



Administrator Guide for Cisco Trust Agent, Release 2.1

Released For Use With
Network Admission Control Framework 2.1 Program

Revised: May 23, 2008

Americas Headquarters

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

Text Part Number: OL-11310-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn is a service mark; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0805R)

Administrator Guide for Cisco Trust Agent 2.1

© 2008 Cisco Systems, Inc. All rights reserved.



CONTENTS

Preface xi

Cisco Trust Agent 2.1 Release	xi
Qualified Deployments of CTA 2.1	xii
Product Versioning	xii
Audience	xii
Conventions	xiii
Related Documentation	xiv
Obtaining Documentation	xiv
Cisco.com	xiv
Product Documentation DVD	xv
Ordering Documentation	xv
Documentation Feedback	xv
Cisco Product Security Overview	xvi
Reporting Security Problems in Cisco Products	xvi
Product Alerts and Field Notices	xvii
Obtaining Technical Assistance	xviii
Cisco Support Website	xviii
Submitting a Service Request	xix
Definitions of Service Request Severity	xx
Obtaining Additional Publications and Information	xx

CHAPTER 1

Cisco Trust Agent Overview 1-1

Cisco NAC Overview	1-1
--------------------	-----

Posture Validation Process Overview	1-2
Initial Posture Validation Process	1-2
Posture Revalidation Process	1-4
Cisco Trust Agent Deployment Considerations	1-5

CHAPTER 2

Installing the Cisco Trust Agent on Linux Operating Systems 2-1

Verifying System Requirements on Linux	2-2
Installation Files	2-3
Initial Deployments	2-3
CTA Scripting Interface Feature	2-3
Installing Cisco Trust Agent	2-4
General Installation Instructions	2-4
Extracting the Installation File and Accepting the EULA	2-4
Installing CTA from the Command Line	2-5
Creating a Custom CTA Installation Package	2-5
Upgrading to Cisco Trust Agent, Release 2.1	2-7
Verifying Cisco Trust Agent Installation on Linux	2-8
Verifying CTA is Running	2-8
Verifying CTA Package Information	2-9
Uninstalling Cisco Trust Agent on Linux	2-9

CHAPTER 3

Installing the Cisco Trust Agent on Macintosh Operating Systems 3-1

System Requirements for Mac OS X	3-2
CTA Scripting Interface Feature	3-3
Installation Files	3-3
Installing Cisco Trust Agent	3-3
General Installation Instructions	3-3
Extracting the Installation Image and Accepting the EULA	3-4

Installing CTA from the Command Line	3-4
Installing CTA Using an Installation Wizard	3-6
Installing CTA Using a Custom Installation Package	3-12
Repairing or Upgrading an Existing CTA Installation	3-14
Verifying Cisco Trust Agent Installation	3-16
Uninstalling Cisco Trust Agent	3-16
Uninstalling CTA and the CTA Scripting Interface	3-16
Uninstalling Only the CTA Scripting Interface	3-17

CHAPTER 4**Installing the Cisco Trust Agent on Windows Operating Systems 4-1**

System Requirements for Installation	4-2
Optional Features You Can Install with CTA	4-4
CTA 802.1x Wired Client	4-4
CTA Scripting Interface	4-4
Installation Files	4-5
Installing Cisco Trust Agent	4-5
General Installation Instructions	4-6
Installing CTA Using MSI Commands	4-7
Installing CTA Using an Installation Wizard	4-13
Installing CTA Using a Custom Installation Package	4-22
Upgrading to Cisco Trust Agent, Release 2.1	4-27
Upgrading from Cisco Trust Agent, Release 1.0	4-27
Upgrading from Cisco Trust Agent, Release 2.0.0.30	4-28
Upgrading from Cisco Trust Agent, Release 2.0.1	4-29
Upgrading from CTA 2.1 Selective Availability and Beta Releases to CTA 2.1.103.0	4-29
Verifying the Cisco Trust Agent Installation	4-30
Uninstalling Cisco Trust Agent on Windows	4-31

CHAPTER 5

Configuring Cisco Trust Agent 5-1

- The ctad.ini Configuration File 5-2
 - Editing the ctad.ini Configuration File 5-3
 - ctad.ini Configuration Parameters 5-4
- Configuring EAP over UDP Communication 5-12
- Configuring Posture Plugins 5-13
 - Configuring CTA and Posture Plugin Interaction 5-13
 - Configuring the Default Posture Plug-in Message Size 5-16
 - Configuring an Application-Specific Posture Plug-in Message Size 5-17
 - Configuring PPMsgSize for Host Posture Plugin 5-18
 - Configuring PPMsgSize for Symantec Posture Plugin 5-19
 - Configuring Asynchronous Posture Status Query 5-19
- Configuring User Notifications 5-20
 - Configuring Windows User Notifications 5-20
 - Configuring Linux User Notifications 5-21
 - Configuring Mac OS X User Notifications 5-22
 - Configuring Clickable URL and Browser Auto-Launch Features 5-23
 - Logging Notifications 5-24
- Certificate Distinguished Name Matching 5-25
 - DN Matching Rule Syntax 5-25
 - Configuring Certificate DN Matching 5-27
- Configuring the Scripting Interface 5-27
- Sample Windows ctad.ini File 5-28
- Sample Linux ctad.ini File 5-32
- Sample Mac OS X ctad.ini File 5-37

CHAPTER 6

Cisco Trust Agent Event Logging 6-1

- How Logging Works 6-2
- CTA Log Files 6-2

Log File Format	6-3
Logging Considerations	6-4
The clogcli Logging Utility	6-4
Logging Levels	6-11
Configuring CTA Logging For Large Deployments	6-11
Sample ctalogd-temp.ini File	6-13

CHAPTER 7**Posture Plugins 7-1**

Types of Posture Plugins Installed by Default	7-2
Host Posture Plugin	7-2
Package Information Retrieved by Host Posture Plugin for Mac OS X Platforms	7-5
Cisco Trust Agent Posture Plugin	7-5
CTA Scripting Posture Plugin	7-9
Plugin Installation and Upgrade	7-9

CHAPTER 8**Cisco Trust Agent's Use of Certificates 8-1**

About The ACS Server Root Certificate	8-3
About The ctacert Utility	8-3
Installing or Updating Certificates Using the ctacert Utility	8-4
Installing or Updating a Certificate on Linux Operating Systems	8-4
Installing or Updating a Certificate on Mac OS X Operating System	8-4
Installing or Updating a Certificate on Windows Operating Systems	8-5
Listing Certificates in the Certificate Store	8-6
Listing Certificates in the Certificate Store on Linux Operating Systems	8-6
Listing Certificates in the Certificate Store on Mac OS X Operating System	8-7
Deleting Certificates from the Certificate Store	8-8

Deleting a Certificate from the Certificate Store on Linux Operating Systems	8-8
Deleting a Certificate from the Certificate Store on Mac OS X Operating System	8-9
Clearing Certificates from the Certificate Store	8-9
Clearing All Certificates from the Certificate Store on Linux Operating Systems	8-9
Clearing All Certificates from the Certificate Store on Mac OS X Operating Systems	8-10
Configuring Machine Authentication Using Certificates	8-10
Requesting the Machine Certificate for Machine Authentication	8-11
Configuring User Authentication Using Certificates	8-12
Importing the User Certificate for User Authentication	8-12
Configuring Machine and User Authentication Using Certificates	8-13
Distinguished Name Matching	8-14
Converting DER Formatted Certificates to PEM Formatted Certificates	8-14

CHAPTER 9

Cisco Trust Agent 802.1x Wired Client 9-1

802.1x Wired Client Features	9-3
802.1x Wired Client Administrative and Client Versions	9-4
802.1x Wired Client User Interface	9-4
Administrative 802.1x Wired Client Automatic Startup	9-4
Network Connection Status	9-5
Disabling the 802.1x Wired Client System Tray Icon	9-6
802.1x Wired Client System Tray Shortcut Menu	9-7
802.1x Wired Client Window	9-7
Network Connection Status	9-8
Access Device Connection Status	9-9
Basic 802.1x Wired Client Procedures	9-11
Opening the 802.1x Wired Client	9-11

Manually Connecting To the Network	9-12
Manually Disconnecting From the Network	9-12
Viewing Network Summary	9-12
Viewing Access Device Status	9-13
Getting Started with 802.1x Wired Client Functions	9-16
Administrative 802.1x Wired Client Overview	9-16
Authentication Methods Overview	9-16
Overview of FAST Connections in a User Logon Context	9-17
Overview of FAST Connections in a Machine Credentials Context	9-18
User Credentials	9-20
Initial Credential Provisioning	9-20
Machine Credentials	9-21
Pre-PAC or no-PAC Provisioning	9-22
Post-PAC Provisioning	9-22
Credential Revalidation	9-23
Server-Initiated Credential Revalidation	9-23
User-Initiated Credential Revalidation	9-23
Understanding Policies and Profiles	9-24
802.1x Wired Client Policy File	9-24
802.1x Wired Client Network Policy File	9-25
Create Deployment Package Wizard	9-26
User Credentials Area	9-27
Automatically Establishing Machine Connection	9-27
Allow Unprotected Client Cert Area	9-28
Trusted Server Validation Area	9-29
User Identity Protection Area	9-32
Authentication Retries Wired / Ethernet Settings	9-34
Deploying End-User 802.1x Wired Clients	9-35
Creating Deployment Packages	9-36

Installing Server Certificates on the Host	9-42
Installing Deployment Packages on Hosts	9-43
Changing Deployment Packages on Hosts	9-44
Replacing a Deployment Package on a Host	9-44

CHAPTER 10

Cisco Trust Agent 802.1x Wired Client Logging 10-1

Technical Log	10-2
Understanding the Technical Log Status and Error Messages	10-3
Technical Log Message Format	10-3
Technical Log Message Content	10-5
Additional Message <value> Descriptions	10-11
Port Status Values	10-12
System Report	10-15
Creating a System Report	10-16

CHAPTER 11

Using the Scripting Interface 11-1

Scripting Interface Overview	11-3
How the Scripting Interface Relays Posture Credentials to ACS	11-3
ctasi Scripting Interface File	11-4
ctascriptpp Posture Plugin File	11-5
Information Files	11-5
Posture Scripts	11-7
Posture Data Files	11-7
Configuring the NAC Environment to Use Your Posture Script	11-13
Write a Posture Script	11-14
Write an Information File for the Posture Script	11-14
Register Posture Scripts	11-14
Add Script Interface Attributes to the ACS Dictionary	11-15

Configure ACS Rules to Determine Posture Based on the Script's Posture Attributes 11-18

Posture Scripts Invoking ctasi 11-18

Status Change 11-20

Stale Posture Data 11-20

Managing Stale Posture Database with CTA 11-21

Managing Stale Posture Database on the ACS Server 11-22

APPENDIX A

ctastat Diagnostic Tool A-1

Running the ctastat Utility A-2

Running ctastat on a Linux Operating System A-2

Running ctastat on a Mac OS X Operating System A-2

Running ctastat on a Windows Operating System A-2

ctastat Utility Output A-3

General CTA Information A-3

Session Information A-3

Plugins Information A-4

ctastat Utility Sample Output A-4

APPENDIX B

Alternate Methods of Installing CTA B-1

Installing CTA 2.1 Using CSA MC 5.2 B-1

APPENDIX C

Open Source License Acknowledgement C-1

OpenSSL/Open SSL Project C-1

License Issues C-1

Info-ZIP C-4



Preface

Cisco Trust Agent (CTA) collects and reports posture credentials from clients in a Network Admission Control (NAC) environment.

Posture credentials are information about a NAC-compliant software application or a client on which it runs. These are examples of posture credentials that can be collected from the client: software application versions, machine name, operating system, and the client's MAC Address.

CTA reports the posture information it gathers to the Cisco Secure Access Control Server (ACS) which then determines application posture and an overall client posture. Examples of client posture could be "Healthy," "Quarantine," or "Unknown."

Based on the client's posture a NAC-compliant Network Access Device (NAD), such as a Cisco Switch or Cisco Router, provide the client access to a network.

Cisco Trust Agent 2.1 Release

The goals of Cisco Trust Agent, Release 2.1.103.0 (CTA 2.1) are to improve on the CTA 2.1.18.0 selective availability release by resolving outstanding product defects and to provide new functionality from that offered in the CTA 2.0.0.30 release. Cisco Trust Agent release 2.1 is an integral component of the Network Admission Control Framework 2.1 solution.

Qualified Deployments of CTA 2.1

Cisco Trust Agent 2.1.103.0 will be distributed to existing customers of CTA and those customers evaluating the NAC Framework 2.1 programs.

CTA 2.1 is not intended for distribution to new customers of CTA nor new customers of the NAC 2.1 Framework solution. New customers to CTA and NAC should work with their Cisco Account Team representative to evaluate their NAC Framework-qualified infrastructure and use-case scenarios.

We are making an extra effort to qualify our customers' infrastructure and goals to ensure that the components in their network are compatible with the NAC Framework, that their goals will be met by the NAC Framework, and that the deployment of the NAC Framework will be successful.

Product Versioning

The full version number of this release is CTA 2.1.103.0. The full release number is used in installation files names and in the text of the *Administrator Guide for Cisco Trust Agent, Release 2.1* and the *Release Notes for Cisco Trust Agent, Release 2.1* when it is important to distinguish the version of CTA being discussed. Any references in the documentation to CTA 2.1 are referring to CTA 2.1.103.0 unless otherwise noted.

Audience

The *Administrator Guide for Cisco Trust Agent, Release 2.1* provides installation, configuration, and monitoring information to administrators responsible for deploying Cisco Trust Agent to network clients.

Conventions

This document uses the following conventions:

Item	Convention
Commands, keywords, special terminology, and options that should be selected during procedures	boldface font
Variables for which you supply values and new or important terminology	<i>italic</i> font
Displayed session and system information, paths and filenames	screen font
Information you enter	boldface screen font
Variables you enter	<i>italic screen</i> font
Menu items and button names	boldface font
Indicates menu items to select, in the order you select them	Option > Network Preferences

**Tip**

Identifies information to help you get the most benefit from your product.

**Note**

Means *reader take note*. Notes identify important information that you should reflect upon before continuing, contain helpful suggestions, or provide references to materials not contained in the document.

**Caution**

Means *reader be careful*. In this situation, you might do something that could result in equipment damage, loss of data, or a potential breach in your network security.

**Warning**

Identifies information that you must heed to prevent injuring yourself or damaging the state of the software or equipment. Warnings identify definite security breaches that will result if the information presented is not followed carefully.

Related Documentation

**Note**

Although every effort has been made to validate the accuracy of the information in the printed and electronic documentation, you should also review Cisco Trust Agent documentation on [Cisco.com](http://www.cisco.com) for any updates.

The following documentation is available on [Cisco.com](http://www.cisco.com):

- For Cisco Trust Agent, there are two documents available: This guide, the *Administrator Guide for Cisco Trust Agent, Release 2.1*, and the *Release Notes for Cisco Trust Agent, Release 2.1*.
- For documentation of other components in the Cisco NAC solution, see [Network Admission Control \(NAC\) Framework page](#) on Cisco.com.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. This section explains the product documentation resources that Cisco offers.

Cisco.com

You can access the most current Cisco documentation at this URL:

<http://www.cisco.com/techsupport>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

The Product Documentation DVD is a library of technical product documentation on a portable medium. The DVD enables you to access installation, configuration, and command guides for Cisco hardware and software products. With the DVD, you have access to the HTML documentation and some of the PDF files found on the Cisco website at this URL:

<http://www.cisco.com/univercd/home/home.htm>

The Product Documentation DVD is created and released regularly. DVDs are available singly or by subscription. Registered Cisco.com users can order a Product Documentation DVD (product number DOC-DOCDVD= or DOC-DOCDVD=SUB) from Cisco Marketplace at the Product Documentation Store at this URL:

<http://www.cisco.com/go/marketplace/docstore>

Ordering Documentation

You must be a registered Cisco.com user to access Cisco Marketplace. Registered users may order Cisco documentation at the Product Documentation Store at this URL:

<http://www.cisco.com/go/marketplace/docstore>

If you do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

Documentation Feedback

You can provide feedback about Cisco technical documentation on the Cisco Support site area by entering your comments in the feedback form available in every online document.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you will find information about how to do the following:

- Report security vulnerabilities in Cisco products
- Obtain assistance with security incidents that involve Cisco products
- Register to receive security information from Cisco

A current list of security advisories, security notices, and security responses for Cisco products is available at this URL:

<http://www.cisco.com/go/psirt>

To see security advisories, security notices, and security responses as they are updated in real time, you can subscribe to the Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed. Information about how to subscribe to the PSIRT RSS feed is found at this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you have identified a vulnerability in a Cisco product, contact PSIRT:

- For emergencies only—security-alert@cisco.com

An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.

- For nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

**Tip**

We encourage you to use Pretty Good Privacy (PGP) or a compatible product (for example, GnuPG) to encrypt any sensitive information that you send to Cisco. PSIRT can work with information that has been encrypted with PGP versions 2.x through 9.x.

Never use a revoked encryption key or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

If you do not have or use PGP, contact PSIRT to find other means of encrypting the data before sending any sensitive material.

Product Alerts and Field Notices

Modifications to or updates about Cisco products are announced in Cisco Product Alerts and Cisco Field Notices. You can receive these announcements by using the Product Alert Tool on Cisco.com. This tool enables you to create a profile and choose those products for which you want to receive information.

To access the Product Alert Tool, you must be a registered Cisco.com user. Registered users can access the tool at this URL:

<http://tools.cisco.com/Support/PAT/do/ViewMyProfiles.do?local=en>

To register as a Cisco.com user, go to this URL:

<http://tools.cisco.com/RPF/register/register.do>

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Support website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Support Website

The Cisco Support website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day at this URL:

<http://www.cisco.com/en/US/support/index.html>

Access to all tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>



Note

Before you submit a request for service online or by phone, use the **Cisco Product Identification Tool** to locate your product serial number. You can access this tool from the Cisco Support website by clicking the **Get Tools & Resources** link, clicking the **All Tools (A-Z)** tab, and then choosing **Cisco Product Identification Tool** from the alphabetical list. This tool offers three search options: by product ID or model name; by tree view; or, for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.



Tip

Displaying and Searching on Cisco.com

If you suspect that the browser is not refreshing a web page, force the browser to update the web page by holding down the Ctrl key while pressing **F5**.

To find technical information, narrow your search to look in technical documentation, not the entire Cisco.com website. After using the Search box on the Cisco.com home page, click the **Advanced Search** link next to the Search box on the resulting page and then click the **Technical Support & Documentation** radio button.

To provide feedback about the Cisco.com website or a particular technical document, click **Contacts & Feedback** at the top of any Cisco.com web page.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests, or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411

Australia: 1 800 805 227

EMEA: +32 2 704 55 55

USA: 1 800 553 2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—An existing network is “down” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operations are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of the network is impaired while most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- The Cisco Online Subscription Center is the website where you can sign up for a variety of Cisco e-mail newsletters and other communications. Create a profile and then select the subscriptions that you would like to receive. To visit the Cisco Online Subscription Center, go to this URL:

<http://www.cisco.com/offer/subscribe>

- The *Cisco Product Quick Reference Guide* is a handy, compact reference tool that includes brief product overviews, key features, sample part numbers, and abbreviated technical specifications for many Cisco products that are sold through channel partners. It is updated twice a year and includes the latest Cisco channel product offerings. To order and find out more about the *Cisco Product Quick Reference Guide*, go to this URL:

<http://www.cisco.com/go/guide>

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:
<http://www.cisco.com/go/marketplace/>
- Cisco Press publishes a wide range of general networking, training, and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:
<http://www.ciscopress.com>
- *Internet Protocol Journal* is a quarterly journal published by Cisco for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:
<http://www.cisco.com/ipj>
- Networking products offered by Cisco, as well as customer support services, can be obtained at this URL:
<http://www.cisco.com/en/US/products/index.html>
- Networking Professionals Connection is an interactive website where networking professionals share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:
<http://www.cisco.com/discuss/networking>
- “What’s New in Cisco Documentation” is an online publication that provides information about the latest documentation releases for Cisco products. Updated monthly, this online publication is organized by product category to direct you quickly to the documentation for your products. You can view the latest release of “What’s New in Cisco Documentation” at this URL:
<http://www.cisco.com/univercd/cc/td/doc/abtnucd/136957.htm>
- World-class networking training is available from Cisco. You can view current offerings at this URL:
<http://www.cisco.com/en/US/learning/index.html>



CHAPTER 1

Cisco Trust Agent Overview

Cisco Trust Agent (CTA) is a component of the Cisco Network Admission Control (NAC) program. NAC enables network access devices to permit or deny network clients access to the network based on the posture of the software on the host. This process is called posture validation.

This chapter contains the following sections:

- [Cisco NAC Overview, page 1-1](#)
- [Posture Validation Process Overview, page 1-2](#)
 - [Initial Posture Validation Process, page 1-2](#)
 - [Posture Revalidation Process, page 1-4](#)
- [Cisco Trust Agent Deployment Considerations, page 1-5](#)

Cisco NAC Overview

The four main components of the Cisco Trust Agent (CTA) posture validation process are as follows:

- **Network client running Cisco Trust Agent**—Cisco Trust Agent collects security posture information from the NAC-compliant applications running on the network client and reports them to the Cisco Secure Access Control Server (ACS). These are some NAC-compliant applications:
 - Antivirus applications
 - Personal firewalls

- Host-based intrusion protection applications, such as Cisco Security Agent (CSA)
- **Network Access Device (NAD)**—The NAD permits or denies network access. Typically, the NAD is a Cisco router or switch.
- **Authentication Server (Cisco Secure Access Control Server (ACS))**—The authentication server is responsible for obtaining and evaluating the security posture credentials from a network client, determining the overall client posture, and providing the appropriate posture token and network access policy to the NAD based on the client's posture.
- **Posture Validation Servers (optional)**—Posture validation servers support Cisco Secure ACS in determining the overall system posture. They are typically third-party applications that support the validation of the security posture credentials for a specific NAC-compliant application. For example, an antivirus software company may prefer to maintain its own posture validation server rather than store and update posture validation information on Cisco's ACS server.

Posture Validation Process Overview

Initial Posture Validation Process

The following provides an overview of how the posture validation components work together during an initial posture validation process.

1. The computer sends a DHCP (Dynamic Host Configurative Protocol) request to obtain an IP address or an ARP (Address Resolution Protocol) request to convert an IP address into a physical address.
2. The Network Access Device (NAD), a Cisco network router or switch, requests a network access policy from the Cisco Secure Access Control Server (ACS).
3. ACS requests the computer's posture credentials from Cisco Trust Agent (CTA) which was previously installed.
4. CTA receives the security posture credential request and, in turn, prompts the posture plugins installed on the computer to gather the posture credentials from the NAC-compliant applications on the client.

5. CTA aggregates all of the posture credentials from the client and returns the information to the NAD.
 - If the posture credentials are sent using the EAP over UDP protocol, the posture information is sent directly from the CTA to ACS.
 - If the posture credentials are sent using the IEEE 802.1x protocol, CTA hands the posture credentials to the Cisco Trust Agent 802.1x Wired Client, also known as the “supplicant” and the CTA 802.1x Wired Client forwards the information to the ACS. (See [Chapter 9, “Cisco Trust Agent 802.1x Wired Client”](#) for more information on the supplicant.)
6. ACS evaluates the security posture credentials for each application that resides on the computer. ACS can perform the credential evaluation using rules in the local database or can relay application credentials to an application-specific posture validation server for evaluation. The result of the evaluation is an application posture token for each evaluated application and an optional user notification.
7. ACS aggregates the application posture credentials and defines an overall system posture token for the client. The system posture token equals the least trusted posture of all the application posture credentials collected from the client.

These are the default system posture tokens; they are ranked from the most trusted posture to the least trusted posture:

- Healthy
 - Checkup
 - Quarantine
 - Transition
 - Infected
 - Unknown
8. ACS maps the system posture token to a network access policy an, optionally, a user notification.
 9. ACS sends the result of the security posture validation back to the NAD, along with the appropriate network access policy for that client, and any user notification, back to the NAD.

10. The NAD implements the security policy for the client and forwards the posture information back to CTA on the computer. Depending upon how you have configured CTA, the results of the posture validation are logged and any user notifications are displayed on-screen in a dialog box.

Based on the access policy, the network client is permitted on the network, denied access to the network, or quarantined to a remediation network until the NAC-compliant applications have been updated to the required levels.

Posture Revalidation Process

Posture revalidation is caused by one of three events and once it occurs posture is validated following the same workflow as described in [Posture Validation Process Overview](#), starting with step 2.

Network Access Device Requests New Posture. In the case of NAC L2 IP or NAC L3 IP network admission methods, switches and routers maintain the posture for a configured amount of time and then that posture expires. When the posture expires, the NAD requests posture again.

An Internal Timer in CTA Expires. The SQTimer parameter defines the interval at which CTA requests the posture plugins for their status. If a posture plugin reports a change in posture status to CTA, CTA alerts the network access device, which triggers a re-posturing of the host. This is one implementation of the asynchronous posture status query feature. This feature is only available if CTA 802.1x Wired Client is installed on the host machine.

See “[Configuring Asynchronous Posture Status Query](#)” section on page 5-19 for more information on how to configure the SQTimer.

Posture Plugins Detect a Change in Status. Some posture plugins monitor the status of their applications and report status changes to CTA upon detection. Such plugins are considered “asynchronous” plugins. When CTA receives the status change from an asynchronous plugin, CTA alerts the network access device, which triggers a re-posturing of the host. For example, the posture plugin for Cisco Security Agent (CSA) detects when the CSA security has been turned off. This is one implementation of the asynchronous posture status query feature. This feature is only available if CTA 802.1x Wired Client is installed on the host machine.

Cisco Trust Agent Deployment Considerations

Network clients are the hosts on your network. This includes PCs, laptops, workstations, and servers. You may have hundreds or thousands of network clients on which you are going to install CTA. Installing and configuring CTA on that many network clients can be a time-consuming process.

To decrease the time spent installing and configuring CTA throughout your network, you can create a custom installation package. The custom installation package lets you provision the CTA settings at installation time. These settings include:

- Communication settings
- Posture notification settings
- Logging and notification setting
- Authentication policies for the CTA 802.1x Wired Client



CHAPTER 2

Installing the Cisco Trust Agent on Linux Operating Systems

This chapter provides system requirement and installation information for installing Cisco Trust Agent (CTA) on Red Hat Linux operating systems. Read this entire chapter before beginning the installation. There are advanced installation options detailed later in this chapter that you may want to use. For example, before deploying CTA on your network, you can create a custom installation package which allows you to set CTA configuration parameters and provision certificates and plug-ins. Proceeding in this manner could save you configuration time during the CTA deployment process.

See these other chapters for installation instructions for different operating systems:

- [Chapter 3, “Installing the Cisco Trust Agent on Macintosh Operating Systems”](#)
- [Chapter 4, “Installing the Cisco Trust Agent on Windows Operating Systems”](#)

This chapter contains the following sections:

- [Verifying System Requirements on Linux, page 2-2](#)
- [Installation Files, page 2-3](#)
- [Initial Deployments, page 2-3](#)
- [CTA Scripting Interface Feature, page 2-3](#)
- [Installing Cisco Trust Agent, page 2-4](#)
 - [General Installation Instructions, page 2-4](#)

- [Extracting the Installation File and Accepting the EULA, page 2-4](#)
- [Installing CTA from the Command Line, page 2-5](#)
- [Creating a Custom CTA Installation Package, page 2-5](#)
- [Upgrading to Cisco Trust Agent, Release 2.1, page 2-7](#)
- [Verifying Cisco Trust Agent Installation on Linux, page 2-8](#)
 - [Verifying CTA is Running, page 2-8](#)
 - [Verifying CTA Package Information, page 2-9](#)
- [Uninstalling Cisco Trust Agent on Linux, page 2-9](#)

Verifying System Requirements on Linux

Before installing Cisco Trust Agent on a Linux operating system, verify that the target system meets the following requirements:

System Component	Requirement
System	<ul style="list-style-type: none"> • Pentium class processor or better • Network connection
Operating System and Language Support	<p>All available internationalized versions of these Linux operating systems support CTA 2.1.:</p> <ul style="list-style-type: none"> • Red Hat Linux v9 • Red Hat Enterprise Linux v3 (Enterprise, Advanced Server, and Workstation) • Red Hat Enterprise Linux v4 (Enterprise, Advanced Server, and Workstation) <p>Note Support for a localized operating system is different from localized version of CTA. The CTA interface and messages are presented in English.</p>
Linux Installers	Red Hat Package Management (RPM) v4.2 or greater.
Hard Disk Space	20 MB

System Component	Requirement
Memory	256 MB Red Hat Enterprise Linux v3 (Enterprise, Advanced, Workstation)
	256 MB Red Hat Enterprise Linux v4 (Enterprise, Advanced, Workstation)
Listening Port	By default, Cisco Trust Agent listens on UDP port 21862. You can change this port number. See, “The ctad.ini Configuration File” section on page 5-2 for more information.

Installation Files

The installation files for CTA for Linux are contained in the **ctaadminex-linux-2.1.103-0.tar.gz** file. That file may be downloaded from Cisco.com. Follow the procedures in [“Installing Cisco Trust Agent”](#) to use the file.

Initial Deployments

For large enterprise Linux deployments, administrators may want to deploy CTA with a customized package. This way all required certificates and plug-ins are administratively configured with no end-user interaction or interference. The customized packages can be delivered to many users at once using an automated software deployment mechanism.

CTA Scripting Interface Feature

The Scripting Interface feature allows software developers to write their own scripts to relay posture information, collected on the system, to CTA. The scripts would perform the equivalent function of a posture plugin. Users will not need this feature unless they intend to write posture scripts.

The Scripting Interface is installed by default on Linux installations.

Installing Cisco Trust Agent

In order to install Cisco Trust Agent you log in as the administrative user on the computer.

General Installation Instructions

-
- Step 1** Download the ctaadminex-linux-2.1.103-0.tar.gz from Cisco.com. For the sake of this example, we will store the ctaadminex-linux-2.1.103-0.tar.gz file in /tmp.
- Step 2** Follow the procedures in [“Extracting the Installation File and Accepting the EULA” section on page 2-4](#).
- Step 3** Install CTA using either of these methods:
- [Installing CTA from the Command Line, page 2-5](#)
 - [Creating a Custom CTA Installation Package, page 2-5](#)

Extracting the Installation File and Accepting the EULA

After downloading the ctaadminex-linux-2.1.103-0.tar.gz file from Cisco.com, use this procedure to extract the CTA installation files and accept the end-user license agreement (EULA).

-
- Step 1** Open a terminal window.
- Step 2** Change the directory to the one that contains the ctaadminex-linux-2.1.103-0.tar.gz file. In our example, this directory is /tmp.
- Step 3** At the prompt, type the following command and press <Enter>.
- ```
tar -zxvf ctaadminex-linux-2.1.103-0.tar.gz
```
- The ctaadminex-linux-2.1.103-0.sh file is extracted and placed in the same directory as the ctaadminex-linux-2.1.103-0.tar.gz file.
- Step 4** At the prompt, type **./ctaadminex-linux-2.1.103-0.sh** and press <Enter>.

- Step 5** When prompted, accept the EULA by typing “y” and pressing <Enter>. The CTA-2.1.103-0 subdirectory is created and the cta-linux-2.1.103-0.i386.rpm is unpacked and placed in that directory. In our example, this new directory would be /tmp/CTA-2.1.103-0.

## Installing CTA from the Command Line

- Step 1** Follow the procedure in the [“Extracting the Installation File and Accepting the EULA” section on page 2-4](#).
- Step 2** Open a terminal window on the client.
- Step 3** Change the directory to the directory that contains the cta-linux-2.1.103-0.i386.rpm file. In our example, the directory is /tmp/CTA-2.1.103-0.
- Step 4** At the prompt, type **rpm -ivh cta-linux-2.1.103-0.i386.rpm** and press <Enter>. You receive these messages indicating that CTA was installed.
- ```
Preparing... ##### 100%
1:cta-linux ##### 100%
```
- Step 5** Verify CTA installation using the procedures in [“Verifying Cisco Trust Agent Installation on Linux” section on page 2-8](#).
- Step 6** If a CA certificate or a matching root certificate from the Cisco Secure ACS server has not been installed on the network client on which you just installed CTA, you must install one or the other certificates. This enables CTA to establish a secure form of communication with the Cisco Secure ACS server. Refer to [“Installing or Updating a Certificate on Linux Operating Systems, page 8-4”](#) for information.

Creating a Custom CTA Installation Package

Use this section as an example of how to create a customized CTA installation on Linux systems.

The CTA installation file is a Red Hat Packet Manager (rpm) file and is installed with standard RPM commands. To create a custom installation package, you create a directory structure which includes the CTA installation file, .ini files,

plugin subdirectory and certificate subdirectory. This directory structure can then be distributed by a software deployment mechanism, such as a script or a software deployment tool.

After the software deployment mechanism distributes the directory structure to the remote network clients, it runs the CTA installation file. The CTA installation file copies the contents of the directory structure to the proper locations on the remote network client. The software deployment mechanism does not need to recompile the CTA installation file to create a custom installation.

The customization choices in this procedure are optional. However, you will find that including some of these customizations is worthwhile. CTA is not a centrally managed product. If you do not plan to use the product defaults, it is to your benefit to pre-configure all available product settings before deploying CTA.

Please read this entire procedure before beginning. There are options detailed later in the instructions that you should be aware of before beginning.

-
- Step 1** Before you create the custom installation package, install CTA on the client you will use to create the custom package. This will install the template ctad.ini file and give you exposure to the CTA installation process. Begin with the [“General Installation Instructions” section on page 2-4](#) to install CTA.
- Step 2** Perform the procedure in the [“Extracting the Installation File and Accepting the EULA” section on page 2-4](#).
- Step 3** Change the directory to the one that contains the **cta-linux-2.1.103-0.i386.rpm** file. In our example, this is the /tmp/CTA-2.1.103-0 directory.
- Step 4** Create a **certs** subdirectory. For example: /tmp/CTA-2.1.103-0/**certs**
- Copy the root certificate for your Cisco Secure ACS server to this directory. During installation, any certificates in this directory are added to the systems root certificate store.
- If your Cisco Secure ACS server uses self-signed certificates, see the Cisco Secure ACS documentation for information about obtaining the certificate; if you use a CA server, refer to your CA server documentation.

**Note**

This step is optional if a CA certificate or ACS root certificate have already been distributed to the network clients receiving this customized CTA installation. If these certificates have not been distributed, this step is required.

- Step 5** Create a **plugins** subdirectory. For example: /tmp/CTA-2.1.103-0/**plugins**
- Copy any third party plugins that you want to provision at installation time into this directory.
- Step 6** Create a new **ctad.ini** file and store it in the installation directory at the same level as the **cta-linux-2.1.103-0.i386.rpm** file. In our example, this is the /tmp/CTA-2.1.103-0 directory.
- The ctad.ini file is used to configure CTA communication settings, user notifications, and certificate validation rules. If you want to change the default communication settings, such as the port number CTA listens over, the maximum number of sessions, and session time-out values, include this file. Refer to [Chapter 5, “Configuring Cisco Trust Agent”](#) for instructions on how you should create and format this file.
- Step 7** Create a new ctalogd.ini file and store it in the installation directory at the same level as the **cta-linux-2.1.103-0.i386.rpm** file. In our example, this is the /tmp/CTA-2.1.103-0 directory. Refer to [Chapter 6, “Cisco Trust Agent Event Logging”](#) for instructions on how you should create and format this file.
- Step 8** A software deployment mechanism deploys the customized /tmp/CTA-2.1.103-0 directory to the appropriate network clients and saves it in a local directory.
- Step 9** The software deployment mechanism installs CTA and its customizations by following the procedure in the [“Installing CTA from the Command Line”](#) section on page 2-5.

Upgrading to Cisco Trust Agent, Release 2.1

Use this procedure to upgrade your CTA installation:

- Step 1** Download the ctaadminex-linux-2.1.103-0.tar.gz from Cisco.com. For the sake of this example, we will store the ctaadminex-linux-2.1.103-0.tar.gz file in /tmp.

- Step 2** Follow the procedures in “[Extracting the Installation File and Accepting the EULA](#)” section on page 2-4.
- Step 3** Install CTA using either of these methods:
- [Installing CTA from the Command Line, page 2-5](#). However, when you are ready to install the upgrade, change the installation command to the upgrade command. This is an example:
rpm -Uvh cta-linux-2.1.103-0.i386
 - [Creating a Custom CTA Installation Package, page 2-5](#)

Verifying Cisco Trust Agent Installation on Linux

After Cisco Trust Agent has been installed you will find the following directory structures containing CTA’s executable files:

- /opt/CiscoTrustAgent
- /opt/PostureAgent

You may also verify which version of CTA is installed by following this procedure:

-
- Step 1** Open a terminal window on the system.
- Step 2** Type **rpm -q cta-linux** and press <Enter>.
- The version of CTA is returned. In the case of CTA 2.1.103.0, this information will be returned: **cta-linux-2.1.103-0**.

Verifying CTA is Running

-
- Step 1** Open a terminal window on the system.
- Step 2** Type **rpm -q cta-linux** and press <Enter>.
- Step 3** Type **ps -A | grep cta** and press <Enter>.
- Step 4** Verify that the following daemons are running:
- ctad

- ctalogd
- ctapsd
- ctaeoud

If these daemons are not running, try rebooting the system. If the daemons still do not run, try reinstalling the application.

Verifying CTA Package Information

-
- Step 1** Open a terminal window on the system.
- Step 2** At any prompt, type **rpm -q cta-linux** and press <Enter>.
- The full package name is returned, for example, **cta-linux-2.1.103-0**

Uninstalling Cisco Trust Agent on Linux

To uninstall Cisco Trust Agent, follow this procedure:

-
- Step 1** Log in to the client as the root user.
- Step 2** Open a terminal window.
- Step 3** At the prompt, run the following command and press <Enter>.
- #rpm -e cta-linux**
- Cisco Trust Agent is uninstalled. You do not need to reboot the system.



Note

Certificates and plugin files are not deleted when CTA is uninstalled; they remain on the client.



CHAPTER 3

Installing the Cisco Trust Agent on Macintosh Operating Systems

This chapter provides system requirement and installation information for installing Cisco Trust Agent (CTA) on Macintosh operating systems. Read this entire chapter before beginning the installation. There are advanced installation options detailed later in this chapter that you may want to use. For example, before deploying CTA on your network, you can create a custom installation package which allows you to set CTA configuration parameters and install certificates and plug-ins. Proceeding in this manner could save you configuration time during the CTA deployment process.

See these other chapters for installation instructions for different operating systems:

- [Chapter 2, “Installing the Cisco Trust Agent on Linux Operating Systems”](#)
- [Chapter 4, “Installing the Cisco Trust Agent on Windows Operating Systems”](#)

This chapter contains the following sections:

- [System Requirements for Mac OS X, page 3-2](#)
- [CTA Scripting Interface Feature, page 3-3](#)
- [Installation Files, page 3-3](#)
- [Installing Cisco Trust Agent, page 3-3](#)
 - [General Installation Instructions, page 3-3](#)
 - [Extracting the Installation Image and Accepting the EULA, page 3-4](#)
 - [Installing CTA from the Command Line, page 3-4](#)

- Installing CTA Using an Installation Wizard, page 3-6
- Installing CTA Using a Custom Installation Package, page 3-12
- Repairing or Upgrading an Existing CTA Installation, page 3-14
- Verifying Cisco Trust Agent Installation, page 3-16
- Uninstalling Cisco Trust Agent, page 3-16
 - Uninstalling CTA and the CTA Scripting Interface, page 3-16
 - Uninstalling Only the CTA Scripting Interface, page 3-17

System Requirements for Mac OS X

Before installing Cisco Trust Agent on a Mac OS X Operating system, verify that the target system meets the following requirements:

System Component	Requirement
System	<ul style="list-style-type: none"> • G3 processor and later • Network connection
Free Hard Disk Space	20 MB minimum
Memory	256 MB RAM
Listening Port	<p>By default, Cisco Trust Agent listens on UDP port 21862.</p> <p>You can change this port number, if necessary. See “The ctad.ini Configuration File” section on page 5-2.</p>
Operating System and Language Support	<p>All available internationalized versions of Mac OS X 10.3.9 and 10.4 operating systems support CTA 2.1.</p> <p>Note Support for a localized operating system is different from localized version of CTA. The CTA interface and messages are presented in English.</p>

CTA Scripting Interface Feature

The Scripting Interface feature allows software developers to write their own scripts to relay posture information, collected on the system, to CTA. The scripts would perform the equivalent function of a posture plugin. Users will not need this feature unless they intend to write posture scripts.

The Scripting Interface is an optional Feature of CTA.

Installation Files

The installation files for CTA for Mac are contained in the **ctaadminex-darwin-2.1.103.0.tar.gz** file. That file may be downloaded from Cisco.com. Follow the procedures in [“Installing Cisco Trust Agent”](#) for descriptions of the files contained in the **ctaadminex-darwin-2.1.103.0.tar.gz** file and their uses.

Installing Cisco Trust Agent

In order to install Cisco Trust Agent you log in as the administrative user on the machine.

General Installation Instructions

-
- Step 1** Download the ctaadminex-darwin-2.1.103.0.tar.gz from Cisco.com and follow the procedures in [“Extracting the Installation Image and Accepting the EULA”](#) section on page 3-4.
- Step 2** Install CTA using any of these methods:
- [Installing CTA from the Command Line, page 3-4](#)
 - [Installing CTA Using an Installation Wizard, page 3-6](#)
 - [Installing CTA Using a Custom Installation Package, page 3-12](#)

- Step 3** Install Cisco Secure Access Control Server (ACS) root certificate on the end-point if it is not distributed as part of a custom installation package. See [“About The ACS Server Root Certificate” section on page 8-3](#) for information about installing this certificate separately.
- Step 4** Verify CTA installation using the [“Verifying Cisco Trust Agent Installation” section on page 3-16](#).

Extracting the Installation Image and Accepting the EULA

After downloading the `ctaadminex-darwin-2.1.103.0.tar.gz` file from Cisco.com, use this procedure to extract the CTA installation files and accept the end-user license agreement (EULA).

-
- Step 1** Open a terminal window.
- Step 2** Using the `CD` command, change the directory to that which contains the `ctaadminex-darwin-2.1.103.0.tar.gz` file.
- Step 3** At the prompt, type:
`tar zxvf ctaadminex-darwin-2.1.103.0.tar.gz`
and press **<Return>**. The `ctaadminex.sh` file is extracted and placed in the same directory as the `ctaadminex-darwin-2.1.103.0.tar.gz` file.
- Step 4** At the prompt, type **`./ctaadminex.sh`** and press **<Return>**.
- Step 5** When prompted, accept the EULA agreement by entering **“y”** and pressing **<Return>**. The `cta-darwin-2.1.103.0.dmg` is unpacked and placed in the same directory as the `ctaadminex.sh` file.
- Step 6** At the prompt, type **`open cta-darwin-2.1.103.0.dmg`** and press **<Return>**. The CiscoTrustAgent volume icon is placed on the desktop and the `cta-darwin-2.1.103.0.dmg` disk image is visible in Finder.

Installing CTA from the Command Line

-
- Step 1** Follow the procedure in the [“Extracting the Installation Image and Accepting the EULA” section on page 3-4](#).
- Step 2** Open a terminal window on the end point.

- Step 3** At the prompt, use the CD command to change the directory to /Volumes/CiscoTrustAgent.
- Step 4** To install CTA, at the prompt, on one line, type the following:
- ```
sudo installer -verbose -pkg
/Volumes/CiscoTrustAgent/CiscoTrustAgent.mpkg/ -target
/Volumes/Macintosh\ /HD
```
- and press <Return>.
- Step 5** When prompted, enter the Administrative user's password. CTA is installed. At the end of a successful installation you will see the message, "The install was successful."
- Step 6** (Optional) To install Cisco Trust Agent Scripting Interface feature, at the prompt, on one line, type the following:
- ```
sudo installer -verbose -pkg  
/Volumes/CiscoTrustAgent/CiscoTrustAgent.mpkg/CiscoTrustAgentSI.  
pkg/ -target /Volumes/Macintosh\ /HD
```
- and press <Return>. At the end of a successful installation you will see the message, "The install was successful."
- Step 7** Verify the installation of CTA using the ["Verifying Cisco Trust Agent Installation" section on page 3-16](#).
- Step 8** Exit the /Volumes/CiscoTrustAgent directory by typing **CD ..** at the prompt and pressing <Return>.
- Step 9** At the prompt, type **hdiutil detach /Volumes/CiscoTrustAgent** and press <Return>. You will receive messages in the terminal window that the volume was unmounted and ejected. Also the CiscoTrustAgent volume icon will be removed.
- Step 10** If a CA certificate or a matching root certificate from the Cisco Secure ACS server has not been installed on the network client on which you just installed CTA, you must install one or the other certificates. This enables CTA to establish a secure form of communication with the Cisco Secure ACS server. Refer to the ["Installing or Updating a Certificate on Mac OS X Operating System" section on page 8-4](#) for more information.

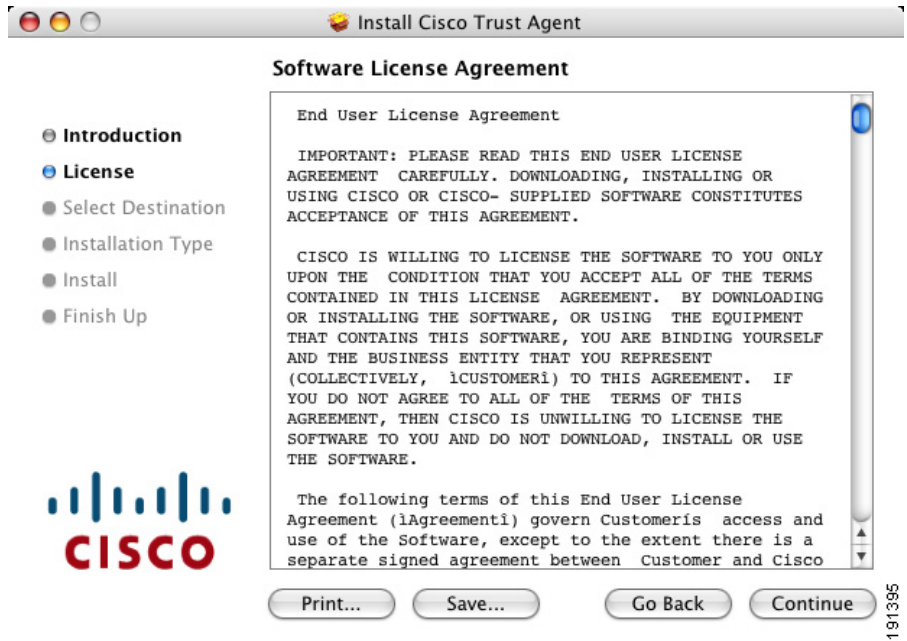
Installing CTA Using an Installation Wizard

- Step 1** Follow the procedure in the [“Extracting the Installation Image and Accepting the EULA”](#) section on page 3-4.
- Step 2** Double-click the **CiscoTrustAgent** volume icon on the desktop.
- Step 3** Double-click the **CiscoTrustAgent.mpkg** icon in the Finder window that opens.
- Step 4** Click **Continue** in the Welcome window.



191397

Step 5 Click **Continue** in the Software License Agreement window.



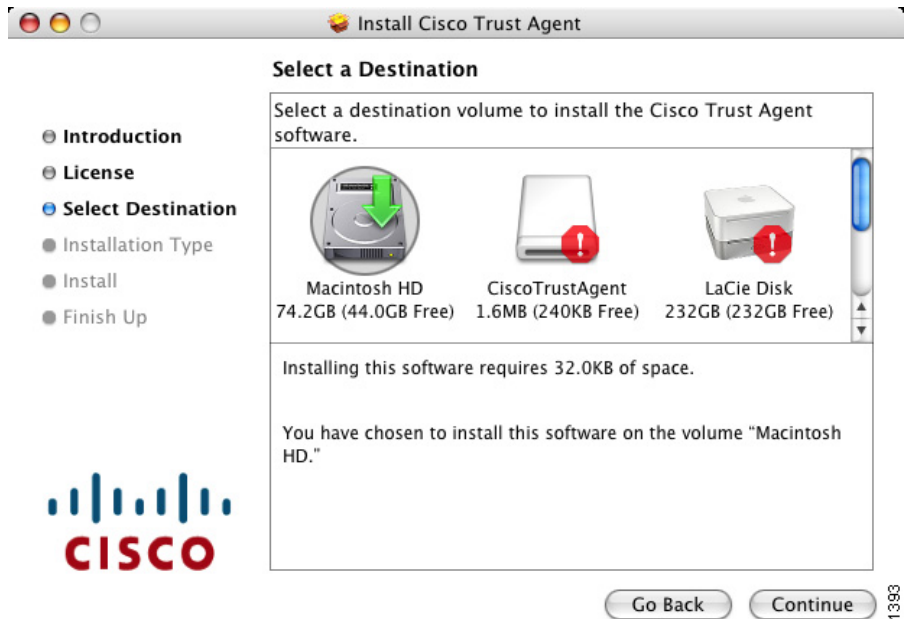
Step 6 Click **Agree** to accept the license for Cisco Trust Agent.



Step 7 Select the drive on which you want to install CTA in the Select a Destination window.

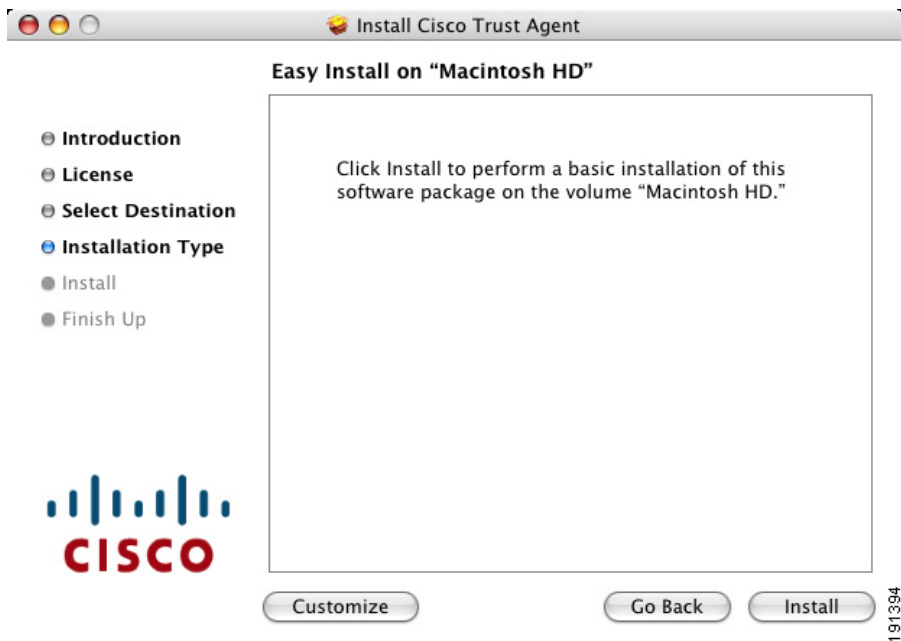
**Note**

You may not install CTA on drives or disk images that display the red exclamation mark symbol.



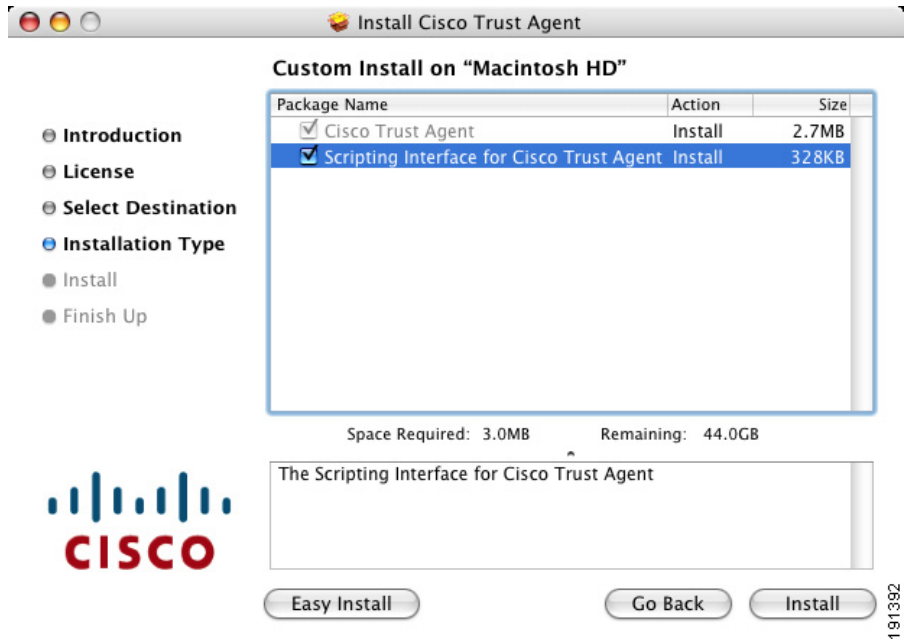
191393

Step 8 Click **Continue**. The Easy Install window opens.



Step 9 Click **Customize** to install the Scripting Interface or skip to [Step 11](#).

- Step 10** In the Custom Install window, check the box next to **Scripting Interface for Cisco Trust Agent** to install that feature.

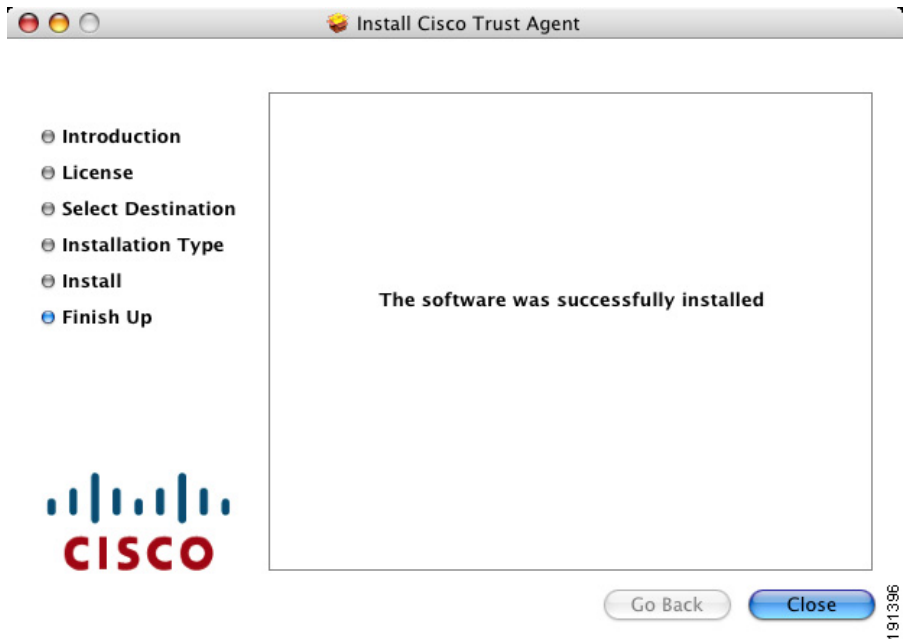


- Step 11** Click **Install** to continue with the installation.

Step 12 Type the Administrator's password when prompted and click **OK**.



Step 13 After CTA has been installed you receive the message, "The software was successfully installed."



Step 14 Click Close.

Step 15 Eject the CiscoTrustAgent volume on the desktop by dragging its icon to the trash.

Installing CTA Using a Custom Installation Package

Use this section as an example of how to create a customized CTA installation on Macintosh systems.

The cta-darwin-2.1.103.0.dmg disk image contains the CiscoTrustAgent.mpkg package which you use to install Cisco Trust Agent. The cta-darwin-2.1.103.0.dmg disk image can be modified by adding configuration .ini files, posture plugins, and certificate files to it. This customized Cisco Trust Agent package can be distributed by a software deployment mechanism, such as a script or a software deployment tool.

After the software deployment mechanism distributes the customized cta-darwin-2.1.103.0.dmg disk image to the remote network clients, it runs the CiscoTrustAgent.mpkg package. The CTA installation file copies the contents of the disk image to the proper locations on the remote network client. The software deployment mechanism does not need to recompile the CTA installation file to create a custom installation.

The customization choices in this procedure are optional. However, you will find that including some of these customizations is worthwhile. CTA is not a centrally managed product. If you do not plan to use the product defaults, it is to your benefit to pre-configure all available product settings before deploying CTA.

Please read this entire procedure before beginning. There are options detailed later in the instructions that you should be aware of before beginning.

Creating a Custom Installation Package

Step 1 Before you create the custom installation package, install CTA on the client you will use to create the custom package. This will install the template ctad.ini file and give you exposure to the CTA installation process. Use the [“Installing CTA Using an Installation Wizard”](#) section on page 3-6 to install CTA.

Step 2 Use the procedure in [“Extracting the Installation Image and Accepting the EULA” section on page 3-4](#) to extract the cta-darwin-2.1.103.0.dmg disk image and accept the EULA for all users.

Step 3 Double-click the **CiscoTrustAgent** volume on the desktop.

Step 4 Customize the **/Volumes/CiscoTrustAgent/certs** directory by copying the root certificate for your Cisco Secure ACS server, or other certificates, to this directory. **During installation, any certificates in this directory are added to the systems root certificate store.**

If your Cisco Secure ACS server uses self-signed certificates, see the Cisco Secure ACS documentation for information about obtaining the certificate; if you use a CA server, refer to your CA server documentation.



Note This step is optional if a CA certificate or ACS root certificate have already been distributed to the network clients receiving this customized CTA installation. If these certificates have not been distributed, and you choose not to add them to the customized disk image, you will need to distribute and install either the CA certificate or ACS root certificate before the client can communicate with the NAC infrastructure.

Step 5 Customize the **/Volumes/CiscoTrustAgent/plugins** directory by copying third party plugins that you want to provision at installation time into this directory.

Step 6 Create a new **ctad.ini** file and copy it into the **/Volumes/CiscoTrustAgent** directory at the same level as the **CiscoTrustAgent.mpkg** package. This file is used to configure CTA communication settings, user notifications, and certificate validation rules. If you want to change the default communication settings, such as the port number CTA listens over, the maximum number of sessions, or session time-out values, include this file. Refer to [Chapter 5, “Configuring Cisco Trust Agent”](#) for instructions on how you should create and format this file.

Step 7 Create a new **ctalogd.ini** file and copy it to the **/Volumes/CiscoTrustAgent** directory at the same level as the **CiscoTrustAgent.mpkg** package. This file is used to enable and disable CTA logging. Refer to [Chapter 6, “Cisco Trust Agent Event Logging”](#) for instructions on how you should create and format this file.

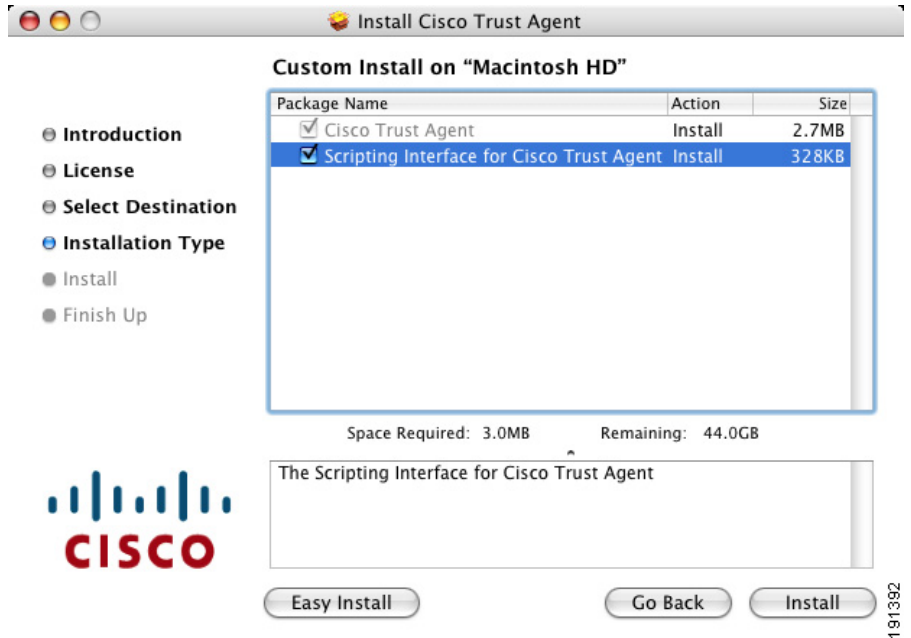
Step 8 A software deployment mechanism deploys the custom-cta-darwin-2.1.103.0.dmg disk image to the appropriate network clients and saves it in a local directory.

- Step 9** The software deployment mechanism installs CTA and its customizations by following the [“Installing CTA from the Command Line”](#) section on page 3-4.

Repairing or Upgrading an Existing CTA Installation

Use this procedure to reinstall CTA or to add the CTA Scripting Interface to an existing CTA installation.

-
- Step 1** Use the procedure in [“Extracting the Installation Image and Accepting the EULA”](#) section on page 3-4 to extract the cta-darwin-2.1.103.0.dmg disk image and accept the EULA.
- Step 2** Open Finder and navigate to the directory where cta-darwin-2.1.103.0.dmg is stored.
- Step 3** Double-click **cta-darwin-2.1.103.0.dmg**. The CiscoTrustAgent volume icon appears on the desktop.
- Step 4** Double-click the **CiscoTrustAgent** volume icon.
- Step 5** Double-click the **CiscoTrustAgent.mpkg** icon in the Finder window that opens.
- Step 6** Click **Continue** in the Welcome window.
- Step 7** Click **Continue** in the Software License Agreement window.
- Step 8** Click **Agree** to accept the license for Cisco Trust Agent.
- Step 9** **Select the drive** on which you want to install CTA in the Select a Destination window.
- Step 10** Click **Continue**. The Easy Install window opens.
- Step 11** To install the Scripting Interface feature, click **Customize** or skip to step [Step 13](#).



- Step 12** In the Custom Install window, check the box next to Scripting Interface for Cisco Trust Agent to install that feature.
- Step 13** Click **Upgrade** to continue with the easy installation of the window.
- Step 14** Type the Administrator's password when prompted and click **OK**.
- Step 15** After CTA has been installed you receive the message, "The software was successfully installed."
- Step 16** Click **Close**.
- Step 17** Eject the CiscoTrustAgent volume on the desktop by dragging its icon to the trash.

Verifying Cisco Trust Agent Installation

After Cisco Trust Agent has been installed you will find the following directory structures containing CTA's executable files:

- /opt/CiscoTrustAgent
- /opt/PostureAgent

To verify that the Cisco Trust Agent is running, follow this procedure:

-
- Step 1** Open a terminal window on the system.
- Step 2** Type `ps -ax | grep cta` and press <Enter>.
- Step 3** Verify that the following daemons are running:
- /opt/CiscoTrustAgent/sbin/ctad
 - ctalogd
 - ctapsd
 - ctaeoud

If these daemons are not running, try rebooting the system. If the daemons still do not run, try reinstalling the application.

Uninstalling Cisco Trust Agent

There are two uninstallation procedures for CTA. You may uninstall CTA and the Scripting Interface or you may uninstall the Scripting Interface alone.

Uninstalling CTA and the CTA Scripting Interface

-
- Step 1** Open a terminal window on the target system.
- Step 2** At the prompt, type `CD /opt/CiscoTrustAgent` and press <Return>.
- Step 3** At the prompt, run the following command:
- ```
sudo ./cta_uninstall.sh
```
- Step 4** Enter the Administrative user's password when prompted.



- Step 5** To uninstall CTA and the CTA Scripting Interface without further prompting, type **y** when prompted.

After Cisco Trust Agent has been successfully uninstalled, you receive the message, “Cisco Trust Agent has been successfully uninstalled.”

**Note**

Certificates, plugin files, and customized configuration files are not deleted when CTA is uninstalled; they remain on the client.

## Uninstalling Only the CTA Scripting Interface

- Step 1** Open a terminal window on the target system.
- Step 2** At the prompt, type **CD /opt/CiscoTrustAgent** and press <Return>.
- Step 3** At the prompt, run the following command:
- ```
sudo ./cta_uninstall.sh --SI
```
- Step 4** Enter the Administrative user's password when prompted.
- Step 5** When prompted, type **y** to uninstall Cisco Trust Agent.
- Step 6** After a successful uninstallation, you receive the message, “Cisco Trust Agent Scripting Interface has been successfully uninstalled.”



CHAPTER 4

Installing the Cisco Trust Agent on Windows Operating Systems

This chapter provides system requirement and installation information for installing Cisco Trust Agent (CTA) on Windows operating systems. Read this entire chapter before beginning the installation. There are advanced installation options detailed later in this chapter that you may want to use. For example, before deploying CTA on your network, you can create a custom installation package which allows you to set CTA configuration parameters and provision certificates, and posture plug-ins. Proceeding in this manner could save you configuration time during the CTA deployment process.

See these other chapters for installation instructions for different operating systems:

- [Chapter 2, “Installing the Cisco Trust Agent on Linux Operating Systems.”](#)
- [Chapter 3, “Installing the Cisco Trust Agent on Macintosh Operating Systems.”](#)

This chapter contains the following sections:

- [System Requirements for Installation, page 4-2](#)
- [Optional Features You Can Install with CTA, page 4-4](#)
 - [CTA 802.1x Wired Client, page 4-4](#)
 - [CTA Scripting Interface, page 4-4](#)
- [Installation Files, page 4-5](#)
- [Installing Cisco Trust Agent, page 4-5](#)
 - [General Installation Instructions, page 4-6](#)

- Installing CTA Using MSI Commands, page 4-7
- Installing CTA Using an Installation Wizard, page 4-13
- Installing CTA Using a Custom Installation Package, page 4-22
- Upgrading to Cisco Trust Agent, Release 2.1, page 4-27
 - Upgrading from Cisco Trust Agent, Release 1.0, page 4-27
 - Upgrading from Cisco Trust Agent, Release 2.0.0.30, page 4-28
 - Upgrading from Cisco Trust Agent, Release 2.0.1, page 4-29
 - Upgrading from CTA 2.1 Selective Availability and Beta Releases to CTA 2.1.103.0, page 4-29
- Verifying the Cisco Trust Agent Installation, page 4-30
- Uninstalling Cisco Trust Agent on Windows, page 4-31

System Requirements for Installation

Before installing Cisco Trust Agent on a Windows operating system, verify that the target system meets the requirements in [Table 4-1](#).



Note

CTA 2.1 does not support Windows NT 4.0 Server or Windows NT 4.0 Workstation. CTA 2.0 was the last release to support Windows NT 4.0.

Table 4-1 CTA System Requirements

System Component	Requirement
System	<ul style="list-style-type: none">• Pentium II class processor or better• Network connection
Windows Installer (MSI)	Version 2.0 or later.
Free Hard Disk Space	20 MB minimum
Memory	256 MB of RAM
Listening Port	By default, Cisco Trust Agent listens on UDP port 21862. You can change this port number, if necessary. See the “ Configuring EAP over UDP Communication ” section on page 5-12.

System Component	Requirement
Windows Operating Systems on which CTA 2.1 and the CTA 802.1x Wired Client Run	<ul style="list-style-type: none">• Windows 2000 Professional and Advanced Server, SP4 and Update Rollup 1• Windows XP Professional, SP1, SP2, and SP3• Windows 2003 Standard, SP1 and R2
Additional Windows operating systems on which CTA 2.1 runs but that do not support CTA 802.1x Wired Client	Windows XP Home, SP1, SP2, and SP3
Language Support for localized operating systems	<p>All available localized versions of these operating systems support this release of CTA.</p> <p>Note Support for a localized operating system is different from localized version of CTA. The CTA interface and messages are presented in English.</p> <ul style="list-style-type: none">• Windows 2000 Professional and Advanced Server, SP4 and Update Rollup 1• Windows XP Professional, SP1, SP2, and SP3• Windows XP Home, SP1, SP2, and SP3• Windows 2003 Standard, SP1 and R2

Optional Features You Can Install with CTA

Windows provides several options for packaging and deploying Cisco Trust Agent. CTA may be packaged with or without the Cisco Trust Agent 802.1x Wired Client (802.1x Wired Client) and Scripting Interface features.

CTA 802.1x Wired Client

CTA can be installed with or without the CTA 802.1x Wired Client feature. The 802.1x Wired Client is CTA's "supplicant." The 802.1x Wired Client sends posture and authentication information, collected by CTA, over the IEEE 802.1x transport protocol through 802.1x-enabled access devices (the Ethernet switch) to the Cisco Secure Access Control Server (ACS). Only after successful client-server authentication will the port access control on the Ethernet switch allow the end-user to connect to the network.

If the NAC deployment in your enterprise uses network routers, or if your network switches communicate with CTA using the EAPoverUDP protocol, you do not need to install CTA with the 802.1x Wired Client.

**Note**

The 802.1x Wired Client is only available for Windows installations and it only supports wired network access.

For more information about the CTA 802.1x Wired client, see [Chapter 9, "Cisco Trust Agent 802.1x Wired Client"](#).

CTA Scripting Interface

The Scripting Interface feature allows software developers to write their own scripts to relay posture information, collected on the system, to CTA. The scripts would perform the equivalent function of a posture plugin. Users will not need this feature unless they intend to write posture scripts.

The Scripting Interface is an optional Feature of CTA.

Installation Files

These are the two installation files for the CTA 2.1.103.0 release for Windows:

- CtaAdminEx-win-2.1.103.0.exe
- CtaAdminEx-supPLICANT-win-2.1.103.0.exe

**Note**

These files are no longer available for download, starting with the 2.1 release.

- ctasetup-win-2.0.x.y.exe
- ctasetup-supPLICANT-2.0.x.y.exe

CtaAdminEx-win-2.1.103.0.exe contains the CTA end-user license agreement (EULA) and the ctasetup-win-2.1.103.0.msi installation file. By running the CtaAdminEx-win-2.1.103.0.exe file, you accept the EULA for all users and extract the ctasetup-win-2.1.103.0.msi installation file. You use the ctasetup-win-2.1.103.0.msi file to install CTA using standard MSI commands. You can use the ctasetup-win-2.1.103.0.msi file to install the CTA Scripting Interface feature, however, you can not use the file to install the 802.1x Wired Client feature.

CtaAdminex-supPLICANT-win-2.1.103.0.exe contains the EULA and the ctasetup-supPLICANT-win-2.1.103.0.msi installation file. By running the CtaAdminEx-supPLICANT-win-2.1.103.0.exe file, you accept the EULA for all users and extract the ctasetup-supPLICANT-win-2.1.103.0.msi installation file. By default, the ctasetup-supPLICANT-win-2.1.103.0.msi file installs Cisco Trust Agent with the CTA 802.1x Wired Client and provides an option to install Scripting Interface feature. If you do not intend to install the CTA 802.1x Wired Client on some end-points, that feature may also be suppressed using standard MSI commands.

Installing Cisco Trust Agent

Cisco Trust Agent installation files are standard Microsoft Windows Installation (MSI) files. Once deployed to the end-point, you can use standard MSI commands to install CTA silently, without user-interaction, or allow users to perform the installation using an installation wizard.

**Note**

The use of “ctasetup-2.1.103.0.msi” in procedures refers generically to either the ctasetup-win-2.1.103.0.msi or the ctasetup-supplciant-win-2.1.103.0.msi file. “ctasetup-2.1.103.0.msi” is not a real installation file.

General Installation Instructions

This is the outline of tasks required to install Cisco Trust Agent.

-
- Step 1** Run the CtaAdminex-win-2.1.103.0.exe or CtaAdminex-supplciant-win-2.1.103.0.exe files and accept the EULA. The ctasetup-win-2.1.103.0.msi or ctasetup-supplciant-win-2.1.103.0.msi file is extracted. See the [“Installation Files” section on page 4-5](#) for an explanation of these files.
- Step 2** (Optional) Create a custom installation package which could contain ACS root certificate, posture plugins, or a customized CTA configuration file. See the [“Installing CTA Using a Custom Installation Package” section on page 4-22](#) for an explanation of these procedures.
- Step 3** Install CTA by distributing ctasetup-win-2.1.103.0.msi or ctasetup-supplciant-win-2.1.103.0.msi files to end-users alone or as part of a custom installation package. You can use standard MSI commands to specify the features installed with CTA and the level of user interaction.
- See the [“Installing CTA Using MSI Commands” section on page 4-7](#) and [“Installing CTA Using an Installation Wizard” section on page 4-13](#) for descriptions of the different installation methods.
- Step 4** Install Cisco Secure Access Control Server (ACS) root certificate on the end-point if not distributed as part of a custom installation package. See [“About The ACS Server Root Certificate” section on page 8-3](#) for information about installing this certificate separately.
- Step 5** Verify CTA installation.

Installing CTA Using MSI Commands

Standard MSI commands can be passed to the Microsoft Windows Installer through command-line options. These commands determine what features to install as well as the level of user interaction.

This section describes the most common MSI commands.

**Note**

For more information on MSI installation commands see the Microsoft Windows Installer SDK at

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/about_windows_installer.asp

- [Installation Command](#)
- [Uninstallation Command](#)
- [Reinstalling or Repairing CTA](#)
- [Creating a Log File While Installing CTA](#)
- [Installing Optional Features During CTA Installation](#)
- [Specifying Installation Directory](#)
- [Specifying Reboot Options](#)
- [Setting User Interface Mode](#)

Installation Command

To install CTA using MSI command line options, you must know the name and path of the ctasetup-2.1.103.0.msi installation file and use the “/I” option with the Msiexec.exe command. The command can be entered from any prompt. See the following example:

```
Msiexec.exe /I "C:\Path_To_MSI\ctasetup-2.1.103.0.msi"
```

This command installs CTA using an installation wizard. Users accept the EULA, choose what features to install, and the installation directory.

Uninstallation Command

To uninstall the CTA using MSI command line options, you must know CTA's ProductCode or "GUID."

To find the GUID, follow this procedure:

-
- Step 1** Open the Windows Registry Editor.
- Step 2** Navigate to **HKEY_LOCAL_MACHINE\Software\Cisco Systems\Cisco Trust Agent**.

The value of the **ProductCode** registry key, including the curly brackets, is the GUID.

To uninstall Cisco Trust Agent, use the /X option with Msiexec.exe command. The command can be entered from any prompt. See the following example:

```
Msiexec.exe /X {GUID}
```

Reinstalling or Repairing CTA

To reinstall or repair CTA from using MSI command line options, run the MSI installation file using the MSI "/F" option. The full command can be run from any prompt. See the following example:

```
Msiexec.exe /fmsu "C:\Path_To_MSI\ctasetup-2.1.103.0.msi"
```

The /fmsu argument performs these actions:

- f – Reinstalls package
- m - Rewrites all required computer-specific registry entries.
- o - Reinstalls if file is missing or if an older version is installed.
- s - Overwrites all existing shortcuts.
- u - Rewrites all required user specific registry entries

Using this command users see messages asking them to wait while CTA is being configured.

Creating a Log File While Installing CTA

To create a log file during installation, run the MSI installation file using the MSI “/L” option. The full command can be entered from any prompt. This logging option requires that the log directory exist and is writable. The log file itself may not exist but the file name must be specified in the command.

See the following example:

```
Msiexec.exe /I "C:\Path_To_MSI\ctasetup-win-2.1.103.0.msi" /L*V  
"C:\ctalogfile.txt"
```

The /L*V option performs these actions:

L – Creates a log file

*V – Specifies verbose logging.

Installing Optional Features During CTA Installation

The ADDLOCAL option allows you to specify which features will be installed along with CTA.

The ctasetup-win-2.1.103.0.msi installs CTA by default. Using the ADDLOCAL command you can install the CTA Scripting Interface as well.

The ctasetup-supplciant-win-2.1.103.0.msi installs CTA and the CTA 802.1x Wired Client by default. However, when using the ADDLOCAL command, CTA is installed but the CTA 802.1x Wired Client interface is not installed by default. When using the ADDLOCAL command you must specify if you are installing either or both the CTA 802.1x Wired Client feature or CTA Scripting Interface feature.

When using the ADDLOCAL command the Scripting Interface features is referred to as “Scripting_Interface” and the CTA 802.1x Wired Client feature is referred to as “8021x_Wired_Client.”

This example shows using the ADDLOCAL option to install only the Scripting Interface feature and not the CTA 802.1x Wired Client feature:

```
Msiexec.exe /I "C:\Path_To_MSI\ctasetup-supplciant-win-2.1.103.0.msi"  
ADDLOCAL=Scripting_Interface
```

This example shows using the ADDLOCAL command to install only the CTA 802.1x Wired Client feature and not the Scripting Interface feature:

```
Msiexec.exe /I "C:\Path_To_MSI\ctasetup-supPLICANT-win-2.1.103.0.msi"  
ADDLOCAL=8021x_Wired_Client
```

These examples show using the ADDLOCAL command to install both features. These examples would be entered on one line.

**Note**

The features are separated by a comma only. There are no spaces before or after the comma.

```
Msiexec.exe /I "C:\Path_To_MSI\ctasetup-supPLICANT-win-2.1.103.0.msi"  
ADDLOCAL=Scripting_Interface,8021x_Wired_Client
```

```
Msiexec.exe /i "C:\Path_To_MSI\ctasetup-supPLICANT-win-2.1.103.0.msi"  
ADDLOCAL=ALL
```

**Note**

The ADDLOCAL command can be used with an interactive installation or a silent installation. (See the [“Setting User Interface Mode”](#) section on page 4-11 for information about “silent” and “interactive” installations.) When used with an interactive installation, users are not given the opportunity to choose what features to install.

Specifying Installation Directory

By default, ctasetup-2.1.103.0.msi installation files install CTA in the “\ProgramFiles\Cisco Systems\” directory of the local drive. You can use the “INSTALLDIR” MSI command to specify a different directory. The directory does not have to exist before you issue the command. See the following example:

```
Msiexec.exe /I "C:\Path_To_MSI\ctasetup-2.1.103.0.msi"  
INSTALLDIR="D:\NewDirectory"
```

This command shows users the installation wizard. During the installation, users will still have an opportunity to change the destination directory.

Specifying Reboot Options

By default the Microsoft Windows Installer determines when a reboot of the system is necessary and automatically prompts the user to reboot at the end of the installation. You can customize this action by using the Microsoft Windows Installer property called “REBOOT.” This property forces or suppresses certain system prompts for a reboot. The behavior of the REBOOT option also depends on whether the end-user is following an installation wizard or the installation is being performed silently.

The REBOOT property has three options:

- **Force** - If end-users perform the installation using an installation wizard, they will be prompted to reboot the system after the installation. If the installation is silent, the system reboots automatically without prompting the user.
- **Suppress** - If end-users perform the installation using an installation wizard, they will not be prompted to reboot the system at the end of the installation. If a reboot is required in the middle of an installation, end-users will be prompted to reboot system. If the installation is silent, end-users will not be prompted to reboot at the end of the installation, however, if a reboot is required in the middle of an installation, the system will be rebooted automatically without prompting the user.
- **ReallySuppress** - All prompts to reboot the system at the end or during an installation, whether the installation is being performed with an installation wizard or is silent, are suppressed

This is an example of using the REBOOT property with the Force option:

```
Msiexec.exe /I "C:\ctasetup-suppllicant-win-2.1.103.0.msi" REBOOT=Force
```

Setting User Interface Mode

By default, CTA’s MSI files provide the users with an installation wizard. Using various MSI commands, you can control how much the user is involved in the CTA’s installation.

For a full description of how the installation wizard works, see the [“Installing CTA Using an Installation Wizard” section on page 4-13](#).

These command options specify the amount of end-user interaction with the CTA installation:

Table 4-2 User Interface MSI Command Line Options

Command Option	Description
/q, /qn	There is no user interaction. This provides a silent installation. Example: <code>Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi" /q</code>
/qb	Users see messages alerting them that CTA is being configured, however, users are not prompted perform any action. Example: <code>Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi" /qb</code>
/qr	Users see some of the installation wizard windows including a progress bar showing installation, however, users are not prompted perform any action. Example: <code>Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi" /qr</code>
/qf	Users are fully involved in the installation of CTA. They install CTA using the installation wizard. Example: <code>Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi" /qf</code>
/qn+	Users receive a pop-up message at the end of the installation specifying the success or failure of the installation. Example: <code>Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi" /qn+</code>
/qb+	Users see messages alerting them that CTA is being configured, however, users are not prompted perform any action during the installation. At the end of the installation users receive a pop-up message that specifies the success or failure of the installation. Example: <code>Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi" /qb+</code>

**Tip**

When combining MSI options, specify the user interface command at the end of the entire command. For example, the following command installs CTA with the Scripting Interface, logging would be turned on, and users would experience basic user interaction with a final pop-up message.

```
Msiexec.exe /I "C:\ctasetup-2.1.103.0.msi"  
ADDLOCAL=Scripting_Interface /L*V "C:\logfile.txt" /qb+
```

Installing CTA Using an Installation Wizard

This section describes installing CTA and its other features by following an installation wizard. The You must have administrator privileges on the network client to install CTA.

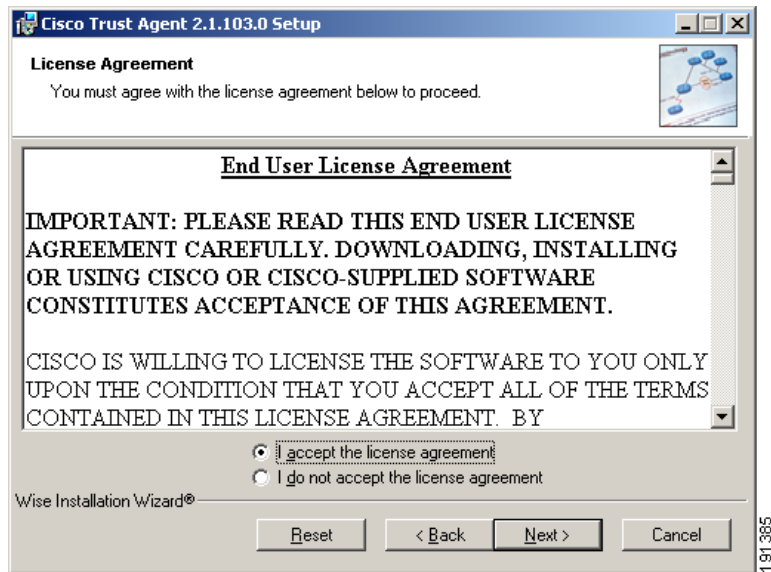
**Note**

If the group policy for the target system allows for elevated privileges for the MSI, then users with Standard or Restricted privileges can install CTA. To use the elevated privileges, MSI 2.0 must be installed before you begin the CTA installation. You cannot use a custom installation package to install the MSI unless you have administrator privileges.

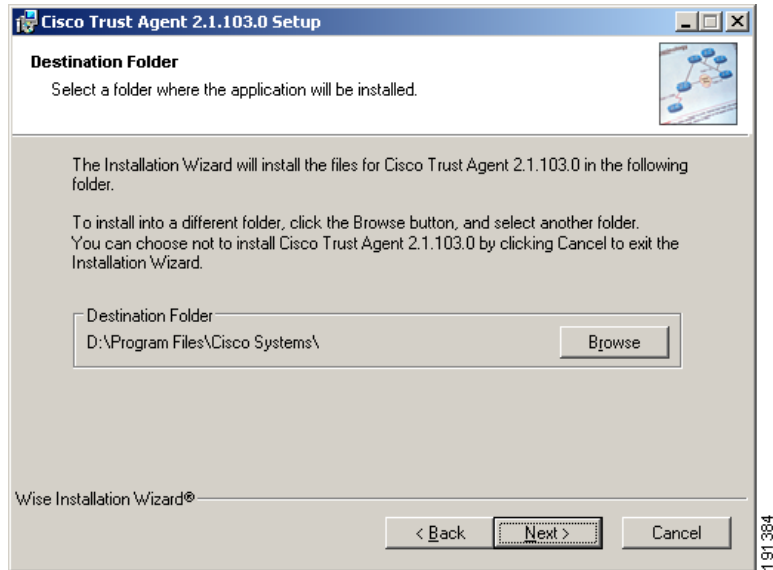
-
- Step 1** Read the [“General Installation Instructions”](#) section on page 4-6.
 - Step 2** Exit all Windows programs and disable any antivirus programs running on the network client.
 - Step 3** Launch the appropriate ctasetup-2.1.103.0.msi file by issuing the proper MSI command line option or by double-clicking the file. The Cisco Trust Agent **Installation Wizard** opens as shown in [Figure 4-1](#).

Figure 4-1 *The Cisco Trust Agent Installation Wizard*

Step 4 Click **Next**. The **License Agreement** window opens as shown in [Figure 4-2](#).

Figure 4-2 *The License Agreement on Windows*

- Step 5** Accept the license agreement by selecting the **I accept the license agreement** radio button and by clicking **Next**. The Destination Folder window opens as shown in [Figure 4-3](#).

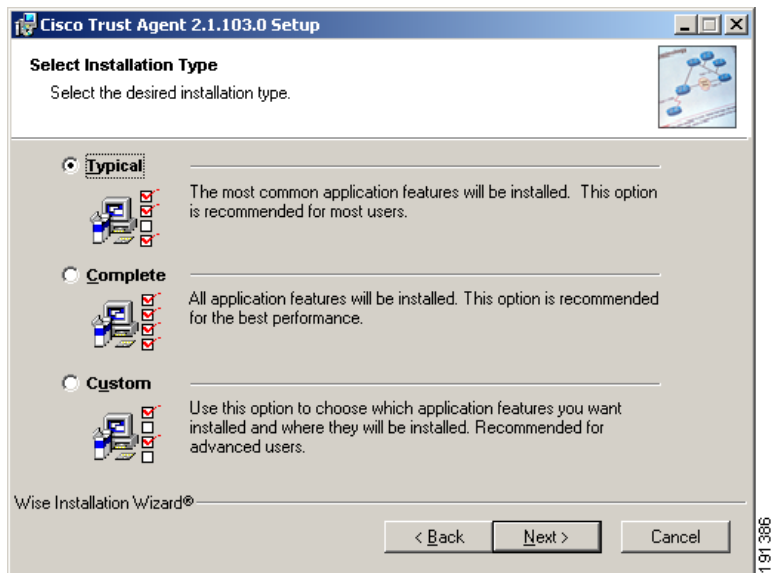
Figure 4-3 *The Destination Window*

Step 6 To change the installation directory:

- a. Click **Browse** to the desired drive and folder, and then click **OK**. The new install location appears in the **Destination Folder** pane.
- b. Click **Next**.

Step 7 The **Select Installation Type** dialog box opens (Figure 4-4).

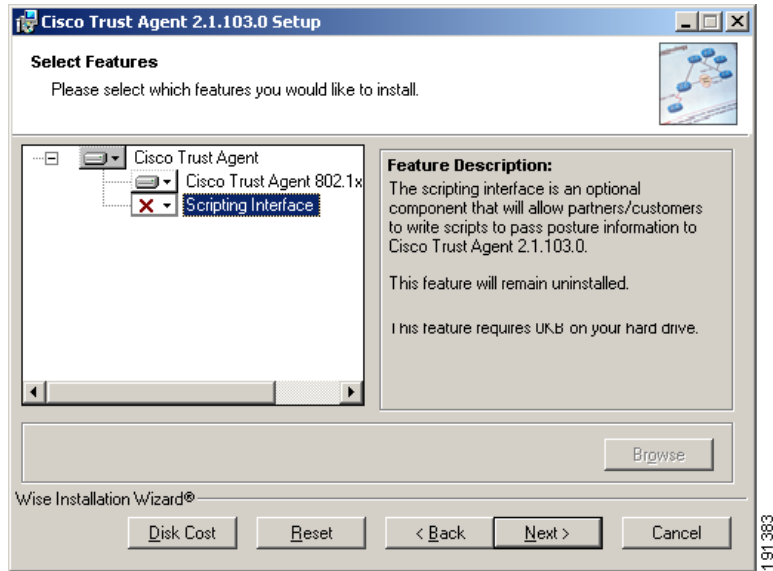
Figure 4-4 *Selecting an Installation Type*



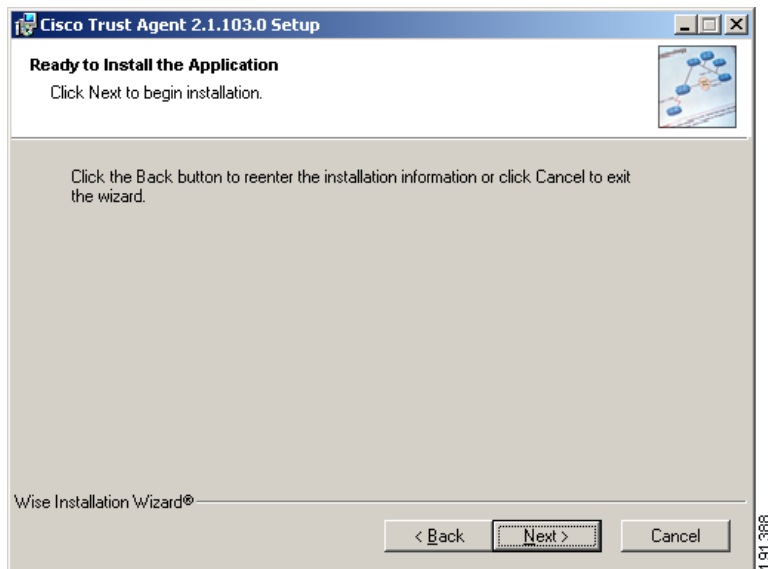
Selecting **Typical** will install the default features included in the installation file. If you are running a ctasetup file which includes the 802.1x Wired Client, that feature will be installed. The Scripting Interface is an optional feature and it is not installed during a **Typical** installation.

Selecting **Complete** installation will install all features available in the installation file. If you are running a ctasetup file which includes the 802.1x Wired Client, that feature will be installed. The Scripting Interface will also be installed during a **Complete** installation.

Selecting **Custom** installation will allow you to include or exclude any features available with the installation file. Figure 4-5 shows how you can select the features to install during a **Custom** installation. You can see that the Scripting Interface is not installed by default when you click **Custom**.

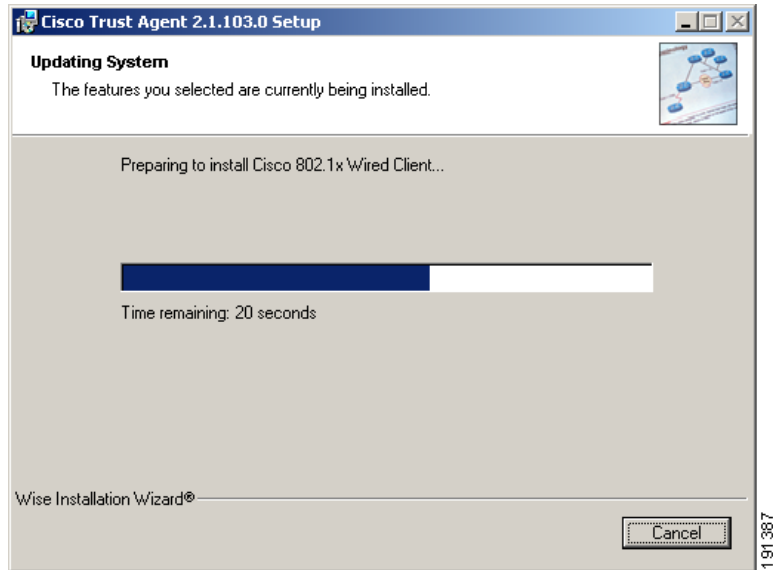
Figure 4-5 Choose Application Features

After choosing an installation type and selecting CTA features, click **Next**. The **Installing the Application** window opens as shown in [Figure 4-6](#).

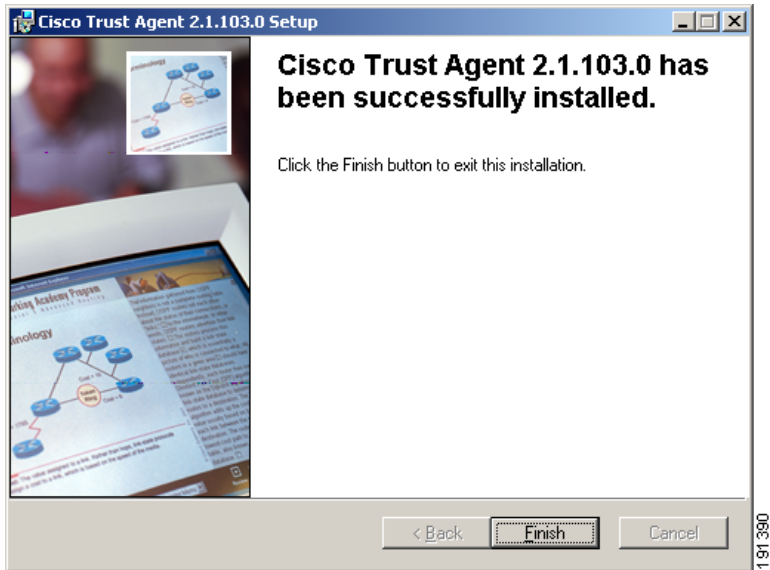
Figure 4-6 *Installing the Application*

Step 8 Click **Next**. The application installs to the selected directory. [Figure 4-7](#) illustrates the window that shows the progress of the installation.

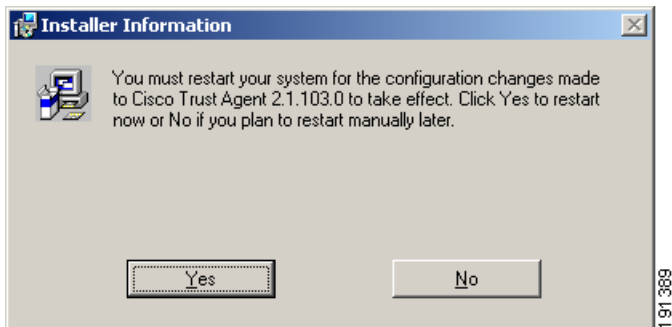
Figure 4-7 *The System is Being Updated*



When the installation is completed, the installer displays the **Installation Completed** window as shown in [Figure 4-8](#).

Figure 4-8 *The Installation is Complete***Step 9** Click **Finish**.

- The installation application closes. If you installed CTA with the 802.1x Wired Client, you will be prompted to restart your system and you will see the Installer Information window as in [Figure 4-9](#).

Figure 4-9 *Restart after Installation*

- If you did not install CTA with the 802.1x Wired Client, enable any antivirus programs you disabled in step 1. You will not be prompted to restart your machine.

Step 10 For CTA to establish a secure form of communication with the Cisco Secure ACS server, you must have either a CA certificate or a matching root certificate from the Cisco Secure ACS server installed on the system. Refer to [“About The ACS Server Root Certificate” section on page 8-3](#) for information.

Installing CTA Using a Custom Installation Package

Use this section as an example of how to create a customized CTA installation on a Windows operating system.

The CTA install application is a single executable file. To create a custom installation package, you create a directory structure which includes the desired CTA installation file, .ini files, plugin subdirectories and certificate subdirectories. This directory structure can then be distributed by a software deployment mechanism, such as a script or a software deployment tool.

After the software deployment mechanism distributes the directory structure to the remote network clients, it runs the CTA installation file with the desired MSI command line options. The CTA installation file copies the contents of the directory structure to the proper locations on the remote network client. The software deployment mechanism does not need to recompile the CTA installation file to create a custom installation.

Please read this entire procedure before beginning. There are options detailed later in the instructions that you should be aware of before beginning.

There are three procedures you need to follow to create a custom installation package:

-
- Step 1** [Install CTA and 802.1x Wired Client on the Administrator’s Client](#)
 - Step 2** [Create the Custom Installation Directory Structure](#)
 - Step 3** [Customize the Installation Directory](#)
 - Step 4** [Run the CTA Installation File to Install the Custom Package](#)

Install CTA and 802.1x Wired Client on the Administrator's Client

Before you create the custom installation package, install CTA and the 802.1x Wired client on the client you will use to create the custom installation package. This will install the template ctad.ini file, give you ready-access to the 802.1x Wired Client for when you create the authentication policies, and familiarize you with CTA's installation process. Use the [“Installing CTA Using an Installation Wizard” section on page 4-13](#) to install CTA and the 802.1x Wired Client.

Create the Custom Installation Directory Structure

For the sake of this and the following two procedures, we describe installing CTA along with the 802.1x Wired Client (the supplicant). To create a custom installation of CTA without the 802.1x Wired Client, the instructions would be the same except that you would start with the CtaAdminEx-win-2.1.103.0.exe file.

-
- Step 1** Create an empty directory on your network client. For example, D:\CTA\Custom_Package
 - Step 2** Open a command prompt.
 - Step 3** CD to the directory which contains the **CtaAdminEx-supPLICANT-win-2.1.103.0.exe** file. (See the [“Installation Files” section on page 4-5](#) for a description of this file.)
 - Step 4** After the prompt, on one line, type the name of the CtaAdminEx-supPLICANT-win-2.1.103.0.exe file followed by a -p switch and the path to the directory you created at the beginning of this procedure. This will extract the ctasetup-supPLICANT-win-2.1.103.0.msi file to the new directory. For example:

```
D:\CTA>CtaAdminEx-supPLICANT-win-2.1.103.0.exe -p D:\CTA\Custom_Package
```

**Note**

If you do not use the -p switch in the command, the ctasetup-supPLICANT-win-2.1.103.0.msi is extracted to the same directory that contains the CtaAdminEx-supPLICANT-win-2.1.103.0.exe file.



Note If you want to create a custom installation package for a wizard-driven installation, copy the ctasetup-win-2.1.103.0.msi or ctasetup-supplciant-win-2.1.103.0.msi file to the Custom_Package directory.

Step 5 When prompted, read and accept the End User License Agreement (EULA) on behalf of all users by typing **Y** and pressing **<Enter>**. After the CTA installation file has been extracted, a message is returned showing the path to which the installation file was extracted. In our example, you should now have a D:\CTA\Custom_Package directory with one file in it:
ctasetup-supplciant-win-2.1.103.0.msi

Step 6 Proceed to “[Customize the Installation Directory](#).”

Customize the Installation Directory

The customization choices in this procedure are optional. However, you will find that including some of these customizations is worthwhile. CTA is not a centrally managed product. If you do not plan to use the product defaults, it is to your benefit to pre-configure all available product settings before deploying CTA.

Step 1 Create a **certs** subdirectory. For example: D:\CTA\Custom_Package\certs
Copy the root certificate for your Cisco Secure ACS server to this directory. During installation, any certificates in this directory are added to the systems root certificate store.

If your Cisco Secure ACS server uses self-signed certificates, see the Cisco Secure ACS documentation for information about obtaining the certificate; if you use a CA server, refer to your CA server documentation.



Note This step is optional if a CA certificate or ACS root certificate have already been distributed to the network clients receiving this customized CTA installation. If these certificates have not been distributed, this step is required.

Step 2 Create a **plugins** subdirectory. For example: D:\CTA\Custom_Package\plugins

Copy any third-party plugins that you want to provision at installation time into this directory.

- Step 3** Create a new **ctad.ini** file and store it in the D:\CTA\Custom_Package directory. This file is used to configure CTA communication settings, user notifications, and certificate validation rules. If you want to change the default communication settings, such as the port number CTA listens over, the maximum number of sessions, and session time-out values, include this file. Refer to [Chapter 5, “Configuring Cisco Trust Agent”](#) for instructions on how you should create and format this file.
- Step 4** Create a new **ctalogd.ini** file and store it in the D:\CTA\Custom_Package directory. This file is used to enable and disable CTA logging. Refer to [Chapter 6, “Cisco Trust Agent Event Logging”](#) for instructions on how you should create and format this file.
- Step 5** If you are installing the 802.1x Wired Client, create the **802_1x** subdirectory; this directory allows you to customize the 802.1x Wired Client settings, for example: D:\CTA\Custom_Package\802_1x. See the [“Creating Deployment Packages” section on page 9-36](#), for information about creating the authentication policies described in this step.
- a. Create the D:\CTA\Custom_Package\802_1x\policies\ directory and include the .xml policy file that defines the machine and user authentication settings for the client.
 - b. Create the D:\CTA\Custom_Package\802_1x\networks\ directory and include the .xml policy file that defines the machine and user authentication settings for the client.
- Step 6** Proceed to [“Run the CTA Installation File to Install the Custom Package.”](#)

Run the CTA Installation File to Install the Custom Package

For the sake of this procedure, we assume that the custom package is deployed to the appropriate network clients by a software deployment mechanism such as a software deployment tool or script.



Note

The custom package consists of the customized installation directories and the CTA installation .msi file. The installation .msi does not recompile the customized files into a new installation .msi file, it installs CTA and the customized files you created in the previous procedure.

- Step 1** The software deployment mechanism deploys the custom package to the appropriate network clients and saves it to a local directory. For the sake of this example, we assume that the custom installation package is saved to the C:\Temp directory. There would now be a C:\Temp\Custom_Package directory on the client.
- Step 2** The software deployment mechanism can then run the CTA installation file and employ whatever MSI command line options you choose. (See the [“Installing CTA Using MSI Commands” section on page 4-7](#) for a summary of common MSI commands and examples of how they are used with the CTA installation files.)

Here are two different examples of the CTA installation:

Run the CTA installation file as it is. This installs CTA, the 802.1x Wired Client, and your customizations. To do so, the software deployment mechanism would run the following commands:

- a. CD to the C:\Temp\Custom_Package directory.
- b. From the prompt, run the **ctasetup-suppllicant-win-2.1.103.0.msi** file. For example:

```
C:\Temp\Custom_Package>ctasetup-suppllicant-win-2.1.103.0.msi
```

Run the CTA installation file with MSI command line options. The command in the procedure installs CTA with the Scripting Interface and CTA 802.1x Wired Client features, it logs the installation and stores the log file in “C:\Custom_Package\logfile.txt, and it is silent installation.

- a. CD to the C:\Temp\Custom_Package directory.
- b. From the prompt, run the **ctasetup-suppllicant-win-2.1.103.0.msi** file with the MSI command line options. For example:

```
C:\Temp\Custom_Package>Msiexec.exe /I ctasetup-suppllicant-win-2.1.103.0.msi  
ADDLOCAL=Scripting_Interface,8021x_Wired_Client /L*V "C:\Custom_Package\logfile.txt" /qn
```



Note

System administrators should be aware that after if the 802.1x Wired Client was installed with CTA, the system needs to be rebooted in order to activate the 802.1x Wired Client.

Upgrading to Cisco Trust Agent, Release 2.1

Cisco Trust Agent supports upgrade installations from versions 1.0, 2.0, 2.0.1, and Selective Availability and Beta 2.1 releases to CTA 2.1.103.

The behavior of an upgrade reflects the kind of installation being used. If the upgrade is performed using an installation wizard, CTA 2.1.103.0 recognizes the previous installation of CTA and prompts users to upgrade. In the case of a silent installation, it is assumed that the user intends to perform an upgrade and the installation proceeds without prompting the user.

**Note**

When upgrading a version of CTA along with the CTA 802.1x Wired Client, to CTA 2.1 with the CTA 802.1x Wired Client, the computer is disconnected from the network at the end of the software upgrade process. The final step of the upgrade procedure is to reboot the computer; rebooting restores the network connection and it is a required step in the upgrade process.

In the case of a silent upgrade, administrators should use MSI commands which limit interruptions to users but still prompt users to reboot their computers at the end of the software upgrade.

When upgrading a version of CTA without the CTA 802.1x Wired Client to the latest version of CTA without the CTA 802.1x Wired Client, you are not required to reboot the computer for the upgrade to take affect. There is no loss of network connectivity during the upgrade process.

Upgrading from Cisco Trust Agent, Release 1.0

During an upgrade installation of CTA from 1.0 to CTA 2.1, existing certificates remain in the certificate store in which they were installed during the CTA 1.0 installation. Posture plugins and the ctalogd.ini file are moved to their new location in the CTA 2.1.103.0 directory structure. The ctad.ini file used in CTA 1.0 remains in the directory in which it was originally installed and CTA 2.1.103.0 recognizes the file in its original location.

Upgrading from Cisco Trust Agent, Release 2.0.0.30

During an upgrade installation of CTA from 2.0.0.30 to CTA 2.1.103.0, certificates, third-party posture plugins, ctad.ini, ctalogd.ini, and log files remain in the directories in which they were installed and they are used by CTA 2.1.103.0.

During an upgrade installation of CTA from 2.0.0.30 to CTA 2.1.103.0, where the 802.1x Wired Supplicant is also being upgraded, certificates, third-party posture plugins, ctad.ini, ctalogd.ini, and log files remain in the directories in which they were installed and they are used by CTA 2.1.103.0.

Deployment profile files used by the 802.1x Wired Client in CTA 2.0.0.30 **are not compatible** with those used by the 802.1x Wired Client in CTA 2.1.103.0. The deployment profile files that define user and machine authentication requirements will need to be recreated and reinstalled after an upgrade from CTA 2.0 with the 802.1x Wired Client to CTA 2.1 with the 802.1x Wired Client.

During an upgrade from CTA 2.0 with the 802.1x Wired Client to CTA 2.1 with the 802.1x Wired Client the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client** directory and all of its contents are deleted and replaced with the upgraded CTA 802.1x Wired Client software.



Note

If, when you created the deployment profile for use with CTA 2.0.0.30, you saved the deployment profile files in a directory outside of \Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client the files will not be deleted by the upgrade procedure, however, you will not be able to use them with CTA 2.1 and its new 802.1x Wired Client. Likewise, there is no advantage in backing-up the deployment profile files used in CTA 2.0.0.30 before you upgrade to CTA 2.1.103.0. The security profiles in CTA 2.0 are not compatible with those used in CTA 2.1.103.0.

For instructions on how to deploy end-user 802.1x Wired Clients which are compatible with Cisco Trust Agent 2.1.103.0, see [“Deploying End-User 802.1x Wired Clients” section on page 9-35](#).

Upgrading from Cisco Trust Agent, Release 2.0.1

Cisco Trust Agent 2.0.1 was a release supported on Windows XP platforms only. During an upgrade installation of CTA from 2.0.1 to CTA 2.1, certificates, third-party posture plugins, ctad.ini, ctalogd.ini, and log files remain in the directories in which they were installed and they are used by CTA 2.1.

During an upgrade installation of CTA from 2.0.1 to CTA 2.1, where the 802.1x Wired Supplicant is also being upgraded, certificates, third-party posture plugins, ctad.ini, ctalogd.ini, log files, and the deployment profile files remain in the directories in which they were installed and they are used by CTA 2.1. These deployment profile files are stored here:

- Drive:\CTA\Custom_Package\802_1x\policies*policy.xml
- Drive: \CTA\Custom_Package\802_1x\networks*networks.xml

For more information about the deployment profile files see, [“Understanding Policies and Profiles” section on page 9-24](#).

Upgrading from CTA 2.1 Selective Availability and Beta Releases to CTA 2.1.103.0

Some customers of Cisco’s Network Admission Control program participated in testing “selective availability” or “limited availability” releases and Beta releases of CTA 2.1 to test its functionality in their NAC environments.

These builds, numbered 2.1.18.0, 2.1.100.0, 2.1.101.0, and 2.1.102.0 may be upgraded to CTA 2.1.103.0 without being uninstalled first. The certificates, third-party posture plugins, ctad.ini, ctalogd.ini, log files, and the deployment profile files remain in the directories in which they were installed and they are used by CTA 2.1.103.0.

Verifying the Cisco Trust Agent Installation

After Cisco Trust Agent has been installed you will find the following directory structures containing CTA's executable files:

- \Program Files\Cisco Systems\CiscoTrustAgent
- \Program Files\Common Files\PostureAgent

If you installed the CTA 802.1x Wired Client along with CTA you will also find the \Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client directory.

After installing CTA, to verify that CTA is running, follow this procedure:

Step 1 Open a command prompt window on the target system.

Step 2 Type **net start** and then click **Enter**.

Step 3 Verify that the following services are running:

Current Service Names:

- Cisco Posture Server Daemon
- Cisco Systems Inc. CTA Posture State Daemon
- Cisco Trust Agent EoU Daemon
- Cisco Trust Agent Logger Daemon

If you installed the CTA 802.1x Wired Client (the supplicant) then you should also see these services running.

- Cisco Trust Agent 802.1x wired client
- Cisco Trust Agent 802.1x wired client log

If these services are not running, try rebooting the system and checking again. If the services still do not run, try reinstalling the application.

Uninstalling Cisco Trust Agent on Windows

To uninstall Cisco Trust agent, follow these steps:

Step 1 Choose **Start > Settings > Control Panel > Add/Remove Programs**.

Step 2 Choose **Cisco Trust Agent** from the list of installed applications.

Step 3 Click **Remove**.

A confirmation dialog box appears.

Step 4 Click **Yes** to continue the removal.



Note

Certificates and plugin files are not deleted when CTA is uninstalled; they remain on the client.



CHAPTER 5

Configuring Cisco Trust Agent

Cisco Trust Agent (CTA) may be configured by making changes to its `ctad.ini` file. The tasks in this chapter describe configuring CTA's communication of posture data to the Cisco Secure Access Control Server as well as the communication of that posture to the client. To configure CTA Logging, see [Chapter 6, "Cisco Trust Agent Event Logging"](#).

This chapter contains the following sections:

- [The `ctad.ini` Configuration File, page 5-2](#)
 - [Editing the `ctad.ini` Configuration File, page 5-3](#)
 - [ctad.ini Configuration Parameters, page 5-4](#)
- [Configuring EAP over UDP Communication, page 5-12](#)
- [Configuring Posture Plugins, page 5-13](#)
 - [Configuring CTA and Posture Plugin Interaction, page 5-13](#)
 - [Configuring the Default Posture Plug-in Message Size, page 5-16](#)
 - [Configuring an Application-Specific Posture Plug-in Message Size, page 5-17](#)
 - [Configuring PPMsgSize for Host Posture Plugin, page 5-18](#)
 - [Configuring PPMsgSize for Symantec Posture Plugin, page 5-19](#)
 - [Configuring Asynchronous Posture Status Query, page 5-19](#)
- [Configuring User Notifications, page 5-20](#)
 - [Configuring Windows User Notifications, page 5-20](#)
 - [Configuring Linux User Notifications, page 5-21](#)

- [Configuring Mac OS X User Notifications, page 5-22](#)
- [Configuring Clickable URL and Browser Auto-Launch Features, page 5-23](#)
- [Logging Notifications, page 5-24](#)
- [Certificate Distinguished Name Matching, page 5-25](#)
 - [DN Matching Rule Syntax, page 5-25](#)
 - [Configuring Certificate DN Matching, page 5-27](#)
- [Configuring the Scripting Interface, page 5-27](#)
- [Sample Windows ctad.ini File, page 5-28](#)
- [Sample Linux ctad.ini File, page 5-32](#)
- [Sample Mac OS X ctad.ini File, page 5-37](#)

The ctad.ini Configuration File

The ctad.ini configuration file is a plain text file that contains the parameters for the Cisco Trust Agent’s communication settings, user notifications, certificate filtering rules, and the scripting interface feature.

Some parameters are shared by Linux, Mac OS X, and Windows environments while other are used only for a particular operating system. See the “[Sample Windows ctad.ini File](#)” section on page 5-28, “[Sample Linux ctad.ini File](#)” section on page 5-32, and “[Sample Mac OS X ctad.ini File](#)” section on page 5-37 for examples of ctad.ini files.

The template ctad.ini file is named ctad-temp.ini and is installed during the Linux, Mac OS X, and Windows installations. The ctad-temp.ini file for Linux and Mac OS X are stored in the /etc/opt/CiscoTrustAgent/ directory. The ctad-temp.ini file for Windows is stored in the \\Program Files\\Cisco Systems\\CiscoTrustAgent\\ directory.

Editing the ctad.ini Configuration File

In order to edit the ctad.ini file you must have administrative privileges.

-
- Step 1** Locate the ctad.ini file or the ctad-temp.ini template file on the host.
- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
 - For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.
- Step 2** If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as “ctad.ini”.
- Step 3** Open the ctad.ini file in a plain text editor.
- Step 4** Locate the section you want to edit.
- Step 5** Delete the semicolon (;) in front of the parameter you want to edit.
- Step 6** Adhering to the value ranges defined in the ctad.ini file, change the value of the parameter.
- Step 7** Save the ctad.ini file and close the file.
- Step 8** Activate changes in the CTA environment:
- If your changes were to the [main], [UserNotifies] or [ServerCertDNVerification] sections of the ctad.ini file, the new values will be re-read when they are needed.
 - If your changes were to the [EAPoUDP] section, reboot the system on which CTA runs.

ctad.ini Configuration Parameters

Table 5-1 explains the sections and parameters used in the ctad.ini file by Linux, Mac OS X, and Windows operating systems.

Table 5-1 *ctad.ini Configuration Parameters*

Keyword	Description	Operating System(s)
[main]	Main section of ctad.ini file.	Linux, Mac OS X, Windows
EnableVFT	<p>The “EnableVFT” parameter indicates if Validation-Flag TLV is enabled on the version of IOS running on the Network Access Device (NAD). “EnableVFT” enables CTA to operate with the NAD wether it has support for Validation-Flag TLV or not.</p> <p>Default Value: 0</p> <p>Range of Values: 0, 1</p> <p>0 = IOS does not support Validation-Flag TLV</p> <p>1 = IOS does support Validation-Flag TLV</p>	Linux, Mac OS X, Windows

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
PPInterfaceType	<p>The PPInterfaceType parameter describes how CTA gathers posture plugin information.</p> <p>Default Value: Block</p> <p>Range of Values: Block, NonBlockConcurrent, NonBlockSerial</p> <ul style="list-style-type: none"> • Block: CTA will request posture information from one plug-in at a time. It will not request information from the next plugin until it has received the posture credentials from the current plug-in. • NonBlockConcurrent: CTA requests posture information from all posture plugins simultaneously. The operation of gathering posture credentials ends when all the posture information is returned or the PPWaitTimeout value is reached, whichever is sooner. • NonBlockSerial: CTA requests posture information from one plug-in at a time. It will not request information from the next plugin until it has received the posture credentials from the current plug-in. The operation of gathering posture credentials ends when all the posture information is returned or the PPWaitTimeout value is reached, whichever is sooner. 	Linux, Mac OS X, Windows
PPWaitTimeout	<p>The PPWaitTimeout is only applicable if the PPInterfaceType is set to NonBlockConcurrent or NonBlockSerial. The parameter defines the maximum time allowed, in seconds, to complete the processing of all plugin queries. Should this timer expire while waiting for responses, CTA will send all of the data that it received thus far in the exchange.</p> <p>Default value: 5 seconds</p> <p>Range of values: 1 - 300 seconds</p>	Linux, Mac OS X, Windows

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
PPMsgSize	The PPMsgSize parameter allows the administrator to modify the maximum message size that a posture plugin can send from 1k to a maximum value of 6k. Default value: 1024 bytes Range of values: 1024 – 6144 bytes	Linux, Mac OS X, Windows
PluginName_PPMsgSize	An application-specific posture plugin message size may be added to the ctad.ini file. When added, the posture plugin this parameter references will provide a posture message of this size and it will ignore the value provided by PPMsgSize. See Configuring an Application-Specific Posture Plug-in Message Size, page 5-17 . Range of values: 1024 – 6144 bytes	Linux, Mac OS X, Windows
SQTimer	The status query timer (SQTimer) parameter defines how often CTA queries the posture plugins to detect a change in their status. Default value: 300 seconds Range of values: 5 - 4294967295 seconds	Linux, Mac OS X, Windows
[EAPoUDP]	Section head for Cisco Trust Agent using EAP over UDP protocol to communicate with Cisco Secure ACS.	Linux, Mac OS X Windows
LocalPort	The LocalPort parameter defines the port on which the NAD initiates posture validation with CTA. Changing this value requires changes to the NAD configuration. Default value: 21862 Rang of values: 1 - 65550	Linux, Mac OS X, Windows

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
MaxSession	CTA supports one established session per NAD. But CTA can support concurrent sessions with multiple NADs. MaxSession is the number of sessions you allow CTA to support. Default value: 8 Range of values: 1 - 256	Linux, Mac OS X, Windows
SessionIdleTimeout	The SessionIdleTimeout parameter defines the number of seconds an EAP over UDP session can remain idle before timing out. Default value: 3600 Range of values: 60 - 172800	Linux, Mac OS X, Windows
BootTimeUDPExemptions	The BootTimeUDPExemptions parameter alters the Windows Firewall policy and enable the CTA to receive packets when the Windows XP SP2 or SP3-based computer is starting. Default value: 1 Range of values: 0, 1 0 = Windows Firewall Boot Time Exemptions are disabled. 1 = Windows Firewall Boot Time Exemptions are enabled. Note Use of the BootTimeUDPExemptions parameter is relevant only when used in conjunction with Microsoft's hot fix for Windows XP (KB17730)	Windows
[UserNotifies]	The [UserNotifies] section defines the behavior of pop-up windows containing messages sent from ACS to CTA.	Linux, Mac OS X, Windows

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
SysModal	<p>SysModal specifies whether the user notification dialog boxes are modal or not. If the notification dialog boxes are modal, the user must close the notification dialog box to continue working. Also, if the dialog boxes are modal, and there is more than one notification, only the last notification appears.</p> <p>Default value: 1</p> <p>Range of Windows values: 0, 1</p> <p>0 = Modal dialog boxes are disabled.</p> <p>1 = Modal dialog boxes are enabled.</p>	Windows
UserActionDelayTimeout	<p>If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message. Setting this parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address.</p> <p>Default value: 25 seconds</p> <p>Range of values: 0 - 4294967295 seconds</p>	Linux, Mac OS X, Windows
EnableNotifies	<p>The EnableNotifies parameter enables or disables user notifications. If the EnableNotifies parameter is enabled, a clickable URL may also be presented in the pop-up browser window that contains the posture result. The end user can click the URL link to go to a browser page that contains additional information provided by the ACS administrator.</p> <p>This parameter applies to logged-in users.</p> <p>Default value: 1</p> <p>Range of values 0, 1</p> <p>0 = User notifications are disabled.</p> <p>1 = User notifications are enabled.</p>	Linux, Mac OS X, Windows

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
MsgTimeout	<p>The MsgTimeout parameter specifies how long, in seconds, user notification dialog boxes are displayed.</p> <p>Default value: 300</p> <p>Range of values: 30 - 4294967</p> <p>Special value: 0 (disables the timeout)</p>	Linux, Mac OS X, Windows
EnableLogonNotifies	<p>Enables or disables user notification received before the user is logged on.</p> <p>Default value: 1</p> <p>Range of values: 0, 1</p> <p>0 = User notifications received before the user is logged on are discarded.</p> <p>1 = User notifications received before the user is logged on are saved and displayed to the user when they log on.</p>	Linux, Mac OS X, Windows
LogonMsgTimeout	<p>Specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies enabled.</p> <p>Default value: 86400</p> <p>Range of values: 30 - 4294967</p> <p>Special value: 0 (disables the timeout)</p>	Linux, Mac OS X, Windows
DisplayType	<p>The DisplayType parameter determines if messages sent from ACS to CTA are displayed in a graphic user interface or in a terminal window.</p> <p>Default value: gui</p> <p>Range of Linux values: term, gui</p> <p>Range of Mac OS X values: gui</p>	Linux

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
TermFont	<p>The TermFont parameter sets the font in which to display the terminal screen text. Use the default value. The TermFont value for Asian Languages will be implemented in a future release.</p> <p>Default value:</p> <p>-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10636-1</p> <p>TermFont entry below for Asian languages:</p> <p>TermFont=-misc-zy song18030-medium-r-normal--0-0-0-0-c-0-iso10646-1</p>	Linux
ClearOldNotification	<p>The ClearOldNotification parameter clears or saves notification messages.</p> <p>Default value: 1</p> <p>Range of values: 0,1</p> <p>0 = Notification messages are saved.</p> <p>1 = CTA clears the old notification message before displaying the new window.</p>	Linux, Mac OS X
BrowserPath	<p>The BrowserPath parameter specifies the full path of the browser on Linux systems.</p> <ul style="list-style-type: none"> For Red Hat Enterprise Linux v3 (Enterprise, Advanced, Workstation) use this path: /usr/bin/mozilla For Red Hat Enterprise Linux v4 (Enterprise, Advanced, Workstation) use this path: /usr/bin/firefox 	Linux

Table 5-1 *ctad.ini Configuration Parameters (continued)*

Keyword	Description	Operating System(s)
[ServerCertDNVerification]	The [ServerCertDNVerification] section contains configurable parameters for distinguished name (DN) matching. When using CA certificates to validate your Cisco Secure ACS server certificate, you can implement additional security using distinguished name matching. See Configuring Certificate DN Matching, page 5-27 for more information on these parameters.	Linux, Mac OS X, Windows
TotalRules	The TotalRules parameter defines the number of DN matching rules that follow it. If the number of rules is greater than 1 the rules are connected by an OR statement. Default value: (none) Range of values: 0 - 64 Special value: 0 (Disables DN matching)	Linux, Mac OS X, Windows
RuleX	The RuleX parameters are DN matching rules, where X is the index for the rule. These are examples: Rule1=CN*"Server", ISSUER-CN*"Finance" Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"	Linux, Mac OS X, Windows
[Scripting_Interface]	The parameters in the [Scripting_Interface] section define the scripting interface	Linux, Mac OS X, Windows
delta_stale	The delta_stale parameter specifies how long, in minutes, before the posture database record is deemed outdated. Default value: 43200 Range of values: 1-5256000 Special value: 0 (the database will never expire)	Linux, Mac OS X, Windows

Configuring EAP over UDP Communication

CTA can communicate with a router or switch using the Extensible Authentication Protocol over User Datagram Protocol (EAP over UDP). When CTA uses EAP over UDP to communicate with a router, this is referred to as the NAC L3 IP method. When CTA uses EAP over UDP to communicate with a switch, this is referred to as the NAC L2 IP method. In these cases, you can configure the NAC L3 IP and NAC L2 IP communication in the [EAPoUDP] section of the ctad.ini configuration file.

These configurations are optional. You are not required to change the default values of these parameters in order for CTA to run properly after installation.

**Note**

On Windows systems, CTA can also communicate with a switch, through the 802.1x Wired Client, using the IEEE 802.1X protocol. This communication is not configurable in the ctad.ini file.

To configure EAP over UDP communication, use the [Editing the ctad.ini Configuration File, page 5-3](#) procedure, reference the sample ctad.ini files, and reference the [ctad.ini Configuration Parameters, page 5-4](#).

You can configure the following communication settings for CTA:

- **LocalPort** — By default, CTA listens on UDP port 21862. If you change this setting, you need to configure your network access device to initiate the posture validation process on the new port number.
- **MaxSession** — CTA only supports one established session per network access device, but can support sessions from multiple, different network access devices at the same time. CTA can support up to 255 sessions at one time.
- **SessionIdleTimeout** — This specifies the number of seconds an EAPoUDP session can remain idle before timing out.
- **BootTimeUDPExemptions** — The BootTimeUDPExemptions parameter alters the Windows Firewall policy and enables CTA to receive packets when the Windows XP SP2 or SP3-based computer is starting.

By enabling BootTimeUDPExemptions you alter the Windows XP Firewall setting by adding our local EAPoUDP port to the Windows XP Firewall boot time UDP exemptions policy. This enables CTA to communicate with ACS over the network.

**Note**

Use of the BootTimeUDPexemptions parameter is relevant only when used in conjunction with Microsoft's hot fix for Windows XP (KB17730)

Configuring Posture Plugins

These are the aspects of posture plugin behavior that are configurable:

- The interaction between CTA and posture plugins for the collection of posture data, transferring notifications, and status updates.
- The message size a posture plugin provides to CTA is configurable.
- Reporting behavior of legacy posture plugins.

Configuring CTA and Posture Plugin Interaction

CTA and the posture plugins interact for the transfer of posture data, posture notifications, and status updates. The PPInterfaceType and PPWaitTimeout parameters are used together to determine how CTA interacts with the plugins and how long the interaction with all plugins lasts. The procedure below describes how to set the values of PPInterfaceType and PPWaitTimeout.

-
- Step 1** Locate the ctad.ini file or the ctad-temp.ini template file on the host.
- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
 - For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.
- Step 2** If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as “ctad.ini”.
- Step 3** Locate the [Main] section in the ctad.ini file.
- Step 4** Delete the semicolon (;) in front of the **PPInterfaceType** and set the parameter to Block, NonBlockSerial, or NonBlockConcurrent. See the description of these values in the [“ctad.ini Configuration Parameters”](#) section on page 5-4.

- Step 5

Delete the semicolon (;) in front of the **PPWaitTimeout** parameter and set the PPWaitTimeout period to the number of seconds you require for all posture plugins to return their posture information.



Note

The maximum setting for PPWaitTimeout cannot be more than the maximum time that the network access device allows the host to respond to posture requests. If PPWaitTimeout is set at too high a value, the entire posture request will timeout and posture will be re-requested.

- Step 6

Save and close the ctad.ini file.
- See [Example 5-1 on page 5-14](#) for a description of how these parameters would interact during a posture request.

Example 5-1 Interaction of PPIInterfaceType and PPWaitTimeout parameters

The examples that follow describe the effect of the values of PPIInterfaceType and PPWaitTimeout during a request for posture information. The interaction between CTA and the plugin for the transfer of a notification or a status update would also follow the same workflow.

For these examples, assume that these are the posture plugins on the client that return posture credentials and that they take the stated amount of time to return those posture credentials to CTA:

Posture Plugin	Time to collect posture credentials and send them to CTA
CiscoHostPP	0.5 seconds
CTAPP	0.5 seconds
AntivirusPP	3.0 seconds
ApplicationPP	2.0 seconds

Scenario 1—PPIInterfaceType is set to “Block” and PPWaitTimeout is set to 5 seconds.

Because PPIInterfaceType is set to “Block” one plugin must collect its credential information before the next plugin can collect its posture credential information. Once all the posture information is collected it is sent to CTA.

In this case, it would take 6 seconds for the full amount of posture credentials to be collected before being sent to CTA. The PPWaitTimeout has no effect on the collection of this data because that parameter is only relevant when PPInterfaceType is set to either NonBlockConcurrent or NonBlockSerial.

Scenario 2—PPInetrfaceType is set to NonBlockConcurrent and PPWaitTimeout is set to 5 seconds.

Because PPInterfaceType is set to “NonBlockConcurrent” all the plugins can collect their posture data simultaneously.

In this example, it would take 3 seconds for all the posture credential information to be collected. The PPWaitTimeout parameter is greater than 3 seconds, so it has no effect on the transaction.

Scenario 3—PPInterfaceType is set to NonBlockSerial and PPWaitTimeout is set to 5 seconds.

Because PPInetrfaceType is set to NonBlockSerial the plugins collect their posture data one at a time. In this scenario, the following would occur:

1. CTA requests posture credentials from CiscoHostPP.
2. CiscoHostPP collects its posture credentials.
3. CTA request posture credentials from CTAPP.
4. CTAPP collects its posture credentials.
5. CTA requests posture credentials from AntivirusPP.
6. AntivirusPP collects its posture credentials.
7. CTA requests posture credentials from ApplicationPP.
8. ApplicationPP begins collecting its posture credentials.
9. The PPWaitTimeout expires before ApplicationPP can complete collecting posture credentials.
10. CTA sends the posture credentials for CiscoHostPP, CTAPP, and AntivirusPP to the ACS.
11. CTA considers ApplicationPP an “unresponsive plugin” and will not request its posture credentials again until ApplicationPP supplies its posture credentials for the last request. When ApplicationPP does reply with its posture credentials it is no longer considered an “unresponsive plugin.”

**Tip**

These scenarios describe how the posture plugin sends posture credentials to CTA. Once CTA has the posture credentials, it forwards them to ACS. ACS then evaluates the posture credentials against the posture validation network access profile.

In the specific case of scenario 3, if the posture credentials of ApplicationPP are required in order for the client to be granted network access, then access may not be granted because ApplicationPP did not have enough time to report its credentials to CTA and CTA did not have the credentials to send them to ACS.

Configuring the Default Posture Plug-in Message Size

By default, plug-ins are permitted to provide 1024 bytes of information to CTA. This number can be increased to allow all plug-ins to provide up to 6KB of information. However, limiting the size of the posture message to as close to 1KB of information as possible allows more applications to provide a posture message and optimizes the reporting time of all the posture messages. The PPMsgSize parameter in the ctad.ini sets the posture plugin message size for all plugins.

You can also set an application-specific posture plugin message size that is different than the default PPMsgSize value for an application that has its own plugin. See [“Configuring an Application-Specific Posture Plug-in Message Size” section on page 5-17](#) for more information.

The maximum amount of posture data that CTA can send to ACS at one time is 16KB.

**Note**

If there is a Symantec posture plugin installed on the client, the ctad.ini file must be configured in one of two ways:

- PPMsgSize must be set to 1024 bytes.
- The Symantec posture plugin must use an application-specific posture plugin set to 1024 bytes

Without using one of these configurations posture plugin messages from any posture plugin will not be transferred to CTA.

-
- Step 1** Locate the ctad.ini file or the ctad-temp.ini template file on the host.
- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
 - For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.
- Step 2** If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as “ctad.ini.”
- Step 3** Locate the [Main] section in the ctad.ini file.
- Step 4** Delete the semicolon (;) in front of the **PPMsgSize**.
- Step 5** Increase the **PPMsgSize** to up to 6144 bytes (6KB).
- Step 6** Save and close the ctad.ini file.

Configuring an Application-Specific Posture Plug-in Message Size

The posture plugin message size may be customized for any posture plugin. The default posture plugin message size for all plugins is equal to the value of the PPMSize parameter in the ctad.ini file. If that default value is not appropriate for a specific posture plug-in, you can specify an application-specific PPMSize parameter value.

The application-specific PPMSize parameter is added to the ctad.ini file in the [main] section and it uses this naming convention: *PluginName_PPMSize*, where *PluginName* is the name of plugin as specified in the posture plugin's information file.

For example, assume you need to set XYZApplication's unique posture plugin message size to 4096 bytes. Suppose XYZApplication posture plugin's information file defines its posture plugin .dll file name like this:

```
[main]
PluginName=XYZApplicationPP.dll
```

In this example, the name of the new PPMSize parameter for XYZApplication you would create would be **XYZApplicationPP_PPMSize**.

**Note**

If there is a Symantec posture plugin installed on the client, the ctad.ini file must be configured in one of two ways to maintain the transfer of posture plugin messages from any posture plugin to CTA:

- PPMsgSize must be set to 1024 byte
- The Symantec posture plugin must use an application-specific posture plugin, set to 1024 bytes.

-
- Step 1** In the \Program Files\Common Files\PostureAgent\Plugins or /opt/PostureAgent/Plugins directory, find the .inf file for the posture plugin that requires an application-specific PPMsgSize.
- Step 2** Make note of the plugin name in the PluginName field of .inf file.
- Step 3** Locate the ctad.ini file or the ctad-temp.ini template file on the host.
- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
 - For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.
- Step 4** If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as “ctad.ini.”
- Step 5** Locate the [main] section in the ctad.ini file.
- Step 6** Add a new parameter to the ctad.ini file following the naming convention described previously in this section. The minimum and maximum value for all PPMsgSize parameters remains 1024 bytes and 6144 bytes respectively.
- Step 7** Save and close the ctad.ini file.

Configuring PPMsgSize for Host Posture Plugin

If PPMsgSize is less than 4096 bytes and there is no application-specific PPMsgSize parameter set for the host posture plugin, then CTA internally sets the value of CiscoHostPlugin_PPMsgSize at 4096 bytes.

If you create a specific CiscoHostPP_PPMsgSize parameter for the Host posture plugin, it must be set between 4096 bytes and 6144 bytes.

Configuring PPMsgSize for Symantec Posture Plugin

If there is a Symantec posture plugin installed on the client, PPMsgSize must be set to 1024 bytes or the Symantec posture plugin must use an application-specific posture plugin set to 1024 bytes to maintain the transfer of posture plugin messages from any posture plugin to CTA.

Configuring Asynchronous Posture Status Query

CTA can be configured to query posture plugins at regular intervals to determine if there has been a change to their application's status. If a posture plugin alerts CTA that there has been a change in posture status, CTA alerts the network access device which triggers a re-posturing of the host. This is called "asynchronous posture status query." This feature is available on NAC L2 802.1x networks. Asynchronous posture status query can not be used on NAC L2 IP or NAC L3 IP networks. To configure the regular interval at which CTA queries the resident plugins for posture status, perform the following procedure:

-
- Step 1** Locate the ctad.ini file or the ctad-temp.ini template file on the host.
- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
 - For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.
- Step 2** If there is a ctad.ini file in the directory, you can edit that file directly. If there is only a ctad-temp.ini file, make a copy of it and save the copy as "ctad.ini".
- Step 3** Locate the [Main] section in the ctad.ini file.
- Step 4** Delete the semicolon (;) in front of the **SQTimer** parameter.
- Step 5** Change the value of SQTimer to reflect the interval at which you want CTA to query posture plugins for a change in posture status.

Configuring User Notifications

User notifications report the “posture” of the host. The messages are sent from Cisco Secure Access Control Server (ACS) to Cisco Trust Agent (CTA). The notifications are displayed as pop-up messages on the desktop, or login screen, of the system on which CTA is installed. These notifications are the only interactive end-user functionality of CTA.

User notifications are configured in the [UserNotifies] section of the ctad.ini configuration file. Any changes made to the [UserNotifies] section of the ctad.ini configuration file are detected and implemented by CTA the next time a notification is received.

The Windows, Linux, and Mac OS X user notification configurations are optional. You do not need to change the default configuration of user notifications in order for CTA to run properly. After reading [Configuring Windows User Notifications](#) or [Configuring Linux User Notifications](#), follow the [Editing the ctad.ini Configuration File, page 5-3](#) procedure to make the appropriate changes to the ctad.ini file. Use the sample ctad.ini files and the section on [ctad.ini Configuration Parameters, page 5-4](#) as references.

Configuring Windows User Notifications

To configure user notifications, use the [Editing the ctad.ini Configuration File, page 5-3](#) procedure, reference the sample ctad.ini files, and reference the [ctad.ini Configuration Parameters, page 5-4](#).

You can configure the following notification properties on Windows platforms:

- Where the notifications appear.
 - If the **EnableNotifies** parameter is enabled then user notifications are displayed on the desktop, after the user has logged in.
 - If the **EnableLogonNotifies** parameter in the ctad.ini is enabled, then user notifications received before the user is logged on are saved and displayed to the user when they log on.
- How long the notification dialog box displays before it closes automatically.
 - The **MsgTimeout** parameter defines how long the message is displayed on the desktop.

- The **LogonMsgTimeout** Specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies is enabled.
- The **UserActionDelayTimeout** parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message.
- **SysModal** window behavior is enabled by default. The behavior requires a user to close a notification dialog box to continue working. You can change this behavior by creating a ctad.ini file and disabling the parameter in the file.

**Note**

If the user notification dialog box is set to system modal, and there is more than one notification message, only the newest message appears. Responding to the message closes all of the message dialog boxes.

Configuring Linux User Notifications

To configure user notifications, use the [Editing the ctad.ini Configuration File, page 5-3](#) procedure, reference the sample Linux ctad.ini file, and reference the [ctad.ini Configuration Parameters, page 5-4](#).

You can configure the following notification properties:

- If the **EnableLogonNotifies** parameter in the ctad.ini is enabled, then user notifications received before the user is logged on are saved and displayed to the user when they log on.

If a GUI is not installed with the Linux operating system, these notifications are written to a message file in the /var/opt/CiscoTrustAgent/msg directory.
- How long the notification dialog box displays before it closes automatically.
 - The **MsgTimeout** parameter defines how long the message is displayed on the desktop.
 - The **LogonMsgTimeout** specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies is enabled.

- The **DisplayType** parameter allows you to choose between receiving messages in a terminal window or in the GUI.
- The **TermFont** parameter specifies the font that will be displayed in the terminal window.
- The **ClearOldNotifications** parameter clears or saves old notification messages. If ClearOldNotifications is enabled, an old notification is cleared before showing a new notification.
- The **BrowserPath** parameter specifies the full path to the browser CTA should use.
- The **UserActionDelayTimeout** parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message.

Configuring Mac OS X User Notifications

To configure user notifications, use the [Editing the ctad.ini Configuration File, page 5-3](#) procedure, reference the sample Mac OS X ctad.ini file, and reference the [ctad.ini Configuration Parameters, page 5-4](#).

You can configure the following notification properties:

- If the **EnableLogonNotifies** parameter in the ctad.ini is enabled, then user notifications received before the user is logged on are saved and displayed to the user when they log on.
- How long the notification dialog box displays before it closes automatically.
 - The **MsgTimeout** parameter defines how long the message is displayed on the desktop.
 - The **LogonMsgTimeout** specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies is enabled.
- The **ClearOldNotifications** parameter clears or saves old notification messages. If ClearOldNotifications is enabled, an old notification is cleared before showing a new notification.

- The **UserActionDelayTimeout** parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message.
- **SysModal** window behavior is enabled by default. The behavior requires a user to close a notification dialog box to continue working. You can change this behavior by creating a ctad.ini file and disabling the parameter in the file.

**Note**

If the user notification dialog box is set to system modal, and there is more than one notification message, only the newest message appears. Responding to the message closes all of the message dialog boxes.

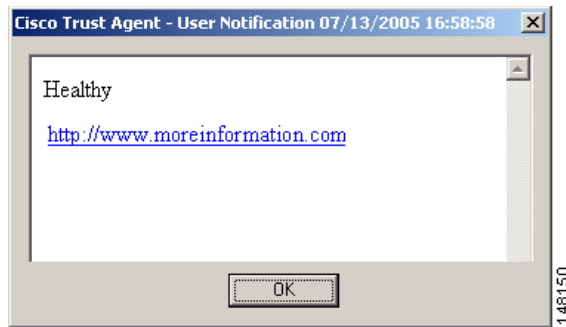
Configuring Clickable URL and Browser Auto-Launch Features

After the overall posture of the client has been determined, the Cisco Secure ACS can be configured to notify the user of that posture. Notifications may be a message in a pop-up window or the user can be directed to a particular URL.

- **Clickable URL**

The pop-up window that notifies the user will contain a posture value, such as “Healthy” or “Quarantine.” It can also contain a clickable URL for the user to follow. For example, if the client requires remediation, the clickable URL can direct the user to an area where a particular remediation solution is available.

Figure 5-1 Clickable URL Notification Pop-up



- **Browser Auto-launch**

Instead of receiving a pop-up window with a clickable URL, the notification can be configured to automatically open a browser window which is already pointing to the URL the user requires.

Logging Notifications

If logging is enabled, the following notification events are logged:

- Notification failures, such as a failure to create the notification dialog box.
- User notification messages.

The messages that are logged are filtered by the logging level assigned to the notification component of CTA. This logging level is specified by the CTAMsg parameter in the [LogLevel] section of the ctalogd.ini configuration file. If this parameter does not appear in the configuration file, or if the configuration file does not exist, then a default level of 3-High is used. This level allows all informational, warning, and critical messages to be written to the log file. The message severity is assigned by the component or plugin that sent the message.

For more information about logging, see [Chapter 6, “Cisco Trust Agent Event Logging”](#).

Certificate Distinguished Name Matching

Certificate distinguished name (DN) matching is performed when CTA communicates with ACS using the NAC L2 IP method. It is not used when CTA communicates with Cisco Secure ACS using NAC L2 802.1x method.

When using CA certificates to validate your ACS server certificate to CTA, you can implement additional security using DN matching to validate the certificate. This prevents other servers or processes that may be using the same root certificate from gaining a trust relationship with the host.

DN matching occurs at the end of the transport layer security (TLS) handshake, after the certificate chain is built. After CTA has received ACS's certificate, this rule is employed to ensure that the properties of the certificate fields have the expected values.

If any rule in the [ServerCertDNVerification] section of the ctad.ini file succeeds then the ACS server is authenticated to CTA.

**Note**

There are no default settings for the [ServerCertDNVerification] section; that section in the sample file contains example information. If you use this sample file as the basis for your own ctad.ini file, delete the section or change it to suit your environment.

DN Matching Rule Syntax

Each rule can contain multiple sub-rules, separated by a comma. If a single sub-rule within a rule fails, then the entire rule fails. The maximum length for a single rule is 255 characters.

The following shows the general format for a rule:

```
Rule1=subrule, subrule, subrule, ...
```

Each sub-rule consists of a certificate subject attribute followed by a value:

```
attribute[operator]"value"
```

The following operators are supported:

- **= (equals)**—The value must exactly match the value given in the subrule.

- ***** (**contains**)—The value must contain the value given in the subrule.

The following DN attributes are supported:

- **CN**—Common Name
- **SN**—Surname
- **GN**—Given Name
- **N**—Unstructured Name
- **I**—Initials
- **GENQ**—Generation Qualifier
- **C**—Country
- **L**—Locality
- **SP**—Province
- **ST**—State
- **O**—Organization
- **OU**—Organization Unit
- **T**—Title
- **EA**—E-mail Address



Note

You can also create sub-rules that check certificate Issuer attributes by adding the **ISSUER-** prefix to the attributes listed above. For example, to validate the Common Name of certificate issuer, you would use the attribute **ISSUER-CN**.

Keeping the syntax in mind, here is an example of two rules:

```
Rule1=CN*"Server", ISSUER-CN*"Finance"
Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"
```

At the end of the TLS handshake the Subject field or Issuer field in the Cisco Secure ACS server certificate are compared to these rules. The DN matching will succeed if one or the other rules succeed.

Rule 1 indicates that the Subject field CN (Common Name) must contain “Server” and the Issuer field CN must contain “Finance.” If both fields do not have the correct information the rule fails.

Rule 2 indicates that the Subject field CN must equal “Finance posture Cert”, the Organization Unit (OU) field must contain “Finance”, and the Issuer CN field must contain “ACME”. If all of these subrules do not have the correct information, the rule fails.

Configuring Certificate DN Matching

-
- Step 1** Locate the ctad.ini file or the ctad-temp.ini template file on the host.
- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
 - For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.
- Step 2** If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as “ctad.ini”.
- Step 3** Locate the [ServerCertDNVerification] section in the ctad.ini file.
- Step 4** Delete the semicolon (;) in front of the TotalRules parameter and equate the parameter to the number of rules you are going to write.
- Step 5** Following the example in the ctad.ini file create a new line for each rule. Begin each line with “RuleX=” where X is the number of the rule.
- Step 6** Write each rule following the [“DN Matching Rule Syntax” section on page 5-25](#).
- Step 7** Save and close the file. The new rules will be implemented the next time DN matching occurs.

Configuring the Scripting Interface

The delta_stale parameter in the [ScriptingInterface] section of the ctad.ini file is the only configurable parameter for the Scripting Interface. This parameter defines the time, in minutes, before the posture data file is considered out of date.

To configure the CTA Scripting Interface, use the [Editing the ctad.ini Configuration File, page 5-3](#) procedure, reference the sample ctad.ini files, and reference the [“ctad.ini Configuration Parameters” section on page 5-4](#).

Sample Windows ctad.ini File

```
; *****
; CTAD.INI FILE DEFINITION
; This file defines communication parameters between a
; Network Access Device (NAD) and Cisco Trust Agent (CTA).
; It also defines variables for notifications and certificate
; filtering rules.
;
; This file can be edited with a plain text editor and used
; in a custom installation of CTA.
;
; The default location for the ctad.ini file is
; the Program Files\CiscoSystems\CiscoTrustAgent\
; directory.
;
; GENERAL EDITING INSTRUCTIONS
; To "turn on" a parameter in a section, delete the ; before
; the ParameterName.
; To "turn off" a parameter, type a ; before the ParameterName.
; *****

; *****
[main]

;The EnableVFT parameter indicates if Validation-Flag TLV
; is enabled on the version of IOS running on the Network Access
; Device (NAD). "EnableVFT" enables CTA to operate with the NAD
; whether it has support for Validation-Flag TLV or not.
;Default Value: 0
;Range of Values: 0, 1
; 0 = IOS does not support Validation-Flag TLV
; 1 = IOS does support Validation-Flag TLV
;EnableVFT=0

;The PPInterfaceType parameter describes how CTA gathers posture
; plugin information.
; Default value: Block
; Range of values: Block, NonBlockConcurrent, NonBlockSerial
; Block = CTA will request posture information from one plug-in at a time.
; NonBlockConcurrent = Request posture information from all posture plugins
; simultaneously.
; NonBlockSerial = Requests posture information from posture plugins one at
; a time and waits for either the return of the posture credentials
; or the end of the PPWaitTimeout value before
; requesting posture credentials from the next posture plugin.
;PPInterfaceType=Block
```

```
;The PPWaitTimeout parameter represents the maximum time allowed,
; in seconds, to complete the processing of all plug-ins.
; Default value: 5 seconds
; Range of values: 1 - 300 seconds
;PPWaitTimeout=5

; The PPMsgSize parameter allows the administrator to modify
; the maximum message size that a posture plugin can send
; from 1k to a maximum value of 6k.
; Default value: 1024 bytes
; Range of values: 1024 - 6144 bytes
;PPMsgSize=1024

;The SQTimer (status query timer) parameter defines
; how often CTA queries the posture plugins to detect a change
; in their status.
; Default value: 300 seconds
; Range of values: 5 - 4294967925 seconds
;SQTimer=300

;The [EAPoUDP] section defines the communication settings between CTA
; and a Layer 3 Network Access Device (NAD), such as a router.
[EAPoUDP]

;The LocalPort parameter defines the port on which the NAD initiates
; posture validation with CTA. Changing this value requires
; changes to the NAD configuration.
; Default value: 21862
; Range of values: 1 - 65550
;LocalPort=21862

;CTA supports one established session per NAD. But CTA can support
; concurrent sessions with multiple NADs. MaxSessions is
; the number of sessions you allow CTA to support.
; Default value: 8
; Range of values: 1 - 256
;MaxSession=8

;The SessionIdleTimeout parameter defines the number of seconds
; an EAP over UDP session can remain idle before timing out.
; Default value: 3600
; Range of values: 60 - 172800
;SessionIdleTimeout=3600

;The BootTimeUDPExemptions parameter alters the Windows Firewall
; policy and enable the CTA to receive packets when the
; Windows XP SP2 or SP3-based computer is starting.
```

Sample Windows ctad.ini File

```
; Default value: 1
; Range of values: 0, 1
; 0 = Windows Firewall Boot Time Exemptions are disabled.
; 1 = Windows Firewall Boot Time Exemptions are enabled.
;BootTimeUDPExemptions=1

;The [UserNotifies] section defines the behavior of pop-up windows
; containing messages sent from ACS to CTA
[UserNotifies]

;SysModal specifies whether the user notification dialog boxes are
; modal or not. If the notification dialog boxes are modal,
; the user must close the notification dialog box to continue
; working. Also, if the dialog boxes are modal, and there is
; more than one notification, only the last notification appears.
; Default value: 1
; Range of values: 0, 1
; 0 = Modal dialog boxes are disabled.
; 1 = Modal dialog boxes are enabled.
;SysModal=1

;Setting this UserActionDelayTimeout parameter allows you to delay the launch of
; the browser window so that the host has more time to obtain an IP address.
; If the browser that displays the posture message is launched before the host
; obtains an IP address, the browser will fail to open the URL contained in the
; posture message.
; Default value: 25 seconds
; Range of values: 0 - 4294967295 seconds
;UserActionDelayTimeout=25

;The EnableNotifies parameter enables or disables user
; notifications. This parameter applies to logged-in users.
; Default value: 1
; Range of values 0, 1
; 0 = User notifications are disabled.
; 1 = User notifications are enabled.
;EnableNotifies=1

;The MsgTimeout parameter specifies how long, in seconds,
; user notification dialog boxes are displayed.
; Default value: 300
; Range of values: 30 - 4294967
; Special value: 0 (disables the timeout)
;MsgTimeout=300

;The EnableLogonNotifies parameter enables or disables user
; notifications received before the user is logged on.
; Default value: 1
```



```
; Range of values: 0, 1
; 0 = User notifications received before the user is logged on are
; discarded.
; 1 = User notifications received before the user is logged on are
; saved and displayed to the user when the log on.
;EnableLogonNotifies=1

;The LogonMsgTimeout Specifies how long, in seconds, a message
; is saved when no user is logged on and when
; EnableLogonNotifies is enabled.
; Default value: 86400
; Range of values: 30 - 4294967
; Special value: 0 (disables the timeout)
; LogonMsgTimeout=86400

;The [ServerCertDNVerification] section contains configurable
; parameters for distinguished name (DN) matching. When
; using CA certificates to validate your Cisco Secure ACS
; server certificate, you can implement additional security
; using distinguished name matching.
[ServerCertDNVerification]

;The TotalRules parameter defines the number of DN matching rules
; that follow it. If the number of rules is greater than 1
; the rules are connected by an OR statement.
; Default value: (none)
; Range of values: 0 - 64
; Special values: 0 (Disables DN matching)
;TotalRules=2

;The RuleX parameters are DN matching rules, where X is the index
; for the rule.
; NOTE: THE RULES BELOW ARE EXAMPLES ONLY. DO NOT USE THEM
; WITHOUT MODIFYING THEM TO SUIT YOUR ENVIRONMENT.
;Rule1=CN*"Server", ISSUER-CN*"Finance"
;Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"

;The parameters in the [Scripting_Interface] section define the
; scripting interface behavior.
[Scripting_Interface]

; The delta_stale parameter specifies how long, in minutes, before
; the posture database record is deemed outdated.
; Default value: 43200
; Range of values: 1-5256000
; Special value: 0 (the database will never expire)
;delta_stale=43200
; *****
```

Sample Linux ctad.ini File

```
; *****
; CTAD.INI FILE DEFINITION
; This file defines communication parameters between a
; Network Access Device (NAD) and Cisco Trust Agent (CTA).
; It also defines variables for notifications and certificate
; filtering rules.
;
; This file can be edited with a plain text editor and used
; in a custom installation of CTA.
;
; The default location of the ctad.ini file is the
; /etc/opt/CiscoTrustAgent/ directory.

; GENERAL EDITING INSTRUCTIONS
; To "turn on" a parameter in a section, delete the ; before
; the ParameterName.
; To "turn off" a parameter, type a ; before the ParameterName.
; *****

; *****
[main]
;The "EnableVFT" parameter indicates if Validation-Flag TLV
; is enabled on the version of IOS running on the Network Access
; Device (NAD). "EnableVFT" enables CTA to operate with the NAD
; whether it has support for Validation-Flag TLV or not.
;Default Value: 0
;Range of Values: 0, 1
; 0 = IOS does not support Validation-Flag TLV
; 1 = IOS does support Validation-Flag TLV
;EnableVFT=0

;The PPInterfaceType parameter describes how CTA gathers posture
; plugin information.
; Default value: Block
; Range of values: Block, NonBlockConcurrent, NonBlockSerial
; Block = CTA will request posture information from one plug-in at a time.
; NonBlockConcurrent = Request posture information from all posture plugins
; simultaneously.
; NonBlockSerial = Requests posture information from posture plugins one at
; a time and waits for either the return of the posture credentials
; or the end of the PPWaitTimeout value before
; requesting posture credentials from the next posture plugin.
;PPInterfaceType=Block
```

```
;The PPWaitTimeout parameter represents the maximum time allowed,
; in seconds, to complete the processing of all plug-ins.
; Default value: 5 seconds
; Range of values: 1 - 300 seconds
;PPWaitTimeout=5

; The PPMsgSize parameter allows the administrator to modify
; the maximum message size that a posture plugin can send
; from 1k to a maximum value of 6k.
; Default value: 1024 bytes
; Range of values: 1024 - 6144 bytes
;PPMsgSize=1024

;The SQTimer (status query timer) parameter defines how often CTA queries the posture
; plugins to detect a change in their status.
; Default value: 300 seconds
; Range of values: 5 - 4294967925 seconds
;SQTimer=300

; The [EAPoUDP] section defines the communication settings
; between CTA and the Network Access Device (NAD).
[EAPoUDP]

;The LocalPort defines the port on which the NAD initiates posture
; validation with CTA. Changing this value requires changes
; to the NAD configuration.
; Default value: 21862
; Rang of values: 1 - 65550
;LocalPort=21862

;CTA supports one established session per NAD. But can support
; concurrent sessions with multiple NADs. MaxSessions is
; the number of sessions you allow CTA to support.
; Default value: 8
; Range of values: 1 - 256
;MaxSession=8

;SessionIdleTimeout defines the number of seconds an EAP over UDP
; session can remain idle before timing out.
; Default value: 3600
; Range of values: 60 - 172800
;SessionIdleTimeout=3600

;The [UserNotifies] section defines the behavior of pop-up windows
; containing messages sent from ACS to CTA
[UserNotifies]
```

Sample Linux ctad.ini File

```
;Setting this UserActionDelayTimeout parameter allows you to delay the launch of
; the browser window so that the host has more time to obtain an IP address.
; If the browser that displays the posture message is launched before the host
; obtains an IP address, the browser will fail to open the URL contained in the
; posture message.
; Default value: 25 seconds
; Range of values: 0 - 4294967295 seconds
;UserActionDelayTimeout=25

;The EnableNotifies parameter enables or disables user
; notifications. This parameter applies to logged-in users.
; Default value: 1
; Range of values 0, 1
; 0 = User notifications are disabled.
; 1 = User notifications are enabled.
;EnableNotifies=1

;The MsgTimeout parameter specifies how long, in seconds,
; user notification dialog boxes are displayed.
; Default value: 300
; Range of values: 30 - 4294967
; Special value: 0 (disables the timeout)
;MsgTimeout=300

;The EnableLogonNotifies parameter enables or disables user
; notifications received before the user is logged on.
; Default value: 1
; Range of values: 0, 1
; 0 = User notifications received before the user is logged on are
; discarded.
; 1 = User notifications received before the user is logged on are
; saved and displayed to the user when the log on.
;EnableLogonNotifies=1

;The LogonMsgTimeout Specifies how long, in seconds, a message
; is saved when no user is logged on and when
; EnableLogonNotifies is enabled.
; Default value: 86400
; Range of values: 30 - 4294967
; Special value: 0 (disables the timeout)
; LogonMsgTimeout=86400

;The DisplayType parameter determines if messages sent from
; ACS to CTA are displayed in a graphic user interface or
; in a terminal window.
; Default value: gui
; Range of values: term, gui
;DisplayType=gui
```

```
;The TermFont parameter sets the font in which to display
; the terminal screen text. Use the default value until
; Cisco Secure ACS can forward localized messages to CTA.
; Default value: -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10636-1
;TermFont=-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10636-1
;TermFont entry below for Asian languages
;TermFont=-misc-zysong18030-medium-r-normal--0-0-0-0-c-0-iso10646-1

; The BrowserPath parameter specifies the full path of the browser on Linux systems.
; For Red Hat Enterprise Linux v3 (Enterprise, Advanced, Workstation)
; use this path: /usr/bin/mozilla
; For Red Hat Enterprise Linux v4 (Enterprise, Advanced, Workstation)
; use this path: /usr/bin/firefox
; BrowserPath=

;The ClearOldNotification parameter clears or saves notification
; messages.
; Default value: 1
; Range of Values:
; 0 = Notification messages are saved.
; 1 = CTA clears the old notification message before displaying
; the new window.
;ClearOldNotification=1

;The [ServerCertDNVerification] section contains configurable
; parameters for distinguished name (DN) matching. When
; using CA certificates to validate your Cisco Secure ACS
; server certificate, you can implement additional security
; using distinguished name matching.
[ServerCertDNVerification]

;The TotalRules parameter defines the number of DN matching rules
; that follow it. If the number of rules is greater than 1
; the rules are connected by an OR statement.
; Default value: (none)
; Range of values: 0 - 64
; Special values: 0 (Disables DN matching)
;TotalRules=2

;The RuleX parameters are DN matching rules, where X is the index
; for the rule.
; NOTE: THE RULES BELOW ARE EXAMPLES ONLY. DO NOT USE THEM
; WITHOUT MODIFYING THEM TO SUIT YOUR ENVIRONMENT.
;Rule1=CN*"Server", ISSUER=CN*"Finance"
;Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER=CN*"ACME"
```

Sample Linux ctad.ini File

```
;The parameters in the [Scripting_Interface] section define the
; scripting interface behavior.
[Scripting_Interface]

; The delta_stale parameter specifies how long, in minutes, before
; the posture database record is deemed outdated.
; Default value: 43200
; Range of values: 1-5256000
; Special value: 0 (the database will never expire)
;delta_stale=43200
```

Sample Mac OS X ctad.ini File

```
; *****
; CTAD.INI FILE DEFINITION
; This file defines communication parameters between a
; Network Access Device (NAD) and Cisco Trust Agent (CTA).
; It also defines variables for notifications and certificate
; filtering rules.
;
; This file can be edited with a plain text editor and used
; in a custom installation of CTA.
;
; The default location of the ctad.ini file is the
; /etc/opt/CiscoTrustAgent/ directory.

; GENERAL EDITING INSTRUCTIONS
; To "turn on" a parameter in a section, delete the ; before
; the ParameterName.
; To "turn off" a parameter, type a ; before the ParameterName.
; *****

; *****
[main]
;The "EnableVFT" parameter indicates if Validation-Flag TLV
; is enabled on the version of IOS running on the Network Access
; Device (NAD). "EnableVFT" enables CTA to operate with the NAD
; whether it has support for Validation-Flag TLV or not.
;Default Value: 0
;Range of Values: 0, 1
; 0 = IOS does not support Validation-Flag TLV
; 1 = IOS does support Validation-Flag TLV
;EnableVFT=0

;The PPInterfaceType parameter describes how CTA gathers posture
; plugin information.
; Default value: Block
; Range of values: Block, NonBlockConcurrent, NonBlockSerial
; Block = CTA will request posture information from one plug-in at a time.
; NonBlockConcurrent = Request posture information from all posture plugins
; simultaneously.
; NonBlockSerial = Requests posture information from posture plugins one at
; a time and waits for either the return of the posture credentials
; or the end of the PPWaitTimeout value before
; requesting posture credentials from the next posture plugin.
;PPInterfaceType=Block
```

Sample Mac OS X ctad.ini File

```
;The PPWaitTimeout parameter represents the maximum time allowed,
; in seconds, to complete the processing of all plug-ins.
; Default value: 5 seconds
; Range of values: 1 - 300 seconds
;PPWaitTimeout=5

; The PPMsgSize parameter allows the administrator to modify
; the maximum message size that a posture plugin can send
; from 1k to a maximum value of 6k.
; Default value: 1024 bytes
; Range of values: 1024 - 6144 bytes
;PPMsgSize=1024

;The SQTimer (status query timer) parameter defines how often CTA queries the posture
; plugins to detect a change in their status.
; Default value: 300 seconds
; Range of values: 5 - 4294967925 seconds
;SQTimer=300

; The [EAPoUDP] section defines the communication settings
; between CTA and the Network Access Device (NAD).
[EAPoUDP]

;The LocalPort defines the port on which the NAD initiates posture
; validation with CTA. Changing this value requires changes
; to the NAD configuration.
; Default value: 21862
; Rang of values: 1 - 65550
;LocalPort=21862

;CTA supports one established session per NAD. But can support
; concurrent sessions with multiple NADs. MaxSessions is
; the number of sessions you allow CTA to support.
; Default value: 8
; Range of values: 1 - 256
;MaxSession=8

;SessionIdleTimeout defines the number of seconds an EAP over UDP
; session can remain idle before timing out.
; Default value: 3600
; Range of values: 60 - 172800
;SessionIdleTimeout=3600

;The [UserNotifies] section defines the behavior of pop-up windows
; containing messages sent from ACS to CTA
[UserNotifies]
```



```
;Setting this UserActionDelayTimeout parameter allows you to delay the launch of
; the browser window so that the host has more time to obtain an IP address.
; If the browser that displays the posture message is launched before the host
; obtains an IP address, the browser will fail to open the URL contained in the
; posture message.
; Default value: 25 seconds
; Range of values: 0 - 4294967295 seconds
;UserActionDelayTimeout=25
```

```
;The EnableNotifies parameter enables or disables user
; notifications. This parameter applies to logged-in users.
; Default value: 1
; Range of values 0, 1
; 0 = User notifications are disabled.
; 1 = User notifications are enabled.
;EnableNotifies=1
```

```
;The MsgTimeout parameter specifies how long, in seconds,
; user notification dialog boxes are displayed.
; Default value: 300
; Range of values: 30 - 4294967
; Special value: 0 (disables the timeout)
;MsgTimeout=300
```

```
;The EnableLogonNotifies parameter enables or disables user
; notifications received before the user is logged on.
; Default value: 1
; Range of values: 0, 1
; 0 = User notifications received before the user is logged on are
; discarded.
; 1 = User notifications received before the user is logged on are
; saved and displayed to the user when the log on.
;EnableLogonNotifies=1
```

```
;The LogonMsgTimeout Specifies how long, in seconds, a message
; is saved when no user is logged on and when
; EnableLogonNotifies is enabled.
; Default value: 86400
; Range of values: 30 - 4294967
; Special value: 0 (disables the timeout)
; LogonMsgTimeout=86400
```

```
;The ClearOldNotification parameter clears or saves notification
; messages.
; Default value: 1
; Range of Values:
; 0 = Notification messages are saved.
; 1 = CTA clears the old notification message before displaying
```

Sample Mac OS X ctad.ini File

```
;
; the new window.
;ClearOldNotification=1

;The [ServerCertDNVerification] section contains configurable
; parameters for distinguished name (DN) matching. When
; using CA certificates to validate your Cisco Secure ACS
; server certificate, you can implement additional security
; using distinguished name matching.
[ServerCertDNVerification]

;The TotalRules parameter defines the number of DN matching rules
; that follow it. If the number of rules is greater than 1
; the rules are connected by an OR statement.
; Default value: (none)
; Range of values: 0 - 64
; Special values: 0 (Disables DN matching)
;TotalRules=2

;The RuleX parameters are DN matching rules, where X is the index
; for the rule.
; NOTE: THE RULES BELOW ARE EXAMPLES ONLY. DO NOT USE THEM
; WITHOUT MODIFYING THEM TO SUIT YOUR ENVIRONMENT.
;Rule1=CN*"Server", ISSUER-CN*"Finance"
;Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"

;The parameters in the [Scripting_Interface] section define the
; scripting interface behavior.
[Scripting_Interface]

; The delta_stale parameter specifies how long, in minutes, before
; the posture database record is deemed outdated.
; Default value: 43200
; Range of values: 1-5256000
; Special value: 0 (the database will never expire)
;delta_stale=43200
```



CHAPTER 6

Cisco Trust Agent Event Logging

CTA logging is disabled by default because CTA is intended to be a transparent application to end users. If you enable logging, CTA logs events generated by CTA components and the posture plugins for the NAC-compliant applications that reside on the system.

This chapter contains the following sections:

- [How Logging Works, page 6-2](#)
- [CTA Log Files, page 6-2](#)
 - [Log File Format, page 6-3](#)
 - [Logging Considerations, page 6-4](#)
- [The clogcli Logging Utility, page 6-4](#)
 - [Logging Levels, page 6-11](#)
- [Configuring CTA Logging For Large Deployments, page 6-11](#)
- [Sample ctalogd-temp.ini File, page 6-13](#)

How Logging Works

Event logging is implemented as a service on the network client. When the CTA Event Logger service starts, it reads the logging configuration file, `ctalogd.ini` and uses any logging levels and settings that are specified. If no logging levels or settings are specified in the `ctalogd.ini` file, the logging service uses its default values.

If logging is enabled, the events are evaluated against the default logging levels or those defined in the `ctalogd.ini` file. Events that meet the logging level are formatted and written to the log file.

**Note**

CTA logging is disabled by default and can be enabled by using the `clogcli` utility, see [“The clogcli Logging Utility” section on page 6-4](#) for more information.

CTA Log Files

CTA log files are text files created by the Cisco Trust Agent Event Logging service. They are ASCII text files that you can view using any text editor. Log file names are automatically generated by the event logger whenever a log file is created. A new log file is created when one of the following events occur:

- Logging is changed from disabled to enabled.
- The log file is cleared while logging is enabled.
- The current log file reaches the maximum file size.

**Note**

The creation of the log file does not occur until the first event is received while logging is enabled.

On **Windows** operating systems, this is the default location of log files:

```
\Program Files\Cisco Systems\CiscoTrustAgent\Logging\Logs
```

On **Linux** and **Mac OS X** operating systems, this is the default location of log files:

```
/var/log/CiscoTrustAgent
```

These directory locations can be changed using the `clogcli` utility. See, “[The clogcli Logging Utility](#)” section on page 6-4 for this procedure.

The logfile names use the date and time the event logger was started to create unique file names. The log file names contain the following format:

CTALOG-YYYY-MM-DDTHH-MM-SS_N.txt.

- **CTALOG**—A fixed prefix indicating that CTA created the log.
- **YYYY**—A four-digit value for the year.
- **MM**—A two-digit value for the month.
- **DD**—A two-digit value for the day.
- **T**—A fixed separator between the date and time.
- **HH**—A two-digit value for the hours, specified in 24-hour time.
- **MM**—A two-digit value for the minutes.
- **SS**—A two-digit value for the seconds.
- **N**—The n^{th} log file created since the event logger was started. This occurs when the current log file reaches the maximum size or when logging is disabled and then enabled.

For example, if the event logger was started on September 20, 2006 at 5:12:58 p.m. the generated log file name would be named

CTALOG-2006-09-20T17-12-58_1.txt.

Log File Format

The log file contains the following fields:

- **Logging Instance**—An incremental number for the log entry.
- **Date/Time**—The date and time the entry was logged.
- **Severity**—The severity of the logged event. Valid severity values are:
 - Critical
 - Info
 - Warning
- **Error Code**—The error code associated with the event.
- **Message Body**—Text describing the event.

The following displays examples of log file entries:

Example 6-1 CTA Log File Sample Entries

```
Cisco Systems Trust Agent Version 2.0
Copyright (C) 1998-2003 Cisco Systems, Inc. All Rights Reserved.
Trust Agent Type(s): Windows, WinNT Running on: 5.1.2600
```

```
1 15:29:57.748 07/13/05 Sev=Info/3 PADaemon/0x6300000C
Starting service:
```

```
2 15:30:40.719 07/13/05 Sev=Info/3 NetTrans/0x63100016
NAD proposed AssociationID 56789
```

Logging Considerations

Log files remain on the disk until a user deletes them, the allotted disc space for logs is used up, or the log file reaches a certain age.

If you enable a high-level of logging, you can potentially fill the disk over time. If you enable logging by default, ensure that a policy for regular log file removal or archival is in place.

The clogcli Logging Utility

CTA provides a utility, **clogcli**, which can be run locally on the end user system. clogcli provides a way for users to enable, disable, and configure logging. This utility is useful in situations where local system troubleshooting is required. The clogcli utility runs at the command line.

On Windows operating systems, clogcli is stored in this directory:

```
\Program Files\Cisco Systems\CiscoTrustAgent\
```

On Linux and Mac OS X operating systems, clogcli is stored in this directory:

```
/opt/CiscoTrustAgent/sbin
```

To run the clogcli utility, follow this procedure:

- Step 1** Open a Command Prompt Window (Windows) or a Terminal Window (Linux and Mac OS X).
- Step 2** Change the directory to the one in which clogcli is stored:
- For Windows operating systems, change the directory to:
 \Program Files\Cisco Systems\CiscoTrustAgent\
 - For Linux and Mac OS X operating systems, change the directory to:
 /opt/CiscoTrustAgent/sbin
- Step 3** At the prompt type **clogcli**, followed by the proper command, and press <Enter>. The command syntax is the same for Windows and Linux operating systems and slightly different for Mac OS X operating systems.

Table 6-1 describes all the commands and options for the clogcli utility.

Table 6-1 *clogcli Utility Command Option*

Option	Description and example
clear	Clears the current log file. A new log file is created if logging is enabled. Linux and Windows example: #> clogcli clear Mac OS X example: \$./clogcli clear
disable	Disables logging. Parameter name in ctagd.ini: EnableLog=0 Linux and Windows example: #> clogcli disable Mac OS X example: \$./clogcli disable

Table 6-1 *clogcli Utility Command Option (continued)*

Option	Description and example
enable	<p>Enables logging.</p> <p>Parameter name in ctagd.ini: EnableLog=1</p> <p>Linux and Windows example:</p> <pre>#> clogcli enable</pre> <p>Mac OS X example:</p> <pre>\$./clogcli enable</pre>
enable -t	<p>Enables logging until the machine is rebooted.</p> <p>Parameter name in ctagd.ini: EnableLog=2</p> <p>Linux and Windows example:</p> <pre>#> clogcli enable -t</pre> <p>Mac OS X example:</p> <pre>\$./clogcli enable -t</pre>

Table 6-1 *clogcli Utility Command Option (continued)*

Option	Description and example
logdir	<p>Sets the log file location for CTA logs.</p> <p>Parameter name in ctalogd.ini: LogDir</p> <p>Default Windows Location: \Program Files\Cisco Systems\CiscoTrustAgent\Logging\Logs</p> <p>Default Linux Location: /var/log/CiscoTrustAgent</p> <p>Range of Values: Any existing directory location.</p> <p>Windows example:</p> <pre>> clogcli logdir c:\Temp\CTALogs</pre> <p>Linux example:</p> <pre># clogcli logdir /tmp/CTALogs</pre> <p>Mac OS X example:</p> <pre>\$./clogcli logdir /tmp/CTALogs</pre> <p>Note This note is for Linux systems only.</p> <p>For security reasons, ctalogd does not run with “administrator” or “root” permissions. Therefore, it is possible for you to specify a directory that ctalogd does not have permission to write to.</p> <p>If ctalogd is not able to create the log file in the directory you specify, then it will create the log file in the default logging directory. You should also see an error message in the syslog. After you begin logging, check both your chosen and the default directories to verify the location of your log file.</p>

Table 6-1 *clogcli Utility Command Option (continued)*

Option	Description and example
loglevel	<p>Sets the log level for all CTA components at once and to the same level. See, “Logging Levels” section on page 6-11 for descriptions of the logging levels.</p> <p>Default Value: 3</p> <p>Range of values: 1-3, 15</p> <p>Linux and Windows example:</p> <pre>#> clogcli loglevel 2</pre> <p>Mac OS X example:</p> <pre>\$./clogcli loglevel 2</pre>
maxdisk	<p>Sets the maximum number of megabytes of space that may be used for logging.</p> <p>Parameter name in ctalogd.ini: MaxDiskSize</p> <p>Default Value: 50</p> <p>Range of Values: 50-100</p> <p>Special Value: 0 - If maxdisk is set to zero then there is no disk space limit for log files.</p> <p>Linux and Windows example:</p> <pre>#> clogcli maxdisk 60</pre> <p>Mac OS X example:</p> <pre>\$./clogcli maxdisk 60</pre>

Table 6-1 *clogcli Utility Command Option (continued)*

Option	Description and example
maxfile	<p>Sets the minimum log file size in megabytes.</p> <p>Parameter name in ctalogd.ini: MaxFileSize</p> <p>Default Value: 4 MB</p> <p>Range of Values: 0 - 50 MB.</p> <p>Special Value: 0 - If maxfile is set to zero, then there is no limit on the log size.</p> <p>Linux and Windows example:</p> <pre>#> clogcli maxfile 5</pre> <p>Mac OS X example:</p> <pre>\$./clogcli maxfile 5</pre>

Table 6-1 *clogcli Utility Command Option (continued)*

Option	Description and example
zipit	<p>This command retrieves the following information and inserts the files into a zip file which is stored on the desktop on Windows and in the user's home directory on Linux or Mac OS X operating systems.</p> <p>Linux and Windows example:</p> <pre>#> clogcli zipit</pre> <p>Mac OS X example:</p> <pre>\$./clogcli zipit</pre> <p>The zip file is named: <i>NAC-TS-YYYY-MM-DDTHH-MM-SS.zip</i></p> <p>Where “NAC-TS” stands for Network Admission Control - Technical Support. The log file follows the Year-Month-DayTHours-Minutes-Seconds convention of the ctalog file.</p> <p>These are the files collected by the clogcli zipit command:</p> <ul style="list-style-type: none"> CTA log files: <ul style="list-style-type: none"> \Program Files\Cisco Systems\CiscoTrustAgent\Logging\Logs\CTALOG*.txt CTA 802.1x Wired Client log files, if CTA 802.1x Wired Client is installed: <ul style="list-style-type: none"> \Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\system\log\apiDebug*.txt \Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\system\log\clientDebug*.txt \Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\log*.txt Output from ctastat: ctastat-output.txt ctad.ini file ctalogd.ini file

Logging Levels

Setting the logging level determines which events are added to the log file. Each logging level provides an incremental addition to the logging information provided by the level below it.

- **1-LOW** — Low-level logging includes critical events. The intent of the low setting is to ensure that your log file does not grow too large while still logging the events that are likely most worth your attention.
- **2-MEDIUM** — Medium-level logging includes warnings and the critical events provided in the low setting.
- **3-HIGH** — High-level logging includes informational events, such as success messages, warnings, and critical events.
- **15-EVERYTHING** — This is the most verbose level of logging. It captures all messages.

HIGH is the default level used when logging is enabled. You can change the logging level for each CTA component using the `clogcli` utility.

As a general guideline, when troubleshooting problems on systems running CTA, keep the logging level set to **3** to receive the most information about the system operation. If you configure logging to be enabled and there are no issues, you should set the logging level to Medium or Low. Setting the logging level to medium or low prevents the log file from growing rapidly and consuming disk space, yet provides enough information to diagnose any possible problems with CTA, posture plugins, or system posture.

Configuring CTA Logging For Large Deployments

To deploy a specialized level of logging state, logging size, log file locations and log file size, you can create a `ctalogd.ini` file that can be distributed in a custom installation package.

The `ctalogd.ini` file contains the configuration parameters for CTA logging. The `ctalogd.ini` file is not delivered at the time of installation; it is created when you use the `clogcli` utility to enable logging and change logging attributes.

CTA logging is disabled by default. Once logging is enabled, the default logging attributes are used unless you specify different attributes in the `ctalogd.ini` file.

To configure CTA logging, follow this procedure:

-
- Step 1** Use the simplest method to install CTA without the 802.1x Wired Client, on a test machine. See, [Chapter 2, “Installing the Cisco Trust Agent on Linux Operating Systems”](#), [Chapter 3, “Installing the Cisco Trust Agent on Macintosh Operating Systems”](#), or [Chapter 4, “Installing the Cisco Trust Agent on Windows Operating Systems”](#) for the appropriate installation method.
- Step 2** Verify that CTA has been installed and is running by using the “Verifying Cisco Trust Agent Installation” procedure in the installation chapter.
- Step 3** Enable logging by using the **clogcli enable** command as described in the “[The clogcli Logging Utility](#)” section on page 6-4.
- Step 4** On the test machine, start the resident file management application and change the directory to the /etc/opt/CiscoTrustAgent directory on Linux and Mac OS X operating systems or the \Programs\Cisco Systems\CiscoTrustAgent\Logging directory on Windows operating systems. You should see the ctalogd.ini file in that directory.
- Step 5** Read the ctalogd.ini file. You should see one section and one entry in the file:
- ```
[main]
EnableLog=1
```
- Step 6** Close the ctalogd.ini file.
- Step 7** Continuing to use the clogcli logging utility, specify the logging attributes and logging levels you desire.
- Step 8** When you are done configuring the logging attributes, you can distribute the ctalogd.ini file to an individual machine, by storing it in the /etc/opt/CiscoTrustAgent directory on Linux and Mac OS X operating systems or the \Programs\Cisco Systems\CiscoTrustAgent\Logging directory on Windows operating systems. You can also distribute the ctalogd.ini file to many machines when you perform a CTA installation with a custom installation package. These machines will all use the logging attributes you specified in the file.

# Sample ctaglogd-temp.ini File

CTA logging is disabled by default. The clogcli utility is designed to create the ctaglogd.ini file which defines logging parameters for a local installation. There is also a ctaglogd-temp.ini file that is shipped with CTA as a template file. Advanced users can edit the file directly if they so choose.

The ctaglogd-temp.ini file is delivered to this location on Windows operating systems: \Program Files\Cisco Systems\CiscoTrustAgent\Logging

The ctaglogd-temp.ini file is delivered to this location on Linux and Mac OS X operating systems: /etc/opt/CiscoTrustAgent

See [Example 6-2 on page 6-13](#) for an example of the ctaglogd-temp.ini file.

## Example 6-2 Sample ctaglogd-temp.ini File

```
; This file contains Cisco Trust Agent log settings.
; To use these settings, save a copy of this file as ctaglogd.ini
; and edit that file.
; This file may be used for Linux, Mac OS X, and Windows operating
; systems.

[main]
; This section turns logging on or off and defines the size, age and
; location of CTA log files.
EnableLog=0
; 0 = disable logging
; 1 = enable logging
MaxFileSize=4
MaxDiskSize=50
FileDeleteAge=30
LogDir=/var/log/CiscoTrustAgent
;LogDir=D:\Program Files\Cisco Systems\CiscoTrustAgent\Logging\Logs

[LogLevel]
; This section allows you to set the logging levels
; for various components of the Cisco Trust Agent.
; 0 = disable
; 1 = log critical events only
; 2 = log critical and warning events
; 3 = log critical, warning, and informational events
; 15 = log all events and messages
```

**Sample ctalogd-temp.ini File**

```
PADaemon=3
NetTrans=3
PAPugin=3
CTAMsg=3
CTAD=3
PEAP=3
EAPTLV=3
EAPSQ=3
PPMgr=3
PSDaemon=3
HostPP=3
CTASC=3
CTASTATE=3
```





## CHAPTER 7

# Posture Plugins

---

Posture plugins are the means by which Cisco Trust Agent (CTA) retrieves posture credentials from NAC-compliant applications installed on a client.

Typically, two files comprise a posture plugin. A posture plugin for Windows consists minimally of a dynamic link library “.dll” file and an information “.inf” file. A posture plugin for Linux consists of a shared object “.so” file and an information “.inf” file.

Posture plugins gather posture credentials from NAC-compliant applications. Posture credentials are information about an application that determines the trust the network should have in the security of that application. Posture credentials may include these kinds of attributes: application name, application version, application release date, or proprietary application settings or configurations. Posture credentials can be different for each NAC-compliant application.

Once the posture plugin gathers the posture credentials, it sends them to the CTA. CTA sends the posture credentials to Cisco Secure Access Control Server (ACS) which determines a posture token for each application that provided credentials and an overall posture token for the entire client. The posture token is communicated to the client in a posture notification message.

In the Network Admission Control (NAC) environment, the value of a client’s posture token determines the level of network access the client is allowed. When we refer to a client’s “posture,” we are referring to the value of the client’s posture token.

See [“Initial Posture Validation Process” section on page 1-2](#) and [“Posture Revalidation Process” section on page 1-4](#) for a description of the workflows and conditions surrounding these events.

This chapter contains the following sections:

- [Types of Posture Plugins Installed by Default, page 7-2](#)
  - [Host Posture Plugin, page 7-2](#)
  - [Cisco Trust Agent Posture Plugin, page 7-5](#)
  - [CTA Scripting Posture Plugin, page 7-9](#)
- [Plugin Installation and Upgrade, page 7-9](#)

Also see the “[Configuring Posture Plugins](#)” section on [page 5-13](#) for information about configuring posture plugins.

## Types of Posture Plugins Installed by Default

When CTA is installed these posture plugins are installed by default.

- [Host Posture Plugin](#)
- [Cisco Trust Agent Posture Plugin](#)

The [CTA Scripting Posture Plugin](#) posture plugin is installed if the scripting interface was installed.

Other plugins may also be installed, such as the posture plugin for Cisco Security Agent or a partner in the Network Admission Control program. These plugins are not discussed in this chapter.

## Host Posture Plugin

The Host Posture Plugin retrieves basic information about the host and returns it to the ACS.

The default location of the Windows host posture plugin is the `\Program Files\Common Files\PostureAgent\Plugins` directory. The plugin consists of two files: `CiscoHostPP.dll` and `CiscoHostPP.inf`.

The default location of the host posture plugin on Linux and Mac OS X operating systems is the `/opt/PostureAgent/Plugins` directory. The plugin consists of two files: `CiscoHostPP_unix.inf` and `cischohostpp.so`.

These are the attributes that are returned by the host posture plugin:

**Table 7-1**      **Host Posture Plugin Attributes**

| Condition name in Posture Validation Policies page of ACS | Attribute Description                                        | Used by these operating system |
|-----------------------------------------------------------|--------------------------------------------------------------|--------------------------------|
| Cisco:Host:ServicePacks                                   | Windows service packs that have been installed on the client | Windows                        |
| Cisco:Host:HotFixes                                       | Windows hot fixes that have been installed on the client     | Windows                        |
| Cisco:Host:HostFQDN                                       | Machine name                                                 | Windows                        |
| Cisco:Host:Package                                        | Version and string version of rpm packages                   | Linux and Mac OS X             |
| Cisco:Host:MACAddress                                     | Machine's MAC address                                        | Linux, Mac OS X, Windows       |

**Note**

The Host posture plugin only reports Windows operating system hotfixes. It does not report application hotfixes.

## MAC Address Information Returned by Host Posture Plugin

The Host Posture Plugin reports basic information about the client running CTA to the ACS. With the release of CTA 2.1, the Host Posture Plugin can now return the MAC address of the client running CTA, provided that the MacAddress attribute has been added to the Posture-Validation Attribute Definition File employed by the ACS CSUtil database utility. (For more information about the ACS CSUtil database utility and the Posture-Validation Attribute Definition File, see the *User Guide for Cisco Secure ACS for Windows Server*.)

The attribute information for MACAddress is below.

```
[attr#n]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00009
attribute-name=MACAddress
attribute-profile=in
```

```
attribute-type=string
```

The plugin will return all the MAC addresses available on the client running CTA and combine them into one string; the MAC addresses will be separated by pipes ( | ). For example, a wireless network card and a wired network card will each return a MAC address.

If you are defining a posture validation rule in ACS based on only one of these MAC addresses, the posture attribute should “contain” the MAC address you are verifying rather than “equal” or “start with” the MAC address you are verifying.

## Package Information Retrieved by Host Posture Plugin for Linux Platforms

In addition to the information defined by the host posture plugin attributes, the host posture plugin for Linux and Mac OS X platforms allows you to retrieve the version number of certain packages pre-defined in ACS.

The version number of the package may be expressed in one of these forms:

- A string.
- A representation of the version number in the form of x.x.x.x. In this case, the version numbers is converted to a 4-octet version number using an algorithm.

Here is an example of how CTA returns package version number using each format: Assume there is a posture validation rule from ACS that requests the version number of OpenSSL. When requested as a string, the version number would be returned as a combination of numbers and letters, such as 0.9.7a. When requested as a 4-octet number, the version number that would be return is 0.9.7.97.

Creating posture validation rules on ACS that request package version information as numbers rather than strings allows you to apply operators in the ACS rule such as “greater than or equal to,” “greater than,” “less than,” or “less than or equal to.” These operators do not apply to strings. The string format is beneficial when too much information is lost when using the numeric format. For example, a package might have a version “rockie-mnt-rel-Feb”, in this case the converted numeric version is reported as “0.0.0.0”. In this case, the string version of the package is more meaningful.

When using the Host Posture Plugin on Linux operating systems to retrieve package information, the string value of a package version can be determined in advance of making the ACS rule by running the “rpm -q package-name” command, for example, “rpm -q openssl”. The 4-octet value of a package version is determined from the same output of the “rpm -q package-name” command.

## Package Information Retrieved by Host Posture Plugin for Mac OS X Platforms

For Mac OS X, there are two types of applications that are of concern to CTA: system applications which have receipts in /Library/Receipts/ and user applications which are installed in /Applications directory.

System applications are identified by the first level folder name under /Library/Receipts, like "Danish.pkg", "X11SDK.pkg". User applications are identified by the application name under /Applications directory as displayed in Finder. For example, “Firefox”, “DVD\ Player”.

The applications located in the subfolders of /Applications directory can also be queried, in these cases the package name looks like the relative path to /Applications. For example, "Utilities/Disk\ Utility", "Zinio/Zinio\ Reader".



### Note

---

White spaces in package names must be escaped with backslash (“\”).

---

The version information of system applications is parsed out of the Contents/version.plist file under the package's directory under the /Library/Receipts directory. Version information is in the form of "a.b.c.d". The first three fields of version are from the CFBundleShortVersionString key, and the fourth field is from SourceVersion key. For user application packages, the version information is retrieved from the Info.plist file under the Contents/ directory in the application's directory. We first look for the value of CFBundleShortVersionString key. If this key is not present we will return the value of CFBundleVersion key. If both keys are missing no information will be returned for the package.

## Cisco Trust Agent Posture Plugin

By default, CTA installs and registers a posture plugin that provides information about itself. The CTA posture plugin returns information such as CTA's name, version number, and the name of the operating system on which it runs.

Types of Posture Plugins Installed by Default

The default location of the Windows CTA posture plugin is the \Program Files\Common Files\PostureAgent\Plugins directory. The plugin consists of two files: ctapp.dll and ctapp.inf.

The default location for the CTA posture plugin on Linux and Mac OS X operating systems is the /opt/PostureAgent/Plugins directory. The plugin consists of two files: ctapp.so and ctapp\_unix.inf.

The CTA plugin returns many of the same attributes for all operating systems. You can use the information in table 7-2 when defining rules for the default Cisco Trust Agent plugin in ACS:

Table 7-2 CTA Posture Plugin Attributes and Definitions

| Condition name in Posture Validation Policies page of ACS | Attribute Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Used by this operating system |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Cisco:PA:Application-Posture-Assessment                   | This is the value of the Application-posture token from CTA.                                                                                                                                                                                                                                                                                                                                                                                                                                  | Linux, Mac OS X, Windows      |
| Cisco:PA:Kernel-Version                                   | Kernel version, which is the same as the output of the “uname -r” command.                                                                                                                                                                                                                                                                                                                                                                                                                    | Linux, Mac OS X,              |
| Cisco:PA:MachinePostureState                              | Contains the running status of the machine.<br>Linux and Mac OS X return these values: <ul style="list-style-type: none"> <li>Booting (ACS value = 1)</li> <li>Running (ACS value = 2)</li> </ul> Windows platforms return these values: <ul style="list-style-type: none"> <li>Booting (ACS value = 1)</li> <li>Running (ACS value = 2)</li> <li>Logged in (ACS value = 3)</li> </ul> See <a href="#">Machine Posture State, page 7-8</a> for more information on the use of this attribute. | Linux, Mac OS X, Windows.     |
| Cisco:PA:OS-Release                                       | A string that contains the OS Kernel name, version, and hardware platform.                                                                                                                                                                                                                                                                                                                                                                                                                    | Linux, Mac OS X,              |

| Condition name in Posture Validation Policies page of ACS | Attribute Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Used by this operating system  |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| Cisco:PA:OS-Type                                          | <p>Contains the name of the operating system running on the client.</p> <p>Linux platforms return these names:</p> <ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux WS</li> <li>• Red Hat Enterprise Linux ES</li> <li>• Red Hat Enterprise Linux AS</li> </ul> <p>Mac OS X returns these names:</p> <ul style="list-style-type: none"> <li>• Mac OS Panther</li> <li>• Mac OS Tiger</li> </ul> <p>Windows platforms return these names:</p> <ul style="list-style-type: none"> <li>• Windows Server 2003 Enterprise Edition</li> <li>• Windows Server 2003 Web Edition</li> <li>• Windows Server 2003 Standard Edition</li> <li>• Windows XP Home Edition</li> <li>• Windows XP Professional</li> <li>• Windows 2000 Advanced Server</li> <li>• Windows 2000 Server</li> </ul> | Linux,<br>Mac OS X,<br>Windows |
| Cisco:PA:OS-Version                                       | The operating system version number in the format <i>major.minor.sustaining.build</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Linux,<br>Mac OS X,<br>Windows |
| Cisco:PA:PA-Name                                          | Contains the name of the posture agent running on the client. In this case, the name would be Cisco Trust Agent.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Linux,<br>Mac OS X,<br>Windows |
| Cisco:PA:PA-Version                                       | The Cisco Trust Agent version number in the format <i>major.minor.sustaining.build</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Linux,<br>Mac OS X,<br>Windows |

Types of Posture Plugins Installed by Default

| Condition name in Posture Validation Policies page of ACS | Attribute Description                                                                                             | Used by this operating system |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Cisco:PA: System-Posture-Assessment                       | This indicates the overall posture token of a client on which CTA runs.                                           | Linux, Mac OS X, Windows      |
| Cisco:PA:User-Notification                                | Contains an informational message that the Cisco Trust Agent displays to the user on request from the ACS server. | Linux, Mac OS X, Windows      |

### Machine Posture State

Machine posture state is provided by CTA to inform ACS about the status of the machine when it boots. One of the following states can be reported:

- Booting
- Running
- Logged in (Not supported on Linux)

Booting is the initial state when the machine is started. The state is set to running when all services are started. On Linux, running is the final state because there is no simple way to determine if a user has logged in. On Windows, when a user has logged in, the logged in state is available. The machine posture state is set once a user has logged into the machine. If the user logs out of the machine, the state is set back to the running state until another user logs in.

This state is used with ACS to determine the posture of a machine based on the rules set up for different policies. For example, the following policy could be set on ACS.

| Rule                                                              | Posture Token |
|-------------------------------------------------------------------|---------------|
| Antivirus Enabled = TRUE                                          | Healthy       |
| Antivirus Installed and Cisco Trust Agent Machine State = Booting | Transitional  |
| Antivirus Installed = FALSE                                       | Quarantine    |



## CTA Scripting Posture Plugin

The ctascriptPP retrieves the posture credentials requested by a third party script.

A binary posture plugin consists of two parts, a dynamic link library (.dll) file for Windows systems, or a shared object (.so) file for Linux systems, and a .inf file. The .inf file points to the .dll (or .so) file that retrieves the posture credentials. Typically, these are pairs of files; one .inf file is associated with one .dll or .so file.

A script can be substituted for a binary posture plugin. However, the script still needs the .dll (or .so) file and the .inf file to retrieve the posture credentials. The .inf file supplied by the third party must always point to the ctascriptPP file. In the case of a script, many unique .inf files point to one .dll (or .so) and that is the ctascriptPP file.

See [Chapter 11, “Using the Scripting Interface”](#) for more information on the CTA scripting interface.

## Plugin Installation and Upgrade

Each NAC-compliant application is responsible for installing its own posture plugin on end systems.

Plugins for Windows environments are installed in this directory:

```
\Program Files\Common Files\PostureAgent\Plugins\Install
```

Plugins for Linux and Mac OS X environments are installed in this directory:

```
/opt/PostureAgent/Plugins/install
```

When CTA receives a posture request, it scans the PostureAgent\Plugins\Install directory for new or updated posture plugins. If there are new or updated posture plugins in the PostureAgent\Plugins\Install directory, CTA performs one of the following actions:

- If the .dll (Windows) or the .so (Linux and Mac OS X) plugin **does not exist** in the PostureAgent\Plugins directory, CTA moves the plugin files from the PostureAgent\Plugins\Install directory to the PostureAgent\Plugins directory.

- If the .dll (Windows) or the .so (Linux and Mac OS X) plugins **does exist** in the PostureAgent\Plugins directory, then CTA checks to see if the plugin, in the PostureAgent\Plugins\Install directory, is newer than the one in the Plugins directory. CTA then moves the newer plugin to the PostureAgent\Plugins directory and overwrites the older one. If the plugin in the PostureAgent\Plugins\Install directory is older than the one in the Plugins directory, CTA deletes it, and continues to use the original plugin.
- If the plugin creates an error during registration, CTA moves the plugin to one of the following directories (if the logging is enabled, the error information is logged):

Windows:

```
\Program Files\Common Files\PostureAgent\Plugins\Quarantine
```

Linux and Mac OS X:

```
/opt/PostureAgent/Plugins/Quarantine
```



---

**Note**

---

Quarantined plugins do not participate in posture validation.

---

You do not need to install CTA before other NAC-compliant applications in order for CTA to make use of their plugins. All plugins are stored in a common directory. When CTA receives a request for posture credentials, it checks the common directory for new plugins before it proceeds to retrieve the posture credentials.



## CHAPTER 8

# Cisco Trust Agent's Use of Certificates

---

CTA uses certificates to establish a PEAP and an EAP FAST session with Cisco Secure Access Control Server (ACS). You need to install the ACS root certificate on the client system for this session to be established.

Typically, this certificate is installed as part of a custom Cisco Trust Agent installation package. If it was not installed, CTA provides a the ctacert utility for installing and updating the posture validation server certificate on the client.

If you have installed the 802.1X Wired Client you may perform machine and user authentication using certificates. You can configure CTA for this authentication after the ACS root certificate has been installed.

This chapter contains the following sections:

- [About The ACS Server Root Certificate, page 8-3](#)
- [About The ctacert Utility, page 8-3](#)
- [Installing or Updating Certificates Using the ctacert Utility, page 8-4](#)
  - [Installing or Updating a Certificate on Linux Operating Systems, page 8-4](#)
  - [Installing or Updating a Certificate on Mac OS X Operating System, page 8-4](#)
  - [Installing or Updating a Certificate on Windows Operating Systems, page 8-5](#)
- [Listing Certificates in the Certificate Store, page 8-6](#)
  - [Listing Certificates in the Certificate Store on Linux Operating Systems, page 8-6](#)

- Listing Certificates in the Certificate Store on Mac OS X Operating System, page 8-7
- Deleting Certificates from the Certificate Store, page 8-8
  - Deleting a Certificate from the Certificate Store on Linux Operating Systems, page 8-8
  - When prompted, type y to confirm your desire to delete the certificate., page 8-8
- Clearing Certificates from the Certificate Store, page 8-9
  - Clearing All Certificates from the Certificate Store on Linux Operating Systems, page 8-9
  - Clearing All Certificates from the Certificate Store on Mac OS X Operating Systems, page 8-10
- Configuring Machine Authentication Using Certificates, page 8-10
  - Requesting the Machine Certificate for Machine Authentication, page 8-11
- Configuring User Authentication Using Certificates, page 8-12
  - Importing the User Certificate for User Authentication, page 8-12
- Configuring Machine and User Authentication Using Certificates, page 8-13
- Distinguished Name Matching, page 8-14
- Converting DER Formatted Certificates to PEM Formatted Certificates, page 8-14

# About The ACS Server Root Certificate

For ACS to establish a secure PEAP or an EAP FAST session with Cisco Trust Agent, you must install the ACS root certificate on the network client. This certificate is either the CA certificate used to validate the server certificate, or a self-signed certificate generated by the ACS server. On Windows platforms, CTA supports PEM wrapped Base-64 or DER encoded binary X.509 certificates. On Linux platforms, CTA supports PEM wrapped Base-64 certificates only.

**Note**

The ACS certificate must have “server authentication” as the certificate purpose for the PEAP session to be created.

Before you begin reviewing this chapter, obtain the ACS root certificate. If ACS uses self-signed certificates, obtain the certificate from the server. (Refer to the *User Guide for Cisco Secure ACS for Windows Server* for information about obtaining the certificate.) If you use a CA certificate, obtain the certificate from your certificate server.

Cisco Trust Agent installs a utility on the local client to help you add, delete, and manage certificates. See [“About The ctacert Utility” section on page 8-3](#) for detailed procedures describing the use of this utility.

## About The ctacert Utility

Use the ctacert utility to install, delete, and manage the root certificate used by Cisco Trust Agent for PEAP (EAPoUDP) sessions with ACS or any other certificates you want to install on the client.

The ctacert utility is installed on Linux, Mac OS X, and Windows platforms. The utility's executable file name on Linux and Mac OS X is ctacert. The utility's executable file name on Windows is ctaCert.exe. This section refers to the utility generically as “ctacert.”

On Windows, the ctaCert.exe utility can accept PEM wrapped Base-64 or DER encoded binary X.509 certificates. On Linux platforms, the ctacert utility only accepts PEM wrapped Base-64 certificates. However, on Linux platforms, the certificates can be converted from DER to PEM formats. See, the [“Converting DER Formatted Certificates to PEM Formatted Certificates” section on page 8-14](#) for the command to perform the conversion.

# Installing or Updating Certificates Using the ctacert Utility

The ctacert utility can be used on Linux, Mac OS X, and Windows operating systems to install or update certificates.

## Installing or Updating a Certificate on Linux Operating Systems

- 
- Step 1** Copy the certificate to the client.
- Step 2** Open a terminal window on the network client.
- Step 3** At the prompt type either of the following commands and press <Enter>:
- `ctacert -a /path/cert_name.cer`
  - `ctacert --add /path/cert_name.cer`

In these examples, */path/cert\_name.cer* represents the full path and file name of the certificate.

After the certificate has been installed, you receive the message, “Certificate successfully added to store with Hashed Name *Number*”, where *Number* is the numeric Hashed Name of the certificate.

## Installing or Updating a Certificate on Mac OS X Operating System

- 
- Step 1** Copy the certificate to the client.
- Step 2** Open a terminal window.
- Step 3** Change the directory to the `/opt/CiscoTrustAgent/bin` directory.
- Step 4** At the prompt enter either of these commands and press <Enter>.
- `sudo ./ctacert -a /path/cert_name.cer`
  - `sudo ./ctacert --add /path/cert_name.cer`

In these examples, */path/cert\_name.cer* represents the full path and file name of the certificate.

After the certificate has been installed, you receive the message, “Certificate successfully added to store with Hashed Name *Number*”, where *Number* is the numeric Hashed Name of the certificate.

## Installing or Updating a Certificate on Windows Operating Systems

On Windows operating systems, all certificates are stored in the Microsoft Certificate Store. The ctaCert.exe utility only allows you to add certificates to the Microsoft Certificate Store. All other management of certificates is done through Microsoft's Certificate Management interface.

This is the /add command syntax for ctaCert.exe:

```
ctaCert.exe /ui {2 | 3| 4 | 5} /add "cert_path" /store "cert_store"
```

### Command Parameters

[Table 8-1](#) describes the command parameters for the ctaCert utility.

**Table 8-1** *ctaCert Utility Command Parameters*

| Parameter | Description                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /ui       | Specifies silent or verbose install. Accepts the following values: <ul style="list-style-type: none"><li>• <b>2</b> or <b>3</b>—Silent installation.</li><li>• <b>4</b> or <b>5</b>—Full user interaction installation.</li></ul> Any other value entered is treated as full user interaction. |
| /add      | Specifies the full path to the certificate being added. You can also specify *.cer to all certificates in the specified directory, for example: c:\My_Certs\*.cer.                                                                                                                             |
| /store    | Specifies the system certificate store. Typically this is “Root”.                                                                                                                                                                                                                              |

To install a certificate using the ctaCert.exe utility, follow this procedure:

---

**Step 1** Copy the certificate to the network client.

- Step 2** Open a command prompt on the network client.
- Step 3** Change directory to the location of the ctaCert.exe utility. By default, the location is C:\Program Files\Cisco Systems\CiscoTrustAgent\.
- Step 4** At the prompt, type the following and press <Enter>.
- ```
ctaCert.exe /ui x /add C:\path\cert_name.cer /store Root
```

Where **/ui x** specifies the level of user interaction and where **C:\path\cert_name.cer** is the full path and file name of the certificate.

The certificate is added to the trusted certificate store on the network client.

Listing Certificates in the Certificate Store

The ctacert utility can be used on Linux and Mac OS X to list the certificates in the client certificate store. Use the Microsoft's Certificate Management interface to perform this task on Windows operating systems.

Listing Certificates in the Certificate Store on Linux Operating Systems

- Step 1** Open a terminal window on the network client.
- Step 2** From any prompt enter either of these commands and press <Enter>.
- ctacert -l
 - ctacert --list

This command displays the hashed file name, certificate version, signature algorithm, subject/issuer name, validity period, and MD5 fingerprint information. Output pertaining to different certificates are separated by a string of dashes.

Example 8-1 *ctacert --list command output on Linux*

```
#ctacert --list
hashed file name: 814661db.0
Version: 3 (0x2)
Serial Number: 0 (0x0)
```



```

Signature Algorithm: md5WithRSAEncryption
Issuer: O=Cisco Systems, Inc., CN=Stress
Validity
Not Before: Aug  7 11:38:06 2002 GMT
Not After : Aug 20 05:09:50 2048 GMT
Subject: O=Cisco Systems, Inc., CN=Stress
MD5 Fingerprint=13:5A:A9:B5:98:DE:78:F5:1A:7E:27:FA:E0:8B:1D:D7
-----

```

Listing Certificates in the Certificate Store on Mac OS X Operating System

-
- Step 1** Open a terminal window on the network client.
- Step 2** Change the directory to /opt/CiscoTrustAgent/bin directory.
- Step 3** At the prompt enter either of these commands and press **<Enter>**.
- `sudo ./ctacert -l`
 - `sudo ./ctacert --list`
- Step 4** When prompted, type the root user's password.
- This command displays the hashed file name, certificate version, signature algorithm, subject/issuer name, validity period, and MD5 fingerprint information. Output pertaining to different certificates are separated by a string of dashes.

Example 8-2 *ctacert --list command output on Mac OS X*

```

Hashed Name: 5e8a8166.0
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 1 (0x1)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: CN=Cisco Systems
  Validity
    Not Before: Jul 19 15:12:24 2006 GMT
    Not After : Jul 19 15:12:24 2007 GMT
  Subject: CN=Cisco Systems
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:TRUE

```

```

X509v3 Key Usage:
    Digital Signature, Key Encipherment, Key Agreement, Certificate Sign
X509v3 Subject Key Identifier:
    B5:79:DE:6A:C7:42:47:25:42:BC:68:43:93:04:69:2E:9B:08:0E:64
X509v3 Extended Key Usage:
    TLS Web Server Authentication
Netscape Cert Type:
    SSL Server

```

Deleting Certificates from the Certificate Store

The `ctacert` utility can be used on Linux and Mac OS X to delete certificates in the client certificate store. Use the Microsoft's Certificate Management interface to perform this task on Windows operating systems.

Deleting a Certificate from the Certificate Store on Linux Operating Systems

-
- Step 1** Open a terminal window on the network client.
- Step 2** From any prompt, enter either of these commands and press **<Enter>**.
- `ctacert -d HASHED-CERT-FILENAME`
 - `ctacert --delete HASHED-CERT-FILENAME`

The `hashed-cert-file-name` can be obtained from the `ctacert --list` output. In [Example 8-1](#), the hashed certificate file name is `814661db.0`.

For example:

```
ctacert -d 814661db.0
```



Note

The hashed file name for a certificate may change when other certificates are removed.

- Step 3** When prompted, type **y** to confirm your desire to delete the certificate.

Deleting a Certificate from the Certificate Store on Mac OS X Operating System

-
- Step 1** Open a terminal window on the network client.
- Step 2** Change the directory to /opt/CiscoTrustAgent/bin directory.
- Step 3** At the prompt enter either of these commands and press **<Enter>**.

- `sudo ./ctacert -d HASHED-CERT-FILENAME`
- `sudo ./ctacert --delete HASHED-CERT-FILENAME`

The `hashed-cert-file-name` can be obtained from the `ctacert --list` output. In [Example 8-1](#), the hashed certificate file name is `814661db.0`.

For example:

```
# sudo ./ctacert -d 814661db.0
```

- Step 4** When prompted, type the root user's password.
- Step 5** When prompted, type **y** to confirm your desire to delete the certificate.

**Note**

The hashed file name for a certificate may change when other certificates are removed.

Clearing Certificates from the Certificate Store

The `ctacert` utility can be used on Linux and Mac OS X to clear all the certificates in the client certificate store. Use the Microsoft's Certificate Management interface to perform this task on Windows operating systems.

Clearing All Certificates from the Certificate Store on Linux Operating Systems

-
- Step 1** Open a terminal window on the network client.
- Step 2** From any prompt enter either of these commands:

- `ctacert -c`
- `ctacert --clear`

Clearing All Certificates from the Certificate Store on Mac OS X Operating Systems

-
- Step 1** Open a terminal window on the network client.
- Step 2** Change the directory to `/opt/CiscoTrustAgent/bin` directory.
- Step 3** At the prompt enter either of these commands and press **<Enter>**.
- `sudo ./ctacert -c`
 - `sudo ./ctacert --clear`
- Step 4** When prompted, type the root user's password.
- Step 5** When prompted, type **y** to confirm your desire to clear the certificate store.

Configuring Machine Authentication Using Certificates

CTA can be configured to perform machine authentication using certificates provided that the 802.1x Wired Client has been installed. All IEEE 802.1x authentication methods are currently only supported on Windows platforms. See [“System Requirements for Installation” section on page 4-2](#) for the complete list of supported platforms.

To configure CTA to perform machine authentication using certificates, you must perform these procedures:

-
- Step 1** [Installing or Updating a Certificate on Windows Operating Systems, page 8-5](#)
- Step 2** [Requesting the Machine Certificate for Machine Authentication, page 8-11](#)
- Step 3** [Deploying End-User 802.1x Wired Clients, page 9-35](#) using the “Creating a Machine Authentication Only Deployment Package” section on page 9-40.

Requesting the Machine Certificate for Machine Authentication

**Note**

You will need to request a machine certificate for the client if one was not issued to the client when it joined the domain.

Follow this procedure to request a certificate for machine authentication:

- Step 1** As the Administrator, log on to the host on which you want to request the machine certificate.
- Step 2** Open a command prompt window.
- Step 3** At the prompt, type **mmc** and press <Enter>.
- Step 4** From the File menu, select **Add/Remove Snap-in**.
- Step 5** In the Standalone tab, click **Add**.
- Step 6** Click the **Certificates** icon in the Add Standalone Snap-in window and click **Add**.
- Step 7** Select **Computer Account** in the Certificates Snap-in window and click **Next**.
- Step 8** Select **Local Computer**.
- Step 9** Click **Finish**.
- Step 10** Click **Close**.
- Step 11** Click **OK** to close the Add/Remove Snap-in window.
- Step 12** Expand the “Certificates (Local Computer) certificate icon under Console Root.
- Step 13** Right-click the **Personal** folder, and navigate **All Tasks > Request New Certificate**.
- Step 14** Click **Next** at the Welcome window.
- Step 15** Select **Computer** and click **Next**.
- Step 16** Enter a name for the certificate in the **Friendly Name** field, a description in the **Description** field and click **Next**.
- Step 17** Click **Finish**.
- Step 18** Click **OK**.

Configuring User Authentication Using Certificates

CTA can be configured to perform user authentication using certificates provided that the 802.1x Wired Client has been installed along with CTA. User authentication using certificates is only available on Windows platforms. See [“System Requirements for Installation” section on page 4-2](#) for the complete list of supported platforms.

To configure CTA to perform user authentication using certificates, you must perform these procedures:

-
- Step 1** [Installing or Updating a Certificate on Windows Operating Systems, page 8-5](#)
 - Step 2** [Importing the User Certificate for User Authentication, page 8-12](#)
 - Step 3** [Deploying End-User 802.1x Wired Clients, page 9-35](#) using the “Creating a Machine Authentication Only Deployment Package” section on page 9-40 procedure.

Importing the User Certificate for User Authentication

-
- Step 1** Create or obtain the user certificate.

**Note**

As there are different vendors and methods used to create a user certificate, those procedures are not covered here. See your specific vendor's documentation for information on creating a machine certificate.

- Step 2** Log on to the host, on which you want to import the user certificate, as the Administrator.
- Step 3** Open a command prompt window.
- Step 4** At the prompt, type **mmc** and press <Enter>.
- Step 5** From the Console menu, select **Add/Remove Snap-in**.
- Step 6** In the Standalone tab, click **Add**.
- Step 7** Click the **Certificates** icon in the Add Standalone Snap-in window and click **Add**.

- Step 8** Select **My User Account** in the Certificates Snap-in window and click **Finish**.
- Step 9** Close the Add Standalone Snap-in window.
- Step 10** Click **OK** to close the Add/Remove Snap-in window.
- Step 11** In the MMC Console (Console1), expand the **Certificates - current user** folder in the directory tree in the left pane.
- Step 12** Right-click the Personal folder and select **All Tasks > Import** from the shortcut menu.
- Step 13** Use the Wizard to browse to your certificate and import it. Accept all the default settings offered to you.
- Step 14** In the Certificates - Current User directory tree, open the Personal Folder and select the **Certificates** sub-folder. In the certificate pane, on the right, you will see the user certificate. The name of the certificate will be the full qualified domain name of the PC.

Configuring Machine and User Authentication Using Certificates

CTA can be configured to perform both machine and user authentication using certificates provided that the 802.1x Wired Client has been installed along with CTA. Authentication using both user and machine certificates is only available on Windows platforms. See [“System Requirements for Installation” section on page 4-2](#) for the complete list of supported platforms.

Before you create a machine and user authentication policy, you must perform these procedures:

-
- Step 1** [Installing or Updating a Certificate on Windows Operating Systems, page 8-5](#)
 - Step 2** [Requesting the Machine Certificate for Machine Authentication, page 8-11](#)
 - Step 3** [Importing the User Certificate for User Authentication, page 8-12](#)
 - Step 4** [Deploying End-User 802.1x Wired Clients, page 9-35](#) using the [Create a Machine and User Authentication Deployment Package, page 9-38](#).

Distinguished Name Matching

When using CA certificates to validate your Cisco Secure ACS server certificate, you can implement additional security using distinguished name (DN) matching to validate the server certificate. This prevents other servers or processes that may be using the same root certificate from gaining a trust relationship with the network client.

DN matching occurs at the end of the TLS handshake, after the certificate chain is built. Invalid DN matching rules are ignored, but logged. Matched rules are logged. Failed rules are not logged.

DN matching rules are configured in the [ServerDNVerification] section of the ctad.ini configuration file. If the [ServerDNVerification] section does not exist, or if there are no rules configured, then the DN matching feature is disabled and the system accepts connections with any validated certificate chain. Otherwise, the server certificate must match one of the DN matching rules for the connection to continue.

If the configuration file does not exist, the default values for these settings are used. To change the value for any of these items, you need to create the configuration file and save it to the appropriate location.

Any changes made to the [ServerDNVerification] section of the ctad.ini configuration file are detected and are implemented by Cisco Trust Agent the next time DN matching occurs.

To learn more about configuring Domain Name matching in the ctad.ini file, see [“Certificate Distinguished Name Matching” section on page 5-25](#).

Converting DER Formatted Certificates to PEM Formatted Certificates

On Linux and Mac OS X platforms, CTA supports PEM wrapped Base-64 certificates but not DER encoded binary X.509 certificates. However DER certificates can be converted to PEM certificates using the following procedure. (For the sake of this procedure, assume that the name of the DER formatted certificate is **ca.der**.)

**Note**

This procedure requires that OpenSSL is installed on the workstation.

Step 1 Log in to the Linux workstation as the root user.

Step 2 Open a terminal window.

Step 3 At the prompt, type the following:

```
openssl x509 -inform DER -outform PEM -in ca.der -out ca.pem
```

■ Converting DER Formatted Certificates to PEM Formatted Certificates



CHAPTER 9

Cisco Trust Agent 802.1x Wired Client

The Cisco Trust Agent 802.1x Wired Client (802.1x Wired Client) is an authentication supplicant for creating secure user connections with an Ethernet switch. The 802.1x Wired Client provides a graphical user interface for monitoring authentication status and managing authorized network access that is protected by the IEEE 802.1x protocol.

The 802.1x Wired Client implementation allows end-user connectivity to the network only after successful client-server authentication via port access control on the 802.1x-enabled access device.

The 802.1x Wired Client is an integral part of the Cisco Network Admission Control (NAC) security environment. In this environment, the Cisco Secure Access Control Server (ACS) requests posture information about NAC-compliant applications running on the system from the Cisco Trust Agent (CTA). The 802.1x Wired Client returns that posture information for CTA.

The Cisco Trust Agent 802.1x Wired Client is available for Windows systems only.

This chapter contains the following sections:

- [802.1x Wired Client Features, page 9-3](#)
- [802.1x Wired Client Administrative and Client Versions, page 9-4](#)
- [802.1x Wired Client User Interface, page 9-4](#)
 - [Administrative 802.1x Wired Client Automatic Startup, page 9-4](#)
 - [Network Connection Status, page 9-5](#)
 - [802.1x Wired Client System Tray Shortcut Menu, page 9-7](#)
 - [802.1x Wired Client Window, page 9-7](#)

- Network Connection Status, page 9-8
 - Access Device Connection Status, page 9-9
- Basic 802.1x Wired Client Procedures, page 9-11
 - Opening the 802.1x Wired Client, page 9-11
 - Manually Connecting To the Network, page 9-12
 - Manually Disconnecting From the Network, page 9-12
 - Viewing Network Summary, page 9-12
 - Viewing Access Device Status, page 9-13
- Getting Started with 802.1x Wired Client Functions, page 9-16
 - Administrative 802.1x Wired Client Overview, page 9-16
 - Authentication Methods Overview, page 9-16
 - Overview of FAST Connections in a User Logon Context, page 9-17
 - Overview of FAST Connections in a Machine Credentials Context, page 9-18
- User Credentials, page 9-20
 - Initial Credential Provisioning, page 9-20
- Machine Credentials, page 9-21
 - Pre-PAC or no-PAC Provisioning, page 9-22
 - Post-PAC Provisioning, page 9-22
- Credential Revalidation, page 9-23
 - Server-Initiated Credential Revalidation, page 9-23
 - User-Initiated Credential Revalidation, page 9-23
- Understanding Policies and Profiles, page 9-24
- Create Deployment Package Wizard, page 9-26
 - User Credentials Area, page 9-27
 - Automatically Establishing Machine Connection, page 9-27
 - User Identity Protection Area, page 9-32
 - Allow Unprotected Client Cert Area, page 9-28
 - Trusted Server Validation Area, page 9-29

- [Deploying End-User 802.1x Wired Clients, page 9-35](#)
 - [Creating Deployment Packages, page 9-36](#)
 - [Installing Server Certificates on the Host, page 9-42](#)
 - [Installing Deployment Packages on Hosts, page 9-43](#)
- [Changing Deployment Packages on Hosts, page 9-44](#)
 - [Replacing a Deployment Package on a Host, page 9-44](#)

802.1x Wired Client Features

The Cisco Trust Agent 802.1x Wired Client (802.1x Wired Client) has the following features:

- **Pre-Configured:** The 802.1x Wired Client operates automatically. The only input required for administrative use of the client is the initial, one-time entry of the administrator's individual user credentials.
- **Wired:** The 802.1x Wired Client supports wired 802.3 Ethernet connections (up to a maximum of 4).
- **Auto-Connection:** Connection to the CTA network environment is automatic, once a physical Ethernet connection has been made between your computer system's wired adapter interface and the network itself. Both machine and user logon context connections are supported.
- **Network Profile:** The 802.1x Wired Client is predefined to support a single network profile, embodying an administrative user, his security credentials, the computer's wired interface adapter(s), an 802.1x Ethernet switch access device, and an authentication server.
- **DHCP Management:** The 802.1x Wired Client ensures IP connectivity after a connection is authenticated.
- **Client Deployment:** The administrative version includes a deployment wizard that allows for creation and pre-configuration of an end user client. Pre-configuring involves supplying a trusted server list and end user credential selection methods.
- **Single Sign-on:** Once deployed, the 802.1x Wired Client can perform user authentication using Single Sign-On for Microsoft and Novell Networks.

- **Request Password When Needed:** Once deployed the 802.1x Wired Client can perform user authentication by requesting the user's password.
- **Machine Authentication Only.** Once deployed, the 802.1x Wired Client can support machine authentication without user authentication.

802.1x Wired Client Administrative and Client Versions

The administrative version of the 802.1x Wired Client is the “out-of-the-box” version of the product. By default, the Administrative 802.1x Wired Client will authenticate its local network using both machine and user authentication. It is intended for use by the IT organization responsible for configuring and deploying the end-user versions of the 802.1x Wired Client.

Once the administrator defines connection policies and those policies are installed along with the 802.1x Wired Client on a user's workstation, the 802.1x Wired Client becomes the end-user version of the product.



Note

Unless otherwise noted, the functions described in this chapter refer only to the administrative version of the 802.1x Wired Client and are not available to end-users running the client version of the 802.1x Wired Client. While end-users will be able to view authentication settings previously deployed by the administrator, they will not be able to configure the client.

802.1x Wired Client User Interface

This section describes the elements of the 802.1x Wired Client interface that are visible to the administrative user.

Administrative 802.1x Wired Client Automatic Startup

After installation, the Cisco Trust Agent 802.1x Wired Client service starts automatically on startup of the Windows operating system. By default, the administrative 802.1x Wired Client is configured to automatically attempt to

authentication the host using machine and user credentials. Confirmation that the 802.1x Wired Client has been successfully started is indicated by the presence of its icon in the Windows task bar icon tray.

Figure 9-1 **802.1x Wired Client Icon**



Network Connection Status

The 802.1x Wired Client icon in the Windows system tray changes color to reflect the state of the network connection.

Icon colors and their associated status indications are detailed in [Table 9-1](#):

Table 9-1 **Icon Status Color Codes**






Icon Color	Status
Green 	The network adapter is connected and has been authenticated.
Yellow 	Authentication is taking place.
Red 	Authentication has failed.

Table 9-1 **Icon Status Color Codes**

Icon Color	Status
Blue 	The network adapter is connected but it does not require authentication. The connection is unauthenticated.
No color 	Connection is idle

Mousing-over the 802.1x Wired Client icon displays a summary connection status message.

Disabling the 802.1x Wired Client System Tray Icon

Step 1 Open the 802.1x Wired Client Interface

Step 2 From the Client menu, uncheck Show System Tray.

This configuration is made on individual clients and cannot be defined in a custom installation package.

802.1x Wired Client System Tray Shortcut Menu

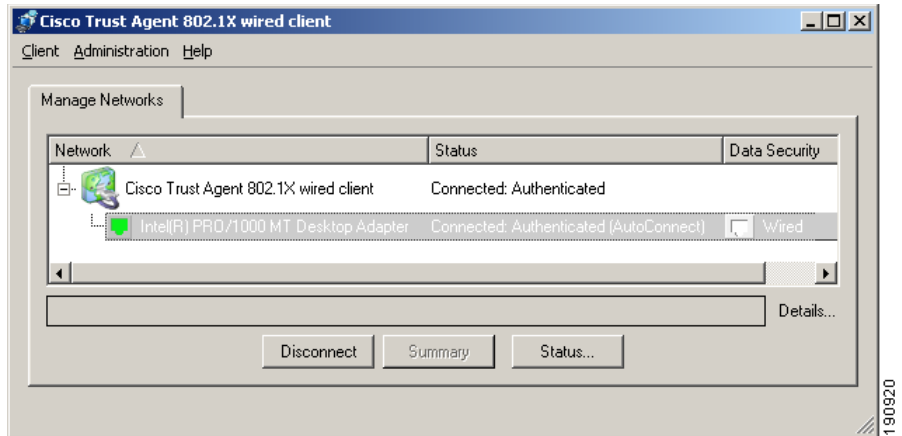
By right-clicking the 802.1x Wired Client status icon in the Windows system tray you see a menu with three menu options.

Menu Choice	Description
Open	Allows you to launch the CTA 802.1x Wired Client interface.
Enable Client	<p>Displays the state of the CTA 802.1x Wired Client, whether it is Enabled or Disabled.</p> <p>Checking the Enable Client menu option enables the 802.1x Wired Client. Unchecking the Enable Client menu choice disables the CTA 802.1x Wired Client.</p> <p>If Enable is checked the 802.1x Wired Client is active and can provide posture and credential information to authenticate the host. If Disable is checked, the 802.1x Wired Client can not provide posture and credential information to authenticate the host.</p> <p>Client:Enabled is the default state of the CTA 802.1x Wired Client.</p>
About	Displays the version number and copyright information for the CTA 802.1x Wired Client.

802.1x Wired Client Window

The 802.1x Wired Client's main window contains a list of controlled networks. The single, pre-named, network connection expands to display all access devices configured within the network connection. In [Figure 9-2](#) the network is named, "Cisco Trust Agent 802.1x wired client."

Figure 9-2 802.1x Wired Client Administrator's main window



Network Connection Status

The color of the network connection's icon reflects the best connection status of all of the access device connections. The network connection icon in [Figure 9-2](#) is named Cisco Trust Agent 802.1x Wired Client.

Table 9-2 Network Icon Status Information

Network Icon Display	Network Status	Description
Fully colored icon	Connected	Indicates that at least one of the access devices is connected and has been authenticated.
Grey icon	Connecting	Indicates that at least one of the access devices is connecting and is authenticating.
No color	Disconnected	Indicates that no access devices is connected or connecting and that no device has been authenticated.

Access Device Connection Status

The main window contains a list of access devices, which includes network adapters and network interfaces. In figure [Figure 9-2](#) there are two access devices in the network:

- Intel(R) PRO/1000 CT Network Connection
- Intel(R) PRO/1000 MT Desktop Adapter

The assigned access device name is taken from the name of the network adapter associated with the access device. If all the members of the ethernet group are 'Not Available,' the name changes to its group '<ethernet>' name.

The color of the icon indicates the connection status of the access device.

Table 9-3 Access Device Icon Status Information

Access Device Icon Display	Access Device Status	Description
None	Disconnected	<ul style="list-style-type: none">• The access device is currently available but not attempting a connection (a transitory state for auto-connection)• Auto-connection has been stopped, possibly caused by one of the following actions:<ul style="list-style-type: none">– Credentials have been cancelled. See the “Credential Revalidation” section on page 9-23.– Connection has been stopped. See the “Manually Connecting To the Network” section on page 9-12.
Yellow	Connecting	The network adapter is attempting to make a network connection.
Green	Connected: Authenticated	The network adapter is connected to the network. The host has been authenticated.
Blue	Connected: Unauthenticated	The network adapter is connected to the network and does require authentication.

Table 9-3 Access Device Icon Status Information

Access Device Icon Display	Access Device Status	Description
Red	Failed (retrying)	This is a temporary state while attempting to authenticate the host and create a connection. The last attempt to authenticate failed.
Red	Failed	Authentication attempts have failed. Selecting the access device will display the reason for the failure in the Message Status bar.
X Overlays gray icon	Not Available	Access device is not available

The **Data Security field** is used only for access devices. It indicates that the security is based on a wired port.

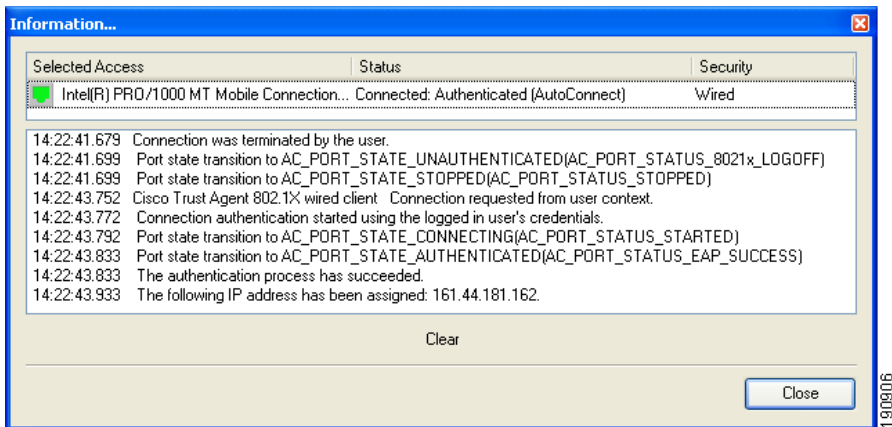
A single **Status Message display line** is located below the Access Port List and adds clarification to the current access status. Often used to clarify the reason for a failure, for example, why a connect attempt might have failed.

To the right of the Status Message display line is the **Details...** hot spot. Clicking the **Details...** hot spot launches the Information window. The Information window displays a real-time feedback of the individual steps of any (manual or auto) connection or disconnection process.

Selecting another access device, or using the “Clear” control clears the current display.

**Note**

The Information dialog is an independent window and will remain open while performing other operations from the main screen.

Figure 9-3 Network Adapter Information box

Basic 802.1x Wired Client Procedures

This section describes the tasks administrators perform from the main window of the 802.1x Wired Client.

Opening the 802.1x Wired Client

Use one of the following methods to launch the 802.1x Wired Client main window:

- Right-click the client icon in the system tray (as shown above in [Figure 9-1](#)), to display the icon menu. Click **Open** to display the 802.1x Wired Client main window.
- Double-click the client icon in the system tray to directly open the 802.1x Wired Client main window.
- Use the Windows Start menu. Navigate Start > Programs > Cisco Systems, Inc. Cisco Trust Agent 802.1x wired client > Cisco Trust Agent 802.1x wired client

Manually Connecting To the Network

- Step 1** From the 802.1x Wired Client main window, select the desired access device icon, displayed without color.
- Step 2** Click **Connect**.

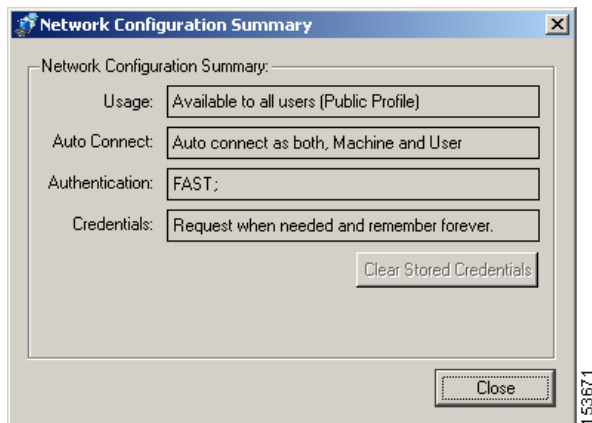
Manually Disconnecting From the Network

- Step 1** From the 802.1x Wired Client main window, select the desired access device icon, displayed in full color.
- Step 2** Click **Disconnect**.

Viewing Network Summary

A summary of the policy that governs the connection and authentication of the host to the network can be viewed but it cannot be modified.

-
- Step 1** In the 802.1x Wired Client main window, select the network for which you want to view the policy summary.
- Step 2** Click **Summary**. The Network Configuration Summary dialog opens.

Figure 9-4 Network Configuration Summary dialog

Field	Description
Usage	802.1x Wired Client profiles are always public
Auto Connect	Indicates if the host connects using machine or user authentication.
Authentication	802.1x Wired Client always uses EAP-FAST authentication
Credentials	Describes where and how user credentials are obtained.
Clear Stored Credentials	Clicking this button allows users to delete their stored credentials and force a re-validation of their credentials.

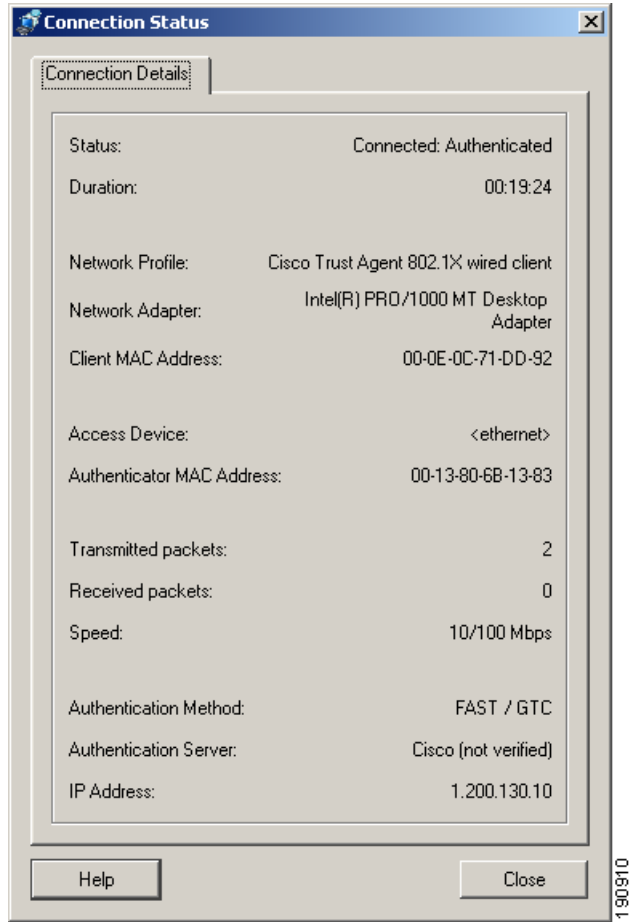
Viewing Access Device Status

This option displays the Connection Details window, which contains specific information about the access device's network connectivity.

- Step 1** In the 802.1x Wired Client main window, select the desired access device icon, for which you want to view the connection status.

Step 2 Click **Status...**The Connection Details window opens.

Figure 9-5 *Connection Details Window*



Field	Description
Status	State of the connection. This is derived from the Main Screen Access Port list
Duration	Defines how long the connection has been operational
Network Profile	Fixed generic name
Network Adapter	Identifies the port's adapter name
Client MAC Address	Displays the MAC address of the associated network adapter.
Access Device	Fixed generic name
Authenticator MAC Address	MAC address of the switch.
Transmitted packets	Actual number of layer 2 frames transmitted
Received packets	Actual number of layer 2 frames received
Speed	Indication of connection maximum data rate
Authentication Method	<p>The authentication method may be EAP-FAST, EAP-GTC, or EAP-TLS.</p> <p>If the field is blank then authentication is not required because the host is not connected to a 802.1x configured port.</p>
Authentication Server	<p>The field shows either the server certificate's name or the server's FAST A-ID - augmented with an indication of either 'not verified' or 'trusted'.</p> <p>'Not verified' only exists for the case of an empty Trusted Server List. (See the “Trusted Server Validation Area” section on page 9-29 for more information on trusted servers.)</p>
IP Address	IP Address of host

Getting Started with 802.1x Wired Client Functions

Administrative 802.1x Wired Client Overview

This administrative version of the client has one pre-defined enterprise network when initially started. A Network is defined by its Network Profile. (See [“Understanding Policies and Profiles” section on page 9-24](#) for more information on network profiles.)

The 802.1x Wired Client utilizes EAP-FAST (the Flexible Authentication via Secure Tunneling method of the Extensible Authentication Protocol). EAP-FAST establishes a protected TLS (Transport LAN Service) tunnel using a pre-shared secret. The pre-shared secret is referred to as the Protected Access Credential, or PAC. (A master key that is kept by the authentication server is used to generate a unique PAC for each user.) The PAC also serves to establish the server's identity to the client. The tunnel is then used to protect weaker authentication methods, typically based on a password such as EAP-GTC or EAP-MSCHAPv2, in which the client authenticates itself to the server. In this way, mutual authentication is achieved. It is also possible that mutual authentication can also be achieved during the tunnel creation if the client provides a client certificate via the TLS protocol.

Authentication Methods Overview

The 802.1x Wired Client authenticates machines and users so that they may connect to the network. Successful user authentication establishes a network connection after a user logs onto the machine. Successful machine authentication connects the machine to the network once the machine's credentials have been authenticated. User and machine authentication may also be used together.

**Note**

- Some applications may not be appropriate choices to provide posture credentials during machine authentication. Such applications may be slow to start, for example, and they will not be ready to provide posture credentials immediately for machine authentication.

In this case, machine authentication could fail, not because of a security problem but because the application was not available to provide its posture credentials in time.

- In order to perform machine authentication, the EAP-FAST Configuration in ACS must allow machine authentication.
- Machine authentication can only be performed on networks where Windows Active Directory is in use.

Each deployed end-user 802.1x Wired Client is configured to support one of the following methods of authentication for network access:

- User authentication only
- Machine and user authentication
- Machine authentication only

The administrative 802.1x Wired Client is pre-configured to connect the host to the network only after authenticating machine and user credentials.

Overview of FAST Connections in a User Logon Context

Initial User Connections - Before the PAC Has Been Provisioned

During the initial authentication session, the FAST protocol provides for the automatic provisioning of the administrative user's unique PAC.

There are two methods of automatically provisioning a user's PAC. The first is a basic automatic provisioning feature that does not require a server certificate to be installed on the host. The second is a more secure automatic provisioning feature that uses a server certificate for providing initial authentication.

**Note**

The initial authentication session may take several authentication cycles to complete depending on the method used and the configuration of the server.

The basic automatic provisioning feature prompts users for their user credentials only.

The automatic provisioning feature that uses a certificate also requires user credentials and the CA certificate used to trust the server certificate must be stored in the proper Windows Certificate Store (User-Trusted Root Store) of the host.

For both the basic or certificate-based automatic provisioning, the FAST authentication server must be configured for auto creation of the administrator's unique user PAC information.

The user credentials needed for the basic and certificate-based automatic provisioning are requested on an as-needed basis. See the [“User Credentials” section on page 9-20](#) for more information.

These are the user credentials that may be required:

- Identity. The format of the information is *UserName@Domain*, where the use of the domain, also referred to as a “realm,” is optional.
- Password. This credential is optional.
- User Certificate Identifier. This credential is optional.

Subsequent User Connections - After the PAC Has Been Provisioned

Once the user (tunnel) PAC has been provisioned and the user's credentials saved, subsequent connections are autonomously made using the PAC data to create the secure tunnel used for passing the user's credentials for authentication. No user intervention is required and authentication is automatic.

Overview of FAST Connections in a Machine Credentials Context

Initial Machine Connections - Before the PAC Has Been Provisioned

The provisioning of the Machine PAC, which is needed for machine context connections, is accomplished using the server certificate or machine security identity (SID). Machine PACs are only supported in newer versions of authentication servers (ACS 4.0 or later) which have been upgraded to support EAP-FAST v1a.

To make a machine connection before the PAC has been provisioned, the CA certificate used to trust the server certificate must be placed in the proper Windows Certificate Store (Local Computer-Trusted Root Store).

The host must also provide these machine credentials:

- Active Directory provided machine certificate. The authentication method must support the use of a certificate to provide machine client credentials - the server must be appropriately configured to call for an inner tunnel method of TLS.
- Active Directory provided SID (password). The authentication method must support the use of a password to provide machine client credentials.

Finally, the FAST authentication server must be configured for auto creation of administrator's unique machine PAC information.

**Note**

The client autonomously seeks the correct credential type based on the EAP method initiated by the authentication server.

**Note**

If a machine certificate or *machine password* (SID) is not *initially pre-installed on the machine*, then an initial machine connection can not be made and the machine PAC provisioning will wait until the initial user logon connection is made. At which point both machine and user (tunnel) PACs will be provisioned.

**Note**

If a machine certificate/password is supported, then an initial machine connection may be made and the machine PAC provisioning may take place depending on whether or not it is supported and configured by the authentication server. Otherwise the machine PAC provisioning will wait until the initial user logon connection is made. At which point both machine and user (tunnel) PACs will be provisioned.

Subsequent Machine Connections - After the PAC Has Been Provisioned

Once the machine PAC has been provisioned, subsequent connections are autonomously made using the PAC data exclusively and authentication is automatic.

User Credentials

The EAP-FAST method used by the 802.1x Wired Client supports the following types of user credentials:

- **User identity.** This is a mandatory element.
- **User password or user certificate.** The use of one of these elements is required.

The single Network Profile of the 802.1x Wired Client is pre-configured to accommodate both types, as needed. The type of credential required for a specific authentication session is determined by the settings of the authentication server and is automatically processed by the 802.1x Wired Client.

Initial Credential Provisioning

After the 802.1x Wired Client is deployed to an individual end-user, specific user credential information may be required to be provisioned. However, there is no pre-provisioning of this information. Rather it is requested, on-demand, by the client at the appropriated time during its first connection attempt after initial startup. A series of appropriate Credential Request Pop-up Dialogs are progressively displayed to the user with prompts and mechanisms for entering the desired information.

Identity Credentials

One of the following identity credential methods has been pre-configured:

- Operating System login (Single-Sign-On) credentials - no provisioning required.
- Request & Store - initial text input provisioning through Enter Your Credentials Pop-ups, encrypted and stored for automatic use thereafter.

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 a “realm,” is optional. The identity specified may contain up to 63 ASCII characters and is case sensitive.

Password Credentials

One of the following password credential methods has been pre-configured:

- Operating System login (Single-Sign-On) credentials - no provisioning required.
- Request & Store - initial text input provisioning through Enter Your Credentials Pop-ups, encrypted and stored for automatic use thereafter. The password specified may contain up to 80 ASCII characters and is case sensitive.

Certificate Credentials

The request and store configuration method for certificates credentials has been pre-configured:

- Request & Store - Initial file selection provisioning through Enter Your Credentials Pop-ups, stored for automatic use thereafter.

The identifying information for the selected certificate in the selection pull-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)
- Expired: Valid to

**Note**

The certificate must not have strong private key protection, since this will cause the connection authentication to fail at logon.

**Note**

The administrator version is pre-configured to support “Request & Store” only.

Machine Credentials

Depending on the EAP method you use, there are two types of machine credentials.

Pre-PAC or no-PAC Provisioning

- **Machine certificate** - This uses the Microsoft Active Directory provided machine certificate, or equivalent, with a TLS-based EAP method.

A machine certificate must be in the appropriate Windows Certificate Store (Personal Certificate Store for the Local Computer).

There are two restrictions to this method:

- There can only be a single certificate stored here - which is the normal usage condition.
- The certificate must not require a PIN or have strong private key protection.

**Note**

The machine identity is provisioned the “dnsName” field, of the Subject Alternative Name, of the machine certificate.

**Note**

Another aspect of machine authentication is that 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.

- **Machine SID** - This uses the Microsoft Active Directory provided machine SID (Security Identifier) with a password-based EAP method, such as, EAP-MSCHAPv2.

**Note**

The SID acts as a machine password.

Post-PAC Provisioning

A Machine PAC is only used with EAP-FAST and only supported in newer versions of authentication servers (e.g., ACS 4 or later) which have been upgraded to support EAP-FAST v1a.

Credential Revalidation

Credentials may expire in which case the server may initiate a revalidation of the credentials or the user may clear the credentials and force a revalidation.

Server-Initiated Credential Revalidation

In addition to cases where the credentials fail to be accepted by the server, the server may employ policies such as password aging that automatically re-prompts the user for their credentials using the appropriate “Enter Your Credentials” pop-up.

User-Initiated Credential Revalidation

The user can manually force a re-prompting for authentication credentials. The re-prompting clears any stored user credentials which causes a new sequence of credential request pop-up dialogs to appear during the next authentication attempt.

**Note**

The control is enabled if the credentials are remembered. When the user clears the credentials, the control will become disabled.

**Note**

This is not valid for Single-Sign-On credentials.

To clear existing credentials, follow this procedure:

-
- Step 1** Launch the 802.1x Wired Client interface.
 - Step 2** In the Connection Status dialog box, select the network that you want to re-prompt for credentials.
 - Step 3** Click **Summary**.
 - Step 4** Click **Clear Stored Credentials**.

Understanding Policies and Profiles

The 802.1x Wired Client is part of a family of 802.1x connection clients. An End-user 802.1x Wired Client creates network connections based on its base features and its operational environment, which is defined by a set of configuration documents.

The 802.1x Wired Client's base feature is that it is limited to a wired (802.3) network media type.

The authentication policies used by the 802.1x Wired Client are based on two configuration documents:

- 802.1x Wired Client policy file
- 802.1x Wired Client network policy file

802.1x Wired Client Policy File

The 802.1x Wired client policy file, policy.xml, which is installed in the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\policies** directory defines these aspects of the operational environment:

- **Authentication methods on how network connections may be created.** 802.1x Wired Client is limited to an EAP-FAST “outer method” with “inner methods” of EAP-TLS, EAP-MSCHAPv2, EAP-GTC
- **User interface configuration variations.** The 802.1x Wired Client is limited to one network profile.
- **Trusted Servers List** which defines how to validate the credentials of servers during mutual EAP authentications.

**Note**

The administrator version is pre-configured with an empty Trusted Server List and therefore never performs server validation.

802.1x Wired Client Network Policy File

The network policy file, **networks.xml**, which is installed in the **Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\networks** directory defines these aspects of the operational environment:

- **Methods for collecting user credentials.** The 802.1x Wired Client supports single sign-on or request authentication-unique password user credentials.

**Note**

The administrator version is pre-configured to support “Request & Store” only.

- **Origination of connections.** The 802.1x Wired Client supports both machine and user originated connection types.
- The administrative version of the 802.1x Wired Client is pre-configured to support “Machine and User authentication.”
- **Use of client certificates.** The 802.1x Wired Client can be configured to require the use of client certificates for machine and user connection types.

**Note**

The administrative version of the 802.1x Wired Client is pre-configured to not require a client certificate for both machine and user authentications.

- **Authentication protocols.** 802.1x Wired Client can be configured for the Identity used for the phase 1 outer (unprotected) tunnel of the EAP-FAST method.

**Note**

The administrative version of the 802.1x Wired Client is pre-configured to use “Anonymous” as the identity.

Using the Administrative 802.1x Wired Client, a system administrator uses the Deployment Package Wizard to create the 802.1x Wired Client policy.xml file and network.xml policy file based on the administrator’s input.

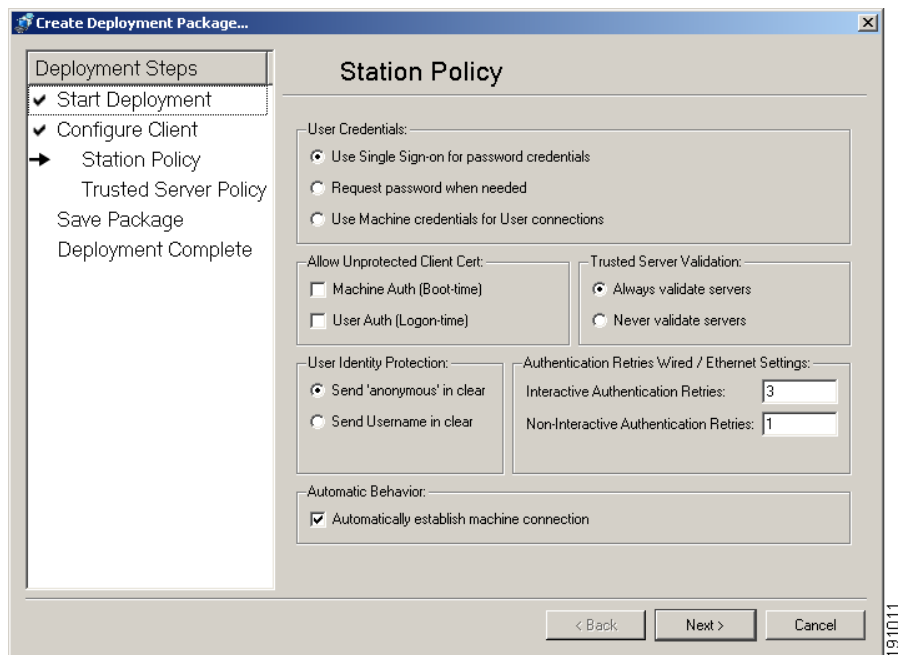
The names for the policy.xml and network.xml files are standardized so that they may be easily recognized and distributed to the appropriate hosts.

Create Deployment Package Wizard

The Create Deployment Package wizard provides an interface which allows you to configure the network connection context. The deployment package you create using the wizard consists of a **policy.xml** and a **networks.xml** file which contain the connection context information. See [“Understanding Policies and Profiles” section on page 9-24](#) for more information about these files and their roles.

The Station Policy window of the Create Deployment Package wizard displays the settings which are configured to create the policy.xml and networks.xml file.

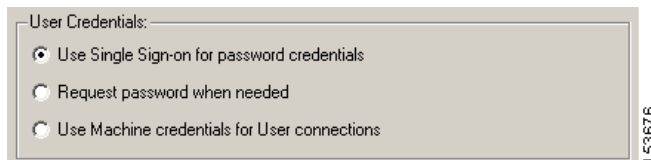
Figure 9-6 *Station Policy Window*



User Credentials Area

The User Credentials area of the Station Policy window defines which user credentials are used to authenticate the host to the ACS server. Radio buttons provide three options.

Figure 9-7 *User Credentials area*



Radio Button	Description
Use Single Sign-on for password credentials	This option uses the user's Windows Operating System password. The user is not prompted for this password.
Request password when needed	This password is recorded in ACS. It is the password defined to authenticate a client to the ACS server. This may be the Windows operating system password or it may be another. The user is prompted for this password.
Use Machine credentials for user connections	This option replaces user credentials with machine credentials. Choose this radio button when performing machine authentication only.

Automatically Establishing Machine Connection

The **Automatically establish machine connection** check box should be checked when you are performing machine authentication or machine and user authentication.

If the box is checked, the client automatically attempts to authenticate itself during boot up. If the box is not checked, machine authentication does not occur until after the GINA (Graphical Identification and Authentication) login.

Figure 9-8 Automatically establish machine connection check box.

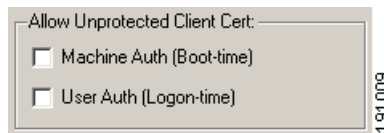


Allow Unprotected Client Cert Area

This area allows you to enable the use of client credentials during the unprotected phase of FAST PAC provisioning. The use of client credentials is optional.

These settings refer to whether or not the 802.1x Wired Client should send a certificate unprotected and not necessarily, whether or not to ever send a certificate.

Figure 9-9 Allow Unprotected Client Cert Area



- **Machine Auth (Boot-time):** This method authenticates the host to the ACS using the Microsoft Active Directory provided machine certificate.
- **User Auth (Logon-time):** This method authenticates the host to the ACS using a pre-deployed user certificate.

Configuring Unprotected Client Cert Area Settings

Checking Machine Auth (Boot-time) or User Auth (Login-time)

Checking either checkbox disables protection for the specific connection time. When requested by the server for a client certificate during the unprotected portion of FAST PAC provisioning:

- If there is a client certificate available the 802.1x Wired Client will send it unprotected or “in the clear.”
- If there is no client certificate, none will be sent and the server policy will determine if authentication continues or fails.

Not Checking Machine Auth (Boot-time) or User Auth (Login-time) checkboxes

Leaving these checkboxes unchecked enables protection for the specific connection time.

When the server requests a client certificate during the unprotected portion of FAST PAC provisioning, the client refuses to send any certificate. The client may do this because it is waiting for the protected phase of the protocol.

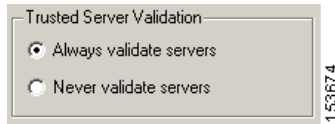
However, before the second phase of the protocol starts, a tunnel is established based on the server’s certificate. The 802.1x Wired Client will send its certificate through this tunnel before phase 2 authentication begins.

**Note**

These check boxes have no effect on the use of a client certificate during the protected portion of FAST PAC provisioning. If the server is configured to request the sending of the client certificate within the secure tunnel (e.g., FAST/TLS), the client will always attempt to use one. If none is available and not sent, the connection attempt will fail.

Trusted Server Validation Area

The Trusted Server Validation area allows you to specify the authentication servers the End-user 802.1x Wired Client can trust with their user and machine credentials.

Figure 9-10 *Trusted Server Validation area*

- Select **Always validate servers** to require authentication server validation during credential authentication. This is the most secure setting and it is the default setting.

**Note**

If you select **Always validate servers** and deploy a client profile with an empty Trusted Servers list, authentication will fail.

- Select **Never validate servers** to ignore authentication server validation during credential authentication. This is the least secure setting.

**Warning**

The “**Never validate servers**” option is not recommended because it reduces the level of security.

**Note**

There is an exception to the warning against using the “**Never validate servers**” radio button in the Trust Server Validation area.

The administrative version of the 802.1x Wired Client is the “out-of-the-box” version of the product. By default, the Administrative 802.1x Wired Client will authenticate itself to the local network using both machine and user authentication. It is intended for use by the IT organization responsible for configuring and deploying the end-user versions of the 802.1x Wired Client.

In order to specify a Trusted Server for a client, you need to create a deployment package and install it on the client. Once you install a deployment package on the client running the Administrative 802.1x Wired Client, the 802.1x Wired Client loses its Administrative capabilities and becomes an ordinary 802.1x Wired Client. That is, after the deployment package is installed, the 802.1x Wired Client will no longer be able to create authentication profiles or other deployment packages.

For that reason, the administrative version of the 802.1x Wired Client is pre-configured to support **Never validate servers** only.

Deploying Trusted Servers

Before you begin deploying trusted servers, the identification of the trusted server(s) that support the initial FAST PAC provisioning of the host needs to be configured. Two server validation methods are supported:

- Basic - requires knowledge of the Authority Identity of the associated Cisco Secure ACS.
- Server Certificate - requires pre-creation of certificate and storage on the associated server.

Initially the Trusted Servers list will be empty.



Note


If you select **Always validate servers** in the Trusted Server Validation area of the Station Policy window, and deploy a client profile with an empty Trusted Servers list, authentication will fail.

-
- Step 1** After clicking **Next** in the Station Policy window, the Trusted Server Policy Window opens.
- Step 2** Click **Add Server Rule** to open the Trusted Server Dialog.
- Step 3** Enter a name in the **Rule name** field. This can be any user-friendly name; it is not used for validation.
- Step 4** In the **Validation method field**, select **Certificate**. This implies that the host will initially provision the PAC using a server certificate. (The PAC validation method is not supported.)



Note

The client, when initially accepting the provisioned PAC, will autonomously add the A-ID to the deployed Trusted Server list - transferring the “trust” in the server certificate to the server's A-ID.

- Step 5** Initially the “Match ANY Certificate Validation Rules” list will be empty. You will need to add a rule by configuring the fields in the Match ANY Certificate Rules area.
- a. 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 from the following list of acceptable certificate attributes: Choose either or both.
 - **Subject Alternative Domain** - searches the DNSName field attribute.
 - **Subject Common Name** - searches the following certificate fields (attributes):
 - Subject: CN (Common Name)
- 

Note If multiple Common Names are specified, only the first one listed in the certificate is used.

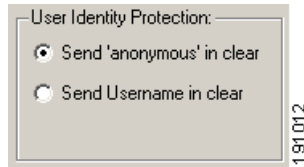
- Subject: DN (Domain Name) - a composite of a set of DC (Domain Component) attributes
- b. Select a validation comparison rule:
 - **Exactly matches** - the certificate field must contain the full contents of the text box
 - **Ends with** - the certificate field must end with the contents of the text box.
 - c. Enter the text: acceptable formats are:
 - **label** - <string>
 - **realm** - <string> . <string> etc.


Step 6 Click **OK**, to return to the Trusted Server Definition Dialog.

Step 7 Click **Next** to continue creating a deployment package.

User Identity Protection Area

The **User Identity Protection** area defines what information is sent during an EAP Identity request.

Figure 9-11 User Identity Protection

Radio Button	Description
Send 'anonymous' in clear	<p>Select the “Send ‘anonymous’ in clear” option to send anonymous as the EAP-FAST outer identity.</p> <p>In this mode the host sends <i>anonymous@Domain</i> for the identity. This is the default setting.</p>
Send Username in clear	<p>Select the Use Username as identity option to send the <i>UserName</i> as the EAP-FAST outer identity.</p> <p>This could be used for login attempt auditing on the switch.</p> <p>In this mode, when also using “Use Single Sign-on for password credentials” for User Credentials, the client sends <i>UserName@Domain</i> for the identity. This information is sent in the clear.</p> <p>If not using “Single Sign-on for password credentials” for User Credentials, the client sends <i>UserName</i> for the identity. This information is also sent in the clear.</p> <div>  <p>Caution This radio button choice is not compatible with ACS 4.0 installations or earlier. Support for this feature will be available in a future release of ACS.</p> </div>

**Note**

This setting does not affect the credentials sent during user and machine authentication. If user credentials are required for authentication *UserName@Domain* will be sent.

**Note**

This authentication choice also needs to be configured on the ACS.

Authentication Retries Wired / Ethernet Settings

Some network access devices have the ability to open a port but switch the user into a special vlan after a failed connection attempt. In order to support these access devices, the client provides the administrator with the capability of adjusting the number of connection retries before disconnecting, allowing the access device to make intelligent decisions based on multiple authentication failures.

Figure 9-12 Authentication Retries Wired / Ethernet Settings Area

Interactive Authentication Retries: Applies for cases in which a user intervention might correct the failed authentication attempt. In general, this applies to connection attempts involving user text entry or user list selection associated with an “Enter Your Credentials” pop-up window, to allow for user corrections. The default setting is four attempts.

Non-interactive Authentication Retries: Applies for cases in which a user intervention would not help to correct the failed authentication attempt. In general, this applies to connection attempts not involving an “Enter Your Credentials” pop-up window, such as, single-sign-on, or all failures associated with a server certificate validation. The default setting is four attempts.

**Note**

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

- Increasing the **Interactive Authentication Retries** count will result in more user dialog prompts.
- Increasing the **Non-Interactive Authentication Retries** count will add a delay to the machine boot/user logon connection to the Windows System in cases where network connectivity fails.

When you decide on the proper settings for these fields, the values of these fields must be at least one greater than the value of the max-attempts value for the “auth fail vlan” feature on your switch. For example, if max-attempts value for “auth fail vlan” is 3 on the switch, both values must be set to 4.

Deploying End-User 802.1x Wired Clients

Deploying the End-user 802.1x Wired Clients requires that you perform three procedures:

-
- Step 1** Create a Deployment Package. In this step you create a deployment package based on one of these authentication strategies:
- User Authentication only
 - Machine and User Authentication
 - Machine Authentication only
- See [“Creating Deployment Packages” section on page 9-36](#) for these procedures.
- Step 2** Installing Server Certificates on the Host. See [“Installing Server Certificates on the Host” section on page 9-42](#).
- Step 3** Installing the End-user 802.1x Wired Client and the Deployment Package on the host. See [“Installing Deployment Packages on Hosts” section on page 9-43](#) for this procedure.

Creating Deployment Packages

This section describes creating these different authentication packages:

- [Creating a User Authentication Deployment Package, page 9-36](#)
- [Create a Machine and User Authentication Deployment Package, page 9-38](#)
- [Creating a Machine Authentication Only Deployment Package, page 9-40](#)

Creating a User Authentication Deployment Package

If you want to provide network access only after a user has logged on to a machine, authenticate user credentials only.

-
- Step 1** Open Cisco Trust Agent 802.1x Wired Client.
- Step 2** From the Administration menu, click **Create Deployment Package**.
- Step 3** At the Welcome window, click **Start**. The Station Policy window opens.
- Step 4** In the Automatic Behavior area at the bottom of the window, uncheck the **Automatically establish machine connection** check box. See the [“Automatically Establishing Machine Connection” section on page 9-27](#) for more information on this checkbox.
- Step 5** In the User Credentials area, select either of these radio buttons:
- **Use Single Sign-on for password credentials**. This option passes the username and password from the Windows logon to the ACS.
 - **Request password when needed**. This option prompts users for their username and password when they are trying to connect to the network. This username and password may be different from the Windows logon information. This value is configured in ACS.
- See the [“User Credentials Area” section on page 9-27](#), for more information on these radio buttons.
- Step 6** In the User Identity Protection area, select the “Send ‘anonymous’ in clear radio button. See the [“User Identity Protection Area” section on page 9-32](#) for more information about the radio buttons in this area.

- Step 7** (Optional) In the Allow unprotected Client Cert area, check **User Auth (Logon-time)** if you want to use a client certificate during Phase 1 of FAST PAC provisioning. See the [“Allow Unprotected Client Cert Area” section on page 9-28](#) for more information about this radio button.
- Step 8** (Optional) In the Authentication Retries Wired/Ethernet Settings area, specify the number of interactive or non-interactive authentication retries. See the [“Authentication Retries Wired / Ethernet Settings” section on page 9-34](#) for more information about these fields.
- Step 9** (Optional) In the Trusted Server Validation area, select one of these options:
- Always validate servers
 - Never Validate servers
- See the [“Trusted Server Validation Area” section on page 9-29](#) for more information about these radio buttons.
- Step 10** Click **Next**. The Trusted Server Policy window opens.
- Step 11** Add a trusted server rule, if it is appropriate:
- If you chose to Always validate servers in the Trusted Server Validation area, add a server rule at this time. See the [“Deploying Trusted Servers” section on page 9-31](#) for this procedure.
 - If you chose Never Validate servers in the Trusted Server Validation area, click **Next**.
- Step 12** In the Save Package window, you can accept the default location in which to save the new deployment package .xml files or click **Browse** to store the files in a new or existing directory.
- Step 13** In the Deployment Package filename field, specify a filename. This name is incorporated in all the authentication profile files. For the sake of an example, we will name the file “common-auth”.
- Step 14** Click **Save**. The deployment package is created and the file names and locations are listed in the Deployment Complete window. For the sake of this example, these were the files that were created:
- C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-policy.xml**
 - C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-networks.xml**

- C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-credentials.xml**

Step 15 Click **Close**.

Step 16 Install the authentication server certificates on the host using the [“Installing Server Certificates on the Host”](#) section on page 9-42.

Create a Machine and User Authentication Deployment Package

This configuration provides network access only after both the user and the machine credentials have been authenticated.

Step 1 Open the Cisco Trust Agent 802.1x Wired Client.

Step 2 From the Administration menu, click **Create Deployment Package**.

Step 3 At the Welcome window, click **Start**. The Station Policy window opens.

Step 4 In the Automatic Behavior area at the bottom of the window, uncheck the **Automatically establish machine connection** check box. See the [“Automatically Establishing Machine Connection”](#) section on page 9-27 for more information on this checkbox.

Step 5 In the User Credentials area, select either of these radio buttons:

- **Use Single Sign-on for password credentials.** This option passes the username and password from the Windows logon to the ACS.
- **Request password when needed.** This option prompts users for their username and password when they are trying to connect to the network. This username and password may be different from the Windows logon information. This value is configured in ACS.

See the [“User Credentials Area”](#) section on page 9-27, for more information on these radio buttons.

Step 6 In the User Identity Protection area, select the **Send ‘anonymous’ in clear** radio button. See the [“User Identity Protection Area”](#) section on page 9-32 for more information about the radio buttons in this area.

Step 7 (Optional) In the Use Client Certificate During area, check **User Authentication** if you want to use a client certificate during Phase 1 of FAST PAC provisioning. See [“Allow Unprotected Client Cert Area”](#) section on page 9-28 for more information.

- Step 8** (Optional) In the Authentication Retries Wired / Ethernet Settings area, specify the number of interactive or non-interactive authentication retries. See the [“Authentication Retries Wired / Ethernet Settings” section on page 9-34](#) for more information.
- Step 9** (Optional) In the Trusted Server Validation area, select one of these options:
- Always validate servers
 - Never Validate servers
- See, [“Trusted Server Validation Area” section on page 9-29](#) for more information.
- Step 10** Click **Next**. The Trusted Server Policy window opens.
- Step 11** Add a trusted server rule, if it is appropriate:
- If you chose to Always validate servers in the Trusted Server Validation area, Add a server rule at this time. See [“Deploying Trusted Servers” section on page 9-31](#) for this procedure.
 - If you chose Never Validate servers in the Trusted Server Validation area, click **Next**.
- Step 12** In the Save Package window, you can accept the default location in which to save the new deployment package .xml files or click **Browse** to store the files in a new or existing directory.
- Step 13** In the Deployment Package filename field, specify a filename. This name is incorporated in all the connection context files. For the sake of an example, we will name the file “common-auth”.
- Step 14** Click **Save**. The deployment package is created and the file names and locations are listed in the Deployment Complete window. For the sake of this example, these were the files that were created:
- C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-policy.xml**
 - C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-networks.xml**
 - C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-credentials.xml**

**Note**

The credentials.xml file is created but it is only meaningful when it is used in a wireless network environment. The CTA 802.1x Wired Client is only used in wired network environments.

Step 15 Click **Close**.

Step 16 Install the authentication server certificates on the host using the [“Installing Server Certificates on the Host”](#) section on page 9-42.

Creating a Machine Authentication Only Deployment Package

This procedure creates a policy for machine authentication only. For machine authentication, the machine credentials are substituted for user credentials. The machine connection to the network may be made automatically or manually.

During an automatic machine authentication, the host boots and the machine credentials are immediately submitted for authentication.

During a manual machine authentication, the host boots but its machine credentials are not submitted for authentication. Machine credentials are sent after the user logs on.

Step 1 Open the Cisco Trust Agent 802.1x Wired Client.

Step 2 From the Administration menu, click **Create Deployment Package**.

Step 3 At the Welcome window, click **Start**. The Station Policy window opens.

Step 4 In the User Credentials area, select **Use Machine credentials for User Connections** radio button. See [“User Credentials Area”](#) section on page 9-27 for more information.

Step 5 Make the machine connection automatic or manual:

- To establish the machine connection upon boot, **check** the **Automatically establish machine connection** check box in the Automatic Behavior area at the bottom of the window.
- To establish the machine connection after GINA login, **uncheck** the **Automatically establish machine connection** check box in the Automatic Behavior area at the bottom of the window.

- Step 6** In the User Identity Protection area, select the **Send ‘anonymous’ in clear** radio button. See the [“User Identity Protection Area” section on page 9-32](#) for more information about the radio buttons in this area.
- Step 7** (Optional) In the Use Client Certificate During area, check **User Auth (Logon-time)** if you want to use a client certificate during Phase 1 of FAST PAC provisioning. See [“Allow Unprotected Client Cert Area” section on page 9-28](#) for more information.
- Step 8** (Optional) In the Authentication Retries Wired / Ethernet Settings area, specify the number of interactive or non-interactive authentication retries. See the [“Authentication Retries Wired / Ethernet Settings” section on page 9-34](#) for more information.
- Step 9** (Optional) In the Trusted Server Validation area, select one of these options:
- Always validate servers
 - Never Validate servers
- See, [“Trusted Server Validation Area” section on page 9-29](#) for more information.
- Step 10** Click **Next**. The Trusted Server Policy window opens.
- Step 11** Add a trusted server rule, if it is appropriate:
- If you chose to Always validate servers in the Trusted Server Validation area, Add a server rule at this time. See [“Deploying Trusted Servers” section on page 9-31](#) for this procedure.
 - If you chose Never Validate servers in the Trusted Server Validation area, click Next.
- Step 12** In the Save Package window, you can accept the default location in which to save the new deployment package .xml files or click **Browse** to store the files in a new or existing directory.
- Step 13** In the Deployment Package filename field, specify a filename. This name is incorporated in all the connection context files. For the sake of an example, we will name the file “common-auth”.
- Step 14** Click **Save**. The deployment package is created and the file names and locations are listed in the Deployment Complete window. For the sake of this example, these were the files that were created:
- C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-policy.xml**

- C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-networks.xml**
- C:\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client**common-auth-credentials.xml**

**Note**

The credentials.xml file is created but it is only meaningful when it is used in a wireless network environment. The CTA 802.1x Wired Client is only used in wired network environments.

Step 15 Click **Close**.

Step 16 Install the authentication server certificates on the host using the [“Installing Server Certificates on the Host”](#) section on page 9-42.

Installing Server Certificates on the Host

Install server certificates on the host in order to support machine connections or user connections.

After the server certificate have been installed on the host, continue with the [“Installing Deployment Packages on Hosts”](#) section on page 9-43.

Machine Connection Support

If the 802.1x Wired Client for the end-user is using the Server Certificate method of initially provisioning the machine PAC, then perform these two steps:

Step 1 The CA certificate used to trust the server certificate must be deployed and placed in the proper Windows Certificate Store (Local Computer-Trusted Root Store).

Step 2 The end-user 802.1x Wired Client should be configured to be using the Microsoft Active Directory method of providing the machine certificate.

**Note**

If a machine certificate is not supported, then no initial machine connection will be made and the machine PAC provisioning will wait until the initial user logon connection is made.

User Connection Support

If the end-user client is using the Server Certificate method of initially provisioning the end-user's PAC, then the CA certificate used to trust the server certificate must be deployed and placed in the proper Windows Certificate Store (User-Trusted Root Store).

Installing Deployment Packages on Hosts

Before you begin, test the deployment package before distributing it to the end-users. It is recommended to load the deployment package on an alternate machine so as to maintain the administrator version of the client on the current machine.

**Note**

If you must test on the same machine, be sure to save the administrator configuration files, so that you can reconstruct the administrator version after the test.

-
- Step 1** Install Cisco Trust Agent with the 802.1x Wired Client on the host that you want to use this connection context. Do not restart the host when prompted.
 - Step 2** Copy the deployment package files you created to the host.
 - Step 3** Copy the policy file to the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\policies** directory. (In our example the policy file is named common-auth-policy.xml.)
 - Step 4** Copy the networks file to the **Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\networks** directory. (In our example the network file is named common-auth-networks.xml.)
 - Step 5** The credentials.xml file is not required for the CTA 802.1x Wired Client.
 - Step 6** Reboot the host.

Changing Deployment Packages on Hosts

Changing the deployment package on an End-user 802.1x Wired Clients is very similar to deploying the original package.

-
- Step 1** Create a new deployment package. Create a deployment package based on **one** of these authentication strategies:
- [Creating a User Authentication Deployment Package, page 9-36](#)
 - [Create a Machine and User Authentication Deployment Package, page 9-38](#)
 - [Creating a Machine Authentication Only Deployment Package, page 9-40](#)
- Step 2** Follow the [Replacing a Deployment Package on a Host](#) procedure.

Replacing a Deployment Package on a Host

Using this procedure you can change the way a host authenticates itself to the network.

-
- Step 1** Open the Windows Services window and stop the Cisco Trust Agent 802.1x Wired Client.
- Step 2** Delete the policy file from the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\policies** directory. (In our example the policy file is named common-auth-policy.xml.)
- Step 3** Copy the policy file from the new deployment package to the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\policies** directory.
- Step 4** Delete the networks file from the **Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\networks** directory. (In our example the network file is named common-auth-networks.xml.)
- Step 5** Copy the networks file from the new deployment package to the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\networks** directory.
- Step 6** Delete the user PAC from the **\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\users** directory. This file is named *UserName@hostname.xml*

- Step 7** Delete the user PAC from the `\Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\profiles\policies\users\allusers` directory. The file is named **machineSettings.xml**.
- Step 8** Reboot the host or restart the Cisco Trust Agent 802.1x Wired Client in the Windows Services window.



CHAPTER 10

Cisco Trust Agent 802.1x Wired Client Logging

In the event of operational problems, with local hardware, with a network access device or authentication server, or internally, these features are available to aid a user or support technician debug an unexpected event in the client:

- Technical Log
- System Report



Tip

When using any antivirus software with the client, it is best to configure the antivirus software, if possible, to ignore scanning/processing current “active” log files in order to avoid consuming processing resources during an authentication.

This chapter contains the following sections:

- [Technical Log, page 10-2](#)
- [Understanding the Technical Log Status and Error Messages, page 10-3](#)
 - [Technical Log Message Format, page 10-3](#)
 - [Technical Log Message Content, page 10-5](#)
 - [Additional Message <value> Descriptions, page 10-11](#)
 - [Port Status Values, page 10-12](#)
- [System Report, page 10-15](#)
 - [Creating a System Report, page 10-16](#)

Technical Log

The technical log file is a time-stamped, Unicode text file that is the destination for log messages capable of being viewed with Notepad (or equivalent) on Windows 2000 and Windows XP. These are the characteristics of the Technical Log:

- The “file” is actually a series of files. The client stops using the current log file and creates a new log file whenever the client starts up or the maximum file size is reached (1MB).
- The set of files have a maximum amount of allocated non-volatile (disk) space of approximately 5 MB. When the maximum storage level is reached, the oldest log file is deleted.
- The current log file has the format: log_current.txt.
- Each archived file has the following naming format: log_<date>_<time>.txt, where <date> has the format YYYY-MM-DD, where YYYY is the year, MM is the month and DD is the day and <time> has the format: hh.mm.ss, where hh is the hour, mm is the minute and ss is the second.

The date/time indicates when the file was archived. Archived files therefore contain events prior to this time.

- The set of files are located in a folder named ‘log’, below the main install folder. This would be **Program Files\Cisco Systems\Cisco Trust Agent 802_1x Wired Client\log** for the default install folder.
- Each log file contains a list of single line entries, where each entry defines a single log event.

The technical log level is intended for those users who have training in 802.1x, 802.11i, EAP, EAP methods, PKI and understand the profiles and policies of the client.

See [“Understanding the Technical Log Status and Error Messages”](#) section on page 10-3 for a description of all the error codes.

Understanding the Technical Log Status and Error Messages

This section describes the format and contents of the Technical Log status and error messages.

Technical Log Message Format

Every log entry has the format:

<date & time> [process] <log message id> <log message class> <context IDs> <grammatical log message>

These are the descriptions of each part of a log entry:

Table 10-1 **Log Entry Field Descriptions**

Log Entry Field	Description
<date & time>	Provided in the format MM/DD/YYYY HH:mm:SS.sss, where: <ul style="list-style-type: none">• MM - numeric month (01-12)• DD - numeric day (01-31)• YYYY - numeric year (e.g. 2005)• HH - numeric hour on a 24 hour clock (00-23)• mm - numeric minutes past the hour (00-59)• SS.sss - numeric seconds with SS being seconds, and sss being fractions of a second.
[process]	An internal process identifier developers use in troubleshooting problems.
<log message id>	The unique number for the log message.

Table 10-1 Log Entry Field Descriptions (continued)

Log Entry Field	Description
<log message class>	<p>Determines the type of log message and is one of the following:</p> <ul style="list-style-type: none"> • I – informational log message - used to indicate a client state that is part of normal processing. • W – warning log message - used to indicate a client state that is insecure or unexpected but which still allows processing. • E – error log message - used to indicate an exception that prevents normal processing.
<context IDs>	<p>Conveys zero or more identifiers to define the context of this log event. Each has the following format:</p> <ul style="list-style-type: none"> • <code><unique string/number> where: • n < code> is a two letter code that indicates the class of the term. • n < unique string/number> is string or number that is guaranteed to be unique. • Adapter Identifier - AD<MAC address in hexadecimal for the adapter> • Access Identifier - AC<MAC/BSSID for the access device> • Media Type Identifier - MT<Ethernet WiFi> (Note: MT may or may not be explicitly indicated) • Connection Identifier - CN<an incrementing integer> • Profile Identifier - PR<profile name truncated to 16 characters>
<grammatical log message>	A sentence that describes the event. It may also contain a variable <value>.
<value>	The <value> in the <grammatical log message> is a placeholder for a variable value to be placed in the message.

Example 10-1 Technical log content:

```
04/20/2006 15:28:47.859 [ 432. 728] 103 I CN<3> Cisco Trust Agent 802.1X wired client
AD<000cf1aeddfe> AC<000cf1aeddfe> Connection Requested automatically from user context.
04/20/2006 15:28:47.875 [ 432.1716] 109 I CN<3> AD<000cf1aeddfe> Connection
Authentication Started in user context.
```

```

04/20/2006 15:28:47.875 [ 432.1136] 29 I CN<3> AD<000cflaeddffc> Port State Machine
transition to AC_PORT_STATE_CONNECTING(AC_PORT_STATUS_STARTED)
04/20/2006 15:28:48.812 [ 432.1136] 29 I CN<3> AD<000cflaeddffc> Port State Machine
transition to AC_PORT_STATE_UNAUTHENTICATED(AC_PORT_STATUS_EAP_FAILURE)
04/20/2006 15:28:48.812 [ 432.1136] 77 E CN<3> AD<000cflaeddffc> Connection
Authentication Failed.
04/20/2006 15:28:49.828 [ 432.1136] 29 I CN<3> AD<000cflaeddffc> Port State Machine
transition to AC_PORT_STATE_AUTHENTICATING(AC_PORT_STATUS_8021x_ACQUIRED)
04/20/2006 15:28:49.843 [ 432.1716] 24 I CN<3> AD<000cflaeddffc> Identity requested.
04/20/2006 15:28:58.968 [ 432.1136] 25 I CN<3> AD<000cflaeddffc> Identity sent.
04/20/2006 15:28:58.984 [ 432.1532] 28 I CN<3> AD<000cflaeddffc> Authentication method
started: EAP-FAST, level 0
04/20/2006 15:28:59.000 [ 432.1136] 26 I CN<3> AD<000cflaeddffc> EAP method suggested by
server: EAP-FAST
04/20/2006 15:28:59.000 [ 432.1136] 27 I CN<3> AD<000cflaeddffc> EAP methods requested by
client: EAP-FAST
04/20/2006 15:28:59.015 [ 432. 728] 73 I CN<3> Client is validating the server.
04/20/2006 15:28:59.015 [ 432. 728] 140 I CN<3> Server AID validated:
57dda0ae0004a74f8c7c959d687c4ed2
04/20/2006 15:28:59.062 [ 432.1532] 28 I CN<3> AD<000cflaeddffc> Authentication method
started: EAP-GTC, level 1
04/20/2006 15:28:59.062 [ 432.1136] 26 I CN<3> AD<000cflaeddffc> EAP method suggested by
server: EAP-GTC
04/20/2006 15:28:59.062 [ 432.1136] 27 I CN<3> AD<000cflaeddffc> EAP methods requested by
client: EAP-GTC
04/20/2006 15:28:59.062 [ 432.1532] 24 I CN<3> AD<000cflaeddffc> Identity requested.
04/20/2006 15:28:59.078 [ 432.1136] 25 I CN<3> AD<000cflaeddffc> Identity sent.
04/20/2006 15:29:04.078 [ 432.1136] 29 I CN<3> AD<000cflaeddffc> Port State Machine
transition to AC_PORT_STATE_AUTHENTICATED(AC_PORT_STATUS_8021x_ACQUIRED)

```

Technical Log Message Content

These are the messages that can be recorded in the technical log file.



Note

See [“Additional Message <value> Descriptions”](#) section on page 10-11 for the descriptions of the message <value> fields.



Note

See [“Port Status Values”](#) section on page 10-12 for the list of expanded descriptions of a <Port State> value.

Understanding the Technical Log Status and Error Messages

Table 10-2 Technical Log Messages and Codes

Class	ID	Context IDs	Message
Client processing messages			
I	1		Client Service Auto Started. <Client's service name>, <version number>, <OS Name>
I	101		Client Service Manually Started. <Client's service name>, <version number>, <OS Name>
I	2		Client Service Normal Shutdown. <Client's service name>, <version number>, <OS Name>
E	133		Client Service Fatal Error Shutdown. <Client's service name>, <version number>, <OS Name> <i>Recovery Action:</i> Manually stop and start the service or in extreme cases, uninstall and reinstall the client (your configuration files will be maintained).
I	3		Boot processing initiated.
Client environment processing messages			
I	85		Entering power save mode. <i>Note:</i> Entering standby/hibernate mode.
I	86		Exiting power save mode (automatic) <i>Note:</i> Exiting standby mode - will be followed with Error Msg #87.
I	87		Exiting power save mode. <i>Note:</i> Exiting standby mode if preceded by Error Msg #86, otherwise exiting hibernate mode.
User Logon processing messages			
I	4		User logon processing initiated.
I	134		Manual user <logon type> logon processing initiated by user <user id>.
I	129		User single sign-on credentials obtained from Novell GINA
I	130		User single sign-on credentials obtained from Microsoft GINA

Table 10-2 Technical Log Messages and Codes (continued)

Class	ID	Context IDs	Message
I	5		User logoff processing initiated
Adapter processing messages			
I	6	AD<> MT<>	Adapter Detected.
I	8	AD<>	Adapter Controlled.
E	30	AD<>	Adapter startup failed because driver is in use. <i>Recovery Action:</i> Manually disable competing utility.
I	14	AD<>	Control has been released for this adapter.
I	135	AD<>	Wired Access device disappeared.
I	7	AD<>	Adapter Removed.
I	95	AD<>	User: User requested client to manage adapter
I	96	AD<>	User: User requested client to not manage adapter
Access device processing messages			
I	15	AC< >	Wired Access device detected.
Connection processing messages			
I	16	CN<> PR<> AD<> AC<>	Connection Requested automatically from machine context.
I	103	CN<> PR<> AD<> AC<>	Connection Requested automatically from user context.
I	104	CN<> PR<> AD<> AC<>	Connection Requested by user from user context.
I	94	PR<>	User: User requested disconnect for network.
I	17	CN<>	Connection Terminated by user request.
I	105	CN<>	Connection Terminated due to service shutdown.
I	106	CN<>	Connection Terminated because adapter was removed.
I	107	CN<>	Connection Terminated because access device disappeared.

Table 10-2 Technical Log Messages and Codes (continued)

Class	ID	Context IDs	Message
E	108	CN< >	Connection Terminated due to fatal error number <error number>: <error text>. <i>Recovery Action:</i> Manually restart the Cisco Trust Agent 802.1x Wired Client service.
Connection processing - IP specific messages			
I	82	CN< >	DHCP: Sending DHCP request.
E	84	CN< >	DHCP: Request failed because of time out. <i>Recovery Action:</i> Verify network readiness - failure outside of client.
E	110	CN< >	DHCP: Server responded with failure. <i>Recovery Action:</i> Verify network readiness - failure outside of client.
E	111	CN< >	DHCP: Unknown failure has occurred. <i>Recovery Action:</i> Verify network readiness - failure outside of client.
I	78	CN < >	Connection IP Address Received: Address: <IP Address>.
Authentication processing messages			
I	23	CN< > AD< >	Connection Authentication Started in machine context.
I	109	CN< > AD< >	Connection Authentication Started in user context.
I	24	CN< > AD< >	Identity requested.
I	25	CN< > AD< >	Identity sent.
I	26	CN< > AD< >	EAP method suggested by server: <Authentication Method name>.
I	27	CN< > AD< >	EAP methods requested by client: (<Authentication Method name>, ..., <Authentication Method name>).
I	28	CN< > AD< >	Authentication method started: <tunnel depth>, <sequence number>, <Authentication Method name>.
I	29	CN< > AD< >	Port State Machine transition to <Port State>(<Port status>).
I	76	CN< > AD< >	Connection Authentication Success.

Table 10-2 Technical Log Messages and Codes (continued)

Class	ID	Context IDs	Message
E	77	CN< > AD< >	Connection Authentication Failed. <i>Recover Action:</i> Verify consistency of client, access point and server configuration.
EAP Notification messages			
I	143	CN< >	EAP Notification message received from: <ssid> <EAP Notification>
Authentication processing - FAST specific messages			
W	125	CN< > AD< >	FAST: unauthenticated provisioning supported.
Authentication processing - server validation specific messages			
W	72	CD< >	Trusted Server list empty, server can not be validated.
I	73	CN< >	Client is validating the server.
I	74	CD< >	Server certificate validated: <Authentication Server Id>.
W	142	CD< >	Profile does not require server validation.
E	75	CD< >	Server certificate invalid because unknown CA. <i>Recovery Action:</i> Verify that the correct CA certificate is in the Windows trusted root certificate store.
E	115	CD< >	Server certificate invalid because CN mismatch in Subject: <CN name from server cert>. <i>Recovery Action:</i> Verify the server validation rule configuration.
E	116	CD< >	Server certificate invalid because DC mismatch in Subject: <DC name from server cert>. <i>Recovery Action:</i> Verify the server validation rule configuration.
E	117	CD< >	Server certificate invalid because Subject Alternative Name mismatch: <Alternative name from server cert>. <i>Recovery Action:</i> Verify the server validation rule configuration.
I	140	CN< >	Server AID validated: <AID-info>
E	141	CN< >	Server not trusted because AID mismatch: <AID-info> <i>Recovery Action:</i> Verify the server validation rule configuration.

Table 10-2 Technical Log Messages and Codes (continued)

Class	ID	Context IDs	Message
User profile configuring - manage trusted servers messages			
I	97		User: User added certificate based trusted server <Rule name>: <certificate-based trusted server rule>
I	112		User: User added pac based trusted server <Rule name>: with AID: <AID-info>
I	98		User: User removed all trusted servers.
I	99		User: User modified trusted server list, <certificate-based trusted server rule>.
License processing messages			
I	89		Licensing: License file found.
E	90		Licensing: License file not found. <i>Recovery Action:</i> verify existence of the <install folder>\licenseTransport.txt file.
I	91		Licensing: License read: <License string>.
W	92		Licensing: License invalid because trial period expired <License string>, <trial period>.
W	118		Licensing: License invalid because termination date reached: <License string>, <termination date>.
W	119		Licensing: License invalid because operating system mismatch: <License string>, <licensed os>.
W	120		Licensing: License invalid because product id does not match: <License string>, <licensed product id>.
W	121		Licensing: License invalid because OEM id does not match: <License string>, <licensed OEM id>.
W	122		Licensing: License invalid because maintenance date reached: <License string>, <maintenance date>.
W	123		Licensing: License invalid due to unknown problem: <License string>, <termination date>.
W	131		Licensing: Ignoring trial license. Tampering detected: <License string>.

Table 10-2 Technical Log Messages and Codes (continued)

Class	ID	Context IDs	Message
I	93		Licensing: License is valid and accepted: <License string>.
Internal messages			
W	0		Technical log message ID[<msgId>] not found.

Additional Message <value> Descriptions

Table 10-3 Message <value> Variables and Descriptions

Variables in log messages	Description
<Client's service name>	The Windows service name for the client.
<version number>	The version number of the client.
<OS Name>	The operating system for which the client was built: Windows 2K/XP
<logon type>:	Novell, Windows
<user id>	User id for user logging on to endpoint.
<error number>	An internal error number.
<error text>	If the <error number> has a text equivalent.
<Authentication Method name>	EAP-PEAP, EAP-TTLS, EAP-TLS, EAP-LEAP, EAP-MD5, EAP-GTC, EAP-FAST, EAPSIM, EAP-MSCHAPv2, MSCHAPv2, MSCHAP, CHAP, PAP.
<tunnel depth>	A number indicating authentication tunnel depth starting at 0 for outer most and 1 for the inner nested method.
<sequence number>	A number indicating where in a chain of authentications this authentication is beginning.
<port state>	The adapter authentication AC_PORT_STATE values: _STOPPED, _CONNECTING, _AUTHENTICATING, _AUTHENTICATED, _REAUTHENTICATING, _UNAUTHENTICATED, _AUTH_NOT_REQD.

Table 10-3 *Message <value> Variables and Descriptions (continued)*

Variables in log messages	Description
<port status>	More detailed information on the success/failure of the authentication (and other associated state changes). It often acts as a sub-status of a particular AC_PORT_STATE. See “Port Status Values” section on page 10-12 for the description of these values.
<AID-info>	The AID (Authority/Server Identifier) in the PAC.
<Authentication Server Identifier>	The fully qualified domain name for the server or the PAC info field truncated to 16 characters.
<EAP Notification>	Unsolicited messages from the authentication server.
<IP Address>	IP address that the end station will use in the standard IP format xxx.xxx.xxx.xxx.
<rule name>	Trusted server rule name.
<certificate-based trusted server rule>	Defines the trusted server rule.
<License string>	The license string read from the license file.
<trial period>	The number of days in trial period.
<termination date>	Date in format yyyy-mm-dd that the license expired.
<licensed os>	The name of the operating systems that the license allows.
<licensed product id>	The product id that the license allows.
<licensed OEM id>	The OEM id that the license allows.

Port Status Values

Some messages describe a port’s state and a port status, for example, “Port State Machine transition to <Port State>(<Port status>).” This section describes the possible port status values.

Status codes related to running state

AC_PORT_STATUS_UNKNOWN

AC_PORT_STATUS_STOPPED

AC_PORT_STATUS_STARTED

Status codes related to link state

AC_PORT_STATUS_LINK_DOWN

AC_PORT_STATUS_LINK_UP

AC_PORT_STATUS_LINK_RESET

Status codes related to 802.1x state machine

AC_PORT_STATUS_8021x_START

AC_PORT_STATUS_8021x_FAILED

AC_PORT_STATUS_8021x_ACQUIRED

AC_PORT_STATUS_8021x_LOGOFF

AC_PORT_STATUS_8021x_TIMEOUT

Error and status codes during 802.1x authentication

AC_PORT_STATUS_ERR_CLIENT_EAP_METHOD_REJECTED

AC_PORT_STATUS_ERR_CLIENT_GENERIC_REJECTED

AC_PORT_STATUS_ERR_CLIENT_IDENTITY_REJECTED

AC_PORT_STATUS_ERR_CLIENT_TLS_CERTIFICATE_REJECTED

AC_PORT_STATUS_ERR_CHALLENGE_TO_AP_FAILED

AC_PORT_STATUS_ERR_ROGUE_AUTH_TIMEOUT

AC_PORT_STATUS_ERR_SERVER_TLS_CERTIFICATE_REJECTED

AC_PORT_STATUS_ERR_UNKNOWN

AC_PORT_STATUS_ERR_RESTRICTED_LOGON_HOURS

AC_PORT_STATUS_ERR_ACCT_DISABLED

AC_PORT_STATUS_ERR_NO_DIALIN_PERMISSION

AC_PORT_STATUS_ERR_CHANGING_PASSWORD

AC_PORT_STATUS_ERR_INVALID_TLV

AC_PORT_STATUS_ERR_UNKNOWN_TLV
AC_PORT_STATUS_ERR_TLV_NAK_RECEIVED
AC_PORT_STATUS_ERR_INVALID_CMAC
AC_PORT_STATUS_ERR_NO_CRYPTOBINDING
AC_PORT_STATUS_EAP_FAST_PROVISIONING
AC_PORT_STATUS_ERR_EAP_FAST_INVALID_PAC_OPAQUE
AC_PORT_STATUS_ERR_EAP_FAST_INVALID_PAC_KEY

Status codes related to EAP

AC_PORT_STATUS_EAP_FAILURE
AC_PORT_STATUS_EAP_SUCCESS
AC_PORT_STATUS_WRN_CLEARTEXT_EAP_FAILURE
AC_PORT_STATUS_WRN_CLEARTEXT_EAP_SUCCESS

Status codes related to credentials

AC_PORT_STATUS_ERR_WRONG_PIN
AC_PORT_STATUS_ERR_PIN_REQUIRED
AC_PORT_STATUS_ERR_NO_DEVICE
AC_PORT_STATUS_ERR_NO_CARD
AC_PORT_STATUS_ERR_SIM_FAILURE

Status codes related to CCX

AC_PORT_STATUS_POSSIBLE_ROGUE_AP_START
AC_PORT_STATUS_POSSIBLE_ROGUE_AP_STOP
AC_PORT_STATUS_CCX_CCKM_ROAM

System Report

The System Report utility provides end users a simple way to automatically gather data needed by support personnel to troubleshoot any problems. It captures the following information:

- Current end-user technical log contents.
- Current internal application activity log.
- Information on the machine's hardware and software environment.

The System Report utility is packaged with the CTA 802.1x Wired Client and automatically installed with the CTA 802.1x Wired Client, however, it is a separate utility and it operates whether the CTA 802.1x Wired Client is active or not.

The System Report utility creates a single compressed file, the System Report, that contains information about the end station's hardware and software environment, the CTA 802.1x Wired Client, as well as the gathered technical and developer logs. The compressed file has these features:

- A consolidated and compressed collection of files
- Uses a non-configurable file name:
CiscoLiteSysRepLog<YYYYMMDD_hhmm>.zip, where YYYY is the year, MM is the month, DD is the day, hh is the hour and mm are the minutes. Hours are stated in 24-hour time.
- The System Report is saved to the Microsoft Windows Desktop. This location is not configurable.

The System Report utility also creates a companion “System Report log” text file which allows one to view the end station environment information that was collected. This file is part of the System Report. It will be overwritten each time the utility is run with the same date.

**Note**

In the event of a failure during the creation of the System Report zip file, this file reports the failure.

The System Report log text file has these features:

- Uses a non-configurable file name: CiscoLiteSysRepLog<YYYYMMDD>.txt, where YYYY is the year, MM is the month, and DD is the day.
- The System Report log text file is saved to the Microsoft Windows Desktop. This location is not configurable.
- The System Report tool is accessible by navigating through the Windows start menu: Start > Programs > Cisco Systems, Inc. Cisco Trust Agent 802.1x Wired Client > Cisco Trust Agent 802.1x Wired Client System Report.

Creating a System Report

Once you create a System Report, it can be shared with customer support to troubleshoot problems that arise.

-
- Step 1** Run the System Report utility by navigating from the Windows Start menu > Programs > Cisco Systems, Inc. Cisco Trust Agent 802.1x Wired Client > Cisco Trust Agent 802.1x Wired Client System Report.
- Step 2** Check the **Protect sensitive data with following password** to encrypt some of the collected files, such as, your configuration files and license files during the zip consolidation and compression process.
- Step 3** Enter your password in the text box



Note

You will need to provide this password to the recipient of the System Report file.



Tip

Not all “unzip” utilities support a null password (empty password textbox) - it's recommended that you supply one.

- Step 4** Click the **Collect Data** button to initiate the information gathering - this will take approximately 1/2 a minute or so.
- Step 5** Once the report is saved, the user will see the statement, “Report generation done ... Log file has been archived” and the following buttons are enabled:

- **Copy To Clipboard** - copies the contents of companion System Report Log file to the Windows clipboard.
- **Locate Report File** - opens Windows Explorer at the desktop



CHAPTER 11

Using the Scripting Interface

This section refers to the Scripting Interface (SI) feature for Linux, Mac OS X, and Windows platforms.

The SI feature provides an interface between a third party script, which gathers posture information, and Cisco Trust Agent, which relays the posture information to the Cisco Secure Access Control Server (ACS).

The SI is provided for customers who do not want to create a posture plugin for their application but still want the application to provide posture information.

Certain filenames and file locations provided in this chapter are platform-dependent. To keep the text platform impartial, the platform-specific names were removed and replaced with a generic name. [Table 11-1](#) provides the actual platform-independent names and how they correspond to those on specific platforms.

Table 11-1 *Platform-specific Names and Corresponding Platforms*

Impartial platform file Names	Windows file names	Linux and Mac OS X file names
ctasi	ctasi.exe	ctasi
ctascriptpp	ctascriptPP.dll	ctascriptpp.so

This chapter contains the following sections:

- [Scripting Interface Overview, page 11-3](#)
 - [How the Scripting Interface Relays Posture Credentials to ACS, page 11-3](#)

- ctasi Scripting Interface File, page 11-4
 - ctascriptpp Posture Plugin File, page 11-5
 - Information Files, page 11-5
 - Posture Scripts, page 11-7
 - Posture Data Files, page 11-7
- Configuring the NAC Environment to Use Your Posture Script, page 11-13
 - Write a Posture Script, page 11-14
 - Write an Information File for the Posture Script, page 11-14
 - Register Posture Scripts, page 11-14
 - Add Script Interface Attributes to the ACS Dictionary, page 11-15
 - Configure ACS Rules to Determine Posture Based on the Script's Posture Attributes, page 11-18
- Posture Scripts Invoking ctasi, page 11-18
 - Status Change, page 11-20
- Stale Posture Data, page 11-20
 - Managing Stale Posture Database with CTA, page 11-21
 - Managing Stale Posture Database on the ACS Server, page 11-22

Scripting Interface Overview

This section describes how the NAC environment is configured to support posture scripts, how the Scripting Interface sends posture information to Cisco Secure Access Control Server (ACS), and it explains the scripting interface components in more detail.

CTA Scripting Interface components enable third party scripts to relay posture credentials, collected from the system, to CTA. The Scripting Interface functionality requires the following components:

- **ctasi** - The scripting Interface (SI) file.
- **ctascriptpp** - The posture plugin file that sends posture information to ACS.
- **Information** file (.inf) - This is a file that associates the script with ctascriptpp posture plugin.
- **Posture script** - The script that gathers posture data information and creates a posture data file.
- **Posture data files** - The plain text file that contains the posture information retrieved by a posture script.

How the Scripting Interface Relays Posture Credentials to ACS

Once the NAC environment has been configured to retrieve posture data from posture scripts, the CTA Scripting Interface can relay posture credentials to CTA and CTA relays the posture credentials to ACS. (See [“Configuring the NAC Environment to Use Your Posture Script”](#) section on page 11-13 for more information on configuring the NAC environment.)

The Scripting Interface functionality follows this workflow to gather posture data and pass it on to ACS:

To relay posture credentials, third party scripts must follow these steps:

-
- Step 1** The third party script runs and generates a posture data file. The posture data file is a plain text file. See the [“Posture Data Files” section on page 11-7](#).
 - Step 2** The third party script invokes the ctasi file. ctasi stores the information in a posture database record.
 - Step 3** During a posture request, the ACS requests the same posture information that the posture script requested. The ACS can request the same information as the script because the posture attributes of the script were added to the ACS dictionary. See, [“Configuring the NAC Environment to Use Your Posture Script” section on page 11-13](#).
 - Step 4** To fulfill the posture request, ctascriptPP Posture Plugin gathers the information stored in the posture database record and forwards it to CTA.
 - Step 5** CTA sends the requested posture credentials to ACS.
 - Step 6** The ACS checks the posture credentials against its rules and policies and determines a posture for the application and for the system. Examples of posture are Healthy, Transition, Quarantine, Infected, or Unknown.

ctasi Scripting Interface File

The ctasi executable file is a component supplied by Cisco Systems. It provides the SI external interface to posture scripts. Each posture script invokes the ctasi file. This script passes ctasi the full path to the posture data file where the collected posture data is stored. The ctasi file then stores this information in a posture database record. See [“Posture Scripts Invoking ctasi” section on page 11-18](#) for more information about how posture scripts invoke ctasi and [“Status Change” section on page 11-20](#) for information about how ctasi acts on a change in posture status.

ctascriptpp Posture Plugin File

The ctascriptpp file is a Cisco Systems supplied component. It is a Posture Plugin (PP) that interfaces with CTA. The ctascriptpp retrieves posture credentials requested by a posture script, responds to status change queries, and handles posture notifications by CTA. (For more information the ctascriptpp file see, [“CTA Scripting Posture Plugin” section on page 7-9.](#))

An information file created for use with a posture script must set the value of PluginName to ctascriptPP.dll on Windows platforms or ctascriptPP.so for non-Windows platforms. For more information about information files see, [“Information Files”](#).

Information Files

Information files (.inf files) are plain text files with an .inf file extension; they are supplied by the author of the posture script with which they are associated.

Similar to all posture plugins, the ctascriptpp file needs to be associated with at least one information file for CTA to register it as a posture plugin.

The information files associated with the posture scripts must always point to the ctascriptpp file, list VendorID=9, and list VendorIDName=Cisco Systems. The VendorID and VendorIDName identify Cisco Systems as the provider of the credential information.

Sample Information File for Windows

```
[main]
PluginName=ctascriptPP.dll
VendorID=9
VendorIDName=Cisco Systems
Styles=SupportAsync
AppList=script_z
[script_z]
AppType=61440
```

Sample Information File for Linux and Mac OS X

```
[main]
PluginName=ctascriptpp.so
VendorID=9
VendorIDName=Cisco Systems
Styles=SupportAsync
AppList=script_z
```

```
[script_z]
AppType=61440
```

Table 11-2 *information File Parameter Descriptions*

Parameter	Description	Values	Required
[main]	Defines the first section of the .inf file.	[main]	YES
PluginName	Name of the plugin that the third party script uses.	ctascriptPP.dll (Windows) ctascriptpp.so (Linux or Mac OS X)	YES
VendorID	Defines Cisco as the provider of the credential using a number.	9	YES
VendorIDName	Defines Cisco as the provider of the credentials using a name.	Cisco Systems	YES
Styles	Indicates to the CTA that Scripting Interface reports status changes asynchronously.	SupportAsync	NO
AppList	This field lists all the applications defined later in the .inf file. If there is more than one application, defined in the .inf file, list all applications in this field and separate their names with a comma. For example AppList=script_z,script_a,script_b	A string. No spaces are permitted.	YES
[AppListSection]	The name of this section corresponds to a value defined in the AppList parameter. For example [script_z] or [script_a] or [script_b].	A string. No spaces are permitted.	YES
AppType	Numbers in this range are reserved for Cisco Systems, Inc. and indicate to the ACS that a script collected the posture credentials.	61440 (0xF000) - 65535 (0xFFFF).	YES

Posture Scripts

A posture script is written for one application, operating system, or other object. The script defines which properties of that object are going to be used to determine the “posture” of the application. Examples of posture are Healthy, Transition, Quarantine, Infected, or Unknown.

The posture script must adhere to these guidelines and performs these tasks:

- Posture scripts can be written in any scripting language.
- Posture scripts define what attributes to collect.
- The script must produce a posture data file that conforms to the syntax defined in [“Posture Data Files” section on page 11-7](#). The posture data file contains the values of the attributes the posture script collected.
- The script must produce one posture data file for each AppType defined by the script’s information file.
- On Linux and Mac OS X operating systems, the posture script must write the posture data file into the /var/tmp/CiscoTrustAgent/pdata directory.
- On Windows operating systems, the posture script can write the posture data file to any directory.
- The posture script must specify that the posture data file it creates can be read by ciscotouser and that the directory in which it is stored can be read by ciscotouser.
- For each posture data file that a script produces, the script must invoke the ctasi scripting interface file separately so that the posture data file may be converted to the posture database record. See [“Posture Scripts Invoking ctasi” section on page 11-18](#) for more information about how posture scripts invoke ctasi.

Posture Data Files

A posture data file is a repository for the posture information gathered by a posture script. This posture data is stored as **plain text** and grouped into posture validation attributes.

A posture data file can only contain data for a single AppType, as defined by the information file used by the posture script. If the posture script creates a combined posture data file for several AppTypes, ctasi will discard the file as invalid and no avpdata file will be created.

A posture validation attribute is composed of a definition header and attribute-value pairs that describe the characteristics of the attribute. (See [Table 11-3 on page 11-11](#), and [Table 11-4 on page 11-12](#)).

The posture validation attribute is uniquely defined by combining the values of the vendor-id, application-id, and the attribute-id. These values are defined in the posture data file.

**Note**

The vendor-id value in the posture data file is the same as the VendorID value in the information file. The application-id value in the posture data file is the same as the AppType value in the information file.

A line in the posture data file can be no more than 511 characters long. Any character in the line after the 511th character is truncated.

The syntax of the posture data file is derived from the Posture Validation Attribute Definition file employed by the ACS CSUtil database utility. (For more information about the ACS CSUtil database utility, see the *User Guide for Cisco Secure ACS for Windows Server*.) Each individual script must produce posture data files that conforms to the format and the syntax of the posture data file (see the [Sample Posture Data File, page 11-9](#)).

Each time a script runs, prior to invoking the ctasi file, it writes the collected posture information into its corresponding posture data file. The ctasi executable file, when instructed by the posture script, reads the posture information from the posture data file.

Creating Posture Data Files on Linux and Mac OS X Operating Systems

On a Linux or Mac OS X operating system, the posture scripts must create posture data files in the following directory:

```
/var/tmp/CiscoTrustAgent/pdata
```

**Note**

This directory is created during the SI feature installation.

Creating Posture Data Files on Windows Operating Systems

On a Windows system, the posture scripts can create posture data files anywhere on the host disk.

Sample Posture Data File

The sample posture data file is a plain text file that lists the mandatory and the optional keys in each section. The following example shows a sample posture data file to illustrate the required syntax. This sample is valid for all operating systems. [Table 11-3](#) explains each entry and indicates whether it is required or optional.

```
[attr#0]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32768
attribute-name=Script-Name
attribute-profile=in
attribute-type=string
attribute-value=Script "posture_file_01"
```

```
[attr#1]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32769
attribute-name=Unsigned-Integer
attribute-profile=in
attribute-type=unsigned integer
attribute-value=31
```

```
[attr#2]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32770
attribute-name=Host-IP-Address
attribute-profile=in
attribute-type=ipaddr
attribute-value=196.68.1.99
```

```
[attr#3]
```

```
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32772
attribute-name=Date-the-Script-was-written
attribute-profile=in
attribute-type=date
attribute-value=1077771601
```

```
[attr#4]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32773
attribute-name=Script-Version
attribute-profile=in
attribute-type=version
attribute-value=1.0.3.5
```

```
[attr#5]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32774
attribute-name=octet-array-sample
attribute-profile=in
attribute-type=octet-array
attribute-value=0x11 0xbf 0x0a 0x0c
```

```
[attr#6]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32776
attribute-name=Integer-(signed)
attribute-profile=in
attribute-type=integer
attribute-value=-5
```

Table 11-3 **Posture Data File Definitions**

Keyword	Description	Values	Required
[attr#n]	The attribute identifier. Open and closed square brackets denote the key value pairs for a single attribute.	Append letters “attr” with a running index.	YES
vendor-id	A globally unique vendor identifier (used by RADIUS) assigned by the Internet Assigned Numbers Authority (IANA).	Nine corresponds to Cisco Systems.	NO
vendor-name	The vendor name represented by the vendor-id.	Cisco	NO
application-id	The posture App-Type integer.	A value from the SI values of 61440 (0xF000) to 65535 (0xFFFF). If it is not one of these values, the posture data file is discarded as invalid.	YES
application-name	The textual representation of the posture App-Type.	Use the name of the third party script that created the file.	NO
attribute-id	The attribute-code integer value.	Any value from private AVPs (32768 to 65535). Any attribute-id with a value outside of this range is discarded as illegal.	YES
attribute-name	The textual representation of the attribute-name.	Free text. No spaces allowed.	NO
attribute-profile	Consistent with the configuration file format used by ACS.	Always ignored by the ctasi file.	NO

Table 11-3 *Posture Data File Definitions (continued)*

Keyword	Description	Values	Required
attribute-type	The data type.	string integer unsigned integer ipaddr date version octet-array	YES
attribute-value	The attribute value.	The correct syntax for each attribute datatype is described in Table 11-4 .	YES

[Table 11-4](#), [Syntax for Attribute Datatype Values](#) is shown below:

Table 11-4 *Syntax for Attribute Datatype Values*

Data Type	Description	Example
octet-array	A space separated stream of strings representing the values of each octet in the array in hexadecimal format.	Octet array {10,32,17,1} will be represented by 0x0A 0x20 0x11 0x01.
integer	A signed value representing 32-bit signed integer value. Valid Range: [-2147483647 ... 2147483646]	Value of -5 will be represented by -5. Value of 10 can be represented by 10 or +10.
unsigned integer	An unsigned value representing the 32-bit unsigned value.	Value of 31 will be represented by a value 31.
string	A free text string.	Script "posture_file_01"

Table 11-4, [Syntax for Attribute Datatype Values](#) is shown below:

Table 11-4 ***Syntax for Attribute Datatype Values***

Data Type	Description	Example
ipaddr	The conventional decimal representation of an IP address version 4. Each one of the 4 octets of the IP address is represented by the decimal value of the octet, and they are separated by a period.	196.68.1.99
date	32-bit unsigned value representing the number of seconds elapsed since midnight (00:00:00), January 1, 1970, coordinated universal time (UTC) represented by decimal digits. No negative values allowed.	0 corresponding to Midnight (00:00:00), January 1, 1970.
version	A string representation of a version. Ideally, this string conforms with the NAC concept of version, which is comprised of 4 integers separated by periods. major.minor.revision.build	2.1.0

Configuring the NAC Environment to Use Your Posture Script

These are the tasks you need to perform in order for your posture script to return posture information to the ACS. These tasks assume that the NAC environment itself is already configured and that the endpoint can return posture based only on the CTA Posture Plugin and the Host Posture Plugin.

-
- Step 1** [Write a Posture Script](#)
 - Step 2** [Write an Information File for the Posture Script](#)
 - Step 3** [Register Posture Scripts](#)
 - Step 4** [Add Script Interface Attributes to the ACS Dictionary](#)

Step 5 [Configure ACS Rules to Determine Posture Based on the Script's Posture Attributes](#)

Write a Posture Script

Write a script to collect posture information from an application. The script identifies the application's attributes which will determine the application's "posture," the script outputs the posture information in the standard format of the posture data file, and the script invokes CTA's ctasi executable file. The script can be written in any scripting language. See ["Posture Scripts" section on page 11-7](#) to learn about the requirements for writing posture scripts.

Write an Information File for the Posture Script

Write an information file that associates the posture script with the ctascriptpp posture plugin. See the ["Information Files" section on page 11-5](#) for a description of this file.

Register Posture Scripts

Posture scripts must be registered with CTA before they can communicate with this application. To register scripts, place the information file that corresponds to the posture script in the appropriate Linux, Mac OS X, or Windows directory before the script runs. See the ["Information Files" section on page 11-5](#) for more information about the information file.

To register scripts on Linux or Mac OS X systems, copy the information file to the following directory:

```
/opt/PostureAgent/Plugins/install
```

To register scripts on Windows platforms, copy the information files to the following directory:

```
%COMMONPROGRAMFILES%\PostureAgent\Plugins\Install
```


On Windows operating systems, %COMMONPROGRAMFILES% is the profile location for any logged on user. For example, on Windows XP, %COMMONPROGRAMFILES% is assigned the following default directory:

\Program Files\Common Files.

For all operating systems, ensure that **ciscotouser** has permission to read the information file.

**Note**

Choose unique filenames to prevent them from being overwritten by another script's registration process.

Add Script Interface Attributes to the ACS Dictionary

Add the application's posture validation attributes, identified in the script, to the ACS attribute dictionary. ACS will then be able to recognize them when their values are reported by CTA.

The posture data collected by a posture script is identified by the vendor-id, application-id of the application, and attribute-id of the posture attribute. The values of application-id and attribute-id are not assigned to a company or an application; they are chosen from a range of valid values by the author of the posture script.

The ACS dictionary does not contain all combinations of application-id and attribute-id values by default. If your script uses these values, you must add these values to the ACS dictionary.

For a more thorough description of how the CSUtil.exe utility is used in this process, see *Appendix D: CSUtil Database Utility*, in the *User Guide for Cisco Secure Access Control Server*.

To add the application-id and attribute-id values your script uses to the ACS dictionary, perform the tasks in these sections:

-
- Step 1** [Create a Posture-Validation Attribute Definition File](#)
 - Step 2** [Add the Attributes to the ACS Dictionary](#)
 - Step 3** [Verify the Contents of the ACS Dictionary](#)
 - Step 4** [Restart ACS Services](#)

Create a Posture-Validation Attribute Definition File

A posture-validation attribute definition file is a text file that contains one or more posture-validation attribute definitions. Each definition comprises a definition header and several values.



Tip

Use a semicolon (;) to identify lines that are comments.

The only difference between the posture-validation attribute definition file and the posture data file is that the posture data file uses one additional attribute-value pair, `attribute-value=`, to the description of each attribute. See [“Sample Posture Data File” section on page 11-9](#) for an example of this attribute-value pair file and a descriptions of the attribute.

```
[attr#0]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32768
attribute-name=Script-Name
attribute-profile=in
attribute-type=string

[attr#1]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=32769
attribute-name=Unsigned-Integer
attribute-profile=in
attribute-type=unsigned integer

[attr#2]
vendor-id=9
vendor-name=Cisco Systems
application-id=61440
application-name=Script-001
attribute-id=32770
attribute-name=Host-IP-Address
attribute-profile=in
attribute-type=ipaddr
```

Add the Attributes to the ACS Dictionary

The **-addAVP** option imports posture-validation attribute definitions into ACS from the posture-validation attribute definition file described in, [Create a Posture-Validation Attribute Definition File](#), page 11-16.

Before You Begin

Because completing this procedure requires restarting the **CSAuth** service, which temporarily suspends authentication services, consider performing this procedure when demand for ACS services is low. Use the steps in [“Verify the Contents of the ACS Dictionary” section on page 11-17](#) to create a backup of posture-validation attribute definitions. You can also use the exported attribute definition file to double-check the vendor ID, application ID, and attribute ID of current posture-validation attributes.

Add attributes to the ACS dictionary using the CSUtil utility. This utility is provided with your ACS.

Step 1 On the ACS server, open a command prompt window.

Step 2 Connect to the directory \Program Files\CiscoSecure ACS V4.0\bin.

Step 3 At the prompt type: `CSUtil.exe -addAVP filename`

In the -addAVP command above filename indicates the posture-validation attribute definition file you created in the [“Create a Posture-Validation Attribute Definition File” section on page 11-16](#). If the posture-validation attribute definition file is not put in the same directory as the CSUtil.exe application, you must specify the full path to the file.

Verify the Contents of the ACS Dictionary

To verify that your attributes were added to the ACS dictionary, follow this procedure:

Step 1 On the ACS server, open a command prompt window.

Step 2 Connect to the directory \Program Files\CiscoSecure ACS V4.0\bin.

Step 3 At the prompt, type the following:

`CSUtil.exe -dumpAVP avp_new_dump.txt`

In the command above the `avp_new_dump.txt` represents the name you choose for the dump file. The command creates the file you name and saves it in the `\Program Files\CiscoSecure ACS V4.0\bin` directory.

Restart ACS Services

Once you have confirmed that the attributes you intended to add to the ACS dictionary were properly added you need to restart these ACS services for the changes to take affect:

- `csauth`
- `cslog`
- `csadmin`

You can restart these services using the Windows Services window.

Configure ACS Rules to Determine Posture Based on the Script's Posture Attributes

On the ACS, configure the rules and policies that determine the application's posture. These rules and policies are based on the posture attributes you added to the ACS attribute dictionary in [“Add Script Interface Attributes to the ACS Dictionary” section on page 11-15](#). For example, a rule might determine that the application's posture is “Healthy” if the application's version is equal to or greater than “2.0.0.0.” See the *User Guide for Cisco Secure Access Control Server* documentation for more information about Posture Validation rules and Network Access Profiles.

At this point a posture script will be able to relay posture credentials to ACS.

Posture Scripts Invoking ctasi

After the posture script has run and collected the relevant posture information, it invokes the `ctasi` file and provides one of two input parameters.

On Windows platforms, invoke `ctasi` with the command below:

```
ctasi.exe <dir_path>posture_file n
```

ctasi is located in the following Windows directory:
\\Program Files\\Common Files\\PostureAgent

On Linux and Mac OS X platforms, invoke ctasi with the command below:
`ctasi <dir_path>posture_file n`

For non-windows operating systems, ctasi is located in this directory:
/opt/PostureAgent/

**Note**

In either case of the posture script invoking ctasi, there can be no spaces in the directory path. If there are spaces in the name of the directory path, enclose the entire path and file name in quotation marks. For example:

```
ctasi.exe "c:\\Posture Data Files\\posture_file.txt" 1
```

The options for the commands that invoke ctasi have the same function in all operating system environments.

The first command option, <dir_path>posture_file, is required. This is the full path where the corresponding file with the posture data information was saved. On Windows platforms this path may be to any location. On Linux and Mac OS X platforms, this path leads to this directory:

```
/var/tmp/CiscoTrustAgent/pdata
```

The second command option, n, is optional. The second option is a number ranging from 1 to 2. This option allows the third party to do the following:

- When the script passes the value n=1, it instructs ctasi that the script has already detected a status change and CTA should accept this status change irrespective of the posture data file contents.
- When the script passes the value n=2, it instructs ctasi that the script determined that there was NO status change and CTA should accept this fact irrespective of the posture data file contents.

If the second optional parameter is omitted, CTA determines if there was a status change or not. CTA makes this decision by comparing the most recent posture reported by the script with the previous posture reported by the script for the same Application Type.

**Note**

There is one exception where the ctasi ignores the no status change instruction by a posture script. This occurs when the ctasi finds no corresponding pre-existing posture database record on the system. The rationale is because no pre-existing posture database record exists on the system, posture for the Application Type may not have been collected, or it was collected so long ago that it was removed as stale.

Status Change

If a Status Change has been relayed to or detected by the ctasi file, then ctasi file asynchronously notifies CTA about this status change because the SupportAsync notification style is always performed by the Scripting Interface.

However, ctasi can only notify CTA and not the NAD. It is the CTA that has the responsibility to notify the NAD about this status change.

In the layer 2 implementation of NAC, this means that once ctasi reports the status change to CTA, CTA will report the status change to the NAD and a new revalidation will be initiated immediately because the underlying protocol allows the CTA to asynchronously notify the NAD about the Status Change.

In a layer 3 implementation of NAC, the underlying protocol does not allow CTA to asynchronously relay this or any other Status Change to the NAD. CTA has to wait for the Status Change Query request from the NAD in order to report it. Therefore, a new revalidation will not be attempted until the Status Change Query comes from the NAD.

Stale Posture Data

Because the ctasi file may not have been invoked for an extended period of time, you may want to invalidate your posture credentials. For example, if a posture script has been deleted or blocked it can not update the posture data file. As a result, the last collected posture information is preserved in the posture data file and used whether it is current or out of date.

You can use the Write-TimeStamp Attribute (attribute-id=15) type to minimize problems occurring from stale posture data.

**Note**

attribute-id 15 is defined to be a data type date. It contains the time, in UTC format, when the posture database record was last updated by the ctasi file. This AVP is paramount to Stale Posture management.

Each entry in the posture database that corresponds to a unique application-id is associated with a Write-TimeStamp attribute. The attribute value is set by the ctasi file. This file updates the attribute value with the system UTC clock value each time it updates the posture database record that correlates to the particular application-id.

To eliminate stale posture data, see the [Managing Stale Posture Database with CTA, page 11-21](#), and the [Managing Stale Posture Database on the ACS Server, page 11-22](#).

Managing Stale Posture Database with CTA

You can add a configurable time value, `delta_stale`, that is set in the [Scripting Interface] section of the `ctad.ini` file to help manage stale posture data, as shown in example 11-1.

Example 11-1 Sample `ctad.ini` File

```
; ctad.ini sample

[Scripting_Interface]
delta_stale=20
```

This is a universal value that is defined in minutes. It corresponds to all posture database records that are related to the SI. If this entry is missing from the `ctad.ini` file, a default value of $(60 * 24 * 30 = 43200)$ one full, 30-day month is used by default. Immediately before the `ctascriptpp` file opens the relevant posture data record, it uses this value and the current time from the operating system to decide whether the posture data record is stale.

**Note**

Network Administrators should configure the `delta_stale` value to be the largest possible value desired among all application-id's related to the SI.

Based on the fact that Write-TimeStamp (attribute-id=15) denotes the time that the record was last updated, the record is considered stale if it was last updated more than delta_stale minutes before the current time.

If the posture data record is deemed stale, it is deleted. Subsequently, the ctascriptpp file does not return any posture information to the posture request. This feature is beneficial because after the stale time value delta is exceeded, no stale posture data is transmitted from the network client to the ACS. However, it is limited because only one universal delta value can be configured that corresponds to all posture data records for all application-id's related to the SI.

Managing Stale Posture Database on the ACS Server

The ACS can be configured to detect stale posture data per application-id.

Configuring rules on the ACS, per application-id, involving the Write-TimeStamp (attribute-id=15) is an approach that works in coordination with, and in addition to the [Managing Stale Posture Database with CTA](#) approach.

Using the [Managing Stale Posture Database on the ACS Server](#) approach a rule (most likely with different stale threshold values) can be set on the ACS per application-id. Using the approach described in the [Managing Stale Posture Database with CTA](#) section, only one threshold is set for all application-id's within the SI range.

To create a rule with the Write-TimeStamp attribute value on the ACS, add the Write-TimeStamp attribute to the ACS dictionary as you did in the [“Add Script Interface Attributes to the ACS Dictionary”](#) section on page 11-15. This is done in the same way that you would add the SI-specific attributes. Use the CSUtil utility with the appropriate input file as shown below. (See *Appendix D: CSUtil Database Utility*, in the *User Guide for Cisco Secure Access Control Server* for more information about the CSUtil utility):

```
CSUtil -addAVP write_time_stamp.ini
```

See the [Sample write_time_stamp.ini](#) example for a sample write_time_stamp.ini file. This sample file is consistent with other examples in this chapter in that the application-id and the application-name are the same.

To add the Write-TimeStamp for your application-id, copy the file and replace the application-id and the application-name values with the application-id and the application-name values that correspond to your script.

Example 11-2 Sample write_time_stamp.ini

```
[attr#0]
vendor-id=9
vendor-name=Cisco
application-id=61440
application-name=Script-001
attribute-id=15
attribute-name=WriteTimeStamp
attribute-profile=in
attribute-type=date
```

With the Write-TimeStamp attribute added to the ACS dictionary, you can configure rules that apply different access policies for the host based on the age of the posture data. This can be determined from the Write-TimeStamp and the current time on the ACS.



APPENDIX **A**

ctastat Diagnostic Tool

CTA provides a utility for administrators to retrieve diagnostic information from CTA. This information includes a snapshot of the communication taking place between ACS and CTA. You can view posture information using this utility, including the last time a posture check was performed.

If you are troubleshooting an issue on a system, Cisco support (TAC) may ask you to run `ctastat` to retrieve information. You must be working locally on the system in order to run `ctastat`.

This appendix contains these sections:

- [Running the ctastat Utility, page A-2](#)
 - [Running ctastat on a Linux Operating System, page A-2](#)
 - [Running ctastat on a Mac OS X Operating System, page A-2](#)
 - [Running ctastat on a Windows Operating System, page A-2](#)
- [ctastat Utility Output, page A-3](#)
 - [General CTA Information, page A-3](#)
 - [Session Information, page A-3](#)
 - [Plugins Information, page A-4](#)
 - [ctastat Utility Sample Output, page A-4](#)

Running the ctastat Utility

The ctastat utility runs using a command line interface (CLI). The utility is available for both Windows and Linux operating systems.

Running ctastat on a Linux Operating System

-
- Step 1** Open a terminal window.
 - Step 2** Change the directory to the /opt/CiscoTrustAgent/sbin directory.
 - Step 3** At the prompt type **./ctastat** and press <Enter>. ctastat displays its output in the terminal window.

Running ctastat on a Mac OS X Operating System

-
- Step 1** Open a terminal window.
 - Step 2** Change the directory to the /opt/CiscoTrustAgent/sbin directory.
 - Step 3** At the prompt type **./ctastat** and press <Enter>. ctastat displays its output in the terminal window.

Running ctastat on a Windows Operating System

-
- Step 1** Open a command prompt window.
 - Step 2** Change the directory to the Program Files\CiscoSystems\CiscoTrustAgent directory.
 - Step 3** At the prompt type, **ctastat.exe** and press <Enter>. ctastat displays its output in the terminal window.

ctastat Utility Output

The output from the ctastat command provides general information about CTA, session information which describes the communication between CTA and ACS, and plugin information which summarizes the status of the posture plugins running on the system.

For an example of a ctastat output see, [Example A-1 on page A-4](#) and [Example A-2 on page A-5](#).

General CTA Information

The general CTA information provided in ctastat output is the local time the statistics were collected and the CTA version running on the system.

[Example A-1 on page A-4](#) shows that the local time ctastat run was Friday, September 16, 2005 at 15:49:06. The CTA version is 2.0.0.26.

Session Information

The session information describes the communication between CTA and ACS. [Table A-1](#) describes the fields in the Session Information area of the output.

Table A-1 *Session Information fields from ctastat output*

Field name	Description
Session Number (Hex)	Session identification number
Session Type	Indicates communication using 802.1x or EAP over UDP (EoU) protocol.
IP Address	IP address and port id of Router/Switch when using EAP over UDP protocol.
Local MAC Address	MAC addresses of local network card when using 802.1x protocol.
Remote MAC Address:	MAC addresses of Router (or Switch) when using 802.1x protocol.
System Posture Token Value	Last reported posture token of overall system.

Table A-1 *Session Information fields from ctastat output (continued)*

Field name	Description
Received on	Time last system posture token was received.
Total Postures Received	Number of posture requests received.
Last SQ Response	Value of last status query response.
Plugin Vendor/Application:	Identifying number of plugin vendor and application. Value correlates with the information in the Plug-ins section of output.
Application Posture Token Value	Posture of the application
Received	The time the application posture token was received.
Posture Request last received	The last time the application posture credentials were requested.
Length of last response to posture request	Length of response to posture credential request measured in bytes.
Sent	The time the posture credentials were received.

Plugins Information

In the Plugins section of the ctastat output, the product vendor and application ID are listed. These numbers correlate with the information in the PluginVendor/Application field in the Session Information output.

ctastat Utility Sample Output

Example A-1 ctastat output for 802.1x connection between CTA and Cisco Secure ACS

CTA Statistics Reporting Tool

```
Cisco Trust Agent Statistics
Current Time: Fri Sep 16 15:49:06 2005
CTA Version: 2.0.0.26
```

Session Information

```
Session Number (Hex): 02000000
Session Type: 802.1X
    Local MAC Address: 0050DA2C7EBD
    Remote MAC Address: 00115DBE2BFF
System Posture Token Value: Healthy
    Received on: Fri Sep 16 15:48:14 2005
    Total Postures Received: 2
Plugin Vendor/Application: 9/1
    Application Posture Token Value: Healthy
        Received: Fri Sep 16 15:48:14 2005
    Posture Request last received: Fri Sep 16 15:48:14 2005
    Length of last response to Posture Req: 20
    Sent: Fri Sep 16 15:48:14 2005
```

Plug-ins:

```
Vendor: Cisco Systems
    Application ID: 1
        Status: Operational
    Application ID: 2
        Status: Operational
```

Example A-2 ctastat output for EAP over UDP connection between CTA and Cisco Secure ACS

CTA Statistics Reporting Tool

```
Cisco Trust Agent Statistics
Current Time: Tue Sep 27 19:11:18 2005
CTA Version: 2.0.0.26
```

Session Information

```
Session Number (Hex): 01000000
Session Type: EOU
    IP Address: 8.8.0.1:21862
System Posture Token Value: Healthy
    Received on: Mon Sep 26 11:42:14 2005
    Total Postures Received: 12
Last SQ Response was "No Status Change"
Plugin Vendor/Application: 9/1
    Application Posture Token Value: Healthy
        Received: Mon Sep 26 11:42:14 2005
    Posture Request last received: Mon Sep 26 11:42:14 2005
    Length of last response to Posture Req: 49
    Sent: Mon Sep 26 11:42:14 2005
Plugin Vendor/Application: 9/2
    Posture Request last received: Mon Sep 26 11:42:14 2005
```

■ ctastat Utility Output

Length of last response to Posture Req: 39
Sent: Mon Sep 26 11:42:14 2005

Plug-ins:

Vendor: Cisco Systems
Application ID: 1
Status: Operational
Application ID: 2
Status: Operational



APPENDIX **B**

Alternate Methods of Installing CTA

Cisco Trust Agent (CTA) can be installed on its own or it can be bundled with other Cisco or third party products. This appendix describes installing CTA using CSA MC 5.2.

Installing CTA 2.1 Using CSA MC 5.2

Your enterprise can use Cisco Security Agent (CSA) 5.2 to distribute and install CTA 2.1.



Note

CSA MC 5.2 can only distribute CTA 2.1 on Windows operating systems. See [“System Requirements for Installation” section on page 4-2](#) for the versions of Windows operating systems on which CTA 2.1 runs.

To distribute and install CTA by using CSA, follow these procedures:

- Step 1** Perform the “Copying Cisco Trust Agent Installer Files” procedure in the *Installing Management Center for Cisco Security Agents* guide.
- Step 2** Perform the “Creating CSA Agent Kits” procedure in the *Using Management Center for Cisco Security Agents Guide*.



Note

There are no installation files bundled by default with CSA distributions.



APPENDIX **C**

Open Source License Acknowledgement

Cisco Trust Agent uses third-party, open-source software. The following acknowledgements pertain to this software license.

OpenSSL/Open SSL Project

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

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

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

License Issues

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

OpenSSL License:

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

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

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

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

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

Original SSLeay License:

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

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

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

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

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

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

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

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

The word ‘cryptographic’ can be left out if the routines from the library being used are not cryptography-related.

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

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

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

Info-ZIP

This is version 2004-May-22 of the Info-ZIP copyright and license. The definitive version of this document should be available at <ftp://ftp.info-zip.org/pub/infozip/license.html> indefinitely.

Copyright (c) 1990-2004 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.
2. Redistributions in binary form (compiled executables) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled.
3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).
4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.



INDEX

Numerics

802.1x Wired Client [4-4](#), [9-1](#)

- [administrative version](#) [9-4](#)
- [administrator's main window](#) [9-8](#)
- [authentication methods](#) [9-16](#)
- [authentication policies](#) [9-24](#)
- [authentication policy summary](#) [9-12](#)
- [authentication retries](#) [9-34](#)
- [automatically establish machine connection](#) [9-27](#)
- [changing authentication profile](#) [9-44](#)
- [changing deployment packages](#) [9-44](#)
- [client version](#) [9-4](#)
- [connection status description](#) [9-8](#)
- [connection status details](#) [9-10](#)
- [connection status window](#) [9-14](#)
- [creating authentication policies](#) [9-26](#)
- [creating authentication profiles](#) [9-35](#)
- [creating machine and user authentication deployment package](#) [9-38](#)
- [creating machine only authentication deployment package](#) [9-40](#)
- [creating user authentication deployment package](#) [9-36](#)

- [credential revalidation](#) [9-23](#)
- [credentials expiring](#) [9-23](#)
- [deploying end-user 802.1x wired clients](#) [9-35](#)
- [deploying trusted servers](#) [9-31](#)
- [description](#) [9-1](#)
- [enabling connection status](#) [9-9](#)
- [features](#) [9-3](#)
- [forcing credential revalidation](#) [9-23](#)
- [installing](#) [4-17](#)
- [installing authentication profiles](#) [9-43](#)
- [installing deployment packages](#) [9-43](#)
- [installing server certificates for machine connection](#) [9-42](#)
- [installing server certificates for user connection](#) [9-43](#)
- [launching client](#) [9-11](#)
- [logging](#) [10-1](#)
- [Machine Auth \(Boot-time\)](#) [9-28](#)
- [Machine Auth \(Logon time\)](#) [9-28](#)
- [machine authentication description](#) [9-16](#)
- [machine authentication only](#) [9-27](#)
- [machine authentication requirements](#) [9-16](#)
- [machine certificate credentials](#) [9-22](#)
- [machine connections after PAC](#) [9-19](#)

- machine connections before PAC [9-18](#)
- machine credentials [9-21](#)
- machine SID credentials [9-22](#)
- main window description [9-7](#)
- manually connecting to a port [9-12](#)
- manually disconnecting from a port [9-12](#)
- network adapter information box [9-11](#)
- network configuration summary window [9-13](#)
- networks.xml file [9-25](#), [9-26](#)
- opening client [9-11](#)
- overview [9-16](#)
- overview of EAP-FAST connections [9-17](#), [9-18](#)
- policy.xml file [9-24](#), [9-26](#)
- request password [9-27](#)
- role in NAC [9-1](#)
- send anonymous in clear [9-33](#)
- sent username in clear [9-33](#)
- station policy window [9-26](#)
- system report [10-1](#), [10-15](#), [10-16](#)
- technical log [10-1](#), [10-2](#)
- technical log content [10-5](#)
- technical log format [10-3](#)
- troubleshooting [10-1](#)
- trusted server validation [9-29](#)
- user and machine authentication description [9-16](#)
- user authentication description [9-16](#)
- user certificate credentials [9-20](#), [9-21](#)

- user connections after PAC [9-18](#)
- user connections before PAC [9-17](#)
- user credential provisioning [9-20](#)
- user credentials [9-20](#)
- user credentials area [9-27](#)
- user identity protection [9-32](#)
- user interface [9-4](#), [9-26](#)
- user password credentials [9-20](#), [9-21](#)
- use single sign-on [9-27](#)
- viewing access device status [9-13](#)
- Windows operating system support [4-3](#)
- Windows service [4-30](#)

A

ACS

- see Cisco Secure Access Control Server

- asynchronous status query [5-19](#)

- authentication profiles

- adding to custom installation [4-25](#)

- authentication servers [1-2](#)

B

- browser auto-launch feature [5-24](#)

C

- Certificate DN matching [5-25](#)

certificates

- ACS root certificate [8-3](#)
- adding to custom installation [2-6, 3-13, 4-24](#)
- certificate formats allowed [8-3](#)
- certificate utility [8-3](#)
- clearing Linux certificate store [8-9](#)
- clearing Mac OS X certificate store [8-10](#)
- converting DER to PEM format [8-14](#)
- ctaCert.exe utility [8-3](#)
- CTA supported certificates [8-3](#)
- deleting from Linux certificate store [8-8](#)
- deleting from Mac OS X certificate store [8-7](#)
- DN matching [5-25](#)
- for machine and user authentication [8-13](#)
- for machine authentication [8-10](#)
- for user authentication [8-12](#)
- importing user certificate [8-12](#)
- installing [8-3, 8-4](#)
- installing on Linux [8-4](#)
- installing on Mac OS X [8-4](#)
- installing on Windows [8-5](#)
- listing certificates in Linux certificate store [8-6](#)
- listing certificates in store [8-6](#)
- obtaining from Cisco Secure ACS [8-3](#)
- updating [8-3, 8-4](#)
- updating on Linux [8-4](#)
- updating on Mac OS X [8-4](#)
- updating on Windows [8-5](#)

- use with Cisco Trust Agent [8-1](#)

certificate utility

- about [8-3](#)
- adding certificate to Windows store [8-5](#)
- command parameters in Windows [8-5](#)
- deleting certificates in Mac OS X store [8-9](#)
- Linux operating system
 - clearing all certificates from store [8-9](#)
 - deleting certificates from store [8-8](#)
 - listing certificates in store [8-6](#)
- listing certificates in store [8-6](#)
- Mac OS X
 - clearing all certificates from store [8-10](#)
 - listing certificates in store [8-7](#)
- using on Linux [8-6](#)
- using on Windows [8-5](#)

Cisco Secure Access Control Server (ACS) [xi, 1-2](#)

- certificates [8-1](#)
- managing stale posture data [11-22](#)
- role in NAC [1-2](#)
- role in posture validation [1-2 to 1-4](#)
- write_time_stamp.ini file [11-23](#)

Cisco Security Agent (CSA) [1-2](#)

- role in NAC [1-2](#)
- used to install CTA [B-1](#)

ciscotouser

- permissions for posture scripts [11-15](#)

Cisco Trust Agent (CSA)

- use of certificates [8-1](#)

Cisco Trust Agent (CTA) [xi](#)802.1x Wired Client [9-1](#)additional features [4-4](#)alternate methods of installing [B-1](#)configuring [5-1](#)deployment options [1-5](#)installing on Linux operating systems [2-1](#)installing on Mac OS X [3-1](#)installing on Windows operating systems [4-1](#)installing using Cisco Security Agent [B-1](#)

Linux daemons

ctad [2-8](#)ctaeoud [2-9](#)ctalogd [2-9](#)ctapsd [2-9](#)logging [6-1](#)

Mac OS X daemons

ctad [3-16](#)ctaeoud [3-16](#)ctalogd [3-16](#)ctapsd [3-16](#)open-source license acknowledgement [C-1](#)overview [1-1](#)purpose [xi](#)role in NAC [1-1](#)role in posture validation [1-2 to 1-4](#)scripting interface [11-1](#)statistics utility [A-1](#)supplicant [9-1](#)use of posture plugins [7-1](#)

Windows daemons

Cisco Posture Server Daemon [4-30](#)Cisco Systems Inc. CTA Posture State Daemon [4-30](#)Cisco Trust Agent EoU Daemon [4-30](#)Cisco Trust Agent Logger Daemon [4-30](#)Clickable URL feature [5-23](#)clogcli utility [6-4](#)clearing current log files [6-5](#)collecting log files [6-10](#)commands [6-5](#)disabling logging [6-5](#)enabling logging [6-6](#)location on Linux operating system [6-4](#)location on Mac OS X [6-4](#)location on Windows operating system [6-4](#)logging level explanation [6-11](#)running [6-5](#)setting log file location [6-7](#)setting logging level [6-8](#)zipit command [6-10](#)

configuration files

ctad.ini file [5-2](#)ctalogd.ini file [6-13](#)configuring Cisco Trust Agent [5-1, 5-16, 5-17, 5-18, 5-19](#)behavior of posture notifications [5-7](#)blocking or non-blocking plugins [5-5](#)

- clearing or saving old posture notifications [5-10](#)
- configuring status query timer [5-6](#)
- ctad.ini file [5-2](#)
- ctalogd.ini file [6-4](#)
- defining communication port for EAPoUDP [5-6](#)
- displaying posture messages in GUI [5-9](#)
- display time of pop-up notifications [5-9](#)
- distinguished name matching [5-11](#)
- distinguished name matching parameters [5-11](#)
- EAP over UDP communication [5-12](#)
- EAP over UDP session idle timeout [5-7](#)
- editing the ctad.ini file [5-3](#)
- enabling or disabling pop-up posture messages [5-8](#)
- font used to display messages in terminal [5-10](#)
- for Windows XP SP-2 or SP-3 firewall [5-7](#)
- logging [6-4](#)
- maximum EAP over UDP sessions [5-7](#)
- notification pop-up modality [5-8](#)
- parameter descriptions [5-4](#)
- pop-up notifications received before logon [5-9](#)
- posture plugin interaction with CTA [5-13](#)
- posture plugins [5-13](#)
- posture pop-up notifications [5-20](#), [5-23](#)
- posture pop-up notifications on Linux [5-21](#)
- posture pop-up notifications on Mac OS X [5-22](#)
- posture pop-up notifications on Windows [5-20](#)
- query plugin for posture status [5-19](#)
- receive posture message after obtaining IP address [5-8](#)
- saving posture notifications [5-9](#)
- scripting interface parameter [5-11](#)
- setting application-specific posture message [5-6](#)
- setting browser path on Linux [5-10](#)
- setting default posture message size [5-6](#)
- time before posture database record is outdated [5-11](#)
- timeout for non-blocking plugins [5-5](#)
- user notifications [5-20](#), [5-23](#)
- user notifications on Linux [5-21](#)
- user notifications on Mac OS X [5-22](#)
- user notifications on Windows [5-20](#)
- Windows
 - SysModal parameter [5-21](#)
- CSA
 - see Cisco Security Agent
- ctacert utility
 - See certificate utility
- ctad.ini file
 - [EAPoUDP] section description [5-6](#)
 - [Scripting_Interface] section [5-11](#)
 - [ServerCertDNVerification] section [5-11](#), [5-25](#)

[UserNotifies] section [5-7](#)

about [5-2](#)

adding to custom installation [2-7, 3-13, 4-25](#)

configuring Validation-Flag TLV [5-4](#)

ctad-temp.ini [5-2](#)

delta_stale [11-21](#)

editing [5-3](#)

location [5-2](#)

parameter descriptions [5-4](#)

 BootTimeUDPExemptions [5-7](#)

 BrowserPath [5-10](#)

 ClearOldNotification [5-10](#)

 delta_stale [5-11](#)

 DisplayType [5-9](#)

 EnableLogonNotifies [5-9](#)

 EnableNotifies [5-8](#)

 EnableVFT [5-4](#)

 LocalPort [5-6](#)

 LogonMsgTimeout [5-9](#)

 MaxSessions [5-7](#)

 MsgTimeout [5-9](#)

 PPInterfaceType [5-5](#)

 PPMsgSize [5-6](#)

 PPWaitTimeout [5-5](#)

 Rule X [5-11](#)

 SessionIdleTimeout [5-7](#)

 SQTimer [5-6](#)

 SysModal [5-8](#)

 TermFont [5-10](#)

TotalRules [5-11](#)

userActionDelayTimeout [5-8](#)

ctad-temp.ini file

 See ctad.ini file

ctalogd.ini file

 [LogLevel] section [5-24](#)

 adding to custom installation [2-7, 4-25](#)

 including in custom installation [3-13](#)

ctalogd-temp.ini file [6-13](#)

 example of [6-13](#)

 location on Linux operating system [6-13](#)

 location on Mac OS X [6-13](#)

 location on Windows operating system [6-13](#)

 See also ctalogd.ini file

CTA posture plugin [7-2, 7-5](#)

 application posture-token attribute [7-6](#)

 attributes of [7-6](#)

 for Linux operating system [7-6](#)

 for Mac OS X [7-6](#)

 for Windows operating system [7-6](#)

 kernel-version attribute [7-6](#)

 machine posture state attribute [7-6, 7-8](#)

 operating system attribute [7-7](#)

 operating system version attribute [7-7](#)

 operating system release attribute [7-6](#)

 posture agent name attribute [7-7](#)

 posture agent version attribute [7-7](#)

 posture message attribute [7-8](#)

 posture token attribute [7-8](#)

- ctascriptPP.dll [11-1](#)
 - description of [11-5](#)
 - ctascriptpp.so [11-1](#)
 - description of [11-5](#)
 - ctasi [11-1](#)
 - invoked by posture script [11-18](#)
 - location of executable on Linux [11-19](#)
 - location of executable on Mac OS X [11-19](#)
 - ctasi.exe [11-1](#)
 - location on Windows [11-19](#)
 - ctastat utility
 - identifying CTA information [A-3](#)
 - identifying session information [A-3](#)
 - output [A-3](#)
 - overview [A-1](#)
 - running on Linux [A-2](#)
 - running on Mac OS X [A-2](#)
 - running on Windows [A-2](#)
 - sample output [A-4](#)
 - customized installation
 - deployment considerations [1-5](#)
 - Linux operating system [2-5](#)
 - benefits of [2-6](#)
 - including certificates [2-6](#)
 - including ctad.ini file [2-7](#)
 - including ctalogd.ini file [2-7](#)
 - including posture plugins [2-7](#)
 - Mac OS X
 - benefits of [3-12](#)
 - ctad.ini file [3-13](#)
 - including certificates [3-13](#)
 - including ctad.ini file [3-13](#)
 - including ctalogd.ini file [3-13](#)
 - including posture plugins [3-13](#)
 - Windows operating system [4-22, 4-24](#)
 - 802_1x directory [4-25](#)
 - benefits of [4-24](#)
 - including authentication profiles [4-25](#)
 - including certificates [4-24](#)
 - including ctad.ini file [4-25](#)
 - including ctalogd.ini file [4-25](#)
 - including plugins [4-24](#)
 - installation directory [4-24](#)
 - install customized package [4-25](#)
-
- ## D
- delta_stale parameter [11-21](#)
 - deploying Cisco Trust Agent [1-5, 2-3](#)
 - benefit of custom package [2-3](#)
 - initial deployment options [2-3](#)
 - distinguished name matching
 - See DN matching
 - DN matching
 - about [5-25, 8-14](#)
 - about rules [8-14](#)
 - attributes supported [5-26](#)
 - issuer attributes [5-26](#)

parameters in ctad.ini file [5-11](#)

rule length [5-25](#)

rules [5-25](#)

sub-rule operators [5-25](#)

sub-rules [5-25](#)

when occurs [8-14](#)

documentation

additional reading [xiv](#)

text conventions [xiii](#)

E

EAP-FAST connections

machine credentials context [9-18](#)

overview [9-17, 9-18](#)

user logon context [9-17](#)

EAP over UDP

configuring communication [5-12](#)

configuring communication port [5-6](#)

H

host posture plugin [7-2](#)

attributes of [7-3](#)

Linux package attribute [7-3](#)

Linux package information [7-4](#)

location on Linux [7-2](#)

location on Mac OS X [7-2](#)

location on Windows [7-2](#)

MAC address attribute [7-3](#)

MAC address information [7-3](#)

machine name attribute [7-3](#)

Mac OS X package attribute [7-3](#)

Mac OS X package information [7-5](#)

Windows hot fix attribute [7-3](#)

Windows service pack attribute [7-3](#)

installation files

Linux operating system [2-3](#)

Mac OS X operating system [3-3](#)

Windows operating system

CTA with 802.1x Wired Client [4-5](#)

CTA without 802.1x Wired Client [4-5](#)

discontinued versions [4-5](#)

installation procedures

Linux operating system [2-4](#)

accepting EULA [2-4](#)

command line procedure [2-5](#)

customized installation [2-5](#)

extracting install file [2-4](#)

general instructions [2-4](#)

package information [2-9](#)

uninstalling CTA [2-9](#)

upgrading CTA [2-7](#)

verifying installation [2-8](#)

Mac OS X [3-3, 3-16](#)

- accepting EULA [3-4](#)
- command line procedure [3-4](#)
- extracting install file [3-4](#)
- general instructions [3-3](#)
- installation wizard [3-6](#)
- repairing [3-14](#)
- uninstalling scripting interface [3-17](#)
- upgrading [3-14](#)
- verifying installation [3-16](#)
- Windows operating system [4-5](#)
 - accepting EULA [4-6](#)
 - custom installation package [4-22](#)
 - customized installation [4-24](#)
 - extracting MSI file [4-6](#)
 - general instructions [4-6](#)
 - installation directory [4-24](#)
 - installation wizard [4-11, 4-13](#)
 - install customized package [4-25](#)
 - installing 802.1x Wired Client [4-17](#)
 - installing scripting interface [4-17](#)
 - uninstalling [4-31](#)
 - upgrading [4-27](#)
 - upgrading from CTA 1.0 [4-27](#)
 - upgrading from CTA 2.0 [4-28](#)
 - upgrading from CTA 2.0.1 [4-29](#)
 - using MSI commands [4-7](#)
 - verify CTA installation [4-30](#)

L

- licenses
 - for open-source software [C-1](#)
- log files
 - about [6-2](#)
 - collecting all log files [6-10](#)
 - creating [6-2](#)
 - format [6-3](#)
 - location on Linux operating systems [6-2](#)
 - location on Mac OS X [6-2](#)
 - location on Windows operating systems [6-2](#)
 - naming convention [6-3](#)
 - persistance [6-4](#)
 - taking up disk space [6-4](#)
- logging
 - about CTA logging [6-2](#)
 - Cisco Trust Agent [6-1](#)
 - clearing current log files [6-5](#)
 - clogcli utility [6-4, 6-5](#)
 - collecting all log files [6-10](#)
 - configuring for large deployments [6-11](#)
 - ctalogd-temp.ini file [6-13](#)
 - default setting [6-1](#)
 - disabling logging [6-5](#)
 - enabling logging [6-6](#)
 - log files [6-2](#)
 - logging level explanation [6-11](#)
 - notifications [5-24](#)

running [6-5](#)
 setting log file location [6-7](#)
 setting logging level [6-8](#)

M

machine and user authentication
 configuring by using certificates [8-13](#)
 machine authentication
 configuring by using certificates [8-10](#)
 requesting machine certificate [8-11](#)

N

NAC-L2-IP method [5-12](#)
 NAC-L3-IP method [5-12](#)
 NAD
 See network access device
 network access device (NAD) [xi, 1-2](#)
 role in NAC [1-2](#)
 role in posture validation [1-2 to 1-4](#)
 Network Admission Control (NAC) [xi](#)
 objective of [1-1](#)
 overview [1-1](#)
 network client
 definition of [1-5](#)
 network clients [1-1](#)
 notifications
 logging [5-24](#)

posture [7-1](#)

O

open source
 software licenses [C-1](#)
 open-source software
 license acknowledgement [C-1](#)

P

posture credentials [xi, 7-1](#)
 relayed by scripting interface [11-3](#)
 transfer from plugin to CTA to ACS [7-1](#)
 posture data file [11-3, 11-7](#)
 attribute definitions [11-11](#)
 creating files for Linux [11-8](#)
 creating files for Mac OS X [11-8](#)
 creating files for Windows [11-9](#)
 description of [11-7](#)
 location on Linux [11-8](#)
 location on Mac OS X [11-8](#)
 location on Windows [11-9](#)
 requirements of [11-8](#)
 sample [11-9](#)
 syntax for attribute datatype values [11-12](#)
 syntax of [11-8](#)
 posture notifications

- clearing or saving old posture notifications [5-10](#)
- configuring browser auto-launch feature [5-23](#)
- configuring clickable URL feature [5-23](#)
- configuring display time of pop-up notifications [5-9](#)
- configuring pop-up notifications received before login [5-9](#)
- configuring pop-up window [5-8](#)
- displaying posture messages in GUI [5-9](#)
- enabling or disabling pop-up message [5-8](#)
- font used to display messages in terminal [5-10](#)
- how they are sent [5-20](#)
- pop-up box modality [5-8](#)
- pop-up message parameters [5-7](#)
- saving pop-up notifications [5-9](#)
- setting browser path on Linux [5-10](#)
- posture plugins
 - adding to custom installation [2-7, 3-13, 4-24](#)
 - application posture-token attribute [7-6](#)
 - application-specific posture message size [5-17](#)
 - asynchronous status query [5-19](#)
 - configuring application-specific message size [5-6](#)
 - configuring application-specific posture message size [5-17](#)
 - configuring blocking or non-blocking interface [5-5, 5-13](#)
 - configuring default message size [5-16](#)
 - configuring default posture message size [5-6](#)
 - configuring host posture plugin message size [5-18](#)
 - configuring interaction with CTA [5-13](#)
 - configuring status query timer [5-6](#)
 - configuring Symantec posture plugin message size [5-19](#)
 - configuring timeout for non-blicking plugins [5-5](#)
 - configuring to query for status change [5-19](#)
 - CTA posture plugin [7-2, 7-5](#)
 - default message size [5-16](#)
 - definition of [7-1](#)
 - example of blocking and non-blocking interface [5-14](#)
 - host posture plugin [7-2](#)
 - host posture plugin message size [5-18](#)
 - installation process overview [7-9](#)
 - installed by default [7-2](#)
 - installing [7-9](#)
 - kernel-version attribute [7-6](#)
 - Linux host [7-2](#)
 - Linux installation directory [7-9](#)
 - Linux package attribute [7-3](#)
 - MAC address attribute [7-3](#)
 - machine name attribute [7-3](#)
 - machine posture state attribute [7-6, 7-8](#)
 - Mac OS X host [7-2](#)
 - Mac OS X installation directory [7-9](#)
 - Mac OS X package attribute [7-3](#)

- operating system attribute [7-7](#)
- operating system release attribute [7-6](#)
- operating system version number attribute [7-7](#)
- posture agent name attribute [7-7](#)
- posture agent version attribute [7-7](#)
- posture message attribute [7-8](#)
- quarantined plugin [7-10](#)
- scripting interface [11-3](#)
- scripting interface plugin [7-9](#)
- script substituting as [7-9](#)
- system posture token attribute [7-8](#)
- upgrading [7-9](#)
- Windows host [7-2](#)
- Windows hot fix attribute [7-3](#)
- Windows installation directory [7-9](#)
- Windows service pack attribute [7-3](#)
- posture scripts [11-3, 11-7](#)
 - ciscotouser [11-15](#)
 - invoking ctasi [11-18](#)
 - location for new scripts [11-14](#)
 - registering [11-14](#)
 - requirements of [11-7](#)
 - user permissions [11-15](#)
- posture token
 - definition of [7-1](#)
- posture validation [1-1](#)
 - authentication servers [1-2](#)
 - components of [1-1](#)
 - definition of [1-1](#)

- network clients [1-1](#)
- posture validation servers [1-2](#)
- process [1-2 to 1-4](#)
- posture validation servers [1-2](#)
 - role in NAC [1-2](#)

R

- repairing CTA
 - Mac OS X [3-14](#)
 - Windows [4-8](#)

S

- scripting interface [4-4](#)
 - adding attributes to ACS [11-15](#)
 - asynchronous status change notification [11-20](#)
 - creating posture data files [11-8, 11-9](#)
 - ctascript.so [11-1](#)
 - ctascriptPP.dll [11-1](#)
 - ctasi [11-1](#)
 - ctasi.exe [11-1](#)
 - executable file [11-4](#)
 - file name conventions [11-1](#)
 - information file [11-3, 11-5](#)
 - information file parameter descriptions [11-6](#)
 - installing [2-3, 3-5, 3-10, 4-17](#)
 - interaction with CTA [11-1](#)

- invoking ctasi executable [11-18](#)
- invoking on Linux [11-19](#)
- invoking on Mac OS X [11-19](#)
- invoking on Windows [11-18](#)
- making it accept the posture data file [11-19](#)
- making it ignore the posture data file [11-19](#)
- managing stale posture data [11-21](#)
- overview [11-3](#)
- posture data file [11-3, 11-7](#)
- posture data file attributes [11-11](#)
- posture plugin [7-9, 11-3](#)
- posture plugin description [11-5](#)
- posture scripts [11-3, 11-7](#)
- posture-validation attribute definition file [11-16](#)
- registering posture scripts [11-14](#)
- relaying posture credentials [11-3](#)
- reporting status change at layer 2 [11-20](#)
- reporting status change at layer 3 [11-20](#)
- role in NAC [11-1](#)
- sample posture data file [11-9](#)
- stale posture data [11-20](#)
- status change [11-20](#)
- syntax for attribute datatype values [11-12](#)
- uninstalling [3-17](#)
- scripting interface posture plugin [7-9](#)
 - description of [11-5](#)
- supplicant
 - logging [10-1](#)
 - See 802.1x Wired Client

- system requirements
 - Linux installer [2-2](#)
 - Linux operating system [2-2](#)
 - hard disk space [2-2](#)
 - listening port [2-3](#)
 - memory [2-3](#)
 - operating systems version [2-2](#)
 - processor [2-2](#)
 - Mac OS X [3-2](#)
 - hard disk space [3-2](#)
 - listening port [3-2](#)
 - memory [3-2](#)
 - operating systems version [3-2](#)
 - processor [3-2](#)
 - Windows operating system [4-2](#)
 - hard disk space [4-2](#)
 - installer [4-2](#)
 - listening port [4-2](#)
 - memory [4-2](#)
 - processor [4-2](#)
- system requirements [4-2](#)

T

- Text conventions [xiii](#)
- transport layer security (TLS) [5-25](#)

U

uninstalling CTA

- Linux operating system [2-9](#)
- Mac OS X [3-16](#)
- Windows operating system [4-31](#)

upgrading CTA

- compatibility of authentication profiles from CTA 2.0 [4-28](#)
- Linux operating system [2-7](#)
- Mac OS X [3-14](#)
- Windows operating system
 - from CTA 1.0 [4-27](#)
 - from CTA 2.0 [4-28](#)
 - from CTA 2.0.1 [4-29](#)

user authentication

- configuring by using certificates [8-12](#)
- importing user certificate [8-12](#)

user credentials

- expiring [9-23](#)
- forcing revalidation [9-23](#)

user notifications

- logging notifications [5-24](#)
- posture notifications [7-1](#)

installing [4-7](#)

installing optional features [4-9](#)

quiet mode [4-11](#)

reboot options [4-11](#)

reinstalling or repairing [4-8](#)

uninstalling [4-8](#)

Windows operating systems

- support for 802.1x Wired Client [4-3](#)
- support for CTA [4-3](#)

W

Windows MSI commands

[4-7](#)

- changing installation directory [4-10](#)