Cisco Secure Services Client Administrator Guide

Software Release 5.0.0

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-13686-01

Cisco Secure Services Client Administrator Guide, Release 5.0

CONTENTS

Preface

The preface provides an overview of the *Cisco Secure Services Client Administrator Guide, Release 5.0*, 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, “Deployment Example Using the SSC Management Utility GUI,” provides a deployment example that illustrates how to use the Cisco SSC Management Utility to create an enterprise-specific distribution package.

Chapter 3, “Troubleshooting,” describes the Cisco SSC Release 5.0 log file, the log message formats, the log packager utility, and the steps to take when you discover a problem with the SSC client.

Appendix A, "Postprocessing Verification Errors," contains a listing of error types and error messages used with the postprocessing utility.

Appendix B, "Cisco Secure Client Services Release 5.0 Log Messages," lists the log messages produced by the SSC Release 5.0 client.

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 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

Notices

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Copyright © 1998-2007 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS”' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".

   The word 'cryptographic' can be left out if the routines from the library being used are not cryptography-related.
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License].

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 configuration file does the IT Administrator have full access to all settings and network configurations.

- Default download package—contains a default configuration that is configured with a non-expiring, wired only license. You can download a trial wireless license from cisco.com at this URL:

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

When 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 configuration 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.

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
- Additional EAP methods: LEAP, EAP-PEAP, EAP-TTLS, EAP-MD5

Supported Operating System Environments

The supported operating system environments are:

- Windows XP Professional (SP1, SP2), Windows 2000 (SP4), or Windows 2003 server

Note 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 consists of the configuration file which 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 network media support.
  - Network policy

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

- Connection Settings

  Configures the global operational aspects of making network connections.

- Groups

  A group, fundamentally, is a collection of configured connections (networks). Every configured connection must belong to some group or be defined under the *globalNetworks* section in the distribution package.

  Note End-users can add networks only to groups and not to the *globalNetworks* section (because they typically do not have access to the management tool that would allow them to sign the distribution package).

  Classifying connections into groups provides multiple benefits:

  – Improved user-experience when attempting to make a connection. It is important to understand how the client establishes a network connection in order to illustrate this point. The client works through the list of available networks in the order in which they are defined until a successful connection is made.

    For example, an enterprise end-user who travels often outside the business campus might configure connections for public WiFi networks or hotspots. Without groups, a newly configured home network is added to the end of this list, which could be quite large. The client works through the list from the beginning, including all the public networks, before establishing a connection to the home network. This greatly increases the time to get connected to the last added network.

  – Easier management of configured connections. In the previous example, if an end-user attempts to delete some connections to get connected quicker, the deleted connections might be needed at a later time. However, if the connection list is divided into groups, each list would be much smaller. When using groups, it is easy to switch between the groups to obtain faster connectivity.

  A group may be created by an administrator or an end-user. There must be at least one group defined in the configuration. If there are multiple groups, one group must be chosen as the *active* group and the client attempts to make a network connection using the connections defined in the active group. End-users can add or delete networks only from the active group. Groups can be added or deleted by clicking on the *Configure Groups* button on the main screen of the client GUI.

  Networks that are defined in the *globalNetworks* section of the distribution package are available in every group at the top of the list. Because only enterprise administrators can create *globalNetworks*, this provides an administrator with control over the enterprise networks that an end-user can connect to, even in the presence of user-defined networks. An end-user is not able to delete administrator configured networks.

  It is important to note that a typical end-user of an enterprise network does not need to have a knowledge of groups in order to use this client. It is the responsibility of the administrator to always specify a default group in the created distribution package. If there is just one group available, the client selects that as the active group. The end-user can add or delete their own networks without using groups.

  Note A group selection is not maintained across reboots or repairs of the client. When the client is repaired or restarted, the client always goes back to the first configured group in the configuration.xml file.

- Networks

  Networks contain 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" 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

Click **Download Software > Client Adapters and Client Software** and follow the prompts 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, configuration.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.

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 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. 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 to create an XML instance file either from scratch or by cut-and-paste from known examples.

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 serves 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 name of your distribution package must be configuration.xml.

Step 2 Pass the generated package distribution .xml file through the SSC postprocess command line utility, sscManagementUtility.exe. The sscManagementUtility 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 (the distribution .xml file that was just generated). The optional files include the PACs and the CA certificates.
- 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 validated and signed distribution package from scratch
- Import an existing unsigned file to use as a starting point for making changes
- Postprocess an existing distribution package

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. Invoking the utility, starts the GUI

Postprocessing Utility

The syntax of the command-line version of the postprocessing utility is shown below. .

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

sscManagementUtility.com help

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

sscManagementUtility.com 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 A, "Postprocessing Verification Errors" for an overview of errors produced during postprocessing.

Note The utility (sscManagementUtility.com) requires 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.

- configuration.xsd, schema file

  Release numbering is defined in the schema itself. Each instantiated distribution package xml file retains the release numbering scheme of its associated schema file.

- validateRules.xsl, business rules file

  Release numbering is controled 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 maintanence, 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

Release Numbering for SSC

The management toolkit package (.zip) file and previous releases of the installation file (.msi) obtained from Cisco have the following format:

SSCMgmtToolkit_A.B.C.xxxx.zip or Cisco_SSC-{OS}-A_B_C_xxxx.msi

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

SSCMgmtToolkit_A.B.C.xxxx.zip or Cisco_SSC-XP2K-A.msi, where *A* indicates major release change.

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 Toolkit Package	Supports These SSC Releases
SSCMgmtToolkit_5.0.0.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
	Cisco_SSC-XP2K-5.msi

Compatibility Between Distribution Package and SSC

SSC Release 5.0 is a major software release and employs a new schema. This schema is not compatible with the schema of prior SSC releases. To aid in the translation of the old schema to the new schema, a schema conversion tool is provided. For additional information see the "Upgrading SSC Release 4.1.x Installations to SSC Release 5.0" section on page 1-12.

This conversion tool will not convert an administrator created SSC Release 4.1 distribution package (schema version 4.1.x) to the SSC Release 5.0 schema. Instead, it will use SSC Release 4.1 internal configuration files (files in *Program Files\Cisco Systems\Cisco Secure Services Client*) to translate the administrator configured networks to the SSC Release 5.0 schema.

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 enterprise deployment utility (sscPackageGen) 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). The syntax of the utility is:

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

Table 1-3 sscPackageGen Command Elements

Command Elements	Meaning
insert	Command to create a msi 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.

End-User Initial Installation

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

- Enterprise deployment installation method
- Legacy installation method (recommended)

Enterprise Deployment Installation Method

SSC and its companion distribution package are deployed as a single file and installed in a single operation. Recall that any required support files (CA certificates and PACs) have already been added to the distribution package itself.

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-5) and your validated and postprocessed distribution package file (configuration.xml).

```
sscPackageGen insert C:\Cisco_SSC-XP2K-5.msi C:\yourSSCInstallPkg.msi
C:\configuration.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.

(The parameter *norestart* prevents a silent install from rebooting the PC.)

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-5).
2. Update the end-user configuration as outlined in the next section.

Note SSC Release 5.0 and later uses an intermediate driver to control the network adapters. Installation is stopped and the user is informed if it detects the presence of another driver with which SSC is not able to co-exist. You need to either disable or un-install the conflicting application.

Updating End-User Configurations

The legacy update method is used to update an end-user configuration.

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

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

   C:\Documents and Settings\All Users\Application Data\Cisco\
   Cisco Secure Services Client\newConfigFiles

2. Either restart the Cisco Secure Services Client service or from the Help menu, choose **Repair**.

Note SSC also detects and implements the new configuration file whenever it attempts a new connection.

Upgrading SSC Release 4.1.x Installations to SSC Release 5.0

There are two components to upgrading existing SSC 4.1.x releases to SSC Release 5.0:

- All previously deployed administrator (locked) networks from SSC Release 4.1.x must be upgraded to SSC Release 5.0.
- All end-user created networks from SSC Release 4.1.x must be upgraded to SSC Release 5.0

Upgrading Administrator Deployed Networks from SSC Release 4.1.x to SSC Release 5.0

An administrator must have the following SSC Release 5.0 client elements on his PC:

- SSC Release 5.0 installation msi file (Cisco_SSC-XP2K-5.msi)
- Configuration management utility (SSCMgmtToolkit_5.0.0.xxxx.zip)
- Configuration combining tool (ConfigCombiner.exe)
- Configuration conversion tool (ConfigConverter.exe)
- Administrator xslt file (configConvert_3_1_admin.xslt)—used to translate administrator-configured SSC Release 4.1 networks to SSC Release 5.0 schema.
- sscPackageGen that generates a custom installation package

The administrator also must have the current SSC Release 4.x deployment package, translated into SSC Release 4.1.2 internal configuration. This is the *profiles* folder found under the *Program Files\Cisco Systems\Cisco Secure Services Client* folder.

In order to deploy an SSC Release 5.0 client that is equivalently configured to your SSC Release 4.x distribution, you must perform these operations:

1. Use the combining tool (ConfigCombiner.exe) to combine SSC Release 4.1 configuration files into a single file:

   Usage: ConfigCombiner.exe [options]

   Options include:

   --source *directory* or -s *directory*—specifies the source directory path. If the source directory option is not specified, the default value for the source directory is *C:\Program Files\Cisco Systems\Cisco Secure Services Client\profiles*.

   --quiet or -q—do not dump the result

   --help—gives the usage of the tool

   The following illustrates a conbining tool example:

   ```
   ConfigCombiner.exe -q
   ```

   The output of this operation produces a file called *configuration.xml*. The file is located in the folder where the tool was executed. The file contains the information in the multiple folders under *c:\Program Files\Cisco Systems\Cisco Secure Client Services\profiles*.

   Note SSC Release 4.1.x files are not modified in any way as a result of this operation.

2. Use the conversion tool (ConfigConverter.exe) with the administrator XSLT file (configConvert_3_1_admin.xslt) to convert the output of the combining tool into an SSC Release 5.0 configuration.xml file:

   Usage: ConfigConverter.exe [options]

   Options include these values:

   --quiet or -q—specifies do not dump the result

   --output *filename* or -o *filename*—specifies the output XML file

   --input *filename* or -i *filename*—specifies the input XML file

   --xslt *filename* or -xslt *filename*—specifies the XSLT file

   You should specify the *--xslt* file option with the XSLT file name set to **configConvert_3_1_admin.xslt** when you are converting the administrator deployed networks using the ConfigConverter tool. This is the same tool used with a different default xslt file to translate the end-user created networks on end-user systems.

   The following illustrates a conversion tool example:

   ```
   ConfigConverter.exe -i configuration.xml -o configuration.xml
   --xslt configConvert_3_1_admin.xslt
   ```

   The output of this operation is a SSC Release 5.0 schema compatible distribution package with an equivalent configuration of your SSC Release 4.1.x deployed networks.

3. You can now use the management utility to perform these operations:
   - Read in the SSC Release 5.0 configuration.xml (which contains the administrator deployed SSC Release 4.1 networks)
   - If needed, modify the SSC Release 5.0 configuration.xml file and root
   - Sign the SSC Release 5.0 configuration.xml file
4. Run the packageGen tool to bundle the signed configuration.xml file along with the SSC Release 5.0 msi file and then deploy the package.

Upgrading End-User Created SSC Release 4.1.x networks to SSC Release 5.0

When SSC Release 5.0 is installed on a PC as an upgrade, it automatically upgrades the SSC Release 4.1.x end-user created networks to SSC Release 5.0 networks. There is nothing that you, the administrator, or the end-user need to do. The results of the upgrade is as follows:

- SSC Release 5.0 starts running with the deployed administrator configuration file.
- All the end-user created profiles from SSC Release 4.1 are imported into the SSC Release 5.0 client.
- This conversion is done once only during the upgrade.
- SSC Release 4.1 has multiple user xml files on an end-station, but SSC Release 5.0 has only one user-XML file. The conversion tool places the contents of multiple SSC Release 4.1 user-profile files into the single SSC Release 5.0 user XML file. Each user XML file in SSC Release 4.1 corresponds to a group in SSC Release 5.0. The group name is the user xml file name prefixed with *CSSC4_*. The profiles in the *allusers* file is placed in the *CSSC4_allusers* group. It is the responsibility of the end-user to later go through the list of available networks using the GUI and delete any networks they do not want.
- There may be multiple networks created in SSC Release 5.0 for a single network in SSC Release 4.1. This is because the SSC Release 5.0 schema allows only one EAP-method per network, whereas the SSC Release 4.1 schema allows multiple EAP methods per network. This means that a user network from SSC Release 4.1, after conversion to SSC Release 5.0, has a network name that includes both the SSC Release 4.1 network name and the EAP method. This is done to help avoid confusion.
- On an upgrade from SSC Release 4.1 to SSC Release 5.0, all static user credentials are imported into SSC Release 5.0. Also theWEP and PSK credentials input by the user are also imported into SSC Release 5.0. However, any 802.1x credentials are not imported, they need to be re-entered if required.

Pre-Installation of Client Certificates

If the end-user SSC file uses 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 deploy client certificates.

CHAPTER 2

Deployment Example Using the SSC Management Utility GUI

This chapter provides a deployment example that illustrates how to use the Cisco SSC Management Utility to create an enterprise-specific distribution package. This chapter contains this section:

SSC Management Utility GUI Deployment Example

Before you begin using the SSC management utility, remember these points:

- You can click on the *?* symbol next to an entry to obtain context-sensitive help.
- The page that displays when you click *Next* is determined by the choices you made on the current page.

The following steps illustrate how to create an enterprise-specific distribution package using the GUI:

Step 1 Click **sscManagementUtility.exe** and the welcome page displays (see Figure 2-1).

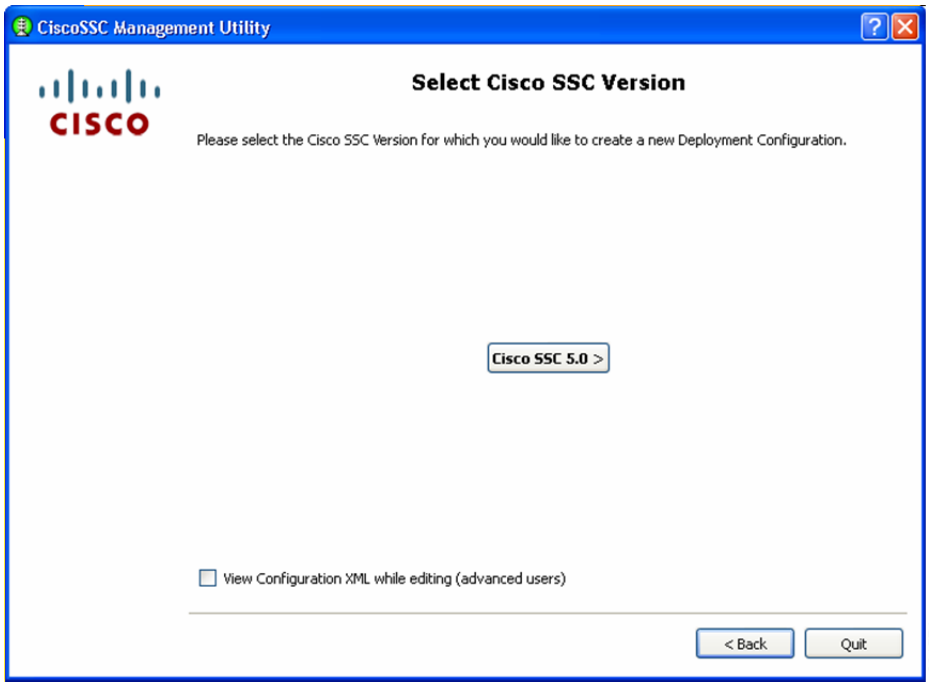
Figure 2-1 SSC Management Utility Welcome Page

There are three choices on this page:

- Create New Configuration Profile—allows you to create a new deployment profile from scratch.
- Modify Existing Configuration Profile—allows you to modify an already created (unprocessed) deployment file.
- Process Existing Configuration Profile—allows you to process an existing unprocessed deployment file. Processing involves these operations:
  - Encrypt credentials and other secrets in the file
  - Pulls in the CA certificates and PAC files that are specified in the file.
  - Signs the resulting file, so that end-users are prevented from tampering with the administrator deployed configuration file.

Step 2 To create a new configuration file from scratch, click **Create New Configuration Profile** and Figure 2-2 displays.

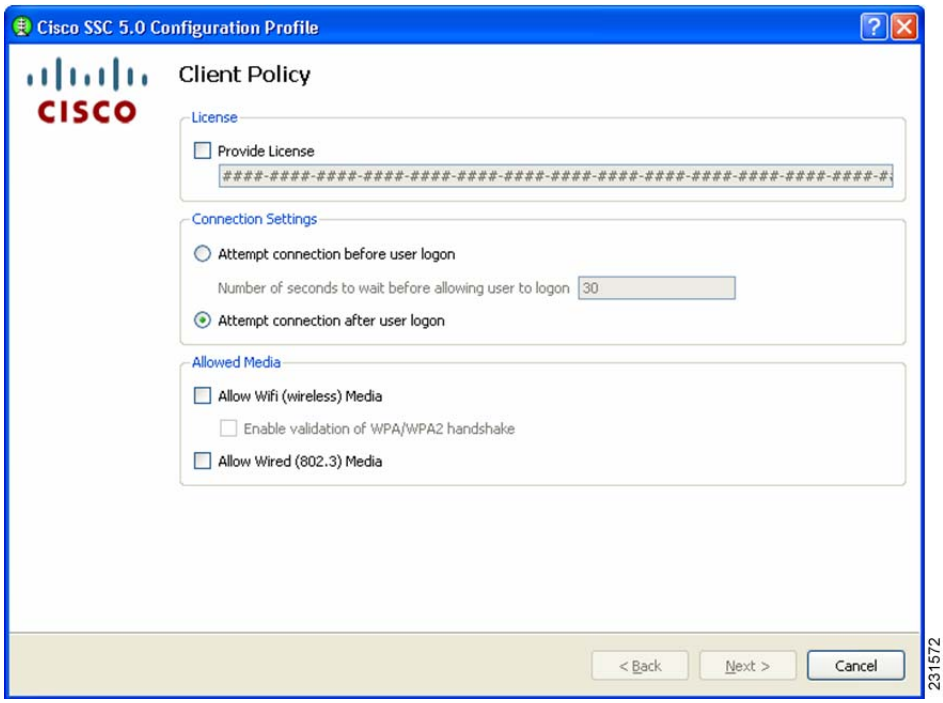
Figure 2-2 Select Cisco SCS Version Page



The SSC management utility enables you to create a configuration file for Cisco SSC releases 5.0, 4.1 and 4.2 (only SCS 5.0 is shown in this illustration).

Step 3 Click **Cisco SSC 5.0** and the Client Policy (Figure 2-3) displays.

Figure 2-3 Cisco Policy Page



Note Cisco SSC release 5.0 does not allow end-users to enter license numbers using the GUI. It is the responsibility of the enterprise administrator to create a distribution package that contains a valid license, so that all end-users have the appropriate licenses.

There are two sections on this page:

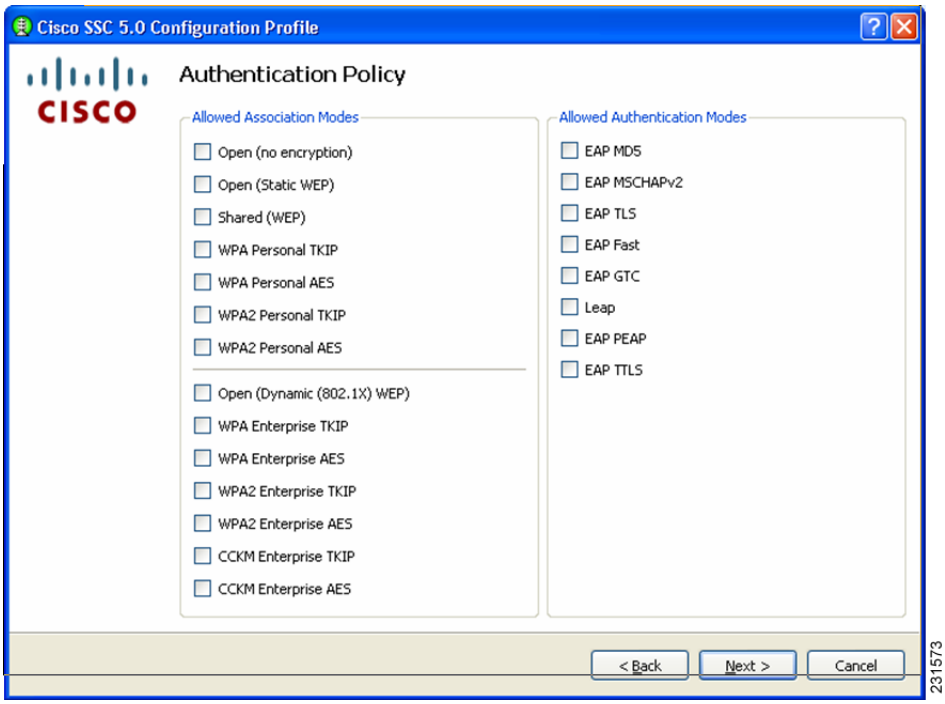
- Connection Settings section—allows you to define whether 802.1x authentication must be attempted before Windows domain authentication, i.e. pre-logon. In the case of pre-logon, you can also specify how long to wait for the connection. If a network connection cannot be established within this time, the Windows logon process continues with user logon.
- Allowed Media section—enables the types of media controlled by the Cisco SSC client.

Note Cisco SSC release 5.0 is single-homed, it allows only one network connection at a time. Also wired connections are prioritized higher than wireless connections.

If wireless media is allowed, you can either enable or disable WPA/WPA2 handshake validation.

Step 4 Choose the desired options on this page and click **Next**. Figure 2-4 displays.

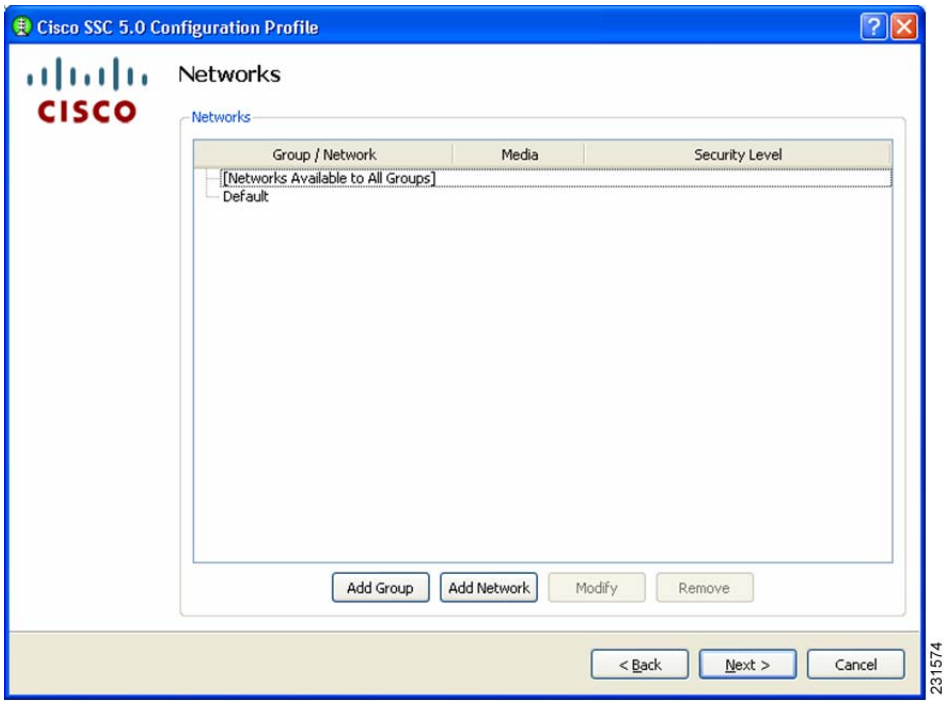
Figure 2-4 Authentication Policy Page



This screen allows you to define network policies - these policies are global. Global policies apply to all networks that you, the administrator, or the user can create.

Step 5 Choose the desired network policy options and click **Next**.The Networks page (Figure 2-5) displays.

Figure 2-5 Networks Page



This screen allows you to configure networks that are pre-define for your enterprise. You can either configure networks that are available across all groups or create groups with specific networks. For additional information on groups, refer to the "Distribution Package" section on page 1-2.

Step 6 To begin creating a group, click **Add Group** and the User Group page (Figure 2-6) displays.

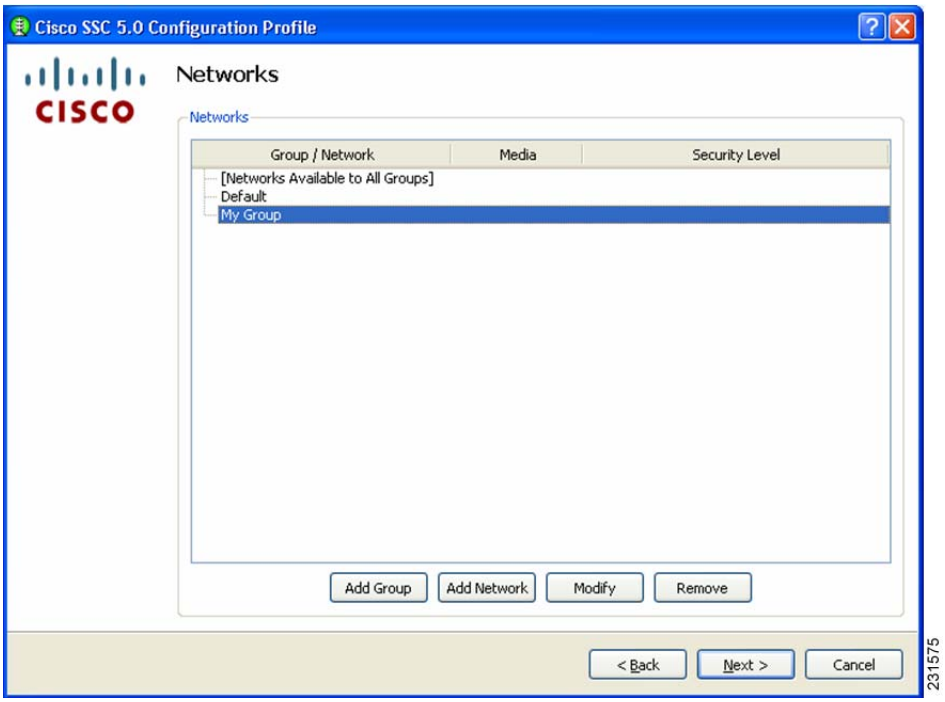
Figure 2-6 Add Group Page

Scan list control— enables you to control whether users can see the scanlist when this group active. There are situations when it might necessary to not allow users to view the scan list, for example, if it is necessary to exclude nearby wireless devices that end-users should not accidentally connect to their networks.

Note This is a per-group setting. For groups created by the end-user using the GUI, the scan list control is set to *Allow the user to see a scan list in this group*.

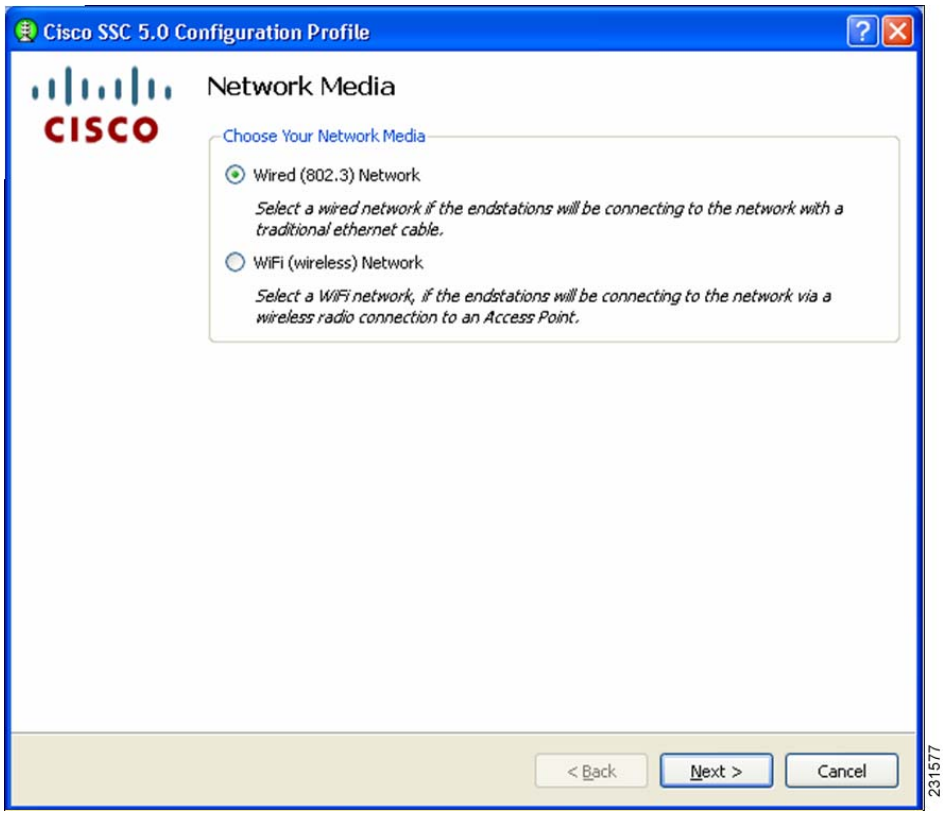
Step 7 Enter the User Group Name and choose the desired scan list control options. When complete, click **OK** and the Networks page redisplays with the new group just created (*My Group* in this example) visible.

Figure 2-7 Network Page with New Group Visible



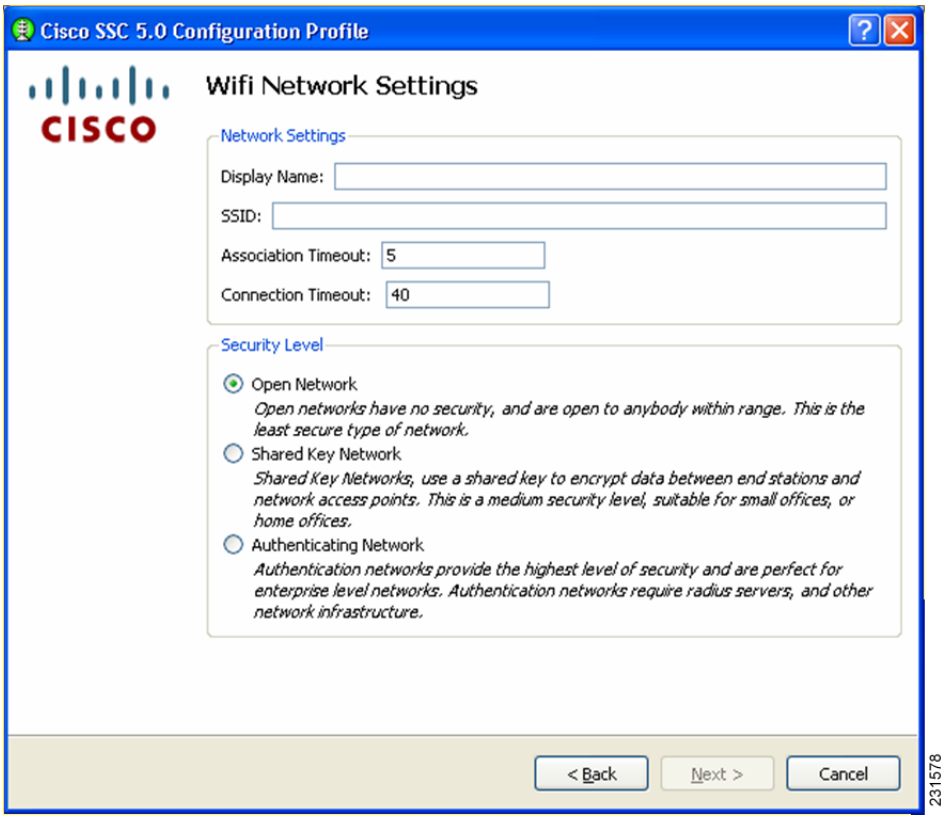
Step 8 To add a network to a newly created group (My Group in this example), click **My Group** to highlight it and click **Add Network**. The Network Media page (Figure 2-8) displays.

Figure 2-8 Network Media Page



Step 9 This page enables you to choose whether you want to add a wired or a wireless network. In this example, choose **Wifi (wireless) Network** to add a wireless network and click **Next**. The WiFi Network Setting page (Figure 2-9) displays.

Figure 2-9 WiFi Network Settings Page



Step 10 This page enables you to create an open (non-secure) network, a shared key network, or an 802.1x authentication network.

Step 11 Enter the network name in the Display Name field.

Step 12 In the SSID field, enter the ssid you want to associate to.

Step 13 Choose a network type, in this example, click **Open Network**.

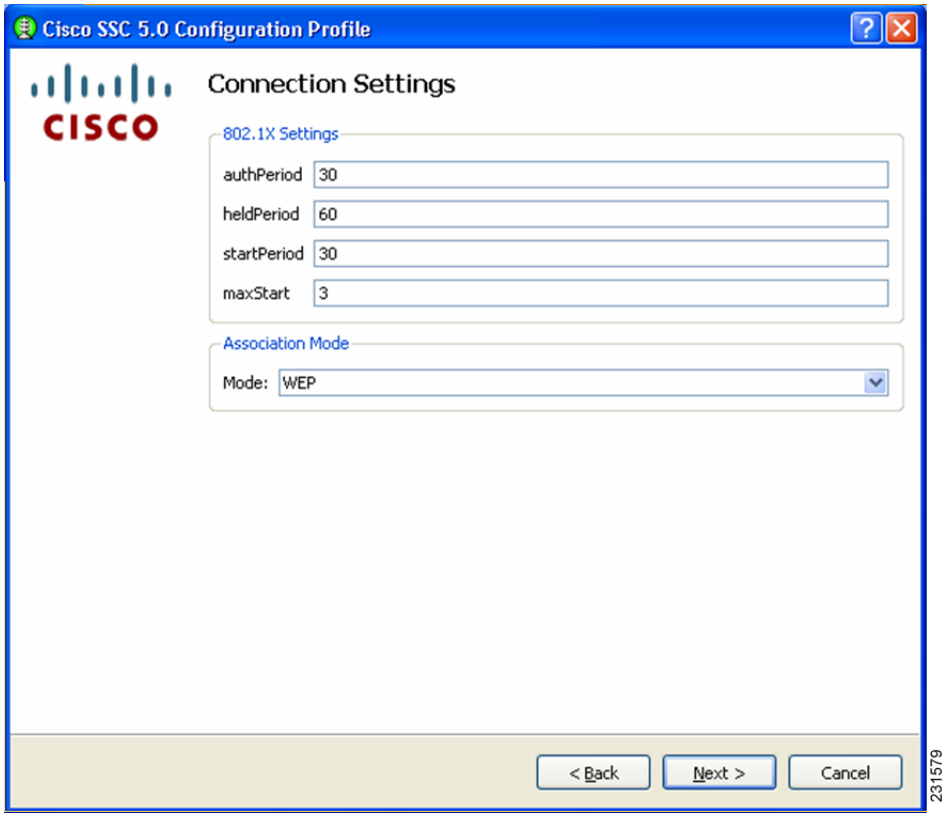
The AssociationTimeout value is the time that the Cisco SSC client waits for association to the ssid before it tries another network.

The Connection Timeout value is the time that the Cisco SSC client waits for a network connection to be established, before it tries another network.

Network connection is considered established if the Cisco SSC client obtains an IP address for that network.

Step 14 Click **Next** and the 802.1x connection settings page displays:

Figure 2-10 802.1X Connection Setting Page



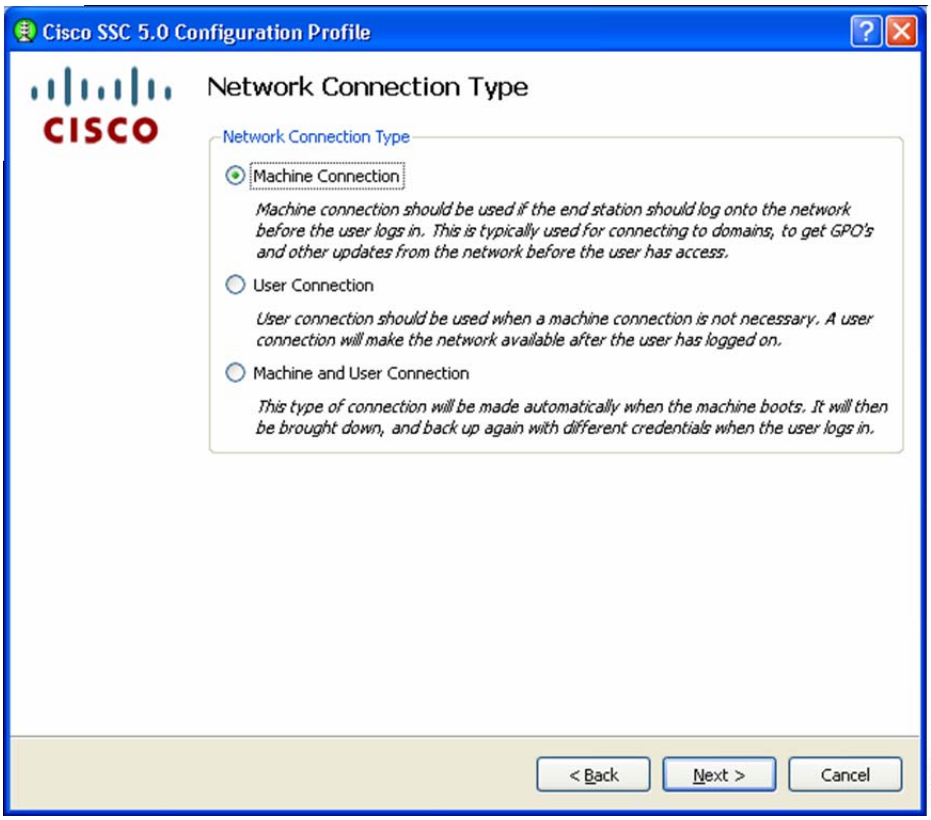
This screen enables you to enter your 802.1x timer values. The default values should work for most networks, however, you have the option to set it to suit your environment.

Step 15 Enter the desired 802.1x timer values. In this example, choose to accept the default values.

Step 16 Choose the association mode for this network, by clicking the drop-down arrow. In this example, choose **WEP**.

Step 17 Click **Next** and the Network Connection Type page (Figure 2-11) displays:

Figure 2-11 Network Connection Type Page



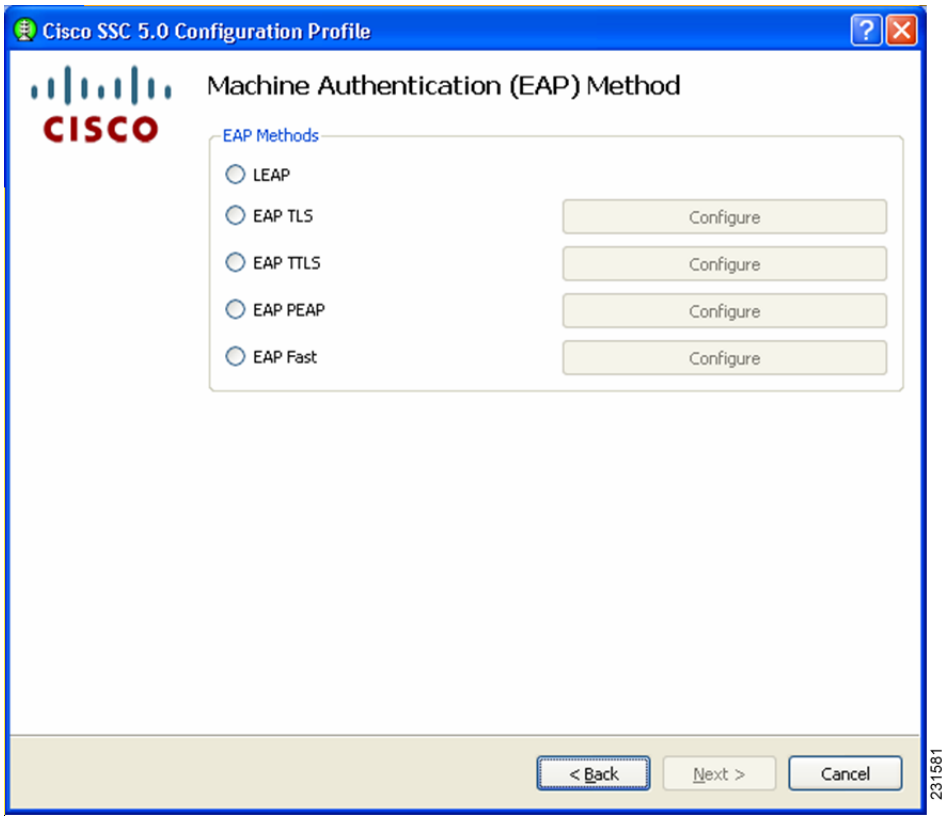
This page enables you to choose the type of network connection. The SSC client defaults to Machine Connection. The User Connections are attempted only during a user session.

A Machine-User ' network contains a machine part and a user part. The ssid is the same for the two parts, but the credential type for machine connection can be different from the credential type for user connection.

Step 18 In this example, choose **Machine and User Connection**.

Step 19 Click **Next** and the Machine Authentication Method page (Figure 2-12) displays.

Figure 2-12 Machine Authentication Method Page



This page enables you to choose the machine authentication method.

Step 20 In this example, to create a Peap-MSChapv2 network, check **EAP PEAP**.

Step 21 To configure PEAP settings, click **Configure** next to EAP PEAP. The EAP Peap Setting page (Figure 2-13) displays.

Figure 2-13 EAP PEAP Setting Page

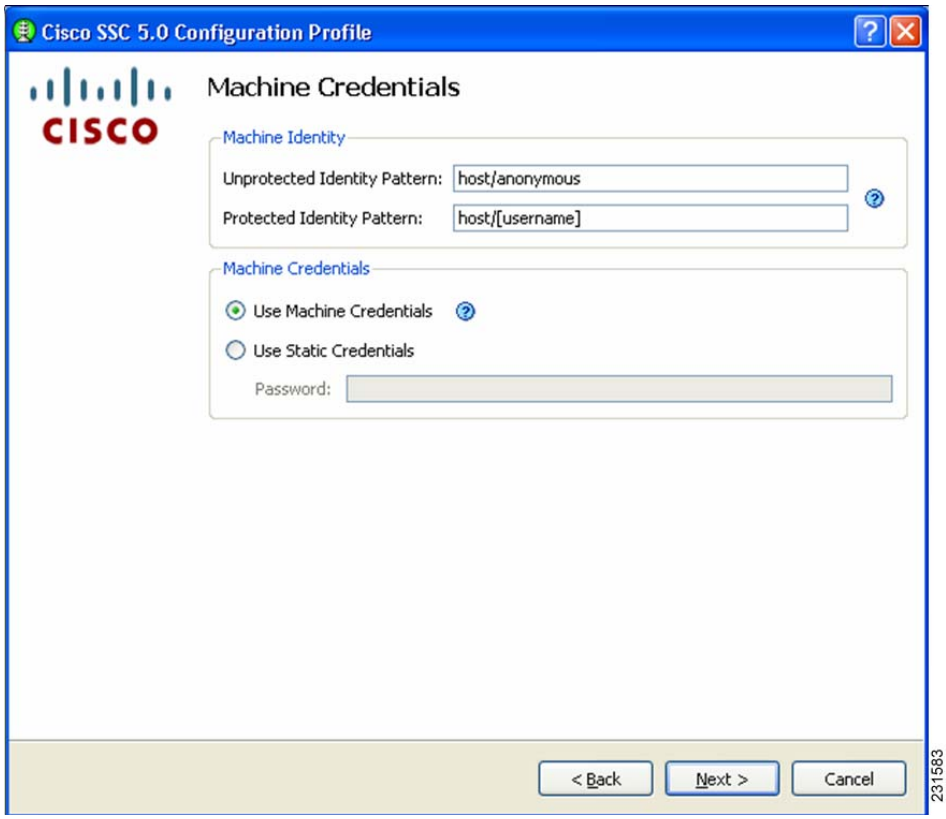
This page enables you to specify these options:

- Validate Server Identity—enables server certificate validation.
- Enable Fast Reconnect—enables session resumption.
- Inner methods based on Credentials Source—enables you to choose authenticate using a password or a certificate.

Step 22 Choose the desired options and click **OK** to return to the Machine Authentication Method (Figure 2-12) page.

Step 23 Click **Next** and the Machine Credentials page (Figure 2-14) displays.

Figure 2-14 Machine Credentials Page



This page enables you to specify the credentials to use to establish this network.

Cisco SSC release 5.0 supports these placeholder patterns when specifying identities:

- [username]
- [domain]

When the [username] and/or [domain] placeholders are used then these conditions apply:

- If a client certificate is used for authentication, then the placeholder's values is obtained from the CN field of the client certificate.
- Otherwise, the credentials are obtained from the operating system and the [username] placeholder represents the assigned machine name.

A typical pattern for machine unprotected identity is *host\anonymous.[domain].*

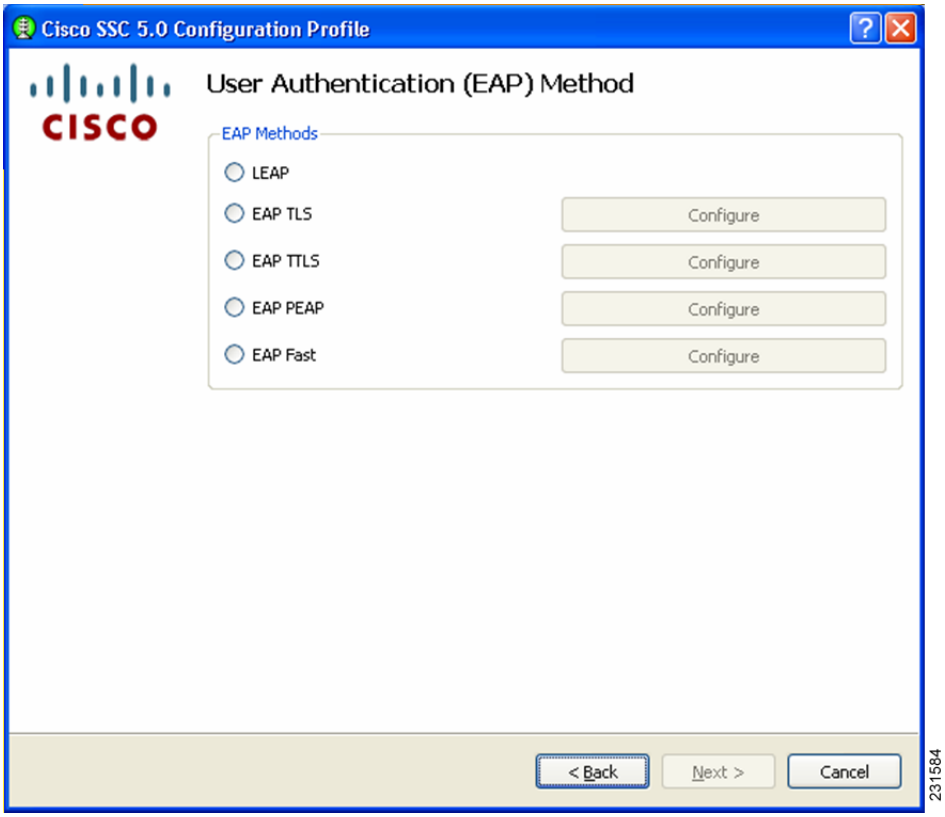
  - If password source is this configured for this profile, then the pattern would be the actual string to send as the username with no placeholders.

A typical pattern for machine protected identity is *host\[username].[domain]*.

- If password source is configured for this profile, then the pattern would be the actual string to send as the username.

Step 24 Enter the desired settings for the machine connection and click **Next**. The User Authentication Method page (Figure 2-15) displays again.

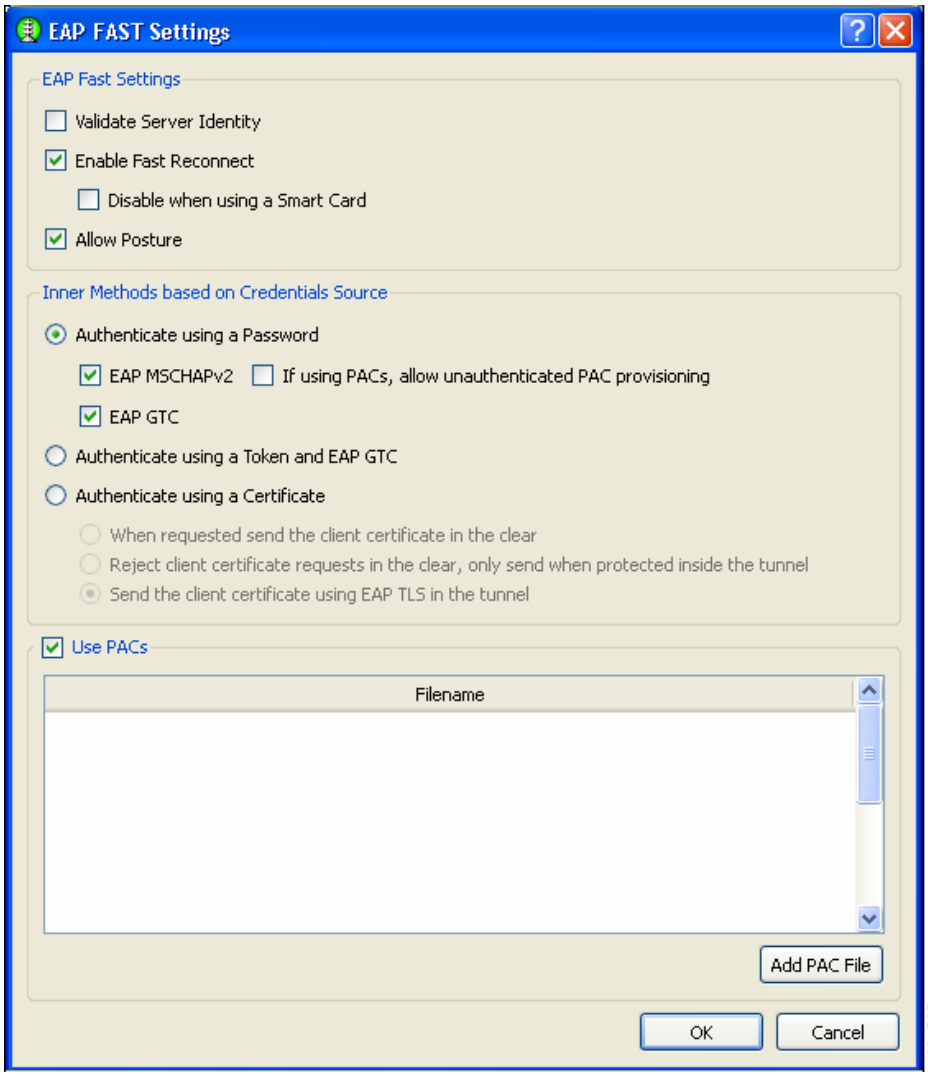
Figure 2-15 User Authentication Method Page



Now you need to specify the credentials for the machine connection.

Step 25 In this example for user authentication, check **EAP Fast** and click **Configure** next EAP-Fast. The EAP Fast Settings page (Figure 2-16) displays.

Figure 2-16 EAP Fast Settings Page

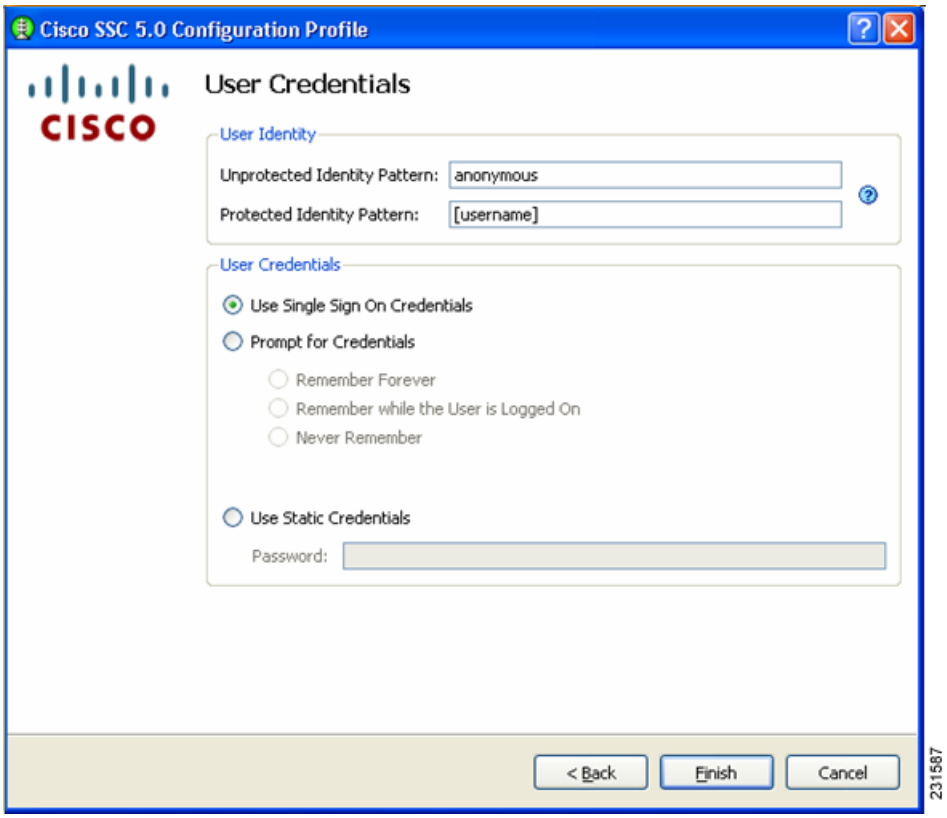


On this page, you have the option to include a manually provisioned PAC by clicking Add PAC File. The contents of the PAC file is added to the distribution package, producing a single deployment file.

Step 26 In this example, check the EAP Fast options shown in Figure 2-16 and click **OK**. The User Authentication Method page (Figure 2-15) displays again.

Step 27 Click **Next** to configure the user credentials. The User Credentials page (Figure 2-17) displays.

Figure 2-17 User Credentials Page



This page enables you to specify the credentials to use to establish this network.

Cisco SSC release 5.0 supports these placeholder patterns when specifying user identities:

- [username]
- [domain]

When the [username] and/or [domain] placeholders are used then these conditions apply:

- If a client certificate is used for authentication, then the placeholder's values is obtained from the CN field of the client certificate.
  - If the credential source is the end-user, then the placeholder's values is obtained from the information the user enters.
  - If the credentials are obtained from the operating system, then the placeholder's value is obtained from the logon information.

  A typical pattern for user unprotected identity is *anonymous@[domain]* for tunneled methods or *[username]@[domain]* for non-tunneled methods.

If the credential source is this profile, then the pattern would be the actual string to send as the username (no placeholders). A typical pattern for user protected identity is *[username]@[domain]*.

If the password source is this profile, then the pattern would be the actual string to send as the username (no placeholders).

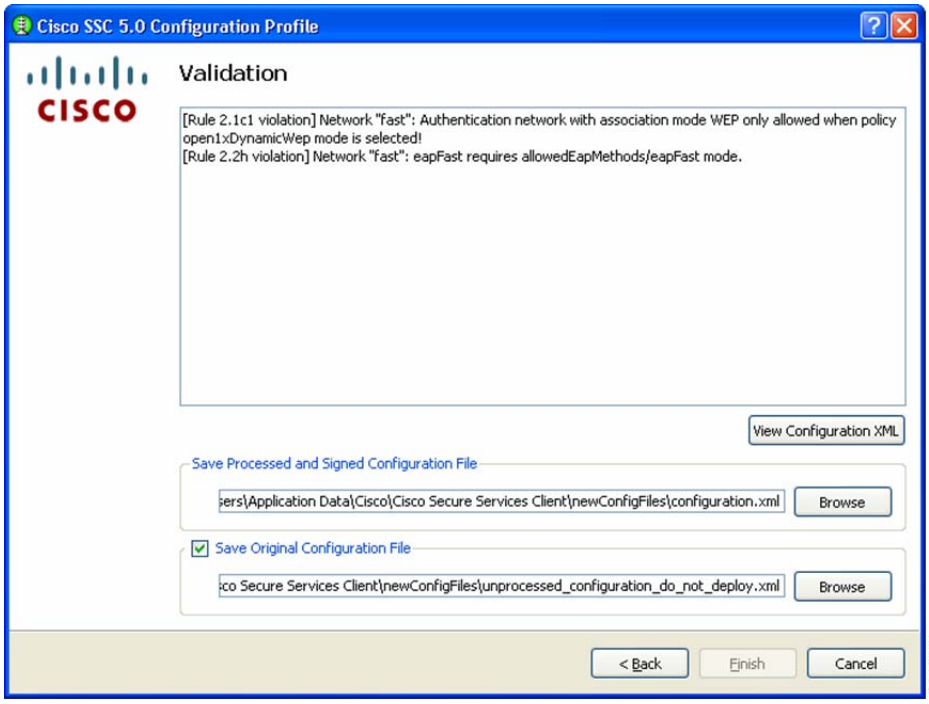
When you have specified the identity pattern, you can then specify the credential source. You can either prompt the user for credentials or use single signon credentials (the SSC client obtains these from the operating system) or specify the actual credentials to be sent in the deployment file.

When you have completed your selections, click on Finish. You have finished creating a network.

Now you can add as many networks as needed and when done, click on *Nex*t.

At this point, the management tool validates the networks you have defined against your policy settings. If there are any policy violations with the networks you have just created, they are displayed. If errors are indicated, you must fix them before you can save the file. For example, this Validation page might (Figure 2-18) display.

Figure 2-18 Validation Page



When there are no validation errors, you can choose to save the deployment file. The management utility saves two formats of the deployment file in any location you choose. The processed file (with encrypted credentials, PACs and CA certificates and signed) is stored by default in this file location:

C:\Documents and Settings\All Users\Application Data\Cisco
Cisco Secure Services Client\newConfigFiles\configuration.xml

The Cisco SSC client looks in this location for any new distribution package. If you have the client installed on your system, this also allows you to automatically test the configuration that you just created and verify it before deploying it.

The unprocessed deployment file is saved in this location:

C:\Documents and Settings\All Users\Application Data\Cisco\Cisco Secure Services Client
newConfigFiles\unprocessed_configuration_do_not_deploy.xml

Caution This file contains credentials in plain text.

If you need to make changes to the deployment package you just created, you can reopen the management utility and click on *Modify Existing Configuration* on the welcome page (Figure 2-1) and choose the unprocessed deployment file that you just saved.

CHAPTER 3

Troubleshooting

This chapter describes the Cisco SSC Release 5.0 log file, the log message formats, the log packager utility, steps to take when you discover a problem with the SSC client, and frequently asked questions. This chapter contains these sections:

Overview

Cisco SSC Release 5.0 creates log files that contain client action sequences to help you troubleshoot client problems. The log file is called *CurrentLog.txt* and is located in *Documents And Settings/All Users/Application Data/Cisco/Cisco Secure Services Client/logs*. You might also see a file called *PreviousLog.txt* in the same folder, which is the previous log file. When the CurrentLog.txt file exceeds 2 Mb or when the client is restarted, the existing log file is renamed to PreviousLog.txt and a new log file is created.

These files contain different levels of log messages with these formats:

- %CSSC-3-ERROR_MSG—an error log message used to indicate an exception that prevents normal processing.
- %CSSC-4-WARNING_MSG—a warning log message used to indicate a client state that is insecure or unexpected but that still allows processing.
- %CSSC-6-INFO_MSG—an informational log message used to indicate a client state that is part of normal processing.
- %CSSC-7-DEBUG_MSG—a debug log message that is useful for the support team.

You can use any commonly used scripting tool to parse the log file to identify any error, warning or information messages you are interested in. Refer to Appendix B, "Cisco Secure Client Services Release 5.0 Log Messages," for a description of the error, warning and information messages provided by SSC Release 5.0.

Log Packager

SSC Release 5.0 comes with a tool called *Log Packager* that collects all relevant system information, including client information, to aid the support team to help you resolve client problems. The information in the report includes client logs, client configurations, license information and adapter information.

This tool is available as a .msi file called *Cisco_Client_Utilities_2KXP-1_0_0_0.msi* on the Cisco SSC product software download page on Cisco.com.

Note You must register or be a registered user of Cisco.com to download product software.

Follow these instructions to obtain the .msi file:

Step 1 Use your Web browser to browse to this URL:

http://tools.cisco.com/support/downloads/pub/Redirect.x?mdfid=280753707

Step 2 Click **Client Adapters and Client Software > Cisco Secure Services Client v5.0** and login or register to Cisco.com.

Step 3 Click **Windows XP** or **Windows 2000** and the Select a Release page displays.

Step 4 Click **5.0.0** under Latest Releases.

Step 5 Click **Cisco_Client_Utilities_2KXP-1_0_0_0.msi** and the Downloads page displays.

Step 6 Click **Download** and agree to the software license agreement.

Step 7 Enter your username and password at the log-on prompt.

Step 8 Follow the prompts to download the software to your PC.

You can use any of your deployment methods to get this package on your enterprise end-systems.

When you experience a problem with Cisco SSC client Release 5.0 that you are not able to resolve, please perform these operations prior to contacting Cisco support:

1. Install the Log Packager tool on the system experiencing problems. Once installed, the tool is available from **Start > All Programs > Cisco**.
2. Start the Log Packager tool and click the Collect Data button.
3. When the log packager has completed collecting data, it produces a new zip file called *CiscoSupportReport.zip* on your desktop. Alternatively, you can also click on the *Locate Report file* button once it becomes active to go directly to the zip file. This file is needed when you contact Cisco support.

For instructions on contacting Cisco support refer to the "Obtaining Documentation, Obtaining Support, and Security Guidelines" section on page -vi.

Frequently Asked Questions

Q. How do I configure an 802.1x network?

A. Recommended: Download the management utility (refer to the "Distribution Package Utilities" section on page 1-4), unzip the file and run the *sscManagementUtility.exe* included in the package.

Q. How come when I press connect it doesn't connect to the network I have selected?

A. CSSC should always try to connect to the network you've selected. If the connection attempt fails, CSSC moves to the next network in the list. This continues until a connection is established. If you want it to connect to a specific network, right click on the network, and choose **Connect exclusively**.

APPENDIX A

Postprocessing Verification Errors

Command Usage Errors

Note Execution of the sscManagementUtility utility will result in either of the following:

- Success—Confirmation message returned. For sign option, 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 - 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 version attributes from base element

  Erroneous XML input text:

  <configuration>

  Error message:

  Loaded version: ..

  Unknown configuration version.

- Missing element closing tag (<collectionBehavior)

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 49, eliminates all of the reported parsing errors listed below.

Erroneous XML input text:

```
(line 48)  <userAuthentication>
(line 49)    <collectionBehavior
(line 50)      <withPassword>
(line 51)        <cachePasswordFromUser>
(line 52)          <forever/>
(line 53)        </cachePasswordFromUser>
(line 54)      </withPassword>
(line 55)    </collectionBehavior>
```

Error message:

Entity: line 50: parser error : error parsing attribute name <withPassword>

Entity: line 50: parser error : attributes construct error <withPassword>

Entity: line 50: parser error : Couldn't find end of Start Tag collectionBehavior line 49 <withPassword>

Entity: line 55: parser error : Opening and ending tag mismatch: userAuthentication line 48 and collectionBehavior </collectionBehavior>^

Entity: line 84: parser error : Opening and ending tag mismatch: authenticationNetwork line 47 and userAuthentication </userAuthentication>

Entity: line 96: parser error : Opening and ending tag mismatch: wifiNetwork line 39 and authenticationNetwork </authenticationNetwork>

Entity: line 97: parser error : Opening and ending tag mismatch: globalNetworks line 30 and wifiNetwork </wifiNetwork>

Entity: line 98: parser error : Opening and ending tag mismatch: networks line 29 and globalNetworks </globalNetworks>

Entity: line 102: parser error : Opening and ending tag mismatch: configuration line 2 and networks </networks>

Entity: line 104: parser error : Extra content at the end of the document <connectionSettings>

Document not loaded.

- Missing attributes from element

  Erroneous XML input text:

  ```
  <unprotectedIdentityPattern>anonymous</unprotectedIdentityPattern>
  ```

  Error message:

  element unprotectedIdentityPattern: Schemas validity error : Element 'unprotectedIdentityPattern': The attribute 'encryptContent' is required but missing.

  Schema validation failed (1868)

- Elements out-of-order as required by schema

  Erroneous XML input text:

  ```
  <wifiNetwork>
    <connectionTimeout>30</connectionTimeout>
    <displayName>My Corporate Wi-Fi Network</displayName>
  ```

Error message:

element connectionTimeout: Schemas validity error : Element 'connectionTimeout': This element is not expected. Expected is (displayName).

Schema validation failed (1871)

- Missing a required element

  Erroneous XML input text:

  ```
  <wifiNetwork>
    <connectionTimeout>30</connectionTimeout>
    <doNotAllowEapOverUdp/>
  ```

  Error message:

  element connectionTimeout: Schemas validity error : Element 'connectionTimeout': This element is not expected. Expected is (displayName).

  Schema validation failed (1871)

- Missing a required element value

  Erroneous XML input text:

  ```
  <wifiNetwork>
    <displayName></displayName>
    <connectionTimeout>30</connectionTimeout>
  ```

  Error message:

  element displayName: Schemas validity error : Element 'displayName': [facet 'minLength'] The value has a length of '0'; this underruns the allowed minimum length of '1'.

  element displayName: Schemas validity error : Element 'displayName': '' is not a valid value of the atomic type 'NonEmptyString'.

  Schema validation failed (1824)

- Element value data type error

  Erroneous XML input text:

  ```
  <wifiNetwork>
    ....
    <associationTimeout>0</associationTimeout>
  ```

  Error message:

  element associationTimeout: Schemas validity error : Element 'associationTimeout ': '0' is not a valid value of the atomic type 'xs:positiveInteger'.

  Schema validation failed (1824)

- Extra white space with an enumerated value

  Erroneous XML input text:

  ```
  <associationMode>
    <wpa>
    <encryption>TKIP </encryption>
    </wpa>
  </associationMode>
  ```

Error message:

element encryption: Schemas validity error : Element 'encryption': [facet 'enumeration'] The value 'TKIP ' is not an element of the set {'AES', 'TKIP'}.

element encryption: Schemas validity error : Element 'encryption': 'TKIP ' is not a valid value of the atomic type 'WpaEncryption'.

Schema validation failed (1824)

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 annotation descriptions in the schema for more information.

- Rule 1.1 Authenticating networks using a tunneled authentication method require the specification of at least one corresponding inner method. Applies to EAP FAST, EAP PEAP and EAP TLS.

  Erroneous XML input text:

  ```
  <wifiNetwork>
    <displayName>Test 1.1.1</displayName>
    ....
          <eapFast>
          ....
                <methods></methods>
  ```

  Error message:

  [Rule 1.1.1 violation] Network Test 1.1.1 EapFast authentication settings should use at least one of the following methods as inner method: eapMschapv2 or eapGtc.

  See the description for elements: *methods* or *eapMethods.*

- Rules 1.2.1 In a user connection context configured for network connectivity before logon, the source for credentials is limited. Client certificates are supported only through smartcards obtained from the OS - client certificates in the Windows certificate store are not supported. Passwords may not be obtained from the user.

  Case 1—Smartcard certificates from OS (Rule 1.2.1a).

  Erroneous XML input text:

  ```
  <displayName>Test 1.2.1a</displayName>
  ...
    <userAuthentication>
      ...
      <certificateSource>
        <certificateFromUser> {Must be from logon.}
  ...
  <connectionSettings>
    <connectionBehaviorAtLogon>
      <attemptConnectionBeforeUserLogon>
  ```

  Error message:

  [Rule 1.2.1a violation] Network Test 1.2.1a Certificate source for user authentication must be certificateFromLogon!

  Case 2—Password from OS or profile (Rule 1.2.1b).

  Erroneous XML input text:

  ```
  <displayName>Test 1.2.1b</displayName>
  ...
    <userAuthentication>
      ...
  ```

```
        <passwordSource>
          <passwordFromUser> {Must be from logon or profile.}
...
<connectionSettings>
  <connectionBehaviorAtLogon>
    <attemptConnectionBeforeUserLogon>
```

Error message:

[Rule 1.2.1b violationPassword source for user authentication must not be passwordFromUser] Network Test 1.2.1b Collection behavior for user authentication must be smartCardOnlyCertificate!

See the description for element: *attemptConnectionBeforeUserLogon.*

- Rules 1.2.2a-c The collection behavior for user credentials is dependent on the type of credential specified.

  Case 1—Password based credentials.

  Erroneous XML input text:

```
<displayName>Test 1.2.2a</displayName>
...
  <authenticationNetwork>
    ...
    <collectionBehavior>
      <withCertificate> {not consistent with source, withPassword required}
    ...
    <authenticationMethod>
    ...
      <passwordSource>
        <passwordFromUser/>
```

  Error message:

  [Rule 1.2.2a violation] Network Test 1.2.2a Collection behavior for user authentication with passwordFromUser must be authenticateWithPassword!

  Case 2—Certificate based credentials.

  Erroneous XML input text:

```
<displayName>Test 1.2.2b</displayName>
...
  <authenticationNetwork>
    ...
    <collectionBehavior>
      <withPassword> {not consistent with source, withCertificate required}
    ...
    <authenticationMethod>
    ...
      <certificateSource>
        <certificateFromUser/>
```

  Error message:

  [Rule 1.2.2b violation] Network Test 1.2.2b Collection behavior for user authentication with certificateFromUser must be authenticateWithCertificate!

Case 3—Token based credentials.

Erroneous XML input text:

```
<displayName>Test 1.2.2c</displayName>
...
   <authenticationNetwork>
      ...
      <collectionBehavior>
         <withCertificate> {not consistent with source, withToken required}
      ...
      <authenticationMethod>
      ...
         <tokenSource>
```

Error message:

[Rule 1.2.2c violation] Network Test 1.2.2c Collection behavior for user authentication with tokens must be authenticateWithToken!

See the description for element: *collectionBehavior.*

- Rule 2.1 Network policy for Wi-Fi associations must include at least one association mode.

  Erroneous XML input text:

  ```
  <networkPolicy>
     <allowedAssociationModes></allowedAssociationModes> {no child element specified}
  ```

  Error message:

  [Rule 2.1 violation] At least one association mode must be specified for networkPolicy/allowedAssociationModes!

  See the description for element: *allowedAssociationModes.*

- Rule 2.1a Network policy for association mode must include *openNoEncryptionfd* to support networks with no authentication or shared secrets.

  Erroneous XML input text:

  ```
  <networkPolicy>
     <allowedAssociationModes>
        <wpaEnterpriseTkip/> {No open networks configured.}
     </allowedAssociationModes>
  ....
  <networks>
     <wiredNetwork>
        <displayName>Test 2.1a</displayName>
        <openNetwork/> {Not allowed}
     </wiredNetwork>
     <wifiNetwork>
        <displayName>Test 2.1a</displayName>
        ...
        <openNetwork> {Not allowed}
        ...
     </wifiNetwork>
  ```

  Error message:

  [Rule 2.1a violation] Network "Test 2.1a": openNetwork only allowed when openNoEncryption mode is selected!

See the description for element: *openNetwork.*

- Rule 2.1b Network policy for association mode must include *openStaticWep* to support any WEP static key network.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No open WEP configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1b</displayName>
    ...
    <sharedKeyNetwork>
      ...
      <wep>
        ...
        <ieee80211Authentication>open</ieee80211Authentication> {Not allowed}
```

  Error message:

  [Rule 2.1b violation] Networks "Test2.1b": wep with ieee80211Authentication/open only allowed when policy openStaticWep mode is selected!

  See the description for element: *ieee80211Authentication.*

- Rule 2.1c Network policy for association mode must include *openStaticWep* to support any WEP static key network.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No shared WEP configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1b</displayName>
    ...
    <sharedKeyNetwork>
      ...
      <wep>
        ...
        <ieee80211Authentication>shared</ieee80211Authentication> {Not allowed}
```

  Error message:

  [Rule 2.1c violation] Networks "Test2.1c": wep with ieee80211Authentication/shared only allowed when policy sharedStaticWep mode is selected!

  See the description for element: *ieee80211Authentication.*

- Rule 2.1c1 Network policy for association mode must include *open1xDynamicWep* to support any dynamic WEP authenticating network.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpaEnterpriseTkip/> {No WEP configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 2.1c1</displayName>
      ...
      <authenticationNetwork>
        ...
        <associationMode>
          <wep> {Not allowed}
  ```

  Error message:

  [Rule 2.1c1 violation] Network "Test 2.1c1": Authentication network with association mode WEP only allowed when policy open1xDynamicWep mode is selected!

  See the description for element: *associationMode.*

- Rule 2.1d Network policy for association mode must include *wpaPersonalTkip* to support a WPA-Personal shared key network using TKIP encryption.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpaEnterpriseTkip/> {No wpaPersonalTkip configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 2.1d</displayName>
      ...
      <sharedKeyNetwork>
        ...
        <wpa>
          ...
          <encryption>TKIP</encryption> {Not allowed}
  ```

  Error message:

  [Rule 2.1d violation] Network "Test 2.1d": wpa with encryption/TKIP only allowed when policy wpaPersonalTkip mode is selected!

  See the description for element: *encryption.*

- Rule 2.1e Network policy for association mode must include *wpaPersonalAes* to support a WPA-Personal shared key network using AES encryption.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No wpaPersonalAes configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1e</displayName>
    ...
    <sharedKeyNetwork>
      ...
      <wpa>
        ...
        <encryption>AES</encryption> {Not allowed}
```

  Error message:

  [Rule 2.1e violation] Network "Test 2.1e": wpa with encryption/AES only allowed when policy wpaPersonalAes mode is selected!

  See the description for element: *wpa/encryption.*

- Rule 2.1f Network policy for association mode must include *wpa2PersonalTkip* to support a WPA2-Personal shared key network using TKIP encryption.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No wpa2PersonalTkip configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1f</displayName>
    ...
    <sharedKeyNetwork>
      ...
      <wpa2>
        ...
        <encryption>TKIP</encryption> {Not allowed}
```

  Error message:

  [Rule 2.1f violation] Networks "Test 2.1f": wpa2 with encryption/TKIP only allowed when policy wpa2PersonalTkip mode is selected!

  See the description for element: *wpa2/encryption.*

- Rule 2.1g Network policy for association mode must include *wpa2PersonalAes* to support a WPA2-Personal shared key network using AES encryption.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpaEnterpriseTkip/> {No wpa2PersonalAes configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 2.1g</displayName>
      ...
      <sharedKeyNetwork>
        ...
        <wpa2>
          ...
          <encryption>AES</encryption> {Not allowed}
  ```

  Error message:

  [Rule 2.1g violation] Networks "Test 2.1g": wpa2 with encryption/AES only allowed when policy wpa2PersonalAes mode is selected!

  See the description for element: *wpa2/encryption.*

- Rule 2.1h Network policy for association mode must include *wpaEnterpriseTkip* to support a WPA-Enterprise network using TKIP encryption.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpaEnterpriseAes/> {No wpaEnterpriseTkip configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 2.1h</displayName>
      ...
      <authenticationNetwork>
        ...
        <associationMode>
          <wpa>
            <encryption>TKIP</encryption> {Not allowed}
  ```

  Error message:

  [Rule 2.1h violation] Network "Test 2.1h": wpa with encryption/TKIP only allowed when policy wpaEnterpriseTkip mode is selected!

  See the description for element: *associationMode.*

- Rule 2.1i Network policy for association mode must include *wpaEnterpriseAes* to support a WPA-Enterprise network using AES encryption.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No wpaEnterpriseAes configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1i</displayName>
    ...
    <authenticationNetwork>
      ...
      <associationMode>
        <wpa>
          <encryption>AES</encryption> {Not allowed}
```

  Error message:

  [Rule 2.1i violation] Network "Test 2.1i": wpa with encryption/AES only allowed when policy wpaEnterpriseAes mode is selected!

  See the description for element: *associationMode.*

- Rule 2.1j Network policy for association mode must include *wpa2EnterpriseTkip* to support a WPA2-Enterprise network using TKIP encryption.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseAes/> {No wpa2EnterpriseTkip configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1j</displayName>
    ...
    <authenticationNetwork>
      ...
      <associationMode>
        <wpa2>
          <encryption>TKIP</encryption> {Not allowed}
```

  Error message:

  [Rule 2.1j violation] Network "Test2.1j": wpa2 with encryption/TKIP only allowed when policy wpa2EnterpriseTkip mode is selected!

  See the description for element: *associationMode.*

- Rule 2.1k Network policy for association mode must include *wpa2EnterpriseAes* to support a WPA2-Enterprise network using AES encryption.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No wpa2EnterpriseAes configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1k</displayName>
    ...
    <authenticationNetwork>
      ...
      <associationMode>
        <wpa2>
          <encryption>AES</encryption> {Not allowed}
```

  Error message:

  [Rule 2.1k violation] Network "Test2.1k": wpa2 with encryption/AES only allowed when policy wpa2EnterpriseAes mode is selected!

  See the description for element: *associationMode.*

- Rule 2.1l Network policy for association mode must include *cckmEnterpriseTkip* to support a WPA/WPA2-Enterprise network with CCKM key management and TKIP encryption.

  Erroneous XML input text:

```
<networkPolicy>
  <allowedAssociationModes>
    <wpaEnterpriseTkip/> {No cckmEnterpriseTkip configured.}
  </allowedAssociationModes>
....
<networks>
  <wifiNetwork>
    <displayName>Test 2.1l</displayName>
    ...
    <authenticationNetwork>
      ...
      <associationMode>
        <cckm>
          <encryption>TKIP</encryption> {Not allowed}
```

  Error message:

  [Rule 2.1l violation] Network "Test2.1l": cckm with encryption/TKIP only allowed when policy cckmEnterpriseTkip mode is selected!

  See the description for element: *associationMode.*

- Rule 2.1m Network policy for association mode must include *cckmEnterpriseAes* to support a WPA/WPA2-Enterprise network with CCKM key management and AES encryption.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedAssociationModes>
      <wpaEnterpriseAes/> {No cckmEnterpriseAes configured.}
    </allowedAssociationModes>
  ....
  <networks>
    <wifiNetwork>
      <displayName>Test 2.1m</displayName>
      ...
      <authenticationNetwork>
        ...
        <associationMode>
          <cckm>
            <encryption>AES</encryption> {Not allowed}
  ```

  Error message:

  [Rule 2.1m violation] Network "Test2.1m": wpa2 with encryption/AES only allowed when policy wpa2EnterpriseAes mode is selected!

  See the description for element: *associationMode.*

- Rule 2.2 Network policy for EAP methods must include at least one method.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedEapMethods></allowedEapMethods> {no child element specified}
  ```

  Error message:

  [Rule 2.2 violation] At least one eapMethod must be specified for networkPolicy/allowedEapMethods!

  See the description for element: *allowedEapMethods.*

- Rule 2.2a 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 2.2a</displayName>
      ...
      <authenticationNetwork>
        ...
        <authenticationMethod>
          <eapMd5> {Not allowed}
  ```

Error message:

[Rule 2.2a violation] Network "Test 2.2a" : eapMethod/eapMd5 requires allowedEapMethods/eapMd5.

See the description for element: *authenticationMethod* or *machineAuthentication* or *machine.*

- Rule 2.2b 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 2.2b</displayName>
      ...
      <authenticationNetwork>
         ...
         <authenticationMethod>
            <eapMschapv2> {Not allowed}
```

  Error message:

  [Rule 2.2b violation] Network "Test 2.2b" : eapMschapv2 requires allowedEapMethods/eapMschapv2 mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine.*

- Rule 2.2c 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 2.2c</displayName>
      ...
      <authenticationNetwork>
         ...
         <authenticationMethod>
            <eapGtc> {Not allowed}
```

  Error message:

  [Rule 2.2c violation] Network "Test 2.2c" : eapMethod/eapGtc requires allowedEapMethods/eapGtc mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine.*

- Rule 2.2d 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 2.2d</displayName>
      ...
      <authenticationNetwork>
        ...
        <authenticationMethod>
          <leap> {Not allowed}
  ```

  Error message:

  [Rule 2.2d violation] Network "Test 2.2d" : eapMethod/leap requires allowedEapMethods/leap mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine*.

- Rule 2.2e Network policy for EAP methods must include *eapTls* to support authenticating wired or wireless networks configured for EAP-TLS in the outer tunnel.

  Erroneous XML input text:

  ```
  <networkPolicy>
    <allowedEapMethods>
      <eapFast/> {No EAP-TLS configured.}
    </allowedEapMethods>
  ....
  <networks>
    <wiredNetwork>
      <displayName>Test 2.2e</displayName>
      ...
      <authenticationNetwork>
        ...
        <authenticationMethod>
          <eapTls> {Not allowed}
  ```

  Error message:

  [Rule 2.2e violation] Network "Test 2.2e" : eapMethod/eapTls requires allowedEapMethods/eapTls mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine*.

- Rule 2.2f 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 2.2f</displayName>
      ...
      <authenticationNetwork>
         ...
         <authenticationMethod>
            <eapTtls> {Not allowed}
```

  Error message:

  [Rule 2.2f violation] Network "Test 2.2f" : eapMethod/eapTtls requires allowedEapMethods/eapTtls mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine.*

- Rule 2.2g 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 2.2g</displayName>
      ...
      <authenticationNetwork>
         ...
         <authenticationMethod>
            <eapPeap> {Not allowed}
```

  Error message:

  [Rule 2.2g violation] Network "Test 2.2g" : eapMethod/eapPeap requires allowedEapMethods/eapPeap mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine.*

- Rule 2.2h 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 2.2h</displayName>
      ...
      <authenticationNetwork>
        ...
        <authenticationMethod>
          <eapFast> {Not allowed}
  ```

  Error message:

  [Rule 2.2h violation] Network "Test 2.2h" : eapMethod/eapFast requires allowedEapMethods/eapFast mode.

  See the description for element: *authenticationMethod* or *machineAuthentication* or *machine.*

- Rule 3a SSC must be configured for at least one media type.

  Erroneous XML input text:

  ```
  <userControlPolicy>
    ...
    <allowedMedia></allowedMedia> {Missing a child element.}
  ```

  Error message:

  [Rule 3a violation] At least one media type must be specified for userControlPolicy/allowedMedia!

  See the description for element: *allowedMedia.*

- Rule 3b 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 3b</displayName>
  ...
  <userControlPolicy>
    ...
    <allowedMedia>
      <wifi/> {Wired not configured.}
    </allowedMedia>
  ```

  Error message:

  [Rule 3b violation] Network "Test 3b": wiredNetwork may not be present unless userControlPolicy/allowedMedia/wired is present.

  See the description for element: *wiredNetwork.*

- Rule 3c 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 3c</displayName>
  ...
  <userControlPolicy>
    ...
    <allowedMedia>
      <wired/> {Wireless not configured.}
    </allowedMedia>
  ```

  Error message:

  [Rule 3c violation] Network "Test 3c": wifiNetwork 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)

APPENDIX **B**

Cisco Secure Client Services Release 5.0 Log Messages

This appendix lists the log messages produced by Cisco Secure Cilent Services Release 5.0.

- **Starting Cisco_SSCservice.exe:** *version number*—indicates the SSC service is starting.
- **Cisco Trust Agent successfully loaded**
- **Failed to load Cisco Trust Agent**
- **Password sent**
- **Certificate sent**
- **Manual user** *logon type* **logon processing initiated by user** *user id.*
- **Normal Shutdown** *version number*—indicates a normal shutdown.
- **Fatal Shutdown** *version number*—indicates a fatal shutdown.
- **Machine startup**—indicates the client is begining its boot time processing.
- **Account logon**—indicates the client detected a user logon.
- **SSO credentials (Microsoft)**—indicates when the client collects credentials from the Microsoft GINA (whether they are used or not during a network authentication)
- **Account logoff**—indicates the client detects a user logoff
- **Adapter detected** *Adapter Id* —indicates a new adapter is detected in the system. The *Adapter Id* refers to the adapter's globally unique identifier (GUID).
- **Adapter removed** *Adapter Id*—indicates a previously reported adapter is lost (or removed)
- **Adapter controlled** *Adapter Id*—indicates control is taken of a particular adapter (the SSC intermediate driver begins to respond to network frames and attempt to set features of the adapter).
- *Adapter Id* **Adapter control failed** *error code*—indicates when the SSC client attempts to take control of an adapter but fails. The *error code* is an internal error code.
- **{WPA | WPA2} unsupported.** *Adapter Id*—indicates when control is taken of an adapter and if the adapter does or does not support WPA or WPA2.
- **Wireless Zero Config deactivated** *Adapter Id*— indicates when control was taken of an adapter that Wireless Zero Config was detected and automatically deactivated for that adapter.
- **Adapter control released** *Adapter Id*—indicates control was released for a particular adapter.

- **Connection Association Started** (*WiFi Association /Encryption Mode*)—when a connection is requested on a WiFi adapter an association must occur. This log message indicates the SSC client is attempting to associate to an ssid. *WiFi Association/Encryption mode* could be one of these values:
  - Open
  - Shared 40 bit key
  - Shared 128 bit key
  - Static WEP 40 bit key
  - Static WEP 128 bit key
  - Dynamic WEP 40 bit key
  - Dynamic WEP 128 bit key
  - WPA-Personal TKIP encryption
  - WPA-Personal AES encryption
  - WPA-Enterprise TKIP encryption
  - WPA-Enterprise AES encryption
  - WPA2-Personal TKIP encryption
  - WPA2-Personal AES encryption
  - WPA2-Enterprise TKIP encryption
  - WPA2-Enterprise AES encryption
- **Starting wired connection, skipping association**
- *Adapter Id* **Connection Association Success (link up)**—indicates an association has completed
- successfully.
- **Connection Association Failed. (Failure:** *error number*)—indicates an association has not completed successfully. *error number* is an internal error code.
- *Adapter Id* **Connection Authentication Started**—indicates an authentication attempt was started.
- *Adapter Id* **Identity requested** – when an identity request comes in from the AP.
- *Adapter Id* **Identity sent** - whenever an identity is sent.
- *Adapter Id* **EAP suggested by server:** *Authentication Method name*—indicates an EAP authentication method was suggested by the server. *Authentication Method name* is one of these values:
  - EAP-PEAP
  - EAP-TTLS
  - EAP-TLS
  - EAP-LEAP
  - EAP-MD5
  - EAP-GTC
  - EAP-FAST
  - EAP-MSCHAPv2
  - MSCHAPv2
  - MSCHAP

  - CHAP
  - PAP
- *Adapter Id* **EAP requested by client: (***Authentication Method name*, …, *Authentication Method name*)—indicates an EAP authentication method was requested by the client. *Authentication Method name* is one of these values:
  - EAP-PEAP
  - EAP-TTLS
  - EAP-TLS
  - EAP-LEAP
  - EAP-MD5
  - EAP-GTC
  - EAP-FAST
  - EAP-MSCHAPv2
  - MSCHAPv2
  - MSCHAP
  - CHAP
  - PAP
- *Adapter Id* **Port State** *Port State* and **Status** *Port status*—indicates the state and status of the adapter's port.

  Port State is one of values:
  - AC_PORT_STATE_STOPPED – indicates port is stopped
  - AC_PORT_STATE_CONNECTING – when it is waiting to start authentication
  - AC_PORT_STATE_AUTHENTICATING – is actively performing the initial 802.1x authentication
  - AC_PORT_STATE_AUTHENTICATED – successfully completed authentication
  - AC_PORT_STATE_REAUTHENTICATING – is actively performing 802.1x reauthentication
  - AC_PORT_STATE_UNAUTHENTICATED – when port wants to authenticate, but can't because of other conditions such as link is down or incorrect credentials
  - AC_PORT_STATE_AUTH_NOT_REQUIRED – when 802.1x authentication is not required. This state only exists for wired adapters or wireless adapters in WEP mode.

  Port status depends on the Port State value. This indicates a sub-state of the port state.
- *Adapter Id* **FAST: unauthenticated provisioning supported**—indicates FAST unauthenticated provisioning is supported by the adapter.
- *Adapter Id* **FAST: phase 1 tunnel for unauthenticated provisioning**
- *Adapter Id* **Allowing session resumption**—indicates when the SSC client begins a TLS-based authentication (PEAP, TTLS, FAST or TLS) and attempts session resumption with a previous session id.
- *Adapter Id* **Authentication Success**—indicates an authentication completed successfully.
- *Adapter Id* **Authentication Failed**—indicates an authentication completed unsuccessfully.
- *Adapter Id* **IP Address Received:** *IP Address*—indicates a connection received an IP Address.

- *Adapter Id* **DHCP: Sending DHCP request.**
- *Adapter Id* **DHCP: Request failed**.
- *Adapter Id* **Wireless Zero Config reactivated for adapter**
- *Access Id* **WiFi access device has invalid channel number:** *SSID*, *channel*
- *Adapter Id* **Couldn't find pre-shared key in profile**
- *Adapter Id***: EAP-TTLS method requested by client:** *method name*
- **Starting wifi connection, trying ssid** *ssid name*
- **Licensing: No license found**.
- **Licensing: License read:** *License string.*
- *License string***: (do not translate) is the license string read from the license file**.
- **Licensing: License invalid (trial period expired** *License string*, *trial period*).
- **Licensing: License invalid (termination date reached:** *License string*, t*ermination date*). *termination date* is the date in format yyyy-mm-dd that the license expired.
- **Licensing: License invalid because product id does not match:** *License string*, *licensed product id*
- **Licensing: License invalid (OEM id does not match:** *License string*, *licensed OEM id*)
- **Licensing: License invalid (maintenance date reached:** *License string*, *maintenance date*). The *maintenance date* value is *the d*ate in format yyyy-mm-dd that the license's maintenance expired.
- **Licensing: License invalid (unknown problem:** *License string*)
- **Licensing: License is valid and accepted:** *License string.*
- **Licensing: Ignoring trial license. Tampering detected:** *License string*—whenever the license history file fails decryption this message is output with each new trial license that is encountered.
- **Licensing: License invalid, can not decode license:** *License string*
- **The configuration is invalid and will be ignored. Error:** *error string*
- **Trusted Server list empty, server can not be validated**
- **Validating the server:** *Authentication Server Id*
- **Server certificate validated:** *Authentication Server Id*
- *Authentication Session Id* **Server certificate invalid (unknown CA)**
- **Server certificate invalid (name mismatch:** *CN/DC/Alt name from server cert*)
- **Invalid key type in distribution package**
- *Outer method***: invalid/unsupported inner authentication method:** *inner method*
- **Invalid outer EAP method:** *method name*
- *Outer method***: No inner authentication methods configured**
- **Disallowed element in configuration: wireless adapters unlicensed**
- **Disallowed element in configuration: wired adapters unlicensed**
- **Disallowed element in configuration: EAP method:** *method name*
- **Disallowed element in configuration: Association mode:** *association mode*
- **Symbolic name:** *GUID of adapter*, **MacAddr:** (M*AC addrress of adapter*)**, Mtu:** (*MTU size*)**, Media:** (*percentage*)**, Encryption:** (*encryption modes*)**, Auth:** (*auth modes*)

- **Server certificate chain invalid**
- **Server certificate chain is not trusted**
- **Invalid wep key length:** *key length*, **should be %d or %d**
- **The wildcard (***pattern string***) in the pattern is unknown and will be removed**
- **Internal error** *error number*, **contact software manufacturer**—indicates you should contact Cisco support.