



SDG CONVERSION TOKEN SERVICE

2.3.x INSTALLATION/CONFIGURATION – EMBEDDED

sdgc.com

© 2026 SDG Corporation
All Rights Reserved

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by SDG Corporation at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of SDG. This Documentation is confidential and proprietary information of SDG and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and SDG governing your use of the SDG software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and SDG.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all SDG copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to SDG that all copies and partial copies of the Documentation have been returned to SDG or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, SDG CORPORATION PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL SDG BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF SDG IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is SDG Corporation.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) – (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2026 SDG Corporation. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contact Information:

SDG Corporation

U.S. Global Headquarters
75 North Water St. Norwalk, Connecticut, 06854
United States
Phone: +1 (203) 866-8886

SDG Global Tech Center

India (Country HQ)
A-10, Sector 2
Noida, UP
India 201301
Phone: +91 120-4014000

SDG Canada

9131 Keele Street, Suite A4
Vaughan, ON L4K 0G7
Canada

Third Party Libraries and Licensing

Library	Version
Jersey	2.35
jackson	2.11.4
googlehttp	1.38.0
jetty	9.4.44.v20210927
googleOauth	1.33.0
jsonwebtoken	0.9.1
commons-codec	1.15
slf4j	2.0.0-alpha7
springframework	5.3.20
jasyp	1.9.3
commons-io	2.11.0

Jersey – RESTFul Web Services in Java

Used under the GPL License

<https://eclipse-ee4j.github.io/jersey.github.io/license.html>

Jackson

Used under an Open Source license from Apache

<https://www.apache.org/licenses/LICENSE-2.0>

GoogleHTTP

Used under Open Source GPL Licensing

<https://opensource.google/documentation/reference/thirdparty/licenses>

Jetty

Used under the Eclipse Public License and Apache 2.0 License

<http://www.eclipse.org/jetty/licenses.php>

Google OAuth

Used under Open Source GPL Licensing

<https://opensource.google/documentation/reference/thirdparty/licenses>

JSON Web Token

Licensed under the Apache 2.0 License

<http://www.eclipse.org/jetty/licenses.php>

Commons Codec

Used under an Open Source license from Apache
<https://www.apache.org/licenses/LICENSE-2.0>

SLF4J

Used under open source GPL Licensing
<https://www.slf4j.org/license.html>

Spring Framework

Used under the Apache 2.0 Open Source License
<https://www.apache.org/licenses/LICENSE-2.0>

JASPYT

Used under the Apache 2.0 Open Source License
<http://www.jaspyt.org/license.html>

Commons-IO

Used under the Apache 2.0 Open Source License
<https://www.apache.org/licenses/LICENSE-2.0>

Table of Contents

Chapter 1: Overview.....	7
SDG Conversion Token Service (CTS) Overview.....	7
Workflow Integration with Ping Federate.....	7
Platform Support.....	8
Components of the SDG Conversion Token Service.....	8
Component Location.....	9
Which Install Path Should You Choose?.....	9
Chapter 2: SDG Conversion Token Service PingFederate Embedded Install.....	11
Download Components.....	11
Communication Port Requirements.....	11
Stage Installation to Dedicated Folder.....	12
CA SiteMinder Java SDK and Policy Server Process.....	12
Register the SDK Agent Used by the SDG Conversion Token Service.....	12
Create the SiteMinder Policy Objects for CTS.....	14
SiteMinder Policy Base Object Creation.....	14
Update the CTS Domain for Your Environment.....	16
PingFederate.....	18
Generate A Client Certificate for SDG Conversion Token Service.....	18
SDG Conversion Token Service.....	20
JAVA.....	20
Installation of the SDG Conversion Token Service WAR file.....	20
Configuration of the SDG Conversion Token Service.....	21
Copy and Update the 'authenticatedUsers.txt' file.....	21
Copy and Rename the SmHost.conf to cbSmHost.conf.....	22
Copy the CTS Integration Kit JAR File to PingFederate.....	22
Copy the CA SiteMinder Java SDK JAR Files to PingFederate.....	22
Edit the run.properties.....	22
Edit 'cts.properties' file.....	23
Restart PingFederate for SDG Conversion Token Service to Deploy.....	23
Validate the SDG Conversion Token Service.....	23
Testing the SDG Conversion Token Service.....	24
Deploying SDG Conversion Token Service to Additional Nodes.....	24
Chapter 3: Configuring Mutual Client Certificate Authentication.....	25
Client Side Certificate Generation.....	25
Generate a Client Certificate Key Pair.....	26
Export client certificate without Private Key for use on server side (.cer file).....	26
Export the Certificate for Use on the CTS Server.....	26
Convert the Certificate to pem Format.....	27
Extract the Private Key.....	27
Files Generated.....	28
Server Side Certificate Generation and Mutual SSL Configuration.....	28
Generate a Keystore Key Pair.....	28
Copy the Client Certificate to the Server.....	29
Import the Client Certificate into the keystore as a Trusted Certificate.....	29
Verify the Certificate in the Keystore.....	30
Add Client Certificate Serial Number to the authenticateusers.txt File.....	31

CHAPTER 1: OVERVIEW

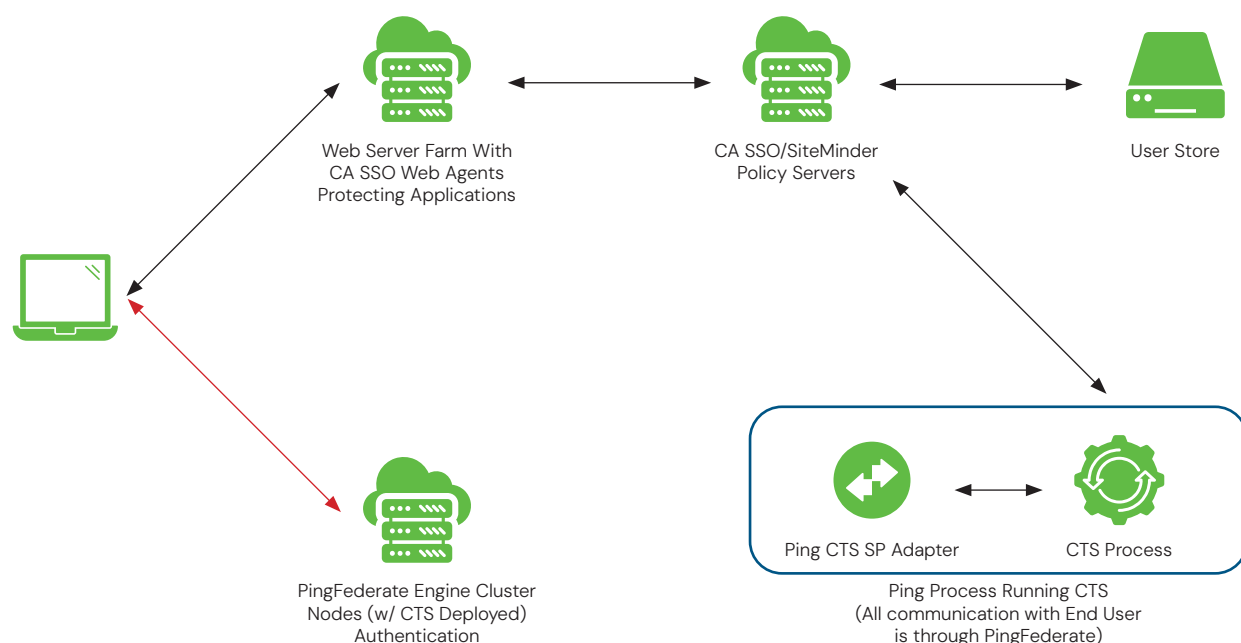
SDG Conversion Token Service (CTS) Overview

The SDG Conversion Token Service (CTS) is a Java REST Web Service that allows the exchange of tokens between different security platforms. In this installation CTS is used to exchange tokens between CA SiteMinder/SSO and PingFederate.

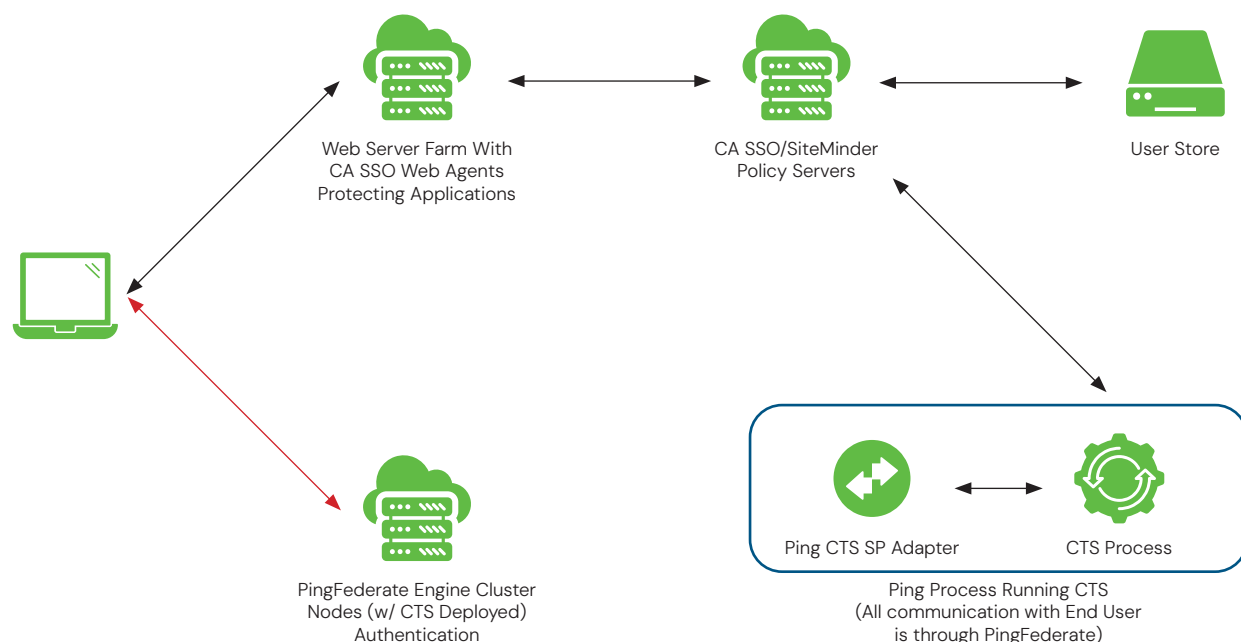
It is built with the CA SSO SDK that allows it to interact with CA SiteMinder Policy Servers. It supports a level of communication that a full featured web agent leverages to talk to multiple policy servers and supports round robin and failover communication.

Workflow Integration with Ping Federate

The SDG Conversion Token Service can be deployed in two different architectures to interoperate with PingFederate. Whether you deploy CTS to run within the PingFederate Jetty Instance or in its Standalone configuration CTS works with PingFederate's SDG Conversion Token Adapters to implement the following use cases.



Use Case 1 – User authenticates to a CA SSO/SiteMinder Resource, existing or newly created SMSESSION Tokens can then be exchanged for access to a Ping Federate protected application with no additional challenge through the use of the SDG Conversion Token Service allowing single sign on to both platforms.



Use Case 2 – User starts at PingFederate and accesses an Application Definition with an Auth Tree with the SDG SP Adapter defined that will send the user to a CA SSO/SiteMinder protected application. The Ping CTS SP Adapter provides trusted information about the user and generates an SMSESSION based on this information. The user now has valid sessions and work between both platforms.

Platform Support

PingFederate Versions	SiteMinder Versions (includes all Service Packs)	Java Versions Validated
<ul style="list-style-type: none"> PingFederate 9.x PingFederate 10.x 	<ul style="list-style-type: none"> SiteMinder 12.52 SiteMinder 12.6 SiteMinder 12.7 SiteMinder 12.8 	<ul style="list-style-type: none"> Oracle Java 1.8 Oracle Java 11* Amazon Corretto 11* OpenJDK 11*

* The SiteMinder Agent SDK is only supported on Oracle 1.8, Oracle (formerly Sun) 1.7, and IBM Java 1.7.x & 1.8.x. SDG cannot guarantee support of other versions of Java since they are unsupported by Broadcom.

Components of the SDG Conversion Token Service

The following components are required as part of the collective solution:

- SDG Conversion Token Service (SDG)
- CTS Integration Kit (Ping Identity)
- Ping Federate (Ping Identity)
- CA SiteMinder (CA/Broadcom)
- CA SiteMinder SDK JARS (Included with CTS)

SDG Conversion Token Service
CTS Integration Kit (Ping Identity)
CA SiteMinder SDK (Included with our kit)
CA SiteMinder (CA/Broadcom)



PingFederate Engine Node

SDG Conversion Token Service
CTS Integration Kit (IDP/SP Adapter JARS)
CA SiteMinder SDK (JARS)



CA SiteMinder Policy Server

CTS Domain/Policy Objects
CTS Agent Configuration Objects
CTS Host Configuration Object
CTS Trusted Host



PingFederate Admin Node

CTS Integration Kit (IDP/SP Adapter Jars)
CA SiteMinder SDK Library Files (Jars)

Component Location

Since the SDG Conversion Token Service requires the use of multiple components please refer to the following table and diagram to plan your installation accordingly.

PingFederate	SDG Conversion Token Service WAR	CA SSO SDK JAR Files	CTS Adapter
Required on PF Stand-Alone	Yes	Yes	(Yes) Configure IDP or SP Adapter as needed
Required on PF Engine	Yes	Yes	(Yes) Configure IDP or SP Adapter as needed
Required on PF Admin Only	No	No	(Yes) Configure IDP or SP Adapter as needed



PingFederate Engine Node(s) or StandAlone

- SDG Conversion Token Service WAR
- cts.properties
- authenticatedUsers.txt
- cbSmHost.conf
- smagentapi.jar
- crypto.jar
- Edit run.properties
- Edit jettyruntime.xml
- SDG Integration Kit JAR



PingFederate Administrator Console

- SDG Integration Kit JAR

Which Install Path Should You Choose?

The SDG Conversion Token Service can be deployed in one of two scenarios, depending on your requirements and infrastructure layout of CA SiteMinder and PingFederate.

If you have a single CA SiteMinder environment to support (B2E) then you should go to Chapter 2 and do the Embedded Installation of CTS in PingFederate. There is a limit that you can only run **ONE** instance of CTS within your PingFederate Installation. All the steps to setup CTS with PingFederate are in that Chapter.

Now, if your infrastructure needs and requirements are that you need to talk to multiple CA SSO instance (B2E, B2B, B2C) from a single PingFederate Infrastructure you will need to **MULTIPLE** Standalone instances of CTS. This is best explained in Chapters 3, 4 and 5. You can run multiple instances of CTS on the same server, you just need to edit the jetty-runtime.xml and change each instance to it's own port.

CHAPTER 2: SDG CTS PINGFEDERATE EMBEDDED INSTALL

Let's begin with what is needed to setup CTS with PingFederate and get it up and running.

Download Components

This document assumes you already have CA SiteMinder installed and configured as per CA/Broadcom documentation. It also assumes that PingFederate has also been installed and baseline configuration done to have it up and running ready to start. In order to complete the solution the following components need to be downloaded:

- SDG Conversion Token Service 2.2.x – <https://sdgc.com/conversion-token-service/>
- Please note a license key for CTS will be emailed to the address provided for download within 1 business day. If you have not received your license key within that time please send an email to solutions@sdgc.com
- CTS Integration Kit (latest version) – <https://www.pingidentity.com/en/resources/downloads/pingfederate.html>
- Click on the Add-ons section to see the integration kits, locate the SDG Integration Kit 2.6.2, as of 08/02/2020

Communication Port Requirements

The following table breaks down the Communication Ports the solution requires in order to fully function.

PingFederate Engine Node(s)

Port	CTS Adapter
9032 (Inbound)	PingFederate installs with a default Port of 9031 so 9032 assumes you would just make that the second port. If you wish to use another port that is fine, the requirement is the HTTP.SECONDARY PORT must be enabled and set.
44441 (Outbound)	CA SiteMinder Accounting Port
44442 (Outbound)	CA SiteMinder Authentication Port
44443 (Outbound)	CA SiteMinder Authorization Port

PingFederate Administrator Console

Port	CTS Adapter
9032 (Inbound)	PingFederate installs with a default Port of 9031 so 9032 assumes you would just make that the second port. If you wish to use another port that is fine, the requirement is the HTTP.SECONDARY PORT must be enabled and set. (<i>Whatever port you specify on the Engine Nodes has to match the Admin Node</i>)

Stage Installation to Dedicated Folder

It is recommended to build a folder where you will expand the CTS files and use that as the base folder for running all commands and configurations. It is best that this folder be located outside of the PingFederate Installation directory but on the same server where PingFederate is located. **Note: This folder will be used frequently during the installation process.**

Example:

Windows: d:\CTS_Install

Linux: /opt/CTS_Install or /apps/CTS_Install

Your CTS Location	
-------------------	--

CA SiteMinder Java SDK and Policy Server Process

As stated earlier we include all the necessary files from CA in order to do what is needed for Host Registration and the JARS needed.

Prior to completing this step please fill out the following table to have the needed pieces of information to complete the host registration:

SiteMinder Policy Server IP Address	
SiteMinder Administrator User (Or User with Agent Registration Permission)	
SiteMinder User Password	
Trusted Host Name for CTS	
SiteMinder HCO (Host Config Object)	

Register the SDK Agent Used by the SDG Conversion Token Service

Below are the steps to do the host registration procedure using the files we provided in the CTS deployment kit you should've downloaded already. The run location is the directory you noted above where you extracted all the files. *(The instructions are going to assume Linux and a base directory of opt/CTS_Install)*

1. In /opt/CTS_Install/CTS_2212_Package/ca-sdk, vi the smreghost.sh or smreghost.bat if on windows to add the required SDK Files Path:

```
#!/bin/ksh
```

```
#####  
#####  
###
```

```
## Copyright (c) 2006 CA. All rights reserved.          ##
## This software may not be duplicated, disclosed or reproduced in whole or      ##
## in part for any purpose except as authorized by the applicable license agreement, ##
## without the express written authorization of CA. All authorized reproductions  ##
## must be marked with this language.          ##
##
##
## TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS      ##
## SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING ##
## WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY,      ##
## FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT      ##
## WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS      ##
## OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS MATERIAL,      ##
## INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS      ##
## INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY      ##
## ADVISED OF SUCH LOSS OR DAMAGE.          ##
#####
#####
###

export JAVA_HOME=Your Java Home
export SM_SMREGHOST_CLASSPATH=/opt/CTS_Install/CTS_2212_Package/ca-sdk/java64
/smagentapi.jar: /opt/CTS_Install/CTS_2212_Package/ca-sdk/java64 /cryptoj.jar export
PATH=$JAVA_HOME/bin:$PATH

java -classpath "$SM_SMREGHOST_CLASSPATH" com.ca.siteminder.sdk.agentapi.SmRegHost "$@"
# The caller needs the exit status from SmRegHost exit $?
```

2. In the same folder where you edited the smreghost.bat (Windows) or smreghost.sh (Linux) run the following command respective of you OS and with the following syntax:
 - a. Windows: smreghost.bat -i ps_ip_addr -u sm_admin -p sm_admin_pw -hn trusted_host_name -hc sm_hco
 - b. Linux: ./smregshost.sh -i ps_ip_addr -u sm_admin -p sm_admin_pw -hn trusted_host_name -hc sm_hco
 - c. Example (Linux): ./smregshost.sh -i 192.168.1.225 -u siteminder -p p@ss1234! -hn cts_pf -hc cts_pf_hco
3. Remember the location of this directory and file as you will need to copy later. (ie. /opt/CTS_Install/CTS_2212_Package/cs-sdk/SmHost.conf)

Create the SiteMinder Policy Objects for CTS

The SDG Conversion Token Service requires a set of policies to exist in the environment. In addition to the configuration steps outlined here, a User Directory is required. This section assumes these items already exist.

There are a couple of steps for configuring the SiteMinder Policies:

1. Run the cts-install.pl script
2. Update the CTS Domain for your environment
3. Configure additional Agent Config Object Settings

SiteMinder Policy Base Object Creation

The cts-install.pl Perl script is used to create the base SiteMinder policies leveraged by the CTS SiteMinder Connector. The following steps outline the process for creating the base objects:

1. Copy the <CTS Home>\script\cts-install.pl or <CTS Home>/scripts/cts-install.pl file to the Policy Server
2. Run the Perl Script to create the SiteMinder Policy Server Objects using the following command (NOTE: SiteMinder's CLI Perl installation should be used to run this command in <SiteMinder Policy Server Home>\CLI\bin or <SiteMinder Policy Server Home>/CLI/bin directory): perl cts-install.pl
3. The installer begins
4. When prompted, enter the username and password of a siteminder admin user (e.g. siteminder)
5. The CTS uses a SiteMinder Domain to create its objects. The Domain name must be unique. Enter the name of the Domain to be created
6. An existing user directory is configured for authentication of user tokens. Specify the user directory to authenticate users that will be using CTS in the Domain Object User Directories
7. For username tokens, a search lookup is required to locate the user identity in the user directory. For details on the format of this lookup, refer to the CA SiteMinder Windows Authentication Scheme documentation. Specify the user search lookup
8. Confirm the installation parameters:

***** SDG Conversion Token Service Policy Server Installer *****

Enter Policy Server Administrator credentials -----

Administrator ID: siteminder

Administrator Password: <siteminder password>

Enter unique CTS Domain Name -----

Name (or X to exit): SDG Conversion Token Service Domain

Select User Directory Used for Authenticating Users ----- [1]

FederationWSCustomUserStore

[2]SAML2FederationCustomUserStore

[3]FedBCCustomUserStore

[4]FedBCCertUserDirectory

[5]Demo User Directory

[X]Exit install script

Enter number or X to exit: 5

Enter the user lookup search query for locating users Use %{UID} for the user ID

Optionally use %{DOMAIN} to further restrict the search For example: (sAMAccountName=%{UID}) -----

User lookup (or X to exit): (uid=%{UID})

Confirm Installation Parameters -----

[1] CTS Domain Name: SDG Conversion Token Service Domain

[2] Selected User Directory: Demo User Directory

[3] User search query: (uid=%{UID})

Enter [Y]es to continue, [X] to exit or number to modify value: y

9. The installation begins:

SDG Conversion Token Service Install and Configuration Guide – Version 1.0 Build V3

Creating CTS objects...Creating CTS Agent Identity .. Done Creating CTS User Attribute Authentication Scheme...Done Creating CTS Agent Configuration Object .. Done

Adding CTS Agent Configuration Parameters...Done Associating User Directory object.

Done Creating CTS Configuration Domain – SDG Conversion Token Service Domain .. Done

Creating ptokenresource Realm Done

Creating vtokenresource Realm Done

Creating uatokenresource Realm .. Done

Creating ptokentresource GET Rule ... Done

Creating vtokentresource GET Rule Done

Creating uatokentresource GET Rule .. Done

Creating CTS Policy...Adding rules to CTS Policy Done

10. The installation is complete

Update the CTS Domain for Your Environment

Once the base policies have been created, the objects can be updated to reflect your specific requirements. To update these policies:

1. Log into the SiteMinder Policy Server Administration Console
2. Navigate to the domain specified in step 5 above

View Domain: SDG Conversion Token Service Domain
[Domains](#) > View Domain: SDG Conversion Token Service Domain


General | Realms | Policies | Responses | Rule Groups | Variables

General

Name SDG Conversion Token Service Domain **Description** SDG Conversion Token Service Domain

Global Policies Apply ☒

User Directories

Name	Description
 Demo User Directory	Demo User Directory


3. Click on Policies:

Modify Domain: SDG Conversion Token Service Domain
Domains > Modify Domain: SDG Conversion Token Service Domain

General Realms **Policies** Responses Rule Groups Variables

• = Required

Policies

Name	Description
 SDG CTS Policy	Map users to resources, define additional attributes and configure other CTS items 

Create

Modify SDG Conversion Token Service Policy:

4. Add the list of authorized CTS users to the Policy

5. Submit the Changes to save the update

Modify Policy: SDG Conversion Token Service Policy
Domains > Modify Domain: SDG Conversion Token Service Domain > Modify Policy: SDG Conversion Token Service Policy

General **Users** Rules Expression

• = Required

User Directories

Demo User Directory

Allow Nested Groups ☐
AND Users/Groups ☐

Name	User Class	Exclude
No results.		

Add Members Add Entry Add All

Modify Policy: SDG Conversion Token Service Policy
Domains > Modify Domain: SDG Conversion Token Service Domain > Modify Policy: SDG Conversion Token Service Policy

General **Users** Rules Expression

• = Required

User Directories

Demo User Directory

Allow Nested Groups ☐
AND Users/Groups ☐

Name	User Class	Exclude
 All	All	 Exclude 

Add Members Add Entry Add All

This completes the configuration of the needed Policies, Objects, and Rules in CA SiteMinder. We now can go back to PingFederate and complete the steps there to get the CTS Service up and talking.

PingFederate

One of the core pieces that needs to be completed is creating a self-signed certificate the CTS uses to talk securely to PingFederate and swap tokens.

Generate A Client Certificate for SDG Conversion Token Service

From the PingFederate Administrative Console Select Security -> SSL Client Keys and Certificates:

1. Create New:

Create a new Certificate and Private Key.

COMMON NAME	SDG CTS
ORGANIZATION	SDG Corporation
ORGANIZATIONAL UNIT	
CITY	
STATE	
COUNTRY	US
VALIDITY (DAYS)	365
KEY ALGORITHM	RSA
KEY SIZE (BITS)	2048
SIGNATURE ALGORITHM	RSA SHA256

Click "Next" and "Done"

2. Record the serial number of the certificate:

CTS Client Certificate Serial Number (Enter with no :s) EX: 0167BED9C9CA	
--	--

3. Click Export to Save the certificate to your desktop:

01:67:BE:D9:C9:CA	CN= SDG CTS, O=SDG Corporation, C=US	Tue Dec 17 17:06:43 PST 2019	RSA 2048	Valid	Export Certificate Signing - Certificate not saved Delete
-------------------	---	------------------------------	----------	-------	--

4. Select “Certificate Only” and click “Next:”

Certificate Management | Export Certificate

Export Certificate Export & Summary

You have a choice of exporting the certificate and the key or just the certificate.

☒ CERTIFICATE ONLY
☐ CERTIFICATE AND PRIVATE KEY

Cancel Next

5. Click “Export” and “Done:”

Certificate Management | Export Certificate

Export Certificate Export & Summary

Click the Export button to export this certificate to the file system.

Export Certificate

Subject DN	CN=SDG CTS, O=SDG Corporation, C=US
Issuer DN	CN=SDG CTS, O=SDG Corporation, C=US
Serial Number	01:67:BE:D9:C9:CA
Expires	Tue Dec 17 17:06:43 PST 2019

Export

Cancel Previous Done

6. From the PingFederate Administrative Console → Security → Trusted CA’s select Import and choose the exported certificate you saved to your desktop in the previous step and click “Next:”

Certificate Management | Import Certificate

Import Certificate Summary

Please select the file containing the desired certificate.

FILENAME 67BED9C9CA.crt Choose file

Cancel Next

7. Verify the certificate and click “Save:”

The screenshot shows a web interface for 'Certificate Management' with a sub-tab 'Import Certificate'. Below the sub-tab are two buttons: 'Import Certificate' and 'Summary'. The 'Summary' tab is active, displaying a table of certificate details. At the bottom right, there are four buttons: 'Cancel', 'Previous', 'Done', and 'Save'. The 'Save' button is highlighted with a red border.

Import Certificate	
Filename	167BED9C9CA.crt
File Size	1084
Serial Number	0167:BED9:C9:CA
Subject DN	CN= SDG CTS, O=SDG Corporation, C=US
Issuer DN	CN= SDG CTS, O=SDG Corporation, C=US
Expires	Tue Dec 17 17:06:43 PST 2019
Signature Algorithm	SHA256withRSA
MD5 Fingerprint	50E5DEC8BC94781F7875CB294F2FC839
SHA1 Fingerprint	7DEB2F5D3D8C7842FACCD165387C991FC4435E7E

SDG Conversion Token Service

In this section we will deploy the SDG Conversion Token Service WAR file to the PingFederate Node, these steps should be performed as the user that owns and will run the PingFederate process.

JAVA

Ensure your JAVA_HOME Environment variables are set before proceeding. In Windows these are set in your system environment variables, in Linux these should be set in the appropriate *shell_profile.sh* script.

Installation of the SDG Conversion Token Service WAR file

1. From your PingFederate engine node folder navigate to the following folder:
 - a. `.../pingfederate_home/pingfederate/server/default/deploy`
 - b. Create a directory called “sdg-conversiontokenservice.war”
 - c. Example: `/opt/PingFederate.10.1.0/pingfederate/server/default/deploy/sdgconversiontokenservice.war`
2. Change into the newly created directory in the previous step:
 - a. `cd /opt/PingFederate.10.1.0/pingfederate/server/default/deploy/sdgconversiontokenservice.war`
(Make sure to use your path this is an example)
3. Run the following command: `jar -xvf /opt/CTS_Install/CTS_2212_Package/sdgconversiontokenservicecombined-2.2.12.war`
4. Verify the directory structure matches the sample below:

```
drwxr-xr-x. 10 pfuser pfadmin 4096 Nov 6 09:03 ..
drwxr-xr-x. 4 pfuser pfadmin 4096 Nov 6 09:03 META-INF
drwxr-xr-x. 10 pfuser pfadmin 4096 Nov 6 09:03 com
-rw-r-----. 1 pfuser pfadmin 938 Nov 6 09:03 log4j.properties
-rw-r-----. 1 pfuser pfadmin 1259 Nov 6 09:03 Launch.class
drwxr-xr-x. 4 pfuser pfadmin 4096 Nov 6 09:03 WEB-INF
drwxr-xr-x. 3 pfuser pfadmin 25 Nov 6 09:03 io
drwxr-xr-x. 2 pfuser pfadmin 32 Nov 6 09:03 groverconfig8491016507689653801
drwxr-xr-x. 11 pfuser pfadmin 4096 Nov 6 09:03 org
drwxr-xr-x. 2 pfuser pfadmin 35 Nov 6 09:03 mozilla
drwxr-xr-x. 5 pfuser pfadmin 46 Nov 6 09:03 javax
-rw-r-----. 1 pfuser pfadmin 2052 Nov 6 09:03 about.html
-rw-r-----. 1 pfuser pfadmin 319 Nov 6 09:03 jetty-dir.css
drwxr-xr-x. 3 pfuser pfadmin 23 Nov 6 09:03 netegrity
-rw-r-----. 1 pfuser pfadmin 1073 Nov 6 09:03 cryptoPerms
-rw-r-----. 1 pfuser pfadmin 740 Nov 6 09:03 smagentapibuild.properties
drwxr-xr-x. 3 pfuser pfadmin 23 Nov 6 09:03 jersey
drwxr-xr-x. 4 pfuser pfadmin 28 Nov 6 09:03 afu
drwxr-xr-x. 13 pfuser pfadmin 4096 Dec 18 05:47 .
```

- Copy the “license.lic” file that you received from SDG to the same directory you are in. The directory structure should now include the license file as shown below:

```
-bash-4.2$ ls -lart
total 52
drwxr-xr-x. 10 pfuser pfadmin 4096 Nov 6 09:03 ..
drwxr-xr-x. 4 pfuser pfadmin 4096 Nov 6 09:03 META-INF
drwxr-xr-x. 10 pfuser pfadmin 4096 Nov 6 09:03 com
-rw-r-----. 1 pfuser pfadmin 938 Nov 6 09:03 log4j.properties
-rw-r-----. 1 pfuser pfadmin 1259 Nov 6 09:03 Launch.class
drwxr-xr-x. 4 pfuser pfadmin 4096 Nov 6 09:03 WEB-INF
drwxr-xr-x. 3 pfuser pfadmin 25 Nov 6 09:03 io
drwxr-xr-x. 2 pfuser pfadmin 32 Nov 6 09:03 groverconfig8491016507689653801
drwxr-xr-x. 11 pfuser pfadmin 4096 Nov 6 09:03 org
drwxr-xr-x. 2 pfuser pfadmin 35 Nov 6 09:03 mozilla
drwxr-xr-x. 5 pfuser pfadmin 46 Nov 6 09:03 javax
-rw-r-----. 1 pfuser pfadmin 2052 Nov 6 09:03 about.html
-rw-r-----. 1 pfuser pfadmin 319 Nov 6 09:03 jetty-dir.css
drwxr-xr-x. 3 pfuser pfadmin 23 Nov 6 09:03 netegrity
-rw-r-----. 1 pfuser pfadmin 1073 Nov 6 09:03 cryptoPerms
-rw-r-----. 1 pfuser pfadmin 740 Nov 6 09:03 smagentapibuild.properties
drwxr-xr-x. 3 pfuser pfadmin 23 Nov 6 09:03 jersey
drwxr-xr-x. 4 pfuser pfadmin 28 Nov 6 09:03 afu
-rw-r-----. 1 pfuser pfadmin 191 Nov 6 09:03 license.lic
drwxr-xr-x. 13 pfuser pfadmin 4096 Nov 6 09:03 .
```

Configuration of the SDG Conversion Token Service

The following sections will outline the steps necessary to complete the installation and configuration of the SDG Conversion Token Service.

Copy and Update the ‘authenticatedUsers.txt’ file

- Copy the ‘authenticatedUsers.txt’ file from:
 ../pingfederate/server/default/deploy/sdgconversiontokenservice.war/WEB-INF/ TO
 ../pingfederate/server/default/conf

2. Edit the 'authenticatedUsers.txt' file and enter the serial number of the certificate created earlier (Should be in a table on Page 18). Be sure to remove all ":" from the serial number. Example: "01:67:BE:D9:C9:CA" will be "0167BED9C9CA"



```
0167BED9C9CA
```

3. Save the file

Copy and Rename the SmHost.conf to cbSmHost.conf

1. Copy the "SmHost.conf" file from /opt/CTS_Install/CTS_2212_Package/ca-sdk/SmHost.conf TO ../pingfederate/server/default/conf/cbSmHost.conf

Copy the CTS Integration Kit JAR File to PingFederate

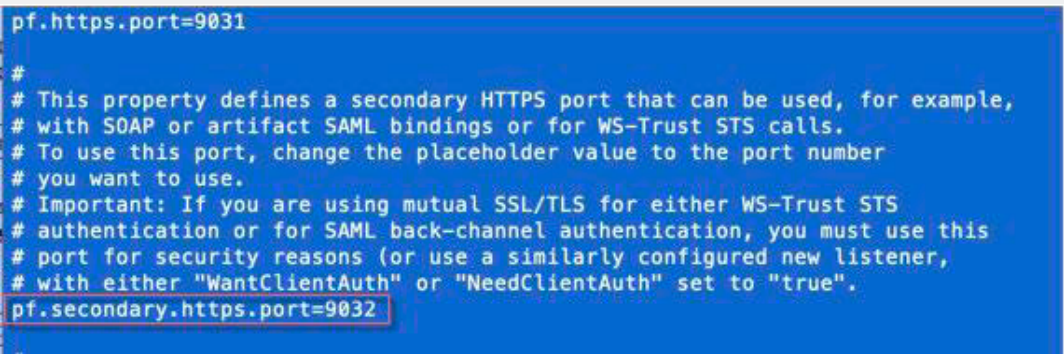
1. From the folder where you downloaded the sdg-integration-kit-2.6.2 file and extracted it, copy the sdg-integration-kit-2.6.36.jar to the following location ../pingfederate/server/default/deploy

Copy the CA SiteMinder Java SDK JAR Files to PingFederate

1. From the /opt/CTS_Install/CTS_2212_Package/cs-sdk/java64 folder:
 - a. cryptoj.jar
 - a. b. smagentapi.jar
2. Copy those files to ../pingfederate/server/default/lib

Edit the run.properties

1. Edit the 'run.properties' file located at ../pingfederate/bin/run.properties
2. Change the line pf.secondary.https.port=-1 to pf.secondary.https.port=9032
 - a. As stated earlier this is an example the port can be whatever you choose or need
3. Save the file



```
pf.https.port=9031
#
# This property defines a secondary HTTPS port that can be used, for example,
# with SOAP or artifact SAML bindings or for WS-Trust STS calls.
# To use this port, change the placeholder value to the port number
# you want to use.
# Important: If you are using mutual SSL/TLS for either WS-Trust STS
# authentication or for SAML back-channel authentication, you must use this
# port for security reasons (or use a similarly configured new listener,
# with either "WantClientAuth" or "NeedClientAuth" set to "true".
pf.secondary.https.port=9032
```

Edit 'cts.properties' file

1. In the folder `../pingfederate/server/default/deploy/sdg-conversiontokenservice.war/WEB-INF/classes` copy the `'_cts.properties'` to `'cts.properties'`
2. Edit the newly created `'cts.properties'` to reflect the configuration, be sure to use exact path, whether using Windows or Linux all directories must be represented with a `"\"`
 - a. `agentSmHostConfigFile` is the path to the config directory where the `cbSmhost.conf` file was copied
 - b. `agentConfigObject` is the name of the ACO created for CTS, this example below should be accurate if you used the perl script to create the policy store objects
 - c. `authenticatedUserSerialNumberPath` is the path to the file `'authenticatedUsers.txt'` file created to store the serial number of the certificate created earlier
 - d. `authResourceURI` is the URL for SDG, this should match the name of folder created under `../deploy` with the `.war` removed
3. Your file should look like the example below:

```
agentSmHostConfigFile=/opt/pingidentity/pf/pingfederate/server/default/conf/cbSmHost.conf
agentConfigObject=sdg_conversiontokenservices_aco
authenticatedUserSerialNumberPath=/opt/pingidentity/pf/pingfederate/server/default/conf/authenticatedUsers.txt
authRequestURI=/sdg-conversiontokenservice/1/token/ forceCertValidation=no useSharedSecret=no altidnumber=0
usealtid=false
ignore_altid_characters=false useTimeouts=true
```

4. Save File

Restart PingFederate for SDG Conversion Token Service to Deploy

1. Restart PingFederate in order to fully deploy the SDG Conversion Token Service.

Validate the SDG Conversion Token Service

1. Once PingFederate has restarted, check the `server.log` for the following line:

```
INFO
[org.eclipse.jetty.server.handler.ContextHandler] Started
o.e.j.w.WebAppContext@2bd02582{/sdg-conversiontokenservice-
2.0,file:///opt/pingidentity/pf/pingfederate/server/default/deploy/sdg-conversiontokenservice.war/,AVAILABLE}
```

2. Once you see this the SDG Conversion Token Service has deployed successfully and should be up and ready to service requests.

Testing the SDG Conversion Token Service

To ensure the SDG Conversion Token Service is up and ready to service requests you can run a test transaction through to validate it is ready and talking to SiteMinder correctly:

1. `curl -k https://localhost:9032/sdg-conversiontokenservice/1/token/test1234`
2. The Port used above in an example, if you used it fine, if not use whatever port you did setup PingFederate for the Secondary HTTPS Port

If a 404 is returned the service may not have deployed or there may be a configuration issue. If you receive 'NO CONNECTION' then you know the SDG Conversion Token Service is deployed, however it is having issues communicating with the SiteMinder Policy Servers.

If you receive 'INVALID TOKEN' then you have successfully deployed the SDG Conversion Token Service and the connection to the CA SiteMinder policy servers was successful.

Deploying SDG Conversion Token Service to Additional Nodes

You may use the following steps to deploy the SDG Conversion Token Service to additional clustered engine nodes:

1. Copy the `sdg-conversiontokenservice.war` directory to the same location on the other nodes in the cluster as you deployed on the first one
2. Copy the `authenticatedUsers.txt` file to the same directory on the other nodes as deployed on the first one
3. Copy the `cbSmHost.conf` to the same location on the other nodes as is located on the first node
4. Open the `cts.properties` file on the node and make sure it is pointing to the correct path on the new nodes for each of those files. This file can also be copied to the other clustered engine nodes.
5. Copy the `cryptoj.jar` and `smagentapi.jar` files into the other nodes located at `../pingfederate/server/default/lib`
6. Restart PingFederate on the new node
7. Repeat validation steps on each node as you complete all needed steps above

CHAPTER 3: CONFIGURING MUTUAL CLIENT CERTIFICATE AUTHENTICATION

To configure the SDG Conversion Token Service for mutual SSL, certificates must be configured both on the client and server side to ensure that the client is authorized to request tokens and also for it to be able to connect securely to the CTS.

The following tools are used to create the and import the certificates in this section:

- Java keytool
- openssl

Other tools can be used for this purpose. However, those tools are not documented in this guide. The keytool can be found at <JDK HOME>\bin or <JDK HOME>/bin depending on the Operating System.

The following steps are leveraged to configure mutual SSL:

Client Side:

1. Generate a client certificate key pair
2. Export client certificate in .cer format
3. Export client certificate in .pem format
4. Export client certificate private key in .pem format

Server Side:

1. Generate a keystore keypair
2. Import client certificate into keystore as a trusted certificate
3. Obtain certificate serial number from client certificate and add to authenticateusers.txt file

Client Side Certificate Generation

Generate a Client Certificate Key Pair

Keytool can be used to generate the client private key and certificate. Keytool is run from the command prompt.

The format for generating the certificate with keytool is as follows:

```
keytool -genkeypair -alias <ALIAS NAME> -keystore <KEY STORE NAME> -storetype pkcs12  
keyalg RSA
```

For example:

```
keytool -genkeypair -alias client -keystore client.p12 -storetype pkcs12 -keyalg RSA
```

After running the command, you will be prompted for several values. These values are specific to your environment. For example:

```
> keytool -genkeypair -alias client -keystore client.p12 -storetype pkcs12 -keyalg RSA
> Enter keystore password: keystorepassword >
Re-enter new password: keystorepassword >
What is your first and last name?
    [Unknown]: Client Certificate
> What is the name of your organizational unit?
[Unknown]: Client Services > What is the name
of your organization? [Unknown]: sdgc.com
> What is the name of your City or Locality? [Unknown]: New
York > What is the name of your State or Province? [Unknown]:
NY > What is the two-letter country code for this unit?
[Unknown]: US
> Is CN=Client Certificate, OU=Client Services, O=sdgc.com, L=New York, ST=NY, C=US correct?
    [no]: yes
```

The result of this example will generate the following file: client.p12

Export client certificate without Private Key for use on server side (.cer file)

The client certificate must then be imported into the Jetty server so that the CTS can validate the client is authorized to make token requests. Keytool is run from the command prompt.

Export the Certificate for Use on the CTS Server

The format for exporting the certificate using keytool is as follows:

```
keytool -exportcert -alias <CERTIFICATE ALIAS> -file <CERTIFICATE FILE NAME> -keystore
<KEY STORE NAME> -storetype pkcs12 -storepass <KEY STORE PASSWORD>
```

So, for our example, the command is:

```
keytool -exportcert -alias client -file client_cert.cer -keystore client.p12 -storetype
pkcs12 -storepass keystorepassword
```

The following file is generated for this example: client_cert.cer

Copy this file to the server where the SDG Conversion Token Service is being executed. This certificate will be used as a trusted certificate in the keystore used by the SDG Conversion Token Service.

Convert the Certificate to pem Format

For our example, openssl is used to convert the certificate to pem format. This certificate will be used later in the document with curl for testing the operation of the CTS. Openssl is run from the command line. On Windows, openssl can be downloaded. Other options also exist for obtaining openssl (e.g. cygwin).

The format for generating the pem certificate is as follows:

```
openssl x509 -inform der -in client_cert.cer -out client_cert.pem
```

So, for our example, the command is:

```
openssl x509 -inform der -in client_cert.cer -out client_cert.pem
```

The following file is generated: client_cert.pem

Extract the Private Key

For our example, openssl is used to extract the private key. This key will be used later in the document with curl for testing the operation of the CTS. openssl is run from the command line. On Windows, openssl can be downloaded. Other options also exist for obtaining openssl (e.g. cygwin).

The format for extracting the private key is as follows:

```
openssl pkcs12 -nodes -in <KEY STORE> -out <KEY FILE NAME>
```

After running the command, you will be prompted for the key store password.

So, for our example, the command is:

```
> openssl pkcs12 -nodes -in client.p12 -out clientkey.pem  
> Enter Import Password: keystorepassword MAC verified  
OK
```

The following file is generated: client_key.pem

Files Generated

The following files were generated for the documented example steps:

- client.p12
- client_cert.cer
- client_cert.pem
- client_key.pem

Server Side Certificate Generation and Mutual SSL Configuration

Generate a Keystore Key Pair

Keytool is used to configure SSL for the Jetty server. This allows clients to connect to the CTS over SSL to ensure that the communication is encrypted. On the server where the SDG Conversion Token Service is to be executed, go to the <CTS HOME>\config or <CTS HOME>/config directory depending on the Operating System.

The format for generating the certificate with keytool is as follows:

```
keytool -genkey -alias <ALIAS> -keyalg RSA -keyStore <KEY STORE NAME> -keysize 2048 sigalg "SHA1withRSA"
```

For example:

```
keytool -genkey -alias cts -keyalg RSA -keyStore cts_server.keystore -keysize 2048 sigalg  
"SHA1withRSA"
```

After running the command, you will be prompted for several values. These values are specific to your environment. For example:

```
> keytool -genkey -alias cts -keyalg RSA -keyStore cts_server.keystore -keysize 2048  
sigalg "SHA1withRSA"  
> Enter keystore password: keystorepassword >  
Re-enter new password: keystorepassword >  
What is your first and last name?  
[Unknown]: SDG Conversion Token Service > What  
is the name of your organizational unit?  
[Unknown]: Services  
> What is the name of your organization?  
[Unknown]: sdgc.com  
> What is the name of your City or Locality? [Unknown]: New  
York  
> What is the name of your State or Province?
```

```
[Unknown]: NY > What is the two-letter country code for
this unit? [Unknown]: US
> Is CN=SDG Conversion Token Service, OU=Services, O=sdgc.com, L=New York, ST=NY, C=US
correct?
[no]: yes
> Enter key password for <cts>
(RETURN if same as keystore password): <press RETURN>
```

The following file is generated: cts_server.keystore

Copy the generated key store to the <CTS HOME>\config or <CTS HOME>/config directory depending on the Operating System. If another location is used, the jetty.xml file must be updated to reflect the new location.

Copy the Client Certificate to the Server

In order validate the client connecting to the CTS, the certificate generated in the client certificate section above. Copy the client certificate into the <CTS HOME>\config or <CTS HOME>/config directory depending on the Operating System. For the example in this document, copy the client_cert.cer file generated in the previous steps into the config directory.

Import the Client Certificate into the keystore as a Trusted Certificate

The client certificate must be imported into the keystore so that the client can be validated during calls to the CTS. Keytool is run from the command line.

The format for importing the certificate is as follows:

```
keytool -importcert -keystore <KEY STORE NAME> -alias <ALIAS> -file <CERTIFICATE FILE> -v
-trustcacerts -noprompt -storepass <KEY STORE PASSWORD>
```

After running the command, you will be prompted for the key store password.

So, for our example, the command is:

```
> keytool -importcert -keystore cts_server.keystore -alias client -file client_cert.cer v
-trustcacerts -noprompt -storepass keystorepassword > Certificate was added to keystore
[Storing cts_server.keystore]
```

Verify the Certificate in the Keystore

Once the certificate is imported, the key store contents should be validated to ensure that the certificate imported correctly. This step is also used for obtaining the client certificate serial number.

This serial number is added to the `authenticatedusers.txt` file. The file contains the list of client certificates allowed to query the CTS. Keytool is run from the command line.

The format for validating the certificate is as follows:

```
keytool -v -list -keystore <KEY STORE NAME> -storepass <KEY STORE PASSWORD>
```

So, for our example, the command is:

```
> keytool -v -list -keystore cts_server.keystore -storepass keystorepassword
```

Keystore type: JKS

Keystore provider: SUN

Your keystore contains 2 entries

Alias name: client

Creation date: Apr 30, 2013

Entry type: trustedCertEntry

Owner: CN=Client Certificate, OU=Client Services, O=sdgc.com, L=New York, ST=NY, C=US

Issuer: CN=Client Certificate, OU=Client Services, O=sdgc.com, L=New York, ST=NY, C=US

Serial number: 51802ddb

Valid from: Tue Apr 30 13:47:23 PDT 2013 until: Mon Jul 29 13:47:23 PDT 2013 Certificate

fingerprints:

MD5: E2:D2:31:B8:FE:87:AE:5F:41:94:FF:DD:73:1D:8F:35

SHA1: BC:F6:09:3A:E6:DE:42:A7:C4:30:3C:A0:98:59:74:CB:2F:4B:EO:ED

Signature algorithm name: SHA1withRSA

Version: 3

```
Alias name: cts
Creation date: Apr 30, 2013
Entry type: PrivateKeyEntry
Certificate chain length: 1 Certificate[1]:
Owner: CN=SDG Conversion Token Service, OU=Services, O=sdgc.com, L=New York, ST=NY, C=US
Issuer: CN=SDG Conversion Token Service, OU=Services, O=sdgc.com, L=New York, ST=NY, C=US
Serial number: 51802e94
Valid from: Tue Apr 30 13:50:28 PDT 2013 until: Mon Jul 29 13:50:28 PDT 2013 Certificate
fingerprints:
    MD5: 81:8B:51:B7:11:3B:A2:45:7D:C6:99:01:FB:35:35:AB
    SHA1: F5:FA:EC:46:45:99:18:12:6C:DF:AA:EB:22:44:3E:01:49:F7:11:B2
    Signature algorithm name: SHA1withRSA
    Version: 3
```

```
*****
*****
```

For this example, the serial number required is bolded in the response above

Add Client Certificate Serial Number to the authenticateusers.txt File

The client certificate serial number is added to the authenticatedusers.txt file. The file contains the list of client certificates allowed to query the CTS. Copy the serial number from the client certificate alias. Open the authenticateusers.txt file and add the value into the file.

For the documented example, the value is: 51802ddb

About SDG

With more than 30 years of experience partnering with global enterprises on complex business and IT initiatives, SDG is a trusted provider of advisory, transformation, and managed services. The firm empowers organizations to strengthen cyber resilience by integrating AI into identity, threat, and risk management solutions that protect digital assets and deliver measurable business value. Learn more at www.sdgc.com.



- 75 North Water Street Norwalk, CT 06854
- 203.866.8886
- sdgc.com