls>
```

Configuring EAP Base Elements

Configure the following elements:

validateServerIdentity

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTtls | eapTls/validateServerIdentity

The mandatory boolean element *validateServerIdentity* specifies whether or not SSC performs client-side validation of the server during authentication.

The element has the following values:

- True—Validate the server certificate.
- False—Do not validate the server certificate.

  This option is *not recommended*, and is usually only used for debugging (to help determine if the server certificate is responsible for a failed authentication) because it reduces the level of security. Using this option implies that you are not Wi-Fi compliant for this wireless network.

Business rule: If the value of *validateServerIdentity* is true, then the optional element *serverValidation* for the same network must be present to provide the server validation details.

Business rule: The value must be true if the network policy specifies mandatory validation. See element *alwaysValidate* in the "Network Policy" section on page 2-6.

enableFastReconnect

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTtls | eapTls/enableFastReconnect

The mandatory boolean element *enableFastReconnect* specifies whether or not SSC performs a fast session resumption using cached credential information during a re-authentication request. (Applies to both outer and inner tunnel methods, where applicable.)

The element has the following values:

- True—Allow fast session resumption.
- False—Disallow fast session resumption.

Note If the network profile specifies multiple EAP authentication methods that involve creation of an SSL session (elements eapFast | eapPeap | eapTtls | eapTls) and the element *enableFastReconnect* is set to True for any method, then fast session resumption will be enabled for all the methods associated with this network profile.

Configuring FAST Client Certificates

Refer to Example 2-45 for the required sequence of these elements.

Step 1 Configure the following element:

protectClientCertificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast/protectClientCertificate

The mandatory boolean element *protectClientCertificate* specifies whether or not SSC will send, when requested, a certificate unprotected.

The element has the following values:

- True—Indicates:

  Enable protection and using a client certificate or

  Not using a client certificate.

  When requested by the server for a client certificate during the unprotected (phase 1) portion of FAST PAC provisioning:

  – SSC refuses at this point to send any certificate (it's allowed this option) because it will wait for the protected Phase 2 of the protocol. (Actually, a tunnel will first be established, based on the server's certificate, and SSC will send its client certificate before Phase 2 begins).

  Cisco ACS, when not configured for using a client certificate, will not ask for the certificate during phase 2. So it is important to use this value to avoid any unwanted client certificate prompts.

- False—Disable protection and using a client certificate.
  When requested by the server for a client certificate during the unprotected (phase 1) portion of FAST PAC provisioning:

  – If there is a client certificate available, it will be sent.

  – If there is no client certificate, none will be sent and the server policy will determine if authentication continues or fails.

This has no affect on the use of a client certificate during the protected (phase 2) portion of FAST PAC provisioning. If the server is configured to request the sending of the client certificate within the secure tunnel, the client will always attempt to use one. If none is available and not sent, the connection attempt will fail.

Step 2 Perform the tasks defined in the "Configuring the Client Certificate Source" section on page 2-69:

Configuring PEAP Client Certificates

Refer to Example 2-46 for the required sequence of these elements.

Step 1 Configure the following element:

protectClientCertificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapPeap | eapTtls/protectClientCertificate

The mandatory boolean element *protectClientCertificate* specifies whether or not SSC will send, when requested, a certificate unprotected.

The element has the following values:

- True—Enable protection.
  When requested by the server for a client certificate during the unprotected (phase 1) portion of the protocol:

  – The client certificate will never be sent to the server (since there is no way to send a client certificate protected for this EAP method). Server policy will determine if authentication continues or fails.

- False—Disable protection.
  When requested by the server for a client certificate during the unprotected (phase 1) portion of the protocol:
  - If there is a client certificate available, it will be sent.
  - If there is no client certificate, none will be sent and the server policy will determine if authentication continues or fails.

Step 2 Perform the tasks defined in the "Configuring the Client Certificate Source" section on page 2-69:

Configuring the Client Certificate Source

Configure the following elements:

certificateSource

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTls/certificateSource

The element *certificateSource* forms the container for specifying the source of a client certificate. No element value is specified.

Restriction: *certificateSource* is required within the *eapTls* element.

Business rule: *certificateSource* is optional within the eapPeap and eapFast elements for the outer tunnel as determined by the policy of the authentication server. However, it is required if the corresponding inner method specified is EAP TLS.

Specify one of the following sources for your client certificate:

- Obtain only from a SmartCard—Use element *smartCardOnlyCertificate*.
- Obtain from either the Windows certificate store or a SmartCard—Use element *smartCardOrOsCertificate*.

SmartCard certificate properties:

- Only a single smartcard reader (the first one detected) is supported.
- Multiple certificates from a single SmartCard are supported.
- SmartCards must support the Microsoft CryptoAPI and SCard interfaces to Cryptographic Service Provider (CSP) functionality. Furthermore, any SmartCard reader and SmartCard combination must inter-operate via the PC/SC interface to provide low level support for these same CSP functions.
- A SmartCard PIN (two-factor authentication) is supported. A prompt will be made through either the Enter Your Credentials dialog or the operating system GINA. PIN behavior depends on the configured user credential collection method for the network.
  - collectionMethod/prompt | static—PINs are not stored and therefore will be subsequently requested again under certain situations such as when a server initiated re-authentication is required (unless the EAP method is configured for fast session resumption), roaming, a failed re-authentication, lost association, resumption from hibernate.
  - collectionMethod/singleSignOn—PINs are stored for the duration of the logon session. Therefore, subsequent pop-up requests are avoided.

Windows certificate properties:

- If a Windows user certificate is required as part of the authentication process, it must be appropriately pre-installed as a separate task.
  - A user certificate is obtained from the Personal Certificate Store for the currently logged in Windows user.
  - A machine certificate is obtained from the Personal Certificate Store for the Local Computer.
- Only valid certificates are displayed for selection. Expired certificates are not listed. Also, valid certificates that are about to expire contain a warning that shows how many days left before the certificate expires.
- When configured for an automatic user connection, a certificate with Strong Private Key Protection will fail at logon and should not be used. Certificates with this property are only supported when configured for always making manual connections at desktop.
- The identifying information for the selected certificate in the selection drop-down list is obtained from the various fields of the certificate as follows:
  - text box name—Subject: CN (Common Name).
  - Issued to:—Subject: CN (Common Name).
  - Issued by:—Issuer: CN (Common Name).
  - Alternative Name:—Subject Alternate Name: DNSName
  - Expires:—Valid to.
  - Extended Key Usage—Extended Key Usage

smartCardOnlyCertificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTtls | eapTls/certificateSource/smartCardOnlyCertificate

The optional element *smartCardOnlyCertificate* specifies that a client certificate must be obtained only from a SmartCard. It is an empty element with no value.

Business rule: If the corresponding connection context is *machineAuthentication*, then this option is not allowed because a machine certificate must be from the OS store.

smartCardOrOsCertificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTtls | eapTls/certificateSource/smartCardOrOsCertificate

The optional element *smartCardOrOsCertificate* specifies that a client certificate may be obtained from either a SmartCard or the Windows certificate store. It is an empty element with no value.

The following example illustrates the distribution package XML for the two choices for the *certificateSource* element and its child element.

Example 2-49 certificateSource

```
<certificateSource>
    <smartCardOrOsCertificate/>
</certificateSource>

<certificateSource>
    <smartCardOnlyCertificate/>
</certificateSource>
```

Configuring Inner Methods

Configure the following element:

innerEapMethods

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods

The mandatory element *innerEapMethods* forms the container for listing the supported inner EAP methods for the outer EAP method. During the EAP negotiation phase of authentication process, if SSC receives a request from the server for a particular EAP method that it is not configured to support, it will respond with an ordered list of alternate EAP methods. The server will process the list to search for an acceptable alternate. If it finds one, it will re-negotiate with the mutually agreed-to method, otherwise authentication will fail. The order of the list is determined by the order in which the chosen child elements are listed in the XML. No element value is specified.

Business rule: at least one child element (at least one inner EAP method) must be specified.

Specify one or more of the following FAST/PEAP inner EAP methods:

- EAP-MSCHAPv2—Use element *eapMschapv2*.
- EAP-GTC—Use element *eapGtc*.
- EAP-TLS—Use element *eapTls*.

eapMschapv2

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods/eapMschapv2

The existence of the optional element *eapMschapv2* specifies the support for the Microsoft Challenge-Handshake Authentication Protocol v2 (EAP-MSCHAPv2) inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

eapGtc

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods/eapGtc

The existence of the optional element *eapGtc* specifies the support for the Generic Token Card (EAP-GTC) inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

eapTls

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods/eapTls

The existence of the optional element *eapTls* specifies the support for the Transport Layer Security (EAP-TLS) inner authentication method for this network.

Business rule: With this option, you must add and configure the elements of the "Configuring the Client Certificate Source" section on page 2-69.

Configuration of the following child element is required:

validateServerIdentity

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods/eapTls/validateServerIdentity

The mandatory boolean element *validateServerIdentity* specifies whether or not SSC performs client-side validation of the server during authentication within the secure inner tunnel.

The element has the following values:

- True—Validate the server certificate.
- False—Do not validate the server certificate.

  This option is not recommended, and is usually only used for debugging (to help determine if the server certificate is responsible for a failed authentication) because it reduces the level of security. Using this option implies that you are not Wi-Fi compliant for this wireless network.

Business rule: If the value of *validateServerIdentity* is true, then the optional element *serverValidation* for the same network must be present to provide the server validation details.

The following example illustrates the distribution package XML for the *innerEapMethods* element and its child elements. The order and number of the child elements is not restricted.

Example 2-50 innerEapMethods

```
<innerEapMethods>
    <eapMschapv2/>
    <eapGtc/>
    <eapTls>
        <validateServerIdentity>true</validateServerIdentity>
    </eapTls>
</innerEapMethods>
```

Configuring TTLS Inner Methods

Configure the following element:

innerMethods

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerMethods

The mandatory element *innerMethods* forms the container for listing the supported inner methods for the outer EAP method. The use of legacy and EAP methods inside the EAP-TTLS are mutually exclusive and cannot be simultaneously specified. No element value is specified.

Specify one the following classes of inner methods:

- Legacy methods—Use element *legacy.*
- EAP methods—Use element *eap*.

legacy

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/legacy

The optional element *legacy* forms the container for configuring one of the legacy TTLS inner methods. No element value is specified.

Specify one of the following legacy TTLS inner methods:

- PAP—Use element *pap*.
- CHAP—Use element *chap*.
- MSCHAP—Use element *mschap*.
- MSCHAPv2—Use element *mschapv2*.

pap

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/legacy/pap

The existence of the optional element *pap* specifies the use of Password Authentication Protocol (PAP) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

chap

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/legacy/chap

The existence of the optional element *chap* specifies the use of Challenge-Handshake Authentication Protocol (CHAP) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

mschap

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/legacy/mschap

The existence of the optional element *mschap* specifies the use of Microsoft Challenge-Handshake Authentication Protocol (MSCHAP) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

mschapv2

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/legacy/mschapv2

The existence of the optional element *mschapv2* specifies the use of Microsoft Challenge-Handshake Authentication Protocol v2 (MSCHAPv2) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

eap

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/eap

The optional element *eap* forms the container for configuring one of the standard EAP methods. During the EAP negotiation phase of authentication process, if SSC receives a request from the server for a particular EAP method that it is not configured to support, it will respond with an ordered list of alternate EAP methods. The server will process the list to search for an acceptable alternate. If it finds one, it will re-negotiate with the mutually agreed-to method; otherwise authentication will fail. The order of the list is determined by the order in which the chosen child elements are listed in the XML. No element value is specified.

Business rule: at least one child element, in other words, at least one inner EAP method, must be specified.

Specify one or more of the following inner EAP methods:

- EAP-MSCHAPv2—Use element *eapMschapv2*.
- EAP-MD5—Use element *eapMd5*.

eapMschapv2

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerMethods/eapMschapv2

The existence of the optional element *eapMschapv2* specifies the support for the Microsoft Challenge-Handshake Authentication Protocol v2 (EAP-MSCHAPv2) inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

eapMd5

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerMethods/eapMd5

The existence of the optional element *eapMd5* specifies the support for the Message Digest 5 (EAP-MD5) inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

The following example illustrates the distribution package XML for the *innerMethods* element and its child elements. The order of the child elements is restricted to that shown.

Example 2-51 innerMethods

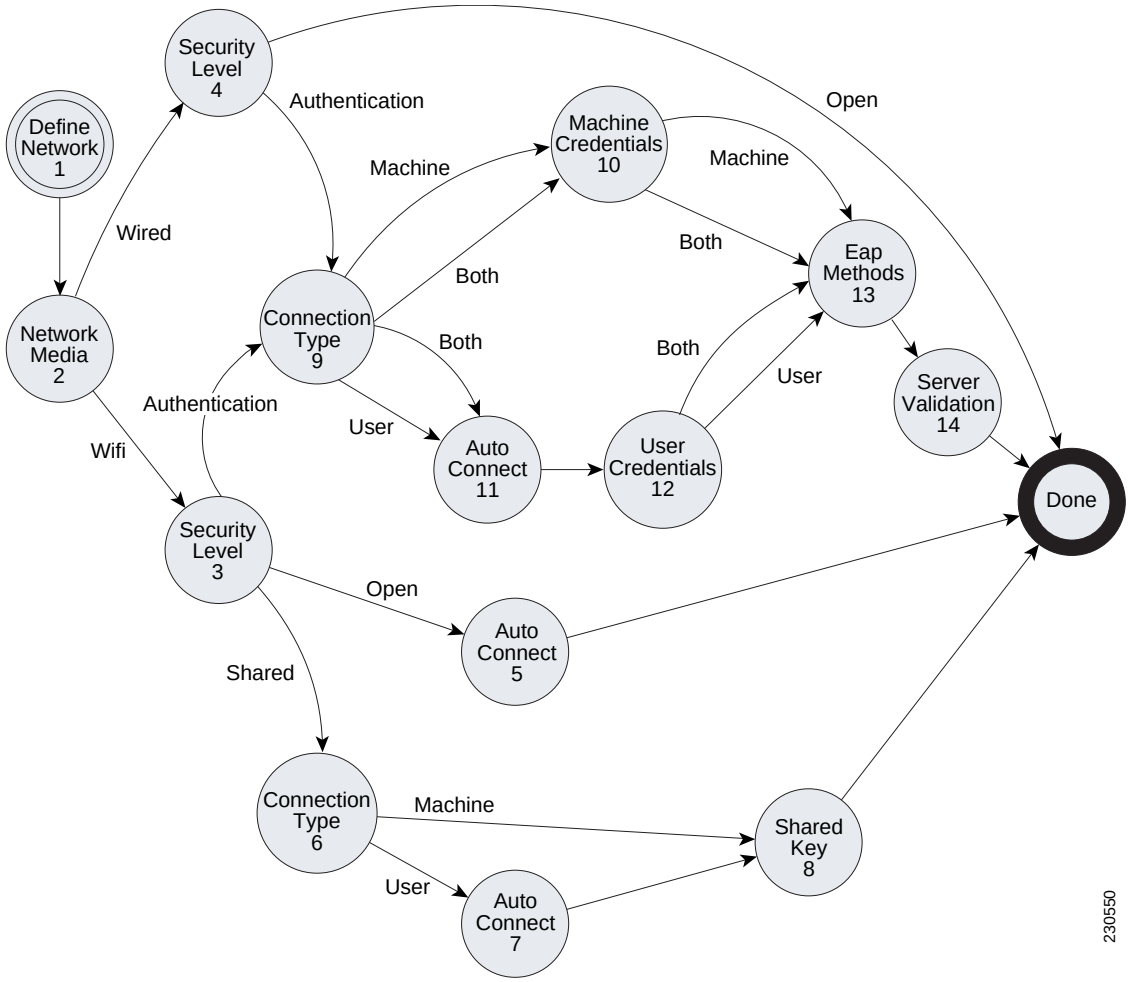
```
<innerMethods>
    <legacy>
        <pap/>
    </legacy>
</innerMethods>

<innerMethods>
    <legacy>
        <chap/>
    </legacy>
</innerMethods>

<innerMethods>
    <legacy>
        <mschap/>
    </legacy>
</innerMethods>

<innerMethods>
    <legacy>
        <mschapv2/>
    </legacy>
</innerMethods>

<innerMethods>
    <eap>
        <eapMd5/>
        <eapMschapv2/>
    </eap>
</innerMethods>
```

APPENDIX A

Network Decision Tree Flow Diagram

The following illustration provides an overview of the XML schema decision tree for configuring a network connection and serves as a graphical index to the corresponding sections in Chapter 2, "Schema Elements." Refer to the legend for a link to the detailed configuring steps and element descriptions.

Figure A-1 Network Configuring High Level Fow Diagram



Legend for Figure A-1.

1. Configure a network
   - Configuring Networks
2. Choosing the network media
   - Choosing a Network Media Type
3. Wi-Fi security class
   - Choosing the Wi-Fi Network's Security Class
4. Wired security class
   - Choosing the Wired Network's Security Class
5. Wi-Fi open network, user connection occurrence
   - Configuring an Open Wi-Fi Network
6. Wi-Fi shared key network, connection context
   - Configuring a Shared-key Wi-Fi Network
   - Configuring a Shared-key, Machine Network
7. Wi-Fi shared key network, user connection occurrence
   - Configuring a Shared-key, User Network
8. Wi-Fi shared key network, shared key
   - Choosing the Shared-key Type
9. Wi-Fi/wired authentication network, connection context
   - Configuring an Authenticating Wi-Fi Network
   - Configuring an Authenticating Wired Network
   - Choosing the Authentication Network's Connection Context
   - Configuring an Authenticating, Machine and User Network
10. Wi-Fi/wired authentication network, machine connection context, credentials
    - Configuring an Authenticating, Machine-only Network
    - Configuring the Authenticating, Machine Credential Source Elements
11. Wi-Fi/wired authentication network, user connection occurrence
    - Configuring an Authenticating, User-Only Network
    - Configuring the Authenticating, User-Only Connection Occurrence Elements
12. Wi-Fi/wired authentication network, user connection context, credentials
    - Configuring the Authenticating, User Credential Source (1) Elements
    - Configuring the Authenticating, User Credential Source (2) Elements
13. Wi-Fi/wired authentication network, EAP methods
    - Choosing Wi-Fi EAP Methods
    - Choosing Wired EAP Methods
14. Wi-Fi/wired authentication network, server validation
    - Configuring Server Validation
    - Configuring Certificate Trusted Server Rules

- Configuring PAC Trusted Server Rules
- Adding CA Certificates

APPENDIX B

Distribution Package Examples

Following are examples of valid .xml distribution package files from typical enterprise environments. The file listings are augmented with crossreferences to the features indicated in the high-level descriptions. The file, sscAdminGuideExXml.zip, also distributed in the SSCAdminUtils zip file, contains all of these examples as individual .xml files, for a convenient starting point and easy text editing.

Note In all of the examples, the license string is functionally invalid. Replace with one appropriate to your application.

High-level Descriptions

- Example B-1—Illustrates only the base elements of a distribution package. No networks are defined in this example. (Use Example B-1.)
- Example B-2—illustrates the addition of minimal, nonauthenticating, open (1) Wi-Fi and (2) wired networks. (Use Example B-2.)
- Example B-3—Illustrates (1) a nonauthenticating, WPA personal Wi-Fi network with the following properties:
  - (2) user connection context
  - (3) WPA-Personal association with TKIP encryption

  Such a network would be applicable to any corporate-supplied home equipment (where you configure the key) that your end-user might have for connecting to your enterprise network remotely. (Use Example B-3.)

Note Any of the following authenticating Wi-Fi network definitions can be extracted and used in a wired authenticating network by removing the *associationMode* element. Extract the following:

```
<authenticationNetwork>
   Retain otherwise: .....
   Remove this: <associationMode>...</associationMode>
</authenticationNetwork>
```

- Example B-4—Illustrates (1) an authenticating Wi-Fi network with the following properties:
  - (2) machine/user connection context

  - (3) user password credentials obtained from an initial, one-time prompt
    (4) machine password obtained automatically from the MS Active Directory setup
  - (5) single, tunneled EAP method
  - (6) server certificated validation based on release 4.0 functionality

  (Use Example B-4.)

- Example B-5—Illustrates (1) an authenticating Wi-Fi network with the following properties:
  - (2) machine/user connection context
  - (3) user password credentials obtained from the operating system (single-signon)
    (4) machine credential obtained automatically from the MS Active Directory setup
  - (5) multiple, tunneled EAP methods
  - (6) server certificate validation based on multiple authentication server rules and release 4.1 (7) CA certificate deployment support

  (Use Example B-5.)

- Example B-6—Illustrates (1) an authenticating, Wi-Fi network with the following properties:
  - Novell domain compatible network
  - (2) user connection context
  - (3) user password credentials obtained from the OS (single-signon)
  - (4) single, tunneled EAP method
  - (5) server certificate validation based on release 4.0 functionality

  (Use Example B-6.)

- Example B-7—Illustrates (1) an authenticating, Wi-Fi network with the following properties:
  - (2) machine connection context
  - (3) machine credentials obtained from release 4.1 static credential support
  - (4) single, tunneled EAP method
  - (5) server certificate validation based on release 4.0 functionality

  (Use Example B-7.)

- Example B-8—Illustrates (1) an authenticating, Wi-Fi network with the following properties:
  - (2) user connection context
  - (3) user client certificate credentials obtained from a smartcard
  - (4) TLS EAP method
  - (5) server certificate validation based on release 4.0 functionality

  (Use Example B-8.)

- Example B-9a—Illustrates (1) an authenticating Wi-Fi network with the following properties:
  - (2) user connection context
  - (3) user password credentials obtained from an initial, one-time prompt
  - (4) EAP-FAST-GTC method (autonomous, authenticated PAC provisioning)
  - (5) server certificate validation for PAC provisioning based on release 4.0 functionality

  (Use Example B-9a.)

- Example B-9b—Illustrates (1) an authenticating Wi-Fi network with the following properties:
  - (2) user connection context
  - (3) user password credentials obtained from an initial, one-time prompt
  - (4) EAP-FAST-GTC method (autonomous, unauthenticated PAC provisioning)
  - (5) server AID validation for PAC provisioning based on release 4.0 functionality

  (Use Example B-9b.)
- Example B-9c—Illustrates (1) an authenticating Wi-Fi network with the following properties:
  - (2) user connection context
  - (3) user password credentials obtained from an initial, one-time prompt
  - (4) FAST EAP-MSCHAPv2 method with release 4.1 manual PAC provisioning support (Cisco ACS server configured for no autonomous PAC provisioning.)
  - (5) no server validation

  (Use Example B-9c.)
- Example B-10—Illustrates (1) an authenticating Wi-Fi network with the following properties:
  - (2) user connection context
  - (3) user password credentials obtained from new release 4.1 static credential support
  - (4) single, tunneled EAP method
  - (5) server certificate validation based on release 4.0 functionality

  (Use Example B-10.)
- Example B-11—Illustrates (1) a wired-only version with the following properties:
  - (2) preset end-user version
  - (3) authenticating network only
  - (4) machine and user connection context
  - (5) FAST EAP method only
  - (6) server certificate validation

  (Use Example B-11.)

File Listings

Example B-1

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
```

```
            <!--legacy WEP shared key and authenticating networks-->
             <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
            <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
            <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-2

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
            <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
            <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(2)     <wiredNetwork>
            <displayName>My Corporate Wired Network</displayName>
            <openNetworkMachineConnection/>
        </wiredNetwork>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
            <openNetworkUserConnection>
                <autoConnect>true</autoConnect>
            </openNetworkUserConnection>
        </wifiNetwork>
    </networks>
```

```
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-3

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
            <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
            <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
        <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <sharedKeyNetwork>
(2)             <userConnection>
                    <keySettings>
(3)                     <wpa>
                            <key>
                                <ascii encrypt="true">mySecret</ascii>
                            </key>
                            <encryption>TKIP</encryption>
                        </wpa>
```

```
                    </keySettings>
                    <autoConnect>true</autoConnect>
                </userConnection>
            </sharedKeyNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-4

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(5)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(6)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
(3)         <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <machineUserAuthentication>
                    <machine>
                        <collectionMethod>
(4)                         <auto/>
                        </collectionMethod>
                        <unprotectedIdentityPattern>host/anonymous</unprotectedIdentityPattern>
                        <protectedIdentityPattern>host/&lt;fqhn&gt;</protectedIdentityPattern>
                    </machine>
```

```
                    <user>
                        <autoConnect>
                            <connectBeforeLogon>true</connectBeforeLogon>
                        </autoConnect>
                        <collectionMethod>
(3)                         <prompt>
                                <credentialsStorage>
                                    <forever/>
                                </credentialsStorage>
                            </prompt>
                        </collectionMethod>
                        <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                        <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                    </user>
                    <eapMethods>
(5)                     <eapFast>
(6)                         <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                                <eapGtc/>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </machineUserAuthentication>
                <serverValidation>
(6)                 <validationRules>
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-5

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="21">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(5)         <eapFast/>
(5)         <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(6)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <machineUserAuthentication>
                    <machine>
                        <collectionMethod>
(4)                         <auto/>
                        </collectionMethod>
                        <unprotectedIdentityPattern>host/anonymous</unprotectedIdentityPattern>
                        <protectedIdentityPattern>host/&lt;fqhn&gt;</protectedIdentityPattern>
                    </machine>
```

```
                    <user>
                        <autoConnect>
                            <connectBeforeLogon>true</connectBeforeLogon>
                        </autoConnect>
                        <collectionMethod>
(3)                         <singleSignOn/>
                        </collectionMethod>
                        <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                        <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                    </user>
                    <eapMethods>
(5)                     <eapFast>
                            <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                                <eapGtc/>
                            </innerEapMethods>
                        </eapFast>
(5)                     <eapPeap>
                            <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>false</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                                <eapGtc/>
                            </innerEapMethods>
                        </eapPeap>
                    </eapMethods>
                </machineUserAuthentication>
                <serverValidation>
(6)                 <validationRules>
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                        <matchSubjectAlternativeName name="Cert Rule 3"
match="endsWith">myCorp2.net</matchSubjectAlternativeName>
                    </validationRules>
                    <trustedRootCACerts>
(7)                     <certificate>
                            <caReference>E:\path\CaCertFile</caReference>
                        </certificate>
                    </trustedRootCACerts>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
```

```
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-6

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(4)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <userAuthentication>
                    <autoConnect>
                        <connectBeforeLogon>true</connectBeforeLogon>
                    </autoConnect>
                    <collectionMethod>
(3)                     <singleSignOn/>
                    </collectionMethod>
                    <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
```

```
                        <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                    <eapMethods>
(4)                     <eapFast>
(5)                         <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                                <eapGtc/>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </userAuthentication>
                <serverValidation>
(5)                 <validationRules>
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-7

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
            <eapFast/>
(4)         <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <machineAuthentication>
                    <collectionMethod>
(3)                     <static/>
                    </collectionMethod>
(3)                 <unprotectedIdentityPattern>anonymous</unprotectedIdentityPattern>
(3)                 <protectedIdentityPattern>machineName</protectedIdentityPattern>
(3)                 <staticPassword encrypt="true">machineSecret</staticPassword>
                    <eapMethods>
```

```
(4)                     <eapPeap>
                            <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                            </innerEapMethods>
                        </eapPeap>
                    </eapMethods>
                </machineAuthentication>
                <serverValidation>
(5)                 <validationRules>
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-8

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(4)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
(3)         <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <userAuthentication>
                    <autoConnect>
                        <connectBeforeLogon>false</connectBeforeLogon>
                    </autoConnect>
                    <collectionMethod>
                        <prompt>
                            <credentialsStorage>
(3)                             <logonSession/>
```

```
                            </credentialsStorage>
                        </prompt>
                    </collectionMethod>
                    <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                    <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                    <eapMethods>
(4)                     <eapFast>
(5)                         <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <certificateSource>
(3)                             <smartCardOnlyCertificate/>
                            </certificateSource>
                            <innerEapMethods>
(4)                             <eapTls>
                                    <validateServerIdentity>true</validateServerIdentity>
                                </eapTls>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </userAuthentication>
                <serverValidation>
(5)                 <validationRules>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-9a

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(4)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
(3)         <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <userAuthentication>
                    <autoConnect>
                        <connectBeforeLogon>false</connectBeforeLogon>
                    </autoConnect>
                    <collectionMethod>
(3)                     <prompt>
                            <credentialsStorage>
                                <forever/>
```

```
                            </credentialsStorage>
                        </prompt>
                    </collectionMethod>
                    <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                    <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                    <eapMethods>
(4)                     <eapFast>
(5)                         <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
(4)                             <eapGtc/>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </userAuthentication>
                <serverValidation>
(5)                 <validationRules>
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-9b

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(4)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
(3)         <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <userAuthentication>
                    <autoConnect>
                        <connectBeforeLogon>false</connectBeforeLogon>
                    </autoConnect>
                    <collectionMethod>
(3)                     <prompt>
                            <credentialsStorage>
                                <forever/>
```

```
                        </credentialsStorage>
                    </prompt>
                </collectionMethod>
                <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                <eapMethods>
(4)                 <eapFast>
(5)                     <validateServerIdentity>true</validateServerIdentity>
                        <enableFastReconnect>true</enableFastReconnect>
                        <protectClientCertificate>true</protectClientCertificate>
                        <innerEapMethods>
(4)                         <eapMschapv2/>
(4)                         <eapGtc/>
                        </innerEapMethods>
                    </eapFast>
                </eapMethods>
            </userAuthentication>
            <serverValidation>
(5)             <trustedServerIds>
                    <trustedServerId name="PAC AID Rule 1">
                        <reference>
                            <aIdReference>E:\path\pacRefFile</aIdReference>
                            <secretKey>1234</secretKey>
                        </reference>
                    </trustedServerId>
                </trustedServerIds>
                <trustAnyRootCaFromOs/>
            </serverValidation>
            <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
            <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
            <associationMode>
                <wpa-Enterprise>TKIP</wpa-Enterprise>
            </associationMode>
        </authenticationNetwork>
    </wifiNetwork>
</networks>
<stationSettings>
    <simultaneousConnections>singleHomed</simultaneousConnections>
    <validateWpaHandshake>true</validateWpaHandshake>
</stationSettings>
<userControlPolicy>
    <clientUIType>configurable</clientUIType>
    <allowLicensing>false</allowLicensing>
    <allowedMedia>
        <wired/>
        <wifi/>
    </allowedMedia>
</userControlPolicy>
</configuration>
```

Example B-9c

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(4)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <allowUserValidationControl/>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
(3)         <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <userAuthentication>
                    <autoConnect>
                        <connectBeforeLogon>false</connectBeforeLogon>
                    </autoConnect>
                    <collectionMethod>
(3)                     <prompt>
                            <credentialsStorage>
                                <forever/>
                            </credentialsStorage>
                        </prompt>
```

```
                    </collectionMethod>
                    <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                    <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
(4)                 <pacs>
                        <pac>
                            <pacReference encrypt="true">E:\path\pacFile</pacReference>
                            <secretKey encrypt="true">pacPassword</secretKey>
                        </pac>
                    </pacs>
                    <eapMethods>
(4)                     <eapFast>
(5)                         <validateServerIdentity>false</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
(4)                             <eapMschapv2/>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </userAuthentication>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-10

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes>
            <!--open network-->
            <open/>
            <!--shared key network-->
            <wpa-Personal/>
            <wpa2-Personal/>
            <!--authenticating network-->
            <wpa-Enterprise/>
            <wpa2-Enterprise/>
           <!--legacy WEP shared key and authenticating networks-->
            <wep/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
(4)         <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
(5)         <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wifiNetwork>
            <displayName>My Corporate Wi-Fi Network</displayName>
            <ssid>MyCorpNet</ssid>
            <associationRetries>3</associationRetries>
            <beaconing>true</beaconing>
(1)         <authenticationNetwork>
(2)             <userAuthentication>
                    <autoConnect>
                        <connectBeforeLogon>false</connectBeforeLogon>
                    </autoConnect>
                    <collectionMethod>
(3)                     <static/>
                    </collectionMethod>
(3)                 <unprotectedIdentityPattern>anonymous@engr.myCompany.com</unprotectedIdentityPattern>
```

```
(3)                 <protectedIdentityPattern>userName</protectedIdentityPattern>
(3)                 <staticPassword encrypt="true">userSecret</staticPassword>
                    <eapMethods>
(4)                     <eapFast>
(5)                         <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </userAuthentication>
                <serverValidation>
(5)                 <validationRules>
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                <associationMode>
                    <wpa-Enterprise>TKIP</wpa-Enterprise>
                </associationMode>
            </authenticationNetwork>
        </wifiNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>true</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
        <clientUIType>configurable</clientUIType>
        <allowLicensing>false</allowLicensing>
        <allowedMedia>
            <wired/>
            <wifi/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

Example B-11

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\distributionPackage.xsd" major_version="4" minor_version="2">
<license>T244-YKGP-UMG5-Y2F2-5KMH-5OYX-DAR4-POND-52Z5-MHJZ-3LOD-SLYL-U5YA-IUKU-M3TC-JNO7-3MEM-LGAA</license>
    <networkPolicy>
        <allowedAssociationModes></allowedAssociationModes>
        <allowedEapMethods>
(5)         <eapFast/>
        </allowedEapMethods>
        <serverValidationPolicy>
(6)         <alwaysValidate>
                <allowUserTrustedServers>false</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsControl>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
        <allowedClientCertificates>
            <noEkuFilter/>
        </allowedClientCertificates>
    </networkPolicy>
    <networks>
(1)     <wiredNetwork>
            <displayName>My Corporate Wired Network</displayName>
(3)         <authenticationNetwork>
(4)             <machineUserAuthentication>
                    <machine>
                        <collectionMethod>
                            <auto/>
                        </collectionMethod>
                        <unprotectedIdentityPattern>host/anonymous</unprotectedIdentityPattern>
                        <protectedIdentityPattern>host/&lt;fqhn&gt;</protectedIdentityPattern>
                    </machine>
                    <user>
                        <autoConnect>
                            <connectBeforeLogon>true</connectBeforeLogon>
                        </autoConnect>
                        <collectionMethod>
                            <singleSignOn/>
                        </collectionMethod>
                        <unprotectedIdentityPattern>anonymous@&lt;domain&gt;</unprotectedIdentityPattern>
                        <protectedIdentityPattern>&lt;username&gt;</protectedIdentityPattern>
                    </user>
                    <eapMethods>
(5)                     <eapFast>
(6)                         <validateServerIdentity>true</validateServerIdentity>
                            <enableFastReconnect>true</enableFastReconnect>
                            <protectClientCertificate>true</protectClientCertificate>
                            <innerEapMethods>
                                <eapMschapv2/>
                            </innerEapMethods>
                        </eapFast>
                    </eapMethods>
                </machineUserAuthentication>
                <serverValidation>
(6)                 <validationRules>
```

```
                        <matchSubjectAlternativeName name="Cert Rule 1"
match="endsWith">myCorp.com</matchSubjectAlternativeName>
                        <matchSubjectName name="Cert Rule 2" match="exactly">My
Corporation</matchSubjectName>
                    </validationRules>
                    <trustAnyRootCaFromOs/>
                </serverValidation>
                <interactiveAuthenticationRetries>4</interactiveAuthenticationRetries>
                <nonInteractiveAuthenticationRetries>4</nonInteractiveAuthenticationRetries>
                </authenticationNetwork>
            </wiredNetwork>
    </networks>
    <stationSettings>
        <simultaneousConnections>singleHomed</simultaneousConnections>
        <validateWpaHandshake>false</validateWpaHandshake>
    </stationSettings>
    <userControlPolicy>
(2)     <clientUIType>preset</clientUIType>
        <allowLicensing>false</allowLicensing>
(1)     <allowedMedia>
            <wired/>
        </allowedMedia>
    </userControlPolicy>
</configuration>
```

APPENDIX C

Postprocessing Verification Errors

Command Usage Errors

Note Execution of the sscManagementUtility utility will result in either of the following:

- Success—No return message. Output file created with processed content.
- Failure—Error message returned. Output file created, but empty.

- Input file must have .xml file extension

  Command syntax example:

  ```
  sscManagementUtility validate -i distPkg
  ```

  Error message:

  ```
  Input file "distPkg" should have the ".xml" extension!
  ```

- Input file has an incorrect file extension

  Command syntax example:

  ```
  sscManagementUtility validate -i distPkg.txt
  ```

  Error message:

  ```
  Input file "distPkg.txt" should have the ".xml" extension!
  ```

- Command line syntax error

  Command syntax example:

  ```
  sscManagementUtility distPkg.xml distPkgSigned.xml
  ```

  Error message:

  ```
  Usage:
    sscManagementUtility [command] [command specific options]

  Command:
    help - print usage
    validate - check for validate configuration Xml file
    sign - validate and sign configuration Xml file
  ```

```
validate options:
  sscManagementUtility validate [-i <input file>]
  -i --in
     path to the original distribution package xml file
sign options:
  sscManagementUtility sign [-i <input file>] [-o <output file]
  -i --in
     path to the original distribution package xml file
  -o --out
     path to the processed and ready to deploy xml file
```

Most command syntax errors will display the command help information, as in this example.

XML Schema Validation Errors

Note Errors found by the utility's built-in XLM schema validation process are displayed as one of the following types:

parser error

Schema validity error

Some examples of schema validation errors are:

- An empty input file, distPkg.xml

  Error message:

  distPkg.xml:1: parser error : Document is empty

  distPkg.xml:1: parser error : Start tag expected, '<' not found

  failed to parse distPkg.xml

- Missing element closing tag (<credentialsStorage)

Tip Parsing errors are hierarchical in nature. Always resolve top-down. The actual error will most likely cause additional by-product errors to appear subsequently in the file.

In this case, fixing the single error in line 56, eliminates all of the reported parsing errors listed below.

Erroneous XML input text:

```
(line 54)   <collectionMethod>
(line 55)      <prompt>
(line 56)         <credentialsStorage
(line 57)            <logonSession/>
(line 58)         </credentialsStorage>
(line 59)      </prompt>
(line 60)   </collectionMethod>
```

Error message:

distPkg.xml:57: parser error : error parsing attribute name <logonSession/>

distPkg.xml:57: parser error : attributes construct error <logonSession/>

distPkg.xml:57: parser error : Couldn't find end of Start Tag credentialsStorage line 56 <logonSession/>

distPkg.xml:58: parser error : Opening and ending tag mismatch: prompt line 55 and credentialsStorage </credentialsStorage>

distPkg.xml:59: parser error : Opening and ending tag mismatch: collectionMethod line 54 and prompt </prompt>

distPkg.xml:60: parser error : Opening and ending tag mismatch: userAuthentication line 50 and collectionMethod </collectionMethod>

distPkg.xml:71: parser error : Opening and ending tag mismatch: authenticationNetwork line 49 and userAuthentication </userAuthentication>

distPkg.xml:83: parser error : Opening and ending tag mismatch: wifiNetwork line 44 and authenticationNetwork </authenticationNetwork>

distPkg.xml:84: parser error : Opening and ending tag mismatch: networks line 43 and wifiNetwork </wifiNetwork>

distPkg.xml:85: parser error : Opening and ending tag mismatch: configuration line 2 and networks </networks>

distPkg.xml:86: parser error : Extra content at the end of the document <stationSettings>

failed to parse distPkg.xml

- Missing attributes from base element

  Erroneous XML input text:

  ```
  <configuration>
  ```

  Error message:

  distPkg.xml:1: element configuration: Schema validity error : Element 'configuration': The attribute 'major_version' is required but missing.

  distPkg.xml:1: element configuration: Schema validity error : Element 'configuration': The attribute 'minor_version' is required but missing.

  distPkg.xml failed schema validation

- Elements out-of-order as required by schema

  Erroneous XML input text:

  ```
  <wifiNetwork>
    <ssid>MyCorpNet</ssid>
    <displayName>My Corporate Wi-Fi Network</displayName>
  ```

  Error message:

  distPkg.xml:45: element ssid: Schema validity error : Element 'ssid': This element is not expected. Expected is (displayName).

  distPkg.xml failed schema validation

- Missing a required element

  Erroneous XML input text:

  ```
  <userAuthentication>
    <autoConnect></autoConnect>
  ```

  Error message:

distPkg.xml:51: element autoConnect: Schema validity error : Element 'autoConnect': Missing child element(s). Expected is (connectBeforeLogon).

distPkg.xml failed schema validation

- Missing a required element value

  Erroneous XML input text:

  ```
  <wifiNetwork>
    <displayName></displayName>
    <ssid>MyCorpNet</ssid>
  ```

  Error message:

  distPkg.xml:45: element displayName: Schema validity error : Element 'displayName': [facet 'minLength'] The value has a length of '0'; this underruns the allowed minimum length of '1'.

  distPkg.xml:45: element displayName: Schema validity error : Element 'displayName': '' is not a valid value of the atomic type 'NonEmptyString'.

  distPkg.xml failed schema validation

- Element value data type error

  Erroneous XML input text:

  ```
  <allowedCredentialStorage>
    <duration>0</duration>
  </allowedCredentialStorage>
  ```

  Error message:

  distPkg.xml:38: element duration: Schema validity error : Element 'duration': '0' is not a valid value of the local atomic type.

  distPkg.xml failed schema validation

- Extra white space with an enumerated value

  Erroneous XML input text:

  ```
  <associationMode>
    <wpa-Enterprise>TKIP </wpa-Enterprise>
  </associationMode>
  ```

  Error message:

  distPkg.xml:81: element wpa-Enterprise: Schema validity error : Element 'wpa-Enterprise': [facet 'enumeration'] The value 'TKIP ' is not an element of the set{'AES', 'TKIP'}.

  distPkg.xml:81: element wpa-Enterprise: Schema validity error : Element 'wpa-Enterprise': 'TKIP ' is not a valid value of the atomic type 'WpaEncryption'.

  distPkg.xml failed schema validation

File Reference Error

The distribution package schema contains several elements that serve as a reference to an external file that is being designated for inclusion in the XML instance file.

Some examples of file reference errors are:

CA Certificate file:

- Incorrect path for file (designated file not present)

XML input text:

<caReference>E:\path\CaCertFile.pem</caReference>

Error message:

CA certificate file: "E:\path\CaCertFile.pem" doesn't exist

- Incorrect file type

  XML input text:

  <caReference>CaCertFile</caReference>

  Error message:

  CA certificate file: "CaCertFile" should be in .pem format

PAC file:

- Incorrect path for file (designated file not present)

  XML input text:

  <aIdReference>E:\path\pacRefFile</aIdReference>

  Error message:

  Pac file "E:\path\pacRefFile" processing error: can not open pac file E:\path\pacRefFile

- PAC password not provided or invalid

  XML input text: optional element, secretKey, not configured.

```
<reference>
  <aIdReference>pacRefFile</aIdReference>
</reference>
```

  XML input text: password value incorrect

```
<reference>
  <aIdReference>pacRefFile</aIdReference>
  <secretKey>1234</secretKey>
</reference>
```

  Error message:

  Pac file "pacRefFile" processing error: Invalid password to access pac file

Business Rules Verification Errors

The list of business rule verification errors, with examples, follows:

See the referenced element descriptions in Chapter 2, “Schema Elements” for more information.

- Rule 1.1 Limits on wired networks - only 1 allowed

  Erroneous XML input text:

```
<networks>
  <wiredNetwork>
    <displayName>Test 1.1-1</displayName>
    ...
  </wiredNetwork>
  <wiredNetwork>
    <displayName>Test 1.1-2</displayName>
```

```
    ...
  </wiredNetwork>
</networks>
```

Error message:

[Rule 1.1 violation] Only one wired network allowed!

See the description for element: *wiredNetwork.*

- Rule 1.2 Limits on networks with same SSID - only 1 allowed

  Erroneous XML input text:

```
<networks>
  <wifiNetwork>
    <displayName>Test 1.2-1</displayName>
    <ssid>SSID1</ssid>
    ...
  </wifiNetwork>
  <wifiNetwork>
    <displayName>Test 1.2-2</displayName>
    <ssid>SSID1</ssid>
    ...
  </wifiNetwork>
</networks>
```

  Error message:

  [Rule 1.2 violation] The following ssid(s) "SSID1" are duplicated!

  See the description for element: *ssid.*

- Rule 2.1.1 Authenticating wireless networks require the specification of at least one authentication method.

  Erroneous XML input text:

```
<wifiNetwork>
  <displayName>Test 2.1.2-1</displayName>
  ....
      <eapMethods/>
```

  Error message:

  [Rule 2.1.1 violation] Wifi authentication Networks "Test 2.1.1-1" should use at least one of the following methods: leap, eapTls, eapTtls, eapPeap or eapFast

  See the description for element: *eapMthods.*

- Rule 2.1.2 Authenticating wired networks require the specification of at least one authentication method.

  Erroneous XML input text:

```
<wiredNetwork>
  <displayName>Test 2.1.2-1</displayName>
  ....
      <eapMethods/>
```

  Error message:

  [Rule 2.1.2 violation] Wired authentication Network "Test 2.1.2-1" should use at least one of the following methods: eapMd5, eapMschapv2, eapGtc, leap, eapTls, eapTtls, eapPeap, eapFast

See the description for element: *eapMthods*.

- Rule 2.1.3 Authenticating networks using a tunneled authentication method require the specification of at least one corresponding inner method.

  Erroneous XML input text:

  Case 1—TTLS specific:

```
<displayName>Test 2.1.3-1</displayName>
...
   <eapMethods>
      <eapTtls>
         ...
         <innerMethods>
            <eap/>
         </innerMethods>
```

  Case 2—FAST, PEAP, TTLS:

```
<displayName>Test 2.1.3-5</displayName>
...
   <eapMethods>
      <eapPeap>
         ...
         <innerMethods/>
```

  Error message:

  [Rule 2.1.3 violation] Networks "Test 2.1.3-1", "Test 2.1.3-5" use a tunneled method without defining an inner method!

  See the description for element: *innerMethods, eap*.

- Rule 2.3 Static credentials (password) must be configured if the collection method is static.

  Erroneous XML input text:

```
<displayName>Test 2.2-2</displayName>
...
   <collectionMethod>
      <static/>
   </collectionMethod>
   <useAnonymousId>true</useAnonymousId>
   <staticIdentity encrypt="true">ItsMe</staticIdentity>
```

  Error message:

  [Rule 2.3 violation] Networks "Test 2.3-2" use static collection method without defining a static password!

  See the description for element: *static*.

- Rule 2.4a For user authentication, static credentials require a password-based EAP method.

  Erroneous XML input text:

  Case 1—Mismatch between settings for *collectionMethod* and *eapMethods*.

```
<displayName>Test 2.4a-1</displayName>
...
   <collectionMethod>
      <static/>
   </collectionMethod>
```

```
...
<eapMethods>
  <eapTls>
```

Case 2—Mismatch between settings for *collectionMethod* and *innerEapMethods*.

```
<displayName>Test 2.4a-3</displayName>
...
  <collectionMethod>
    <static/>
  </collectionMethod>
  ...
  <eapMethods>
    <eapFast>
      ...
      <innerEapMethods>
        <eapTls>
```

Error message:

> [Rule 2.4a violation] Networks "Test 2.4a-1", "Test 2.4a-3" use static credential collection for user authentication and should not define certificate based methods!

See the description for element: *static*.

- Rule 2.4b For machine authentication, static credentials require a password-based EAP method. Additionally, EAP FAST PACs can not be used.

  Erroneous XML input text:

  Case 1—Mismatch between settings for *collectionMethod* and *eapMethods*.

  ```
  <displayName>Test 2.4b-1</displayName>
  ...
    <collectionMethod>
      <static/>
    </collectionMethod>
    ...
    <eapMethods>
      <eapTls>
      ...
      <eapFast>
  ```

  Case 2—Mismatch between settings for *collectionMethod* and *innerEapMethods*.

  ```
  <displayName>Test 2.4b-3</displayName>
  ...
    <collectionMethod>
      <static/>
    </collectionMethod>
    ...
    <eapMethods>
      <eapPeap>
        ...
        <innerEapMethods>
          <eapTls>
  ```

  Error message:

  > [Rule 2.4b violation] Networks "Test 2.4b-1", "Test 2.4b-3" use static credential collection for machine authentication and should not define methods using pac(s) or certificates!

See the description for element: *static.*

- Rule 2.5 Client certificate usage requires configuring a certificate source.

  Erroneous XML input text:

```
<displayName>Test 2.5-1</displayName>
...
   <eapMethods>
      <eapFast>
         ...
         {Missing optional element certificateSource which is required in this case.}
         <innerEapMethods>
            <eapTls>
```

  Error message:

  [Rule 2.5 violation] Networks "Test 2.5-1" missing required certificate!

  See the description for element: *certificateSource.*

- Rule 2.6 In a user-only connection context configured for network connectivity before logon, client certificates are supported only through smartcards - client certificates in the Windows certificate store are not supported.

  Erroneous XML input text:

  Case 1—Outer method.

```
<displayName>Test 2.6-1</displayName>
...
   <userAuthentication>
      <autoConnect>
         <connectBeforeLogon>true</connectBeforeLogon>
      </autoConnect>
      ...
      <eapMethods>
         <eapTls>
            <certificateSource>
               <smartCardOrOsCertificate/> {Must be smartcard only.}
            </certificateSource>
```

  Case 2—Inner method.

```
<displayName>Test 2.6-2</displayName>
...
   <userAuthentication>
      <autoConnect>
         <connectBeforeLogon>true</connectBeforeLogon>
      </autoConnect>
      ...
      <eapMethods>
         <eapFast>
            <certificateSource>
               <smartCardOrOsCertificate/> {Must be smartcard only.}
            </certificateSource>
            <innerEapMethods>
               <eapTls>
```

  Error message:

[Rule 2.6 violation] Networks "Test 2.6-1", "Test 2.6-2" must be smartCardOnlyCertificate!

See the description for element: *connectBeforeLogon.*

- Rule 2.7 Mandating server validation requires the configuring of a trusted server certificate rule.

  Erroneous XML input text:

  Case 1—Outer method.

```
<displayName>Test 2.7-1</displayName>
...
      <eapMethods>
        <eapTls>
          <validateServerIdentity>true</validateServerIdentity>
          ...
    <serverValidation>
      {Missing optional element validationRules which is required in this case.}
      <trustAnyRootCaFromOs/>
    </serverValidation>
```

  Case 2—Inner method.

```
<displayName>Test 2.7-2</displayName>
...
      <eapMethods>
        <eapPeap>
          <validateServerIdentity>true</validateServerIdentity>
          ...
          <innerEapMethods>
            <eapTls>
    ...
    <serverValidation>
      {Missing optional element validationRules which is required in this case.}
      <trustAnyRootCaFromOs/>
    </serverValidation>
```

  Error message:

  [Rule 2.7 violation] Networks "Test 2.7-1", "Test 2.7-2" must be validated with either matchSubjectAlternateName or matchSubjectCommonName!

  See the description for element: *serverValidation.*

- Rule 2.8 Mandating server validation for FAST requires the configuring of either a trusted server certificate rule or a trusted server PAC rule.

  Erroneous XML input text:

```
<displayName>Test 2.8-1</displayName>
...
      <eapMethods>
        <eapFast>
          <validateServerIdentity>true</validateServerIdentity>
          ...
    <serverValidation>
      {Missing optional element validationRules or trustedServerIds, one of which is required
in this case.}
      <trustAnyRootCaFromOs/>
    </serverValidation>
```

Error message:

[Rule 2.8 violation] Networks "Test 2.8-1" must be validated with either matchSubjectAlternateName, matchSubjectCommonName or trustedServerIds!

See the description for element: *serverValidation*.

- Rule 2.9 Providing PACs for a network requires configuring FAST.

  Erroneous XML input text:

```
<displayName>Test 2.9-1</displayName>
...
      <pacs>
        <pac>
          <pacReference encrypt=”true”>pacFile</pacReference>
        </pac>
      </pacs>
      <eapMethods>
        <eapPeap> {Must be eapFast.}
      </eapMethods>
```

  Error message:

  [Rule 2.9 violation] Networks "Test 2.9-1" must have eapFast!

  See the description for element: *pacs*.

- Rule 2.11 Machine certificate must be from OS store.

  Erroneous XML input text:

```
<displayName>Test 2.11-2</displayName>
...
  <machineAuthentication>
    ...
        <certificateSource>
          <smartCardOnlyCertificate/> {Must be from OS.}
        </certificateSource>
```

  Error message:

  [Rule 2.11 violation] Networks "Test 2.11-2" must be configured to use the OS as a certificate source (machine authentication)!

  See the description for element: *smartCardOnlyCertificate*.

- Rule 2.12 The logical expression for the value of identity patterns must use the defined keywords and be properly bracketed.

  Case 1: Incorrect keyword

  Erroneous XML input text:

```
<displayName>Corporate User</displayName>
...
  <userAuthentication>
    ...
    <protectedIdentityPattern>&lt;user&gt;</protectedIdentityPattern>
```

  Error message:

  [Rule 2.12 violation] Network "Corporate User" protected identity pattern "<user>" is not valid: Unexpected token 'user' at character position 2.

Case 2: Incorrect bracketing

Erroneous XML input text:

```
<displayName>Corporate Machine</displayName>
...
   <machineAuthentication>
      ...
      <unprotectedIdentityPattern>host/&lt;fqhn&gt;&gt;</unprotectedIdentityPattern>
```

Error message:

[Rule 2.12 violation] Network "Corporate Machine" unprotected identity pattern "host/<fqhn>>" is not valid: Unexpected token '>' at the end of the identity pattern.

See the description for elements: *unprotectedIdentityPattern* and *protectedIdentityPattern*.

- Rule 2.13 & 2.14 Identity pattern keywords are connection-context and credential collection method dependent.

  Case 1: wrong connection context usage

  Erroneous XML input text:

```
<displayName>Test 2.13a</displayName>
...
   <machineAuthentication>
      ...
      <collectionMethod>
         <auto/>
      </collectionMethod>
      <unprotectedIdentityPattern>host/&lt;username&gt;</unprotectedIdentityPattern>
```

  Error message:

  [Rule 2.13a violation] Network "Test 2.13a" unprotected identity pattern "host/<username>" is not valid: token '<username>' at character position 6 may not be used with the configured machine connection context.

  Case 2: wrong credential collection method

  Erroneous XML input text:

```
<displayName>Test 2.14</displayName>
...
   <userAuthentication>
      ...
      <collectionMethod>
         <static/>
      </collectionMethod>
      <unprotectedIdentityPattern>&lt;username&gt;</unprotectedIdentityPattern>
```

  Error message:

  [Rule 2.14 violation] Network "Test 2.14" unprotected identity pattern "<username>" is not valid: token '<username>' at character position 1 may not be used with the static credential collection method with the configured user connection context.

  See the description for elements: *unprotectedIdentityPattern* and *protectedIdentityPattern*.

- Rule 2.15 Tunneled methods require a protected identity.

  Erroneous XML input text:

```
<displayName>Test 2.15</displayName>
...
   <userAuthentication>
      ...
      ...
      <unprotectedIdentityPattern>&lt;username&gt;@&lt;domain&gt;</unprotectedIdentityP
attern>
      <eapMethods>
         <eapFast>
```

Error message:

[Rule 2.15 violation] Network "Test 2.15" is not valid: a protected identity pattern must be supplied when configured for a tunneled EAP method but is not required when configured only for a non-tunneled method.

See the description for elements: *unprotectedIdentityPattern* and *protectedIdentityPattern.*

- Rule 3.1a Network policy for association mode must include *open* to support networks with no authentication or shared secrets.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedAssociationModes>
      <wpa-Enterprise/> {No open networks configured.}
   </allowedAssociationModes>
....
<networks>
   <wifiNetwork>
      <displayName>Test 3.1a-1</displayName>
      ...
      <openNetworkUserConnection> {Not allowed}
      ...
   </wifiNetwork>
   <wiredNetwork>
      <displayName>Test 3.1a-2</displayName>
      <openNetworkMachineConnection/> {Not allowed}
   </wiredNetwork>
```

  Error message:

  [Rule 3.1a violation] Networks "Test 3.1a-1", "Test 3.1a-2" openNetworkMachineConnection or openNetworkUserConnection only allowed when Open mode is selected!

  See the description for element: *openNetworkUserConnection, openNetworkMachineConnection.*

- Rule 3.1b Network policy for association mode must include *wep* to support any WEP shared key network.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedAssociationModes>
      <wpa-Enterprise/> {No WEP configured.}
   </allowedAssociationModes>
....
<networks>
   <wifiNetwork>
      <displayName>Test 3.1b-1</displayName>
```

```
...
<sharedKeyNetwork>
   ...
   <wep> {Not allowed}
```

Error message:

[Rule 3.1b violation] Networks "Test3.1b-1": wep with either ieee80211Authentication/open or ieee80211Authentication/shared only allowed when policy wep mode is selected!

See the description for element: *wep.*

- Rule 3.1c Network policy for association mode must include *wpa-Personal* to support a WPA-Personal shared key network.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpa-Enterprise/> {No WPA-Personal configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 3.1c-1</displayName>
    ...
    <sharedKeyNetwork>
      ...
      <wpa> {Not allowed}
```

  Error message:

  [Rule 3.1c violation] Networks "Test 3.1c-1": keySettings/wpa only allowed when wpa-Personal mode is selected!

  See the description for element: *wpa.*

- Rule 3.1d Network policy for association mode must include *wpa2-Personal* to support a WPA2-Personal shared key network.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpa-Enterprise/> {No WPA2-Personal configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 3.1d-1</displayName>
    ...
    <sharedKeyNetwork>
      ...
      <wpa2> {Not allowed}
```

  Error message:

  [Rule 3.1d violation] Networks "Test 3.1d-1": keySettings/wpa2 only allowed when wpa2-Personal mode is selected!

  See the description for element: *wpa2.*

- Rule 3.1e Network policy for association mode must include *wep* to support any dynamic WEP authenticating network.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpa-Enterprise/> {No WEP configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 3.1e-1</displayName>
      ...
      <authenticationNetwork>
        ...
        <associationMode>
          <dynamicWep> {Not allowed}
  ```

  Error message:

  [Rule 3.1e violation] Network "Test 3.1e-1": associationMode/dynamicWep only allowed when policy wep mode is selected!

  See the description for element: *dynamicWep*.

- Rule 3.1f Network policy for association mode must include *wpa-Enterprise* to support a WPA-Enterprise network.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpa-Enterprise/> {No WPA-Enterprise configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 3.1f-1</displayName>
      ...
      <authenticationNetwork>
        ...
        <associationMode>
          <wpa-Enterprise>TKIP</wpa-Enterprise> {Not allowed}
  ```

  Error message:

  [Rule 3.1f violation] Network "Test 3.1f-1": associationMode/wpa-Enterprise only allowed when wpa-Enterprise mode is selected!

  See the description for element: *wpa-Enterprise.*

- Rule 3.1g Network policy for association mode must include *wpa2-Enterprise* to support a WPA2-Enterprise network.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpa-Enterprise/> {No WPA2-Enterprise configured.}
    </allowedAssociationModes>
  ```

```
....
<networks>
   <wifiNetwork>
      <displayName>Test 3.1g-1</displayName>
      ...
      <authenticationNetwork>
         ...
         <associationMode>
            <wpa2-Enterprise>AES</wpa2-Enterprise> {Not allowed}
```

Error message:

[Rule 3.1g violation] Network "Test 3.1g-1": associationMode/wpa2-Enterprise only allowed when wpa2-Enterprise mode is selected!

See the description for element: *wpa2-Enterprise*.

- Rule 3.2a Network policy for EAP methods must include *eapMd5* to support authenticating wired networks configured for EAP-MD5.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedEapMethods>
      <eapFast/> {No EAP-MD5 configured.}
   </allowedEapMethods>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.2a</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <eapMd5> {Not allowed}
```

  Error message:

  [Rule 3.2a violation] Network "Test 3.2a" : eapMethod/eapMd5 requires allowedEapMethods/eapMd5.

  See the description for element: *eapMethod/eapMd5*.

- Rule 3.2b Network policy for EAP methods must include *eapMschapv2* to support authenticating wired networks configured for EAP-MSCHAPv2.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedEapMethods>
      <eapFast/> {No EAP-MSCHAPv2 configured.}
   </allowedEapMethods>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.2b</displayName>
      ...
      <authenticationNetwork>
```

```
...
<eapMethods>
   <eapMschapv2> {Not allowed}
```

Error message:

[Rule 3.2b violation] Network "Test 3.2b" : eapMethod/eapMschapv2 requires allowedEapMethods/eapMschapv2.

See the description for element: *eapMethod/eapMschapv2*.

- Rule 3.2c Network policy for EAP methods must include *eapGtc* to support authenticating wired networks configured for EAP-GTC.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedEapMethods>
      <eapFast/> {No EAP-GTC configured.}
   </allowedEapMethods>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.2c</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <eapGtc> {Not allowed}
```

  Error message:

  [Rule 3.2c violation] Network "Test 3.2c" : eapMethod/eapGtc requires allowedEapMethods/eapGtc.

  See the description for element: *eapMethod/eapGtc*.

- Rule 3.2d Network policy for EAP methods must include *leap* to support authenticating wired or wireless networks configured for EAP-LEAP.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedEapMethods>
      <eapFast/> {No EAP-LEAP configured.}
   </allowedEapMethods>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.2d</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <leap> {Not allowed}
```

  Error message:

  [Rule 3.2d violation] Network "Test 3.2d" : eapMethod/leap requires allowedEapMethods/leap.

  See the description for element: *eapMethod/leap*.

- Rule 3.2e Network policy for EAP methods must include *eapTls* to support authenticating wired or wireless networks configured for EAP-TLS.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedEapMethods>
    <eapFast/> {No EAP-TLS configured.}
  </allowedEapMethods>
....
<networks>
  <wiredNetwork>
    <displayName>Test 3.2e</displayName>
    ...
    <authenticationNetwork>
      ...
      <eapMethods>
        <eapTls> {Not allowed}
```

  Error message:

  [Rule 3.2e violation] Network "Test 3.2e" : eapMethod/eapTls requires allowedEapMethods/eapTls.

  See the description for element: *eapMethod/eapTls.*

- Rule 3.2f Network policy for EAP methods must include *eapTtls* to support authenticating wired or wireless networks configured for EAP-TTLS.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedEapMethods>
    <eapFast/> {No EAP-TTLS configured.}
  </allowedEapMethods>
....
<networks>
  <wiredNetwork>
    <displayName>Test 3.2f</displayName>
    ...
    <authenticationNetwork>
      ...
      <eapMethods>
        <eapTtls> {Not allowed}
```

  Error message:

  [Rule 3.2f violation] Network "Test 3.2f" : eapMethod/eapTtls requires allowedEapMethods/eapTtls.

  See the description for element: *eapMethod/eapTtls*.

- Rule 3.2g Network policy for EAP methods must include *eapPeap* to support authenticating wired or wireless networks configured for EAP-PEAP.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedEapMethods>
    <eapFast/> {No EAP-PEAP configured.}
  </allowedEapMethods>
```

```
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.2g</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <eapPeap> {Not allowed}
```

Error message:

[Rule 3.2g violation] Network "Test 3.2g" : eapMethod/eapPeap requires allowedEapMethods/eapPeap.

See the description for element: *eapMethod/eapPeap*.

- Rule 3.2h Network policy for EAP methods must include *eapFast* to support authenticating wired or wireless networks configured for EAP-FAST.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedEapMethods>
      <eapPeap/> {No EAP-FAST configured.}
   </allowedEapMethods>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.2h</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <eapFast> {Not allowed}
```

  Error message:

  [Rule 3.2h violation] Network "Test 3.2h" : eapMethod/eapFast requires allowedEapMethods/eapFast.

  See the description for element: *eapMethod/eapFast.*

- Rule 3.3a Network policy for credential storage must include *forever* to support networks with credentials that are to be saved across logins.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedCredentialStorage>
      {forever not configured.}
   </allowedCredentialStorage>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.3a</displayName>
      ...
      <authenticationNetwork>
```

```
...
   <credentialsStorage>
      <forever> {Not allowed}
```

Error message:

[Rule 3.3a violation] Networks "Test 3.3a": credentialStorage/forever requires that networkPolicy/allowedCredentialStorage/forever be selected.

See the description for element: *credentialStorage.*

- Rule 3.3b Network policy for credential storage must include *logonSession* to support networks with credentials that are to be saved only during the current login session.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedCredentialStorage>
      {logonSession not configured.}
   </allowedCredentialStorage>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.3b</displayName>
      ...
      <authenticationNetwork>
         ...
            <credentialsStorage>
               <logonSession> {Not allowed}
```

  Error message:

  [Rule 3.3b violation] Networks "Test 3.3b": credentialStorage/logonSession requires that networkPolicy/allowedCredentialStorage/logonSession be selected.

  See the description for element: *credentialStorage.*

- Rule 3.3c Network policy for credential storage must include *duration* to support networks with credentials that are to be saved for a preconfigured time period.

  Erroneous XML input text:

```
<networkPolicy>
   <allowedCredentialStorage>
      {duration not configured.}
   </allowedCredentialStorage>
....
<networks>
   <wiredNetwork>
      <displayName>Test 3.3c</displayName>
      ...
      <authenticationNetwork>
         ...
            <credentialsStorage>
               <duration> {Not allowed}
```

  Error message:

  [Rule 3.3c violation] Networks "Test 3.3c": credentialStorage/duration requires that networkPolicy/allowedCredentialStorage/duration be selected.

  See the description for element: *credentialStorage.*

- Rule 3.4 If the network policy requires server validation, then all networks must be configured accordingly.

  Erroneous XML input text:

```
<networkPolicy>
   <serverValidationPolicy>
      <alwaysValidate> {Validation required.}
....
<networks>
   <wifiNetwork>
      <displayName>Test 3.4-1</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <eapFast>
               <validateServerIdentity>false</validateServerIdentity> {Not allowed}
...
   <wifiNetwork>
      <displayName>Test 3.4-2</displayName>
      ...
      <authenticationNetwork>
         ...
         <eapMethods>
            <eapFast>
               <validateServerIdentity>true</validateServerIdentity>
               ...
                  <innerEapMethods>
                     <eapTls>
                        <validateServerIdentity>false</validateServerIdentity> {Not allowed}
```

  Error message:

  [Rule 3.4 violation] Networks "Test 3.4-1", "Test 3.4-2": each Tls, Ttls, Peap, Fast method should require server identity validation in conformance with the policy.

  See the description for element: *validateServerIdentity.*

- Rule 3.5 The logical expression for the client certificate Extended Key Usage filtering must use the defined keywords and be properly parenthesized.

  Case 1—Incorrect keyword

  Erroneous XML input text:

```
<networkPolicy>
   ...
   <allowedClientCertificates>
      <certificateEkuFilterExpression>(SmartCardLogon or not
IpsecTunnel1)</certificateEkuFilterExpression> {keyword misspelled}
   </allowedClientCertificates>
</networkPolicy>
```

  Error message:

  [Rule 3.5 violation] The certificate Extended Key Usage (EKU) expression is not valid: Unexpected token 'IpsecTunnel1' at character position 24

  Case 2—Errored expression

Erroneous XML input text:

```
<networkPolicy>
   ...
   <allowedClientCertificates>
      <certificateEkuFilterExpression>(SmartCardLogon or not
IpsecTunnel</certificateEkuFilterExpression> {missing closing parenthesis}
   </allowedClientCertificates>
</networkPolicy>
```

Error message:

[Rule 3.5 violation] The certificate Extended Key Usage (EKU) expression is not valid: Expected ')' at the end of the expression.

See the description for element: *certificateEkuFilterExpression.*

- Rule 4 End-user is not permitted to override an initial setting of a single-homed network.

  Erroneous XML input text:

  ```
  <networkPolicy>
     <allowUserSimultaneousConnectionsControl>true</allowUserSimultaneousConnectionsCo
  ntrol> {Not allowed.}
  ....
  <stationSettings>
     <simultaneousConnections>singleHomed</simultaneousConnections>
  ```

  Error message:

  [Rule 4 violation] If stationSettings/simultaneousConnections is singleHome, networkPolicy/allowUserSimultaneousConnectionsControl must be false!

  See the description for element: *allowUserSimultaneousConnectionsControl.*

- Rule 5a SSC must be configured for at least one media type.

  Erroneous XML input text:

  ```
  <userControlPolicy>
     ...
     <allowedMedia></allowedMedia> {Missing a child element.}
  ```

  Error message:

  [Rule 5a violation] At least one media type must be specified for userControyPolicy/allowedMedia!

  See the description for element: *allowedMedia.*

- Rule 5b The general policy must be configured to allow wired media to support the configuring of a wired network.

  Erroneous XML input text:

  ```
  <networks>
     <wiredNetwork> {Not allowed.}
        <displayName>Test 5b</displayName>
  ...
  <userControlPolicy>
     ...
     <allowedMedia>
        <wifi/> {Wired not configured.}
     </allowedMedia>
  ```

Error message:

[Rule 5b violation] Network "Test 5b": (wired) may not be present unless userControlPolicy/allowedMedia/wired is present.

See the description for element: *wiredNetwork*.

- Rule 5c The general policy must be configured to allow wireless media to support the configuring of a Wi-Fi network.

  Erroneous XML input text:

```
<networks>
   <wifiNetwork> {Not allowed.}
      <displayName>Test 5c</displayName>
...
<userControlPolicy>
   ...
   <allowedMedia>
      <wired/> {Wireless not configured.}
   </allowedMedia>
```

  Error message:

  [Rule 5c violation] Network "Test 5c": (wifi) may not be present unless userControlPolicy/allowedMedia/wifi is present.

  See the description for element: *wifiNetwork*.

Scripting Errors

Return codes are implemented for identification of failures at each phase of processing. The following lists all the application return codes:

- 0 Success
- 1 Wrong arguments
- 2 Unknown configuration file version
- 3 Schema validation failed
- 4 Business rules validation failed
- 5 Referenced files cannot be found
- -1 Unexpected error (see stderr for details)