CHAPTER 2

Schema Elements

Introduction

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

This chapter contains the following sections:

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

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

configuration/networks/wifiNetwork/

configuration/networks/wiredNetwork/

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

configuration/networks/wifiNetwork/authenticationNetwork/machineAuthentication/

configuration/networks/wifiNetwork/authenticationNetwork/userAuthentication/

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

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

configuration/networks/wiredNetwork/authenticationNetwork/machineAuthentication/

configuration/networks/wiredNetwork/authenticationNetwork/userAuthentication/

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

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

Note Throughout this chapter, where an element has a relational restriction with another element, the requirement is captured in its business rule statement. The concept of a business rule is described in Chapter 1, "Enterprise Deployment", "Schema Validation:".

Configuring the Distribution Package

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

configuration

Schema path:

configuration

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

This element has the following required attributes:

- major_version—Required with value = 4.
- minor_version—Required with value = 1.
- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"—Copy as defined here.
- xsi:noNamespaceSchemaLocation="C:\yourPath\distributionPackage.xsd"—Contains the absolute or relative path to the schema used to instantiate a particular .xml distribution package file; in this case it must point to distributionPackage.xsd.

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

  xsi:noNamespaceSchemaLocation="distributionPackage.xsd"

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

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

Step 1 Perform the tasks defined in "Configuring Your License".

Step 2 Perform the tasks defined in "Configuring Your Policy".

Step 3 Perform the tasks defined in "Configuring Your Connection Settings".

Step 4 Perform the tasks defined in "Configuring Networks".

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

Example 2-1 Base Element

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

Configuring Your License

Configure the following element:

license

Schema path:

configuration/license

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

The following items are affected by the license:

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

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

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

Example 2-2 license

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

Configuring Your Policy

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

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

User Control Policy

Configure the following element:

userControlPolicy

Schema path:

configuration/userControlPolicy

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

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

Step 1 Configure the following policy element for user interface:

clientUIType

Schema path:

configuration/userControlPolicy/clientUIType

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

The element has the following values:

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

Step 2 Configure the following policy element for licensing methods:

allowLicensing

Schema path:

configuration/userControlPolicy/allowLicensing

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

The element has the following values:

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

Step 3 Configure the following policy element for media support:

allowedMedia

Schema path:

configuration/userControlPolicy/allowedMedia

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

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

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

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

wifi

Schema path:

configuration/userControlPolicy/allowedMedia/wifi

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

wired

Schema path:

configuration/userControlPolicy/allowedMedia/wired

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

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

Example 2-3 userControlPolicy

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

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

Network Policy

Configure the following element:

networkPolicy

Schema path:

configuration/networkPolicy

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

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

Step 1 Configure the following policy element for association modes:

allowedAssociationModes

Schema path:

configuration/networkPolicy/allowedAssociationModes

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

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

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

Specify one or more of the following wireless association modes:

The order of the child elements is not restricted.

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

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

open

wpa-Personal

wpa-Enterprise

wpa2-Personal

wpa2-Enterprise

wep

Schema paths:

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

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

Step 2 Configure the following policy element for authentication methods:

allowedEapMethods

Schema path:

configuration/networkPolicy/allowedEapMethods

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

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

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

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

Specify one or more of the following authentication methods:

The order of the child elements is not restricted.

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

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

eapMd5

eapMschapv2

eapGtc

eapFast

eapPeap

eapTtls

eapTls

leap

Schema paths:

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

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

Step 3 Configure the following policy element for trusted servers:

serverValidationPolicy

Schema path:

configuration/networkPolicy/serverValidationPolicy

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

Specify one of the following policies:

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

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

allowUserValidationControl

Schema path:

configuration/networkPolicy/serverValidationPolicy/allowUserValidationControl

The existence of the *allowUserValidationControl* element allows for individualized configuring of server validation. The configuration of whether or not server validation is performed is made on a per EAP method basis within each network.

alwaysValidate

Schema path:

configuration/networkPolicy/serverValidationPolicy/alwaysValidate

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

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

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

allowUserTrustedServers

Schema path:

configuration/networkPolicy/serverValidationPolicy/alwaysValidate/allowUserTrustedServers

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

The element has the following values:

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

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

allowUserSimultaneousConnectionsControl

Schema path:

configuration/networkPolicy/allowUserSimultaneousConnectionsControl

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

The element has the following values:

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

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

Step 5 Configure the following policy element for storing credentials:

allowedCredentialStorage

Schema path:

configuration/networkPolicy/allowedCredentialStorage

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

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

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

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

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

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

forever

Schema path:

configuration/networkPolicy/allowedCredentialStorage/forever

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

logonSession

Schema path:

configuration/networkPolicy/allowedCredentialStorage/logonSession

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

duration

Schema path:

configuration/networkPolicy/allowedCredentialStorage/duration

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

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

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

allowUserWpaHandshakeValidationControl

Schema path:

configuration/networkPolicy/allowUserWpaHandshakeValidationControl

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

The element has the following values:

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

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

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

allowPublicProfileCreation

Schema path:

configuration/networkPolicy/allowPublicProfileCreation

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

The element has the following values:

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

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

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

Example 2-4 networkPolicy

```
<networkPolicy>
    <allowedAssociationModes>
        <!--open network-->
        <open/>
        <!--shared key network-->
        <staticWep/>
        <shared/>
        <wpa-Personal/>
        <wpa2-Personal/>
        <!--authenticating network-->
        <dynamicWep/>
        <wpa-Enterprise/>
```

```
            <wpa2-Enterprise/>
        </allowedAssociationModes>
        <allowedEapMethods>
            <!--wired only-->
            <eapMd5/>
            <eapMschapv2/>
            <eapGtc/>
            <!--wired or wireless-->
            <eapFast/>
            <eapPeap/>
            <eapTls/>
            <eapTtls/>
            <leap/>
        </allowedEapMethods>
        <serverValidationPolicy>
            <alwaysValidate>
                <allowUserTrustedServers>true</allowUserTrustedServers>
            </alwaysValidate>
        </serverValidationPolicy>
        <allowUserSimultaneousConnectionsControl>false</allowUserSimultaneousConnectionsContro
l>
        <allowedCredentialStorage>
            <forever/>
            <logonSession/>
            <duration>5</duration>
        </allowedCredentialStorage>
        <allowUserWpaHandshakeValidationControl>false</allowUserWpaHandshakeValidationControl>
        <allowPublicProfileCreation>false</allowPublicProfileCreation>
</networkPolicy>
```

Configuring Your Connection Settings

Configure the following element:

connectionSettings

Schema path:

configuration/connectionSettings

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

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

Step 1 Configure the following connection setting element:

simultaneousConnections

Schema path:

configuration/networkPolicy/simultaneousConnections

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

The element has the following values:

- singleHomed—restricted to creating only a single connection at a time (prevents multi-homed configurations).

- multiHomed—allows multiple simultaneous connections (allows multi-homed network connections). For the selected network, SSC will attempt to make a connection for all equipped and managed wired and wireless network adapters.

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

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

Step 2 Configure the following connection setting element:

validateWpaHandshake

Schema path:

configuration/networkPolicy/validateWpaHandshake

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

The element has the following values:

- true—enable WPA/WPA2 handshake validation. (recommended)
  Use this setting when your end-stations all have wireless adapters with fully compliant WPA/WPA2 drivers, as required by the standards.
- false—disable WPA/WPA2 handshake validation.
  Use this setting only for special cases in which your wireless adapter's driver is known to have this deficiency.

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

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

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

Example 2-5 connectionSettings

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

Configuring Networks

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

Configure the following element:

networks

Schema path:

configuration/networks

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

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

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

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

Next item: "Choosing a Network Media Type".

Choosing a Network Media Type

Specify one of the following network media types:

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

wiredNetwork

Schema path:

configuration/networks/wiredNetwork

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

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

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

Next item: "Configuring a Wired Network".

wifiNetwork

Schema path:

configuration/networks/wifiNetwork

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

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

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

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

Example 2-6 networks

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

Configuring a Wi-Fi Network

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

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

Configuring a Wired Network

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

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

Wi-Fi Network Base Elements

Configure the following elements:

displayName

Schema path:

configuration/networks/wifiNetwork/displayName

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

ssid

Schema path:

configuration/networks/wifiNetwork/ssid

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

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

Restriction: SSIDs are limited to 32 ASCII characters.

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

associationRetries

Schema path:

configuration/networks/wifiNetwork/associationRetries

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

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

Restriction: the number of retries is limited to 99.

Default: the recommended value is 3.

beaconing

Schema path:

configuration/networks/wifiNetwork/beaconing

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

The element has the following values:

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

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

Example 2-7 partial wifiNetwork

```
<wifiNetwork>
    <displayName>My Corporate Wi-Fi Network</displayName>
    <ssid>MyCorpNet</ssid>
    <associationRetries>3</associationRetries>
    <beaconing>true</beaconing>
```

```
    <!--{your choice of network security class goes here}-->
</wifiNetwork>
```

Choosing the Wi-Fi Network's Security Class

Specify one of the following security classes for the network:

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

openNetworkUserConnection

Schema path:

configuration/networks/wifiNetwork/openNetworkUserConnection

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

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

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

sharedKeyNetwork

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork

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

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

authenticationNetwork

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork

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

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

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

Example 2-8 wifiNetwork

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

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

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

Configuring an Open Wi-Fi Network

Configure the following element:

autoConnect

Schema path:

configuration/networks/wifiNetwork/openNetworkUserConnection/autoConnect

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

The element has the following values:

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

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

Example 2-9 openNetworkUserConnection

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

Configuring a Shared-key Wi-Fi Network

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

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

machineConnection

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection

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

Next item: "Configuring a Shared-key, Machine Network"

userConnection

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection

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

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

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

Example 2-10 sharedKeyNetwork

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

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

Configuring a Shared-key, Machine Network

Configure the following element:

keySettings

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/machineConnection/keySettings

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

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

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

Example 2-11 machineConnection

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

Configuring a Shared-key, User Network

Configure the following elements:

autoConnect

Schema path:

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

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

The element has the following values:

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

keySettings

Schema path:

configuration/networks/wifiNetwork/sharedKeyNetwork/userConnection/keySettings

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

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

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

Example 2-12 userConnection

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

```
    <autoConnect>true</autoConnect>
</userConnection>
```

Choosing the Shared-key Type

Specify one of the following key types for the network:

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

wep

Schema paths:

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

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

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

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

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

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

wpa

Schema paths:

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

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

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

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

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

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

wpa2

Schema paths:

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

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

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

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

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

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

Configuring a WEP Shared-key

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

Step 1 Configure the following elements:

ieee80211Authentication

Schema paths:

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

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

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

Next item: "Choosing the WEP Association".

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

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

Example 2-13 WEP keySettings

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

Choosing the WEP Association

Specify one of the following WEP association modes:

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

open

Schema paths:

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

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

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

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

shared

Schema path:

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

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

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

Example 2-14

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

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

Choosing the WEP Key Format

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

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

wepAscii40

Schema paths:

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

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

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

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

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

wepAscii128

Schema paths:

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

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

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

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

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

wepHex40

Schema paths:

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

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

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

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

Restriction: the value must be ten Hex characters long.

wepHex128

Schema paths:

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

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

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

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

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

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

Example 2-15 wep choices

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

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

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

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

Configuring a WPA/WPA2 Shared-key

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

Step 1 Configure the following element:

encryption

Schema paths:

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

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

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

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

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

The element has the following values:

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

Step 2 Configure the following element:

key

Schema paths:

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

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

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

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

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

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

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

Example 2-16 WPA keySettings

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

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

Choosing the WPA/WPA2 Key Format

Specify one of the following key formats for the network:

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

ascii

Schema paths:

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

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

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

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

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

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

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

hex

Schema paths:

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

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

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

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

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

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

Restriction: the value must be64 Hex characters in length.

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

Example 2-17 key

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

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

Configuring an Authenticating Wi-Fi Network

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

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

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

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

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

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

Example 2-18 partial authenticationNetwork

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

Configuring the Authentication Association Mode

Configure the following element:

associationMode

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork/associationMode

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

Next item: "Choosing the Association Mode".

Choosing the Association Mode

Specify one of the following association modes for the network:

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

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

dynamicWep

Schema path:

configuration/networks/wifiNetwork/authenticationNetwork/associationMode/dynamicWep

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

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

wpa-Enterprise

Schema path:

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

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

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

The element has the following values:

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

wpa2-Enterprise

Schema path:

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

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

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

The element has the following values:

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

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

Example 2-19 associationMode

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

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

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

Configuring the Authenticating Network Base Elements

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

Step 1 Configure the following authentication retry elements:

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

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

interactiveAuthenticationRetries

Schema paths:

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

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

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

Restriction: the number of retries is limited to 99.

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

nonInteractiveAuthenticationRetries

Schema paths:

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

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

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

Restriction: the number of retries is limited to 99.

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

Step 2 Configure the following server validation element:

serverValidation

Schema paths:

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

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

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

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

Next item: "Configuring Server Validation"

Configuring Server Validation

Step 1 Configure your trusted server list as follows:

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

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

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

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

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

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

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

trustAnyRootCaFromOs

Schema paths:

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

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

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

trustedRootCaCerts

Schema paths:

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

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

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

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

Next item: "Adding CA Certificates".

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

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

Example 2-20 serverValidation

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

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

Configuring Certificate Trusted Server Rules

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

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

validationRules

Schema paths:

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

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

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

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

matchSubjectAlternativeName

Schema paths:

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

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

- Subject Alternative Name: DNSName.

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

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

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

Values for required attributes:

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

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

matchSubjectName

Schema paths:

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

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

- Subject: CN (Common Name).

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

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

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

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

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

Values for required attributes:

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

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

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

Example 2-21 validationRules

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

Configuring PAC Trusted Server Rules

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

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

trustedServerIds

Schema paths:

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

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

Configure the following child element:

trustedServerId

Schema path:

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

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

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

Values for required attributes:

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

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

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

reference

Schema paths:

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

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

Configure the child elements.

aidReference

Schema paths:

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

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

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

secretKey

Schema paths:

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

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

aid

Schema paths:

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

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

Restriction: HEX formatted.

aidInfo

Schema paths:

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

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

Restriction: ASCII formatted.

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

Example 2-22 trustedServerId

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

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

Adding CA Certificates

Configure the following elements:

certificate

Schema paths:

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

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

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

Specify and configure the following element:

caReference

Schema paths:

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

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

The following two certificate file formats are supported: .der and .pem. The *content* element's *format* attribute is added and configured appropriately with value pem or der.

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

Example 2-23 certificate

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

Choosing the Authentication Network's Connection Context

Specify one of the following connection contexts for the network:

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

machineAuthentication

Schema paths:

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

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

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

userAuthentication

Schema paths:

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

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

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

machineUserAuthentication

Schema paths:

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

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

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

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

Example 2-24 authenticationNetwork

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

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

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

Configuring an Authenticating, Machine-only Network

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

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

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

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

Example 2-25 machineAuthentication

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

Configuring the Authenticating, Machine Credential Source Elements

Configure the following element:

collectionMethod

Schema paths:

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

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

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

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

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

auto

Schema paths:

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

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

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

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

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

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

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

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

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

It is an empty element with no value.

static

Schema paths:

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

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

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

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

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

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

Example 2-26 collectionMethod (machine)

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

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

Configuring the Authenticating, Connection Independent Base Elements

Configure the following element:

useAnonymousId

Schema paths:

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

The mandatory boolean element *useAnonymousId* specifies the content of the EAP Response/Identity message used in the phase 1, outer (unprotected) tunnel of tunneled EAP methods.

The element has the following values:

- False—With this option SSC will:
  - Allow all EAP methods.
  - Send the UserName in all EAP responses.

  In this mode the client will send UserName@Domain for the Identity response, where the domain routing (@Domain) is optional. The user identity (UserName) is sent in the clear. [UserName] will always be sent for the EAP Identity response of any phase 2, inner tunnel (protected identity).

  Note In general, a Microsoft AAA server using its PEAP method will require this setting.

- True—With this option SSC will:
  - Restrict the set of allowed authentication methods to those that use tunneling.
    Business rule: the corresponding *eapMethods* element must specify only EAP FAST, PEAP or TTLS.
  - Restrict sending the UserName in the EAP Identity response of the outer (unprotected) tunnel.

  In this mode the client will send anonymous@Domain for the Identity response.

  Note If using domains with Cisco ACS 3.3 AAA server you must use this setting.

Configuring the Authentication Static Credential Elements

Configure the following elements:

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

staticIdentity

Schema paths:

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

The value of the optional element *staticIdentity* specifies the content of the EAP Response/Identity message. An identity has a Network Access Identifier (NAI) format and takes the following generalized form: UserName@Domain, where the use of the @Domain (also referred to as the realm) is optional. (The use of a domain is based on the requirements of your specific authentication server.)

The legacy NT4 format, Domain\UserName, is also acceptable.

If a domain is specified no processing is done to it. Whether it is a domain alias (not fully qualified) or a fully qualified domain name, whatever is configured is sent in the EAP Response message.

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

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

staticPassword

Schema paths:

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

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

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

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

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

Configuring an Authenticating, User-Only Network

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

Step 1 Perform the tasks defined in "Configuring the Authenticating, User-Only Connection Occurrence Elements".

Step 2 Perform the tasks defined in "Configuring the Authenticating, User Credential Source (1) Elements".

Step 3 Perform the tasks defined in "Configuring the Authenticating, Connection Independent Base Elements".

Step 4 Perform the tasks defined in "Configuring the Authentication Static Credential Elements".

Step 5 Perform the tasks defined in "Configuring the FAST PAC Elements".

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

Example 2-27 userAuthentication

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

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

Configuring the Authenticating, User-Only Connection Occurrence Elements

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

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

autoConnect

Schema paths:

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

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

Configuration of the following child element is required:

connectBeforeLogon

Schema paths:

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

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

The element has the following values:

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

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

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

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

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

manualConnect

Schema paths:

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

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

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

Example 2-28 autoConnect

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

Configuring the Authenticating, User Credential Source (1) Elements

Configure the following element:

collectionMethod

Schema paths:

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

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

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

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

prompt

Schema paths:

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

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

Next item: "Choosing Prompted Credential Storage".

singleSignOn

Schema paths:

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

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

static

Schema paths:

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

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

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

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

autoMachine

Schema paths:

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

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

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

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

Example 2-29 collectionMethod (user)

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

<collectionMethod>
    <singleSignOn/>
</collectionMethod>

<collectionMethod>
    <static/>
</collectionMethod>

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

Configuring the Authenticating, User Credential Source (2) Elements

Configure the following element:

collectionMethod

Schema paths:

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

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

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

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

prompt

Schema paths:

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

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

Next item: "Choosing Prompted Credential Storage".

singleSignOn

Schema paths:

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

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

static

Schema paths:

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

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

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

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

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

Example 2-30 collectionMethod (machine/user)

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

<collectionMethod>
    <singleSignOn/>
</collectionMethod>

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

Choosing Prompted Credential Storage

Configure the following element:

credentialsStorage

Schema paths:

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

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

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

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

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

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

forever

Schema paths:

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

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

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

logonSession

Schema paths:

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

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

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

duration

Schema paths:

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

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

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

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

Example 2-31 credentialsStorage

```
<credentialsStorage>
```

```
    <forever/>
</credentialsStorage>

<credentialsStorage>
    <logonSession/>
</credentialsStorage>

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

Configuring the FAST PAC Elements

Configure the following element:

pacs

Schema paths:

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

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

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

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

Configuration of the following child element is required:

pac

Schema paths:

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

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

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

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

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

Step 1 Specify and configure the following element:

pacReference

Schema paths:

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

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

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

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

Step 2 Configure the following element:

secretKey

Schema paths:

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

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

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

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

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

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

Example 2-32 pacs

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

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

Configuring an Authenticating, Machine and User Network

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

Step 1 Configure the machine portion.

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

b. Perform the tasks defined in "Configuring the Authenticating, Connection Independent Base Elements".

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

Step 2 Configure the user portion.

a. Perform the tasks defined in "Configuring the Authenticating, User Connection Occurrence Elements".

a. Perform the tasks defined in "Configuring the Authenticating, User Credential Source (2) Elements".

b. Perform the tasks defined in "Configuring the Authenticating, Connection Independent Base Elements".

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

d. Perform the tasks defined in "Configuring the FAST PAC Elements".

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

Example 2-33 userAuthentication

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

Configuring the Authenticating, User Connection Occurrence Elements

Configure the following element:

autoConnect

Schema paths:

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

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

The element has the following values:

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

Wired Network Base Elements

Configure the following element:

displayName

Schema path:

configuration/networks/wiredNetwork/displayName

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

Choosing the Wired Network's Security Class

Specify one of the following security classes for the network:

- Open network—Use element *openNetworkMachineConnection*. Most likely you will not be deploying an open network to your end-users because of the need to pre-specify the SSID. Your end-users, if allowed, can create a network profile for a connection to a specific open network.
- Authentication Network—Use element *authenticationNetwork*. This is the most likely choice because you will want to preconfigure your enterprise network consistent with your authentication server and its policies and with your credentials environment.

openNetworkMachineConnection

Schema path:

configuration/networks/wiredNetwork/openNetworkMachineConnection

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

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

authenticationNetwork

Schema path:

configuration/networks/wiredNetwork/authenticationNetwork

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

Next item: "Configuring an Authenticating Wired Network"

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

Example 2-34 wiredNetwork

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

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

Configuring an Authenticating Wired Network

Follow these steps to configure an authenticating Wired network.

Step 1 Perform the tasks defined in "Configuring the Authenticating Network Base Elements".

Step 2 Perform the tasks defined in "Choosing the Authentication Network's Connection Context".

Step 3 Perform the tasks defined in "Choosing Wired EAP Methods".

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

Example 2-35 partial authenticationNetwork

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

Choosing Wi-Fi EAP Methods

Perform the tasks defined in "Choosing Wi-Fi/Wired EAP Methods".

Choosing Wired EAP Methods

Configure the following element:

eapMethods

Schema paths:

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

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

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

Business rule: Only EAP methods supported by the policy are permitted. See element *allowedEapMethods* in "Network Policy".

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

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

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

eapMd5

Schema paths:

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

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

eapMschapv2

Schema paths:

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

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

eapGtc

Schema paths:

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

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

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

Example 2-36 eapMethods (wired)

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

Choosing Wi-Fi/Wired EAP Methods

Configure the following element:

eapMethods

Schema paths:

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

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

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

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

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

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

eapFast

Schema paths:

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

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

Next item: "Configuring EAP-FAST".

eapPeap

Schema paths:

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

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

Next item: "Configuring EAP-PEAP".

eapTtls

Schema paths:

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

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

Next item: "Configuring EAP-TTLS".

eapTls

Schema paths:

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

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

Next item: "Configuring EAP-TLS".

eapLeap

Schema paths:

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

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

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

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

Example 2-37 eapMethods (wired or wireless)

```
<eapMethods>
    <eapLeap/>
    <eapFast>
        {child elements}
    </eapFast>
```

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

Configuring EAP-FAST

Follow these steps to configure EAP-FAST.

Step 1 Perform the tasks defined in "Configuring EAP Base Elements".

Step 2 Perform the tasks defined in "Configuring FAST Client Certificates".

Step 3 Perform the tasks defined in "Configuring Inner Methods".

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

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

Example 2-38 eapFast

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

Configuring EAP-PEAP

Follow these steps to configure EAP-PEAP.

Step 1 Perform the tasks defined in "Configuring EAP Base Elements".

Step 2 Perform the tasks defined in "Configuring PEAP Client Certificates".

Step 3 Perform the tasks defined in "Configuring Inner Methods".

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

Example 2-39 eapPeap

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

Configuring EAP-TTLS

Follow these steps to configure EAP-TTLS.

Step 1 Perform the tasks defined in "Configuring EAP Base Elements".

Step 2 Perform the tasks defined in "Configuring TTLS Inner Methods".

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

Example 2-40 eapTtls

```
<eapTtls>
    <validateServerIdentity>true</validateServerIdentity>
    <enableFastReconnect>true</enableFastReconnect>
    <innerMethods>
        {child element}
    </innerMethods>
</eapTtls>
```

Configuring EAP-TLS

Follow these steps to configure EAP-TLS.

Step 1 Perform the tasks defined in "Configuring EAP Base Elements".

Step 2 Perform the tasks defined in "Configuring the Client Certificate Source".

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

Example 2-41 eapTls

```
<eapTls>
    <validateServerIdentity>true</validateServerIdentity>
    <enableFastReconnect>true</enableFastReconnect>
    <certificateSource>
        {child element}
    </certificateSource>
</eapTls>
```

Configuring EAP Base Elements

Configure the following elements:

validateServerIdentity

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTtls | eapTls/validateServerIdentity

The mandatory boolean element *validateServerIdentity* specifies whether or not SSC performs client-side validation of the server during authentication.

The element has the following values:

- True—Validate the server certificate.
- False—Do not validate the server certificate.

  This option is *not recommended*, and is usually only used for debugging (to help determine if the server certificate is responsible for a failed authentication) because it reduces the level of security. Using this option implies that you are not Wi-Fi compliant for this wireless network.

Business rule: If the value of *validateServerIdentity* is true, then the optional element *serverValidation* for the same network must be present to provide the server validation details.

Business rule: The value must be true if the network policy specifies mandatory validation. See element *alwaysValidate* in "Network Policy".

enableFastReconnect

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap | eapTtls | eapTls/enableFastReconnect

The mandatory boolean element *enableFastReconnect* specifies whether or not SSC performs a fast session resumption using cached credential information during a re-authentication request. (Applies to both outer and inner tunnel methods, where applicable.)

The element has the following values:

- True—Allow fast session resumption.
- False—Disallow fast session resumption.

Note If the network profile specifies multiple EAP authentication methods that involve creation of an SSL session (elements eapFast | eapPeap | eapTtls | eapTls) and the element *enableFastReconnect* is set to True for any method, then fast session resumption will be enabled for all the methods associated with this network profile.

Configuring FAST Client Certificates

Refer to Example 2-38 for the required sequence of these elements.

Step 1 Configure the following element:

protectClientCertificate

Schema paths:

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

The mandatory boolean element *protectClientCertificate* specifies whether or not SSC will send, when requested, a certificate unprotected.

The element has the following values:

- True—Indicates:

  Enable protection and using a client certificate or

  Not using a client certificate.

  When requested by the server for a client certificate during the unprotected (phase 1) portion of FAST PAC provisioning:

  – SSC refuses at this point to send any certificate (it's allowed this option) because it will wait for the protected Phase 2 of the protocol. (Actually, a tunnel will first be established, based on the server's certificate, and SSC will send its client certificate before Phase 2 begins).

  Cisco ACS, when not configured for using a client certificate, will not ask for the certificate during phase 2. So it is important to use this value to avoid any unwanted client certificate prompts.

- False—Disable protection and using a client certificate.
  When requested by the server for a client certificate during the unprotected (phase 1) portion of FAST PAC provisioning:

  – If there is a client certificate available, it will be sent.

  – If there is no client certificate, none will be sent and the server policy will determine if authentication continues or fails.

This has no affect on the use of a client certificate during the protected (phase 2) portion of FAST PAC provisioning. If the server is configured to request the sending of the client certificate within the secure tunnel, the client will always attempt to use one. If none is available and not sent, the connection attempt will fail.

Step 2 Perform the tasks defined in "Configuring the Client Certificate Source":

Configuring PEAP Client Certificates

Refer to Example 2-39 for the required sequence of these elements.

Step 1 Configure the following element:

protectClientCertificate

Schema paths:

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

The mandatory boolean element *protectClientCertificate* specifies whether or not SSC will send, when requested, a certificate unprotected.

The element has the following values:

- True—Enable protection.
  When requested by the server for a client certificate during the unprotected (phase 1) portion of the protocol:
  - The client certificate will never be sent to the server (since there is no way to send a client certificate protected for this EAP method). Server policy will determine if authentication continues or fails.
- False—Disable protection.
  When requested by the server for a client certificate during the unprotected (phase 1) portion of the protocol:
  - If there is a client certificate available, it will be sent.
  - If there is no client certificate, none will be sent and the server policy will determine if authentication continues or fails.

Step 2 Perform the tasks defined in "Configuring the Client Certificate Source":

Configuring the Client Certificate Source

Configure the following elements:

certificateSource

Schema paths:

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

The element *certificateSource* forms the container for specifying the source of a client certificate. No element value is specified.

Restriction: *certificateSource* is required within the *eapTls* element.

Business rule: *certificateSource* is optional within the eapPeap and eapFast elements for the outer tunnel as determined by the policy of the authentication server. However, it is required if the corresponding inner method specified is EAP TLS.

Specify one of the following sources for your client certificate:

- Obtain only from a SmartCard—Use element *smartCardOnlyCertificate*.
- Obtain from either the Windows certificate store or a SmartCard—Use element *smartCardOrOsCertificate*.

SmartCard certificate properties:

- Only a single smartcard reader (the first one detected) is supported.
- Multiple certificates from a single SmartCard are supported.
- SmartCards must support the Microsoft CryptoAPI and SCard interfaces to Cryptographic Service Provider (CSP) functionality. Furthermore, any SmartCard reader and SmartCard combination must inter-operate via the PC/SC interface to provide low level support for these same CSP functions.
- A SmartCard PIN (two-factor authentication) is supported. A prompt will be made through either the Enter Your Credentials dialog or the operating system GINA. PIN behavior depends on the configured user credential collection method for the network.
  - collectionMethod/prompt | static—PINs are not stored and therefore will be subsequently requested again under certain situations such as when a server initiated re-authentication is required (unless the EAP method is configured for fast session resumption), roaming, a failed re-authentication, lost association, resumption from hibernate.
  - collectionMethod/singleSignOn—PINs are stored for the duration of the logon session. Therefore, subsequent pop-up requests are avoided.

Windows certificate properties:

- If a Windows user certificate is required as part of the authentication process, it must be appropriately pre-installed as a separate task.
  - A user certificate is obtained from the Personal Certificate Store for the currently logged in Windows user.
  - A machine certificate is obtained from the Personal Certificate Store for the Local Computer.
- Only valid certificates are displayed for selection. Expired certificates are not listed. Also, valid certificates that are about to expire contain a warning that shows how many days left before the certificate expires.
- When configured for an automatic user connection, a certificate with Strong Private Key Protection will fail at logon and should not be used. Certificates with this property are only supported when configured for always making manual connections at desktop.
- The identifying information for the selected certificate in the selection drop-down list is obtained from the various fields of the certificate as follows:
  - text box name—Subject: CN (Common Name).
  - Issued to:—Subject: CN (Common Name).
  - Issued by:—Issuer: CN (Common Name).
  - Alternative Name:—Subject Alternate Name: DNSName
  - Expires:—Valid to.
  - Extended Key Usage—Extended Key Usage

smartCardOnlyCertificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/
machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/
eapFast | eapPeap | eapTtls | eapTls/certificateSource/smartCardOnlyCertificate

The optional element *smartCardOnlyCertificate* specifies that a client certificate must be obtained only from a SmartCard. It is an empty element with no value.

Business rule: If the corresponding connection context is *machineAuthentication*, then this option is not allowed because a machine certificate must be from the OS store.

smartCardOrOsCertificate

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/
machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/
eapFast | eapPeap | eapTtls | eapTls/certificateSource/smartCardOrOsCertificate

The optional element *smartCardOrOsCertificate* specifies that a client certificate may be obtained from either a SmartCard or the Windows certificate store. It is an empty element with no value.

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

Example 2-42 certificateSource

```
<certificateSource>
    <smartCardOrOsCertificate/>
</certificateSource>

<certificateSource>
    <smartCardOnlyCertificate/>
</certificateSource>
```

Configuring Inner Methods

Configure the following element:

innerEapMethods

Schema paths:

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

The mandatory element *innerEapMethods* forms the container for listing the supported inner EAP methods for the outer EAP method. During the EAP negotiation phase of authentication process, if SSC receives a request from the server for a particular EAP method that it is not configured to support, it will respond with an ordered list of alternate EAP methods. The server will process the list to search for an acceptable alternate. If it finds one, it will re-negotiate with the mutually agreed-to method, otherwise authentication will fail. The order of the list is determined by the order in which the chosen child elements are listed in the XML. No element value is specified.

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

Specify one or more of the following FAST/PEAP inner EAP methods:

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

- EAP-TLS—Use element *eapTls*.

eapMschapv2

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods/eapMschapv2

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

eapGtc

Schema paths:

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

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

eapTls

Schema paths:

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

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

Business rule: With this option, you must add and configure the elements of "Configuring the Client Certificate Source".

Configuration of the following child element is required:

validateServerIdentity

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerEapMethods/eapTls/validateServerIdentity

The mandatory boolean element *validateServerIdentity* specifies whether or not SSC performs client-side validation of the server during authentication within the secure inner tunnel.

The element has the following values:

- True—Validate the server certificate.
- False—Do not validate the server certificate.

  This option is not recommended, and is usually only used for debugging (to help determine if the server certificate is responsible for a failed authentication) because it reduces the level of security. Using this option implies that you are not Wi-Fi compliant for this wireless network.

Business rule: If the value of *validateServerIdentity* is true, then the optional element *serverValidation* for the same network must be present to provide the server validation details.

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

Example 2-43 innerEapMethods

```
<innerEapMethods>
    <eapMschapv2/>
    <eapGtc/>
    <eapTls>
        <validateServerIdentity>true</validateServerIdentity>
    </eapTls>
</innerEapMethods>
```

Configuring TTLS Inner Methods

Configure the following element:

innerMethods

Schema paths:

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

The mandatory element *innerMethods* forms the container for listing the supported inner methods for the outer EAP method. The use of legacy and EAP methods inside the EAP-TTLS are mutually exclusive and cannot be simultaneously specified. No element value is specified.

Specify one the following classes of inner methods:

- Legacy methods—Use element *legacy*.
- EAP methods—Use element *eap*.

legacy

Schema paths:

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

The optional element *legacy* forms the container for configuring one of the legacy TTLS inner methods. No element value is specified.

Specify one of the following legacy TTLS inner methods:

- PAP—Use element *pap*.
- CHAP—Use element *chap*.
- MSCHAP—Use element *mschap*.
- MSCHAPv2—Use element *mschapv2*.

pap

Schema paths:

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

The existence of the optional element *pap* specifies the use of Password Authentication Protocol (PAP) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

chap

Schema paths:

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

The existence of the optional element *chap* specifies the use of Challenge-Handshake Authentication Protocol (CHAP) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

mschap

Schema paths:

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

The existence of the optional element *mschap* specifies the use of Microsoft Challenge-Handshake Authentication Protocol (MSCHAP) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

mschapv2

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapTtls/innerMethods/legacy/mschapv2

The existence of the optional element *mschapv2* specifies the use of Microsoft Challenge-Handshake Authentication Protocol v2 (MSCHAPv2) for the inner authentication method for this network. No additional configuring is required. It is an empty element with no value.

eap

Schema paths:

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

The optional element *eap* forms the container for configuring one of the standard EAP methods. During the EAP negotiation phase of authentication process, if SSC receives a request from the server for a particular EAP method that it is not configured to support, it will respond with an ordered list of alternate EAP methods. The server will process the list to search for an acceptable alternate. If it finds one, it will re-negotiate with the mutually agreed-to method; otherwise authentication will fail. The order of the list is determined by the order in which the chosen child elements are listed in the XML. No element value is specified.

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

Specify one or more of the following inner EAP methods:

- EAP-MSCHAPv2—Use element *eapMschapv2*.

- EAP-MD5—Use element *eapMd5*.

eapMschapv2

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerMethods/eapMschapv2

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

eapMd5

Schema paths:

configuration/networks/[wifiNetwork | wiredNetwork]/authenticationNetwork/machineAuthentication | userAuthentication | machineUserAuthentication/eapMethods/eapFast | eapPeap/innerMethods/eapMd5

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

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

Example 2-44 innerMethods

```
<innerMethods>
    <legacy>
        <pap/>
    </legacy>
</innerMethods>

<innerMethods>
    <legacy>
        <chap/>
    </legacy>
</innerMethods>

<innerMethods>
    <legacy>
        <mschap/>
    </legacy>
</innerMethods>

<innerMethods>
    <legacy>
        <mschapv2/>
    </legacy>
</innerMethods>

<innerMethods>
    <eap>
        <eapMd5/>
        <eapMschapv2/>
    </eap>
</innerMethods>
```