



CertAgent[®]

Installation, Configuration, and Management Guide

Version 7.0
September 15, 2020

Information in this document is subject to change without notice and does not represent a commitment on the part of Information Security Corporation. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use without the prior written permission of Information Security Corporation.

CertAgent is commercial computer software and, together with any related documentation, is subject to the restrictions on U.S. Government use as set forth below.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 52.227-7013. "Contractor/manufacturer" is Information Security Corporation, 1011 W. Lake Street, Suite 425, Oak Park, IL 60301.

The U.S. International Traffic in Arms Regulations (ITARs) (22 CFR 125.03) prohibits the dissemination of certain types of technical data to foreign nationals.

Protected by U.S. Patent No. 5,699,431.

CertAgent is a trademark of Information Security Corporation. Other product and company names mentioned in this document may be the trademarks of their respective owners.

The cryptographic functionality of CertAgent is provided by CDK 7.0, ISC's FIPS 140-1 validated cryptographic module, via a Java JNI and/or an RMI interface. In addition, CertAgent uses code extracted or derived from the following open source software packages redistributable under the terms of the Apache License Version 2.0 and other licenses.

Apache Tomcat, Version 8.5.57: Copyright © 1999-2020 The Apache Software Foundation

<https://tomcat.apache.org/>

Log4j, Version 2.13.3: Copyright © 1999-2017 The Apache Software Foundation

<https://logging.apache.org/log4j/2.x>

jQuery, Version 3.5.1: Copyright © JS Foundation and other contributors

<https://jquery.com>

jQuery UI, Version 1.12.1: Copyright © jQuery Foundation and other contributors

<https://jqueryui.com>

HyperSQL database, Version 2.5.1: Copyright (c) 2001-2020, The HSQL Development Group

<http://hsqldb.org>

Gson, Version 2.8.6: Copyright © 2008 Google Inc.

<https://github.com/google/gson>

CertAgent Installation Guide, Version 7.0 (Revision 9, September 2020)

Copyright © 1991-2020 Information Security Corporation. All Rights Reserved.

Information Security Corporation

1011 W. Lake Street, Suite 425
Oak Park, IL 60301

Phone: +1 708-445-9415

Fax: +1 708 445-9705

E-mail: tech@infoseccorp.com

Web: www.infoseccorp.com

Table of Contents

1	Introduction.....	8
1.1	CertAgent Architecture.....	8
1.2	Roles	9
1.3	Technical Support	11
2	Installation and Configuration	12
2.1	System Requirements.....	12
2.1.1	Operating System.....	12
2.1.2	CPU	12
2.1.3	Random Number Generator	12
2.1.4	HSM.....	13
2.1.5	Java.....	13
2.1.6	Database.....	13
2.1.7	Servlet Container.....	18
2.1.8	Firewall.....	19
2.2	Installation.....	19
2.2.1	Installation Checklist	19
2.2.2	Unpacking the Software Distribution.....	20
2.2.3	Installation.....	20
2.2.4	Distribution Files	28
2.3	Updating an Existing CertAgent 7.x Installation	29
2.3.1	Checking for Update.....	29
2.3.2	Verifying the Update Package	30
2.3.3	Installing the Update	31
2.4	Advanced Configuration	33
2.4.1	Customizing the Web Page Headers and Footers.....	33
2.4.2	Configuring the Session Time-out	35
2.4.3	Disabling NIAP Conformance Setting on Startup	35
2.4.4	Replacing Administrator Credentials	36
2.4.5	Replacing TLS Credentials.....	39
2.4.6	High Availability and Load Balancing.....	42
2.4.7	Configuring HTTP Port.....	47
2.4.8	Updating Java	48
2.4.9	Configuring EST Port.....	50
2.4.10	Section 508 Compliance	51
3	Managing the CertAgent Server	52
3.1	Using the System Services	52

3.2	Using CertAgent Script.....	52
3.2.1	Starting the Server	52
3.2.2	Starting the Server in SSL Debug Mode	52
3.2.3	Starting the Server in Maintenance Mode.....	53
3.2.4	Stopping the Server.....	53
3.2.5	Checking the Server Status.....	53
3.2.6	Viewing the Server Log.....	53
3.2.7	Viewing the Version Number	53
4	Setting the System PIN	54
4.1	Using the CertAgent Script	54
4.2	Using the PIN Entry Page.....	54
4.2.1	Configuring the Authorized IP Addresses.....	55
4.3	Using the PIN Entry API	56
5	Using the CertAgent Web Interface	57
5.1	Importing Administrator Credentials into Browsers	57
5.1.1	Firefox.....	57
5.1.2	Internet Explorer	58
5.2	Enable Scripting in Browsers	58
5.3	Using the Administrative Site	59
5.4	Using the CA Account Site	60
5.5	Using the Public Site	60
6	Additional Management Tools.....	61
6.1	The CertAgent Command Line Tool.....	61
6.1.1	Command Line Syntax.....	61
6.1.2	Creating a CA Account.....	64
6.1.3	Creating a Profile.....	65
6.1.4	Generating Credentials for a CA Account	65
6.1.5	Generating a Certificate Request for a CA Account	66
6.1.6	Installing a CA Certificate	67
6.1.7	Selecting Existing Credentials for a CA Account.....	67
6.1.8	Listing Account Configuration Settings	68
6.1.9	Listing Profile Configuration Settings	68
6.1.10	Updating Account Configuration Settings.....	69
6.1.11	Listing Accounts and Profiles.....	69
6.1.12	Listing Supported Key Generation Options.....	69
6.1.13	Listing Available Hash Functions for a Specific Key Type and Size	70
6.1.14	Listing the Slots and Labels on an HSM.....	70

6.1.15	Viewing an Account's ACL	70
6.1.16	Adding to an Account's ACL	71
6.1.17	Updating the Permission of a Certificate from an Account ACL	72
6.1.18	Removing a Certificate from an Account's ACL.....	73
6.1.19	Importing Certificates	73
6.1.20	Importing a CRL	73
6.1.21	Exporting Certificates	73
6.1.22	Submitting Certificate Requests.....	74
6.1.23	Deleting a Profile	74
6.1.24	Disabling an Account.....	74
6.1.25	Listing Trust Anchors.....	75
6.1.26	Adding a Trust Anchor.....	75
6.1.27	Removing a Trust Anchor	75
6.1.28	Listing CRLs.....	76
6.1.29	Adding a CRL.....	76
6.1.30	Removing a CRL	76
6.1.31	Displaying System Information	76
6.1.32	Displaying Unique Subject DN Statistics	77
6.1.33	Exporting CA Account and Its Profiles Configurations	78
6.1.34	Importing CA Account and Its Profiles Configurations.....	81
6.2	The Certificate Report Generator.....	82
6.2.1	Command Line Syntax.....	82
6.2.2	Sample commands	83
7	The CertAgent RA Management Interface.....	84
7.1	Establishing a TLS Session with Client Authentication	85
7.2	Submitting a Certificate Request.....	85
7.2.1	Request	86
7.2.2	Response	95
7.3	Revoking a Certificate.....	97
7.3.1	Request	97
7.3.2	Response	98
7.4	Reinstating a Certificate	98
7.4.1	Request	98
7.4.2	Response	99
7.5	Issuing a CRL	100
7.5.1	Request	100
7.5.2	Response	100
7.6	Listing CA Account Names	101
7.6.1	Request	101

7.6.2	Response	102
7.7	Listing Certificate Requests	103
7.7.1	Request	103
7.7.2	Response	104
7.8	Listing Certificates	105
7.8.1	Request	105
7.8.2	Response	106
7.9	Retrieving a Certificate	108
7.9.1	Request	108
7.9.2	Response	109
8	The Database Access Service	110
8.1	Developing a Java Client	111
8.2	Using the DBAccess API	111
8.3	Supported SQL Syntax	117
8.3.1	CertAgent Database Schema.....	118
8.3.2	Sample SQL statements.....	121
9	Retrieving System Information and CA Resources	123
9.1	Retrieving CertAgent Version and System Information	123
9.1.1	Request	123
9.1.2	Response	124
9.2	Retrieving CA Resources.....	125
9.2.1	Request	125
9.2.2	Response	125
10	Backup and Recovery	126
10.1	System Files	126
10.2	Database	127
10.2.1	HyperSQL.....	128
10.2.2	PostgreSQL	129
10.2.3	Oracle	129
10.3	HSM	129

1 Introduction

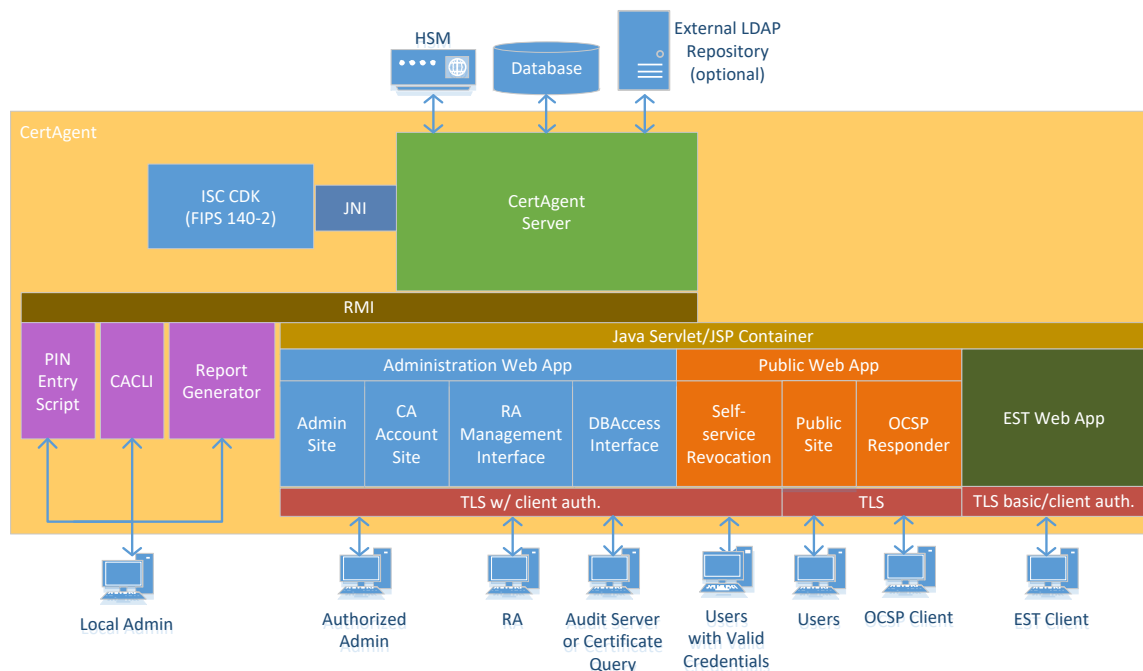
CertAgent is an X.509-compliant certificate authority (CA). It is an easily managed, web-based CA intended for use as the core component of an enterprise public key infrastructure (PKI). Designed to meet the needs of a wide variety of organizations, the current release offers enhanced enrollment over secure transport (EST) services, remote administration, integrated certificate and certificate revocation list (CRL) databases, and an OCSP responder. It supports an unlimited number of root and intermediate CAs, providing support for as complex a certificate hierarchy as the size of your enterprise warrants.

1.1 CertAgent Architecture

CertAgent consists of three separate web applications that must all be “deployed” into a Servlet/JSP container running on a webserver:

- an ‘admin site’ - the web application providing the PIN entry interface, system administration, CA account interfaces, registration authority management interface (RAMI), and DBAccess services
- a ‘public site’ - a web application providing end-user enrollment, certificate management interfaces, and OCSP interface
- an ‘EST site’ - a web application accepting Enrollment over Secure Transport (EST) requests

The following diagram illustrates the basic layout of the CertAgent system.



CertAgent 7 System Architecture

Most CA activities are completed by using a web browser that connects to the CertAgent web interface. The CA supports eight web-based interfaces using different ports or URLs (Admin System PIN Entry Site,

Admin Site, CA Account Site, Public Site, RAMI (Registration Authority Management Interface), DBAccess, EST, and OCSP).

- The Admin System PIN Entry Site channel is secured using HTTPS. Access to this site is limited to localhost and additional authorized IP addresses.
- The Admin Site, CA Account Site, DBAccess, and Registration Authority (RAMI) channels require valid identification and authentication credentials in the form of certificates. This channel is secured using client authenticated HTTPS/TLS.
- The Public Site channel is secured using HTTPS/TLS and HTTP. All pages except the CA Information page are HTTPS/TLS protected. The CA information page, used by relying parties to obtain CRLs, issuer certificates, and CA version information, is available without security over HTTP. All pages except the self-service revocation page are unauthenticated. The self-service revocation page requires valid identification and authentication credentials in the form of certificates.
- The EST channel is secured using HTTPS/TLS. Connections are authenticated with either certificates or a subscriber name and password.
- The OCSP interface is available without security over HTTP or secured using HTTP/TLS. All access is unauthenticated.

To support TLS (Transport Layer Security) encryption and authentication, the host webserver must be provisioned with a TLS certificate and private key and be configured with two TLS ports: one with enforced client authentication (for the admin site) and one without client authentication (for the public site).

In addition, CertAgent employs an independent 'system certificate' to encrypt sensitive information for all CA accounts. The private key corresponding to this certificate must reside on a hardware security module (HSM).

NOTE: All administrators and CAs must also be provided with login credentials (*i.e.*, certificates and private keys) so that they may properly authenticate themselves to the webserver.

CertAgent's OCSP capability is divided into basic OCSP support and enhanced OCSP support. Basic OCSP support provides OCSP responses for issuers managed by the CertAgent instance. If enabled, enhanced OCSP support, known as Dhuma, provides OCSP responses for issuers not managed by the CertAgent instance.

1.2 Roles

CertAgent supports three roles, Administrator, Auditor, and CA Operations Staff, each of which consists of an access control list (ACL) of one or more X.509 certificates and one or more permissions (admin, audit, certify, revoke, RAMI, and DBAccess).

CertAgent has two administrator web sites, each with its set of roles and access control lists.

The CertAgent Administrative webpages, known as the Admin Site, support the following roles and responsibilities:

Role	Permission	Responsibility
Administrator	admin	manage “system” credentials, database configuration settings, manage CA accounts, manage ACLs, trust anchor database, CRL store for path validations, NIAP configuration, run integrity tests, configure audit trails and manage jobs
Auditor	audit	view and export audit trails

The CertAgent CA Account webpages, known as the CA Site, support the following roles and responsibilities:

Role	Permission	Responsibility
Administrator	admin	manage account configurations (issuer credential, certificate profile, CRL issuance, certificate issuance, EST, OCSP, RAMI, and enrollment options)
Auditor	audit	view and export audit trails, and search certificates
CA Operations Staff	certify	issue certificates, reject invalid certificate requests, manage EST subscribers, manage automated certificate issuance option, and manage RAMI enrollment setting
	revoke	revoke certificates, issue CRLs, manage self-service certificate revocation option, manage automated CRL issuance option, manage RAMI CRL issuance, and revocation settings
	RAMI	submit requests via the RA management interface (RAMI)
	DBAccess	submit queries via the DBAccess service

Admin Site Administrators are assumed to have administrator privileges to the physical system on which the CertAgent is installed. They can:

- Install or update CertAgent
- Inject the system PIN
- Start/Stop the CertAgent, Apache Tomcat, and the Database services
- Run the CACLI program (allows the scripting of the creation of a root or issuer)
- Run the Report Generator Program

Both the Admin Site and the CA site maintain their own access control lists (ACL) containing authorized user certificates and permissions. Only users with the appropriate permissions can execute the defined functions. The role restriction is not configurable.

1.3 Technical Support

Information Security Corporation provides technical support for CertAgent during normal business working days, Monday through Friday, 8:00 a.m. to 5:00 p.m. Central Standard Time.

Phone: (708) 445-9415
Fax: (708) 445-9705
Web: www.infoseccorp.com/support/contents.htm
E-mail: tech@infoseccorp.com

2 Installation and Configuration

2.1 System Requirements

The following components are required to run CertAgent:

- Operating system: Microsoft Windows Server 2016 or above, Windows 10 or above, CentOS 7 or above, or Red Hat Enterprise Linux 7 or above
- a CPU supports RDRAND instruction (UNIX system only)
- memory: 3GB RAM (minimum)
- 64-bit Java 8, 11, or above
- a suitable database such as Oracle, PostgreSQL, and HyperSQL, and its JDBC driver
- a suitable Java Servlet container, Apache Tomcat
- a hardware security module (HSM)

2.1.1 Operating System

CertAgent should be installed and maintained by the privileged users of the operating system. Microsoft Windows Server 2016 or above, Windows 10 or above, CentOS 7 or above, or Red Hat Enterprise Linux 7 or above are supported.

2.1.2 CPU

On UNIX, CertAgent is required to run on a system with a CPU that supports RDRAND instruction. Run the following command to test if your CPU supports it:

```
cat /proc/cpuinfo | grep -i rdrand
```

2.1.3 Random Number Generator

On UNIX, the `rngd` daemon must be running to ensure that `/dev/random` has sufficient entropy. Otherwise, CertAgent cannot be run or there may be long delays during key generation. Follow the steps below to install and configure this daemon.

1. If `rngd` daemon has not been installed, run the following command:

```
yum install rng-tools
```

2. The program `rngd` and its service will be installed.
3. To start the service and make it starts and stops automatically upon system start-up and shut-down, run the following commands:

```
systemctl start rngd
systemctl enable rngd
```

2.1.4 HSM

An HSM must be installed and configured with an appropriate 64-bit HSM library prior to the CertAgent installation. You will be prompted for the path of the 64-bit HSM library and HSM access information during the installation.

CertAgent fully supports hardware security modules (HSMs) in the sense that the system key and each CA's keys can be stored on an HSM. While any HSM with a PKCS#11-compliant interface should work with CertAgent, the following HSMs have been successfully tested by ISC and found to be fully compatible:

- ISC Acala¹
- Engage Black BlackVault HSM
- Thales Trusted Cyber Technologies Luna Network, PCIe, and USB HSMs
- nCipher nShield Connect HSMs
- Envieta QFlex HSM
- Futurex Vectera Plus HSM

NOTE: HSM performance can have a direct impact on the responsiveness of CertAgent; slower HSMs may cause long delays and timeouts. This concern may be of critical importance when CertAgent is acting as an OSCP responder as that service requires the HSM to perform at least one additional signature operation for each response. For the latest information, please check ISC's website: <https://infosecorp.com/product/certagent>.

2.1.5 Java

A 64-bit Java 8 (also known as 1.8.0), 11, or above is required to be installed independently before installing CertAgent. Oracle JDK/JRE, OpenJDK, Amazon Corretto, and AdoptOpenJDK are supported.

NOTE: If CertAgent will be installed with the NIAP-compliant option, only Java version 8 update 241+, 11, or above is supported.

2.1.6 Database

The current release of CertAgent uses an auxiliary Oracle, PostgreSQL, or HyperSQL database for the storage of its credentials, account configurations, certificates, certificate request, CRLs, access control lists, and audit trails. Consequently, a compatible JDBC driver or Instant Client must be separately

¹ a software-based HSM that may be used with an "offline" CA.

licensed by the customer and installed on the CertAgent host; they are *not* included in the standard CertAgent software distribution package.

HyperSQL server 2.5.1 and its JDBC driver were included in the CertAgent installer. If this option is selected during the installation, a new HyperSQL server with a CertAgent database, and user account will be created and configured automatically. You can then skip this section.

If you are planning to create a new database or use an existing database, follow the vendor's document for installation and configuration.

The following sections describe how to obtain the JDBC driver of your database and create a database account for CertAgent. During the CertAgent installation, the JDBC driver location, database URL, user name and password will be prompted.

2.1.6.1 PostgreSQL

If you do not already have the PostgreSQL JDBC driver installed, it may be downloaded for free from PostgreSQL:

<https://jdbc.postgresql.org/download.html>

To install, choose the appropriate JDBC driver according to your PostgreSQL database version and save it to a file.

When the CertAgent installer prompts for the JDBC path, specify the location of the JDBC driver file.

2.1.6.1.1 Creating a Database User

A database user is required to have its own database schema to store CertAgent tables. Log in to the PSQL program as a system user and run the following commands to create a new database and a new user for CertAgent.

NOTE: The schema name must be exactly the same as the user name and the password must be wrapped in single quotes.

```

Syntax:
sudo su - postgres
createdb <db name>
psql -d <db name>
create user "<user>" password '<password>';
create schema "<schema>" authorization "<user>";

Example:
sudo su - postgres
createdb certagentdb
psql -d certagentdb
create user "certagent" password 'password';
create schema "certagent" authorization "certagent";

```

2.1.6.1.2 Using JDBC URL

To configure the CertAgent host to use the thin driver for database access, specify a database URI of the following form:

```
jdbc:postgresql://<host>:<port>/<database>
```

The above URL, database user, and password information are required when configuring the database settings from the CertAgent system administrative site. Please pass this information to the system administrator.

2.1.6.2 HyperSQL

If the HyperSQL database will be used and you do not have one installed already, you can select 'Install and configure a HyperSQL database for me' option during the installation. The HyperSQL JDBC driver and database user account will be installed and created within it. Database access information configuration, managed from the system administrative site, will be configured automatically. You can then skip this section.

2.1.6.2.1 Downloading the JDBC Driver

If you do not already have the HyperSQL JDBC driver installed, it may be freely downloaded from sourceforge.net:

<http://sourceforge.net/projects/hsqldb/files/hsqldb/>

To install:

1. Download the latest version.
2. Unzip the package into an appropriate directory (e.g., /usr).

A directory named 'hsqldb-_**<version>**' will be created. The JDBC driver hsqldb.jar is located in the lib directory.

When the CertAgent installer prompts for the JDBC path, specify the location of the JDBC driver file.

2.1.6.2.2 *Creating a Database User*

A database user is required to have its own database schema to store CertAgent tables. Log into the HSQL Database Manager as 'SA' and run the following commands to create a new user for CertAgent.

NOTE: The schema name must be exactly the same as the user name.

```
Syntax:
create user "<user>" password "<password>";
create schema "<schema>" authorization "<user>";
alter user "<user>" set initial schema "<schema>";

Example:
create user "certagent" password "password";
create schema "certagent" authorization "certagent";
alter user "certagent" set initial schema "certagent";
```

2.1.6.2.3 *Using JDBC URL*

To configure the CertAgent host to use the thin driver for database access, specify a database URI of the following form:

```
jdbc:hsqldb:hsqldb://<host>:<port>/<alias>
```

The above URL, database user, and password information are required when configuring the database settings from the CertAgent system administrative site. Please pass this information to the system administrator.

2.1.6.3 *Oracle*

NOTE: Oracle database is not supported in NIAP compliant CertAgent.

2.1.6.3.1 *Using the Oracle JDBC Thin Driver*

If you are planning to use the JDBC thin driver to connect to your Oracle database and do not already have it installed, it may be freely downloaded from the Oracle website:

<https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html>

To install:

1. Select the appropriate database version.
2. Choose the appropriate `ojdbc6.jar` file and save it to a file.

When the CertAgent installer prompts for the JDBC path, specify the location of the `ojdbc6.jar` file.

2.1.6.3.2 Using the Oracle JDBC OCI Driver

If you are planning to use the JDBC-OCI driver to connect to your Oracle database and do not already have the Instant Client installed, it may be freely downloaded from Oracle:

<https://www.oracle.com/database/technologies/instant-client/downloads.html>

To install:

1. Select the appropriate 64-bit Instant Client for your CertAgent host platform.
2. Download the latest 'Instant Client Package – Basic' package.
3. Unzip the package into an appropriate directory (e.g., /usr).

A directory named 'Instantclient_<version>' will be created.

When the CertAgent installer prompts for the JDBC path, specify the location of the ojdbc6.jar file in the Instant Client directory. After completing the CertAgent installation, append the ORACLE_HOME directory to your LD_LIBRARY_PATH by inserting the highlighted text below into <ca home>/certagent.sh:

```
<...>
LIB_HOME=$CA_HOME/lib; export LIB_HOME
ORACLE_HOME=/usr/instantclient_11_2; export ORACLE_HOME
LD_LIBRARY_PATH=$CA_HOME/bin:$LD_LIBRARY_PATH:$ORACLE_HOME; export
LD_LIBRARY_PATH
<...>
```

or <ca home>\certagent.bat:

```
<...>
set ORACLE_HOME=C:\instantclient_11_2
set PATH=%CA_HOME%\bin;%PATH%;$ORACLE_HOME%
<...>
```

Save your changes and restart the CertAgent service.

2.1.6.3.3 Creating a Database User

A database user is required to have its own database schema to store CertAgent tables. Log in to the SQLPLUS program as a system user and run the following command to create a new user for CertAgent.

```
Syntax:
SQL> create user <user> identified by <password>;
SQL> grant connect,resource to <user>;
```

```
Example:
SQL> create user certagent identified by password;
SQL> grant connect,resource to certagent;
```

2.1.6.3.4 Using JDBC URL

To configure the CertAgent host to use the thin driver for database access, specify a database URI of the following form:

```
jdbc:oracle:thin:@//<host>:<port>/<service name>
```

If your database supports OCI, an SQL*Net configuration file may be used to define the necessary database connections. Below is a sample `tnsnames.ora` file that must be located in the `<ORACLE_HOME>/network/admin` directory.

```
RACDB =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = cl6cluster-scan) (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = racdb.infoseccorp.com)
    (FAILOVER_MODE = (TYPE=select) (METHOD=basic))
  )
)
```

When configuring the database connection on the CertAgent host, specify `jdbc:oracle:oci:@racdb` in the URL field.

Alternatively, you may include the above configuration settings in the URI:

```
jdbc:oracle:oci:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=cl6cluster-scan) (PORT=1521)) (CONNECT_DATA=(SERVER=DEDICATED)
(SERVICE_NAME=racdb.infoseccorp.com) (FAILOVER_MODE=(TYPE=select)
(METHOD=basic))))
```

The above URL, database user, and password information are required when configuring the database settings from CertAgent system administrative site. Please pass this information to the system administrator.

2.1.7 Servlet Container

A servlet container is required to deploy CertAgent web applications. Apache Tomcat 8.5.57 was included in the CertAgent installer. **It will be installed and configured automatically during installation.**

2.1.8 Firewall

CertAgent should be installed on a machine that is well-protected behind a properly configured firewall. Only the following ports should be opened to the host system:

- the Admin and CA Account Sites' port for HTTPS with client authentication (default: 8443)
- the Public Site's port for HTTPS without client authentication (default: 443)
- the HTTP port (default: 80) to accept OCSP requests if enabled

2.2 Installation

To make CertAgent conform to the National Information Assurance Partnership (NIAP) requirements:

- **NIAP compliance option must be selected during the installation.**
- All the options specified in the NIAP conformance options page in the Admin Site must be checked. For details, see the section entitled *Managing the NIAP Conformance Options* in the CertAgent Administrator Guide (ca_adminhelp.pdf).

2.2.1 Installation Checklist

The checklist below provides an overview of the installation and configuration tasks required to get a typical CertAgent system up and running.

- ☐ On UNIX, ensure the CPU supports RDRAND instruction
- ☐ On UNIX, ensure the rngd daemon is running
- ☐ Install an HSM and appropriate 64-bit HSM library
- ☐ Install Java
- ☐ Unpack the software distribution and run the installation wizard
- ☐ Configure firewall
- ☐ Install the administrator's credentials into a web browser on the administrator's system (*e.g.*, into Internet Explorer's CAPI store)
- ☐ Enter the system PIN

2.2.2 Unpacking the Software Distribution

2.2.2.1 UNIX

The CertAgent package for UNIX platforms consists of a zip archive that may be unzipped (with directory structure preserved) into any convenient directory on your webserver's hard drive.

A 64-bit HSM library and a JDBC driver (if PostgreSQL or Oracle database will be used) must be installed on the server before installing CertAgent.

The following directories must be specified in the LD_LIBRARY_PATH variable:

- the 64-bit HSM libraries
- the Oracle Instant Client libraries (if OCI driver will be used)

For example:

```
LD_LIBRARY_PATH=/opt/acala
export LD_LIBRARY_PATH
```

If the required components have already been installed, go to the `certagent<version>-install` directory and run `install.sh` to start the installation process of installing CertAgent.

2.2.2.2 Windows

CertAgent for Windows is packaged as a zip archive that may be unzipped into any convenient directory on your webserver's hard drive. After unzipping the archive, run `certagent.<version>.x64.exe` to install CertAgent. Once the installation program begins, just follow the on-screen instructions.

2.2.3 Installation

By default, the installer prompts you for the settings during the installation process. Alternatively, settings can be customized in a configuration file and applied automatically during the installation.

2.2.3.1 Default

The following information may be prompted for during the installation process. A description for each field is found in the table below:

Field	Description
Installation Type	NIAP compliance or non-NIAP compliance
64-bit Java installation directory	64-bit Java installation directory; e.g., <code>/usr/java/jdk-11.0.8</code>
Hostname or IP address	Hostname or IP address of the system; e.g., <code>192.198.0.20</code>

Port number for public HTTPS access (<public port>)	This port will be used for HTTPS without client authentication; default to 443
Port number for admin HTTPS access (<admin port>)	This port will be used for HTTPS with client authentication; default to 8443
Public HTTP option	Option to open an HTTP port for OCSP and CRL retrieval; default: enable
Port number for public HTTP port	Prompt if 'public HTTP port' option is selected. This port will be used for OCSP and CRL retrieval in addition to the public HTTPS access port; default to 80
Component Type	(LINUX only) Install CertAgent only Install Dhuma only Install CertAgent and Dhuma
CertAgent serial number	CertAgent serial number; Required to enable CertAgent's features
Dhuma serial number	Dhuma serial number; Required to enable Dhuma's features
Database option	Install and configure a HyperSQL 2.5.1 Use an existing PostgreSQL database Use an existing Oracle database Use an existing HyperSQL database
Port number for HyperSQL database server	Prompt if 'Install and configure a HyperSQL database' option is selected; this port will be used for accessing the HyperSQL database; default to 9001
Password for the default SA account	Prompt if 'Install and configure a HyperSQL database' option is selected; the HyperSQL server is managed by the default SA account. This password will be set for the SA account
Database URL	Prompt if 'Use an existing database' option is selected; database access URL
Database user name	Database user name; If 'Use an existing database' option is selected; user name to manage the 'certagent' database created in the HyperSQL server; default to certagent Otherwise: user name of the existing database
Database user password	Password of the specified database user name
64-bit HSM library	Path of the 64-bit HSM library; e.g., /opt/acala/libacala.so
HSM Partition	HSM Partition; Prompt to select one of the partitions found in the HSM
HSM PIN	HSM PIN
Password for credentials (<p12 password>)	PKCS#12 password for TLS and administrator credentials that will be generated by the installer
Organization	Organization; e.g., ISC. If prompts, this value will be used to populate the subject DN base: O=<organization>, C=US

Subject DN base (<base>)	Subject DN base; e.g., O=ISC, C=US. This value will be used to populate the subject DN of the initial certificates; <dn>: <prefix>, <base>
Subject DN Prefix (<prefix>)	Subject DN prefix; e.g. CN=CertAgent 7.0.9 Root CA E275; This value will be used to populate the subject DN of the initial certificates; <dn>: <prefix>, <base>
Validity Period	Validity period for the initial certificate
Key type	Key type of the initial certificate; Select either RSA 3072 or NIST P-384
DN encoding	DN encoding for the initial certificate; PrintableString (default) or UTF8String

Once the installation program begins, just follow the on-screen instructions.

Accepting all default installation options is recommended. The default installation directory is `C:\Program Files\CertAgent7` on Windows and `/usr/local/certagent7` on UNIX. Throughout this document <ca home> will be used to refer to the CertAgent root installation directory.

If NIAP compliance installation type is selected, the installer will:

- create a HyperSQL database server or use an existing PostgreSQL database
- generate an HSM-based TLS credential

If non-NIAP compliance installation type is selected, the installer will:

- create a HyperSQL database or use an existing PostgreSQL, Oracle, or HyperSQL database
- generate a software-based TLS credential

For both installation types, the installer will:

- install Apache Tomcat 8.5.57
- install the program files
- generate credentials for the system and root CA on an HSM
- generate software-based credentials for an administrator, an auditor, a CA operations staff

During the installation, it prompts for the server IP or hostname, admin TLS port, public TLS port, HSM access information, organization, and PKCS#12 password. Be sure to make note of the PKCS#12 password (<p12 pass>), admin TLS port (<admin port>) and public TLS port (<public port>) you enter during installation.

If 'HyperSQL database' option is selected, the installer will prompt for a listening port, new SA password, and new user name and password information. The installer will install a HyperSQL database server with a CertAgent database. The default SA account password will be updated, and a user account with the specified name and password will be created for the CertAgent database. Be sure to make a note of the database information you enter during installation.

If 'HyperSQL database' option is not selected, the installer will prompt for which existing database (PostgreSQL, HyperSQL, or Oracle) and its JDBC driver to be used. The database access information (URL, user name and password) will be prompted.

A CA account (ca7) and three profiles (ca7clientauth, ca7tls and ca7ocsp) will be created during the installation. Root CA, System, TLS, administrator, auditor, CA operations staff, and OCSP signer credentials will also be generated.

By default, the subject DN of the certificates generated are in the following form:

CN=CertAgent <version> System Key <4 random characters>, <base>

CN=CertAgent <version> Root CA <4 random characters>, <base>

CN=<server IP>, <base>

CN=CertAgent <version> Administrator, <base>

CN=CertAgent <version> Auditor, <base>

CN=CertAgent <version> CA Operations Staff, <base>

CN=CertAgent <version> OCSP Signer, <base>

For example:

CN=CertAgent 7.0.9 System Key E275, O=ISC, C=US

CN=CertAgent 7.0.9 Root CA E275, O=ISC, C=US

CN=192.168.0.86, O=ISC, C=US

CN=CertAgent 7.0.9 Administrator, O=ISC, C=US

CN=CertAgent 7.0.9 Auditor, O=ISC, C=US

CN=CertAgent 7.0.9 CA Operations Staff, O=ISC, C=US

CN=CertAgent 7.0.9 OCSP Signer, O=ISC, C=US

The subject DN, key type and validity period of these certificates are customizable. The installer will prompt for a DN base which will apply to all the DNs and a DN prefix for each certificate.

NOTE: These credentials should be considered temporary and only used to facilitate initial system setup. They should be replaced with properly issued credentials before making the system operational. For details, see sections 2.4.4 Replacing Administrator Credentials and 2.4.5 Replacing TLS Credentials.

Once the installation is completed, CertAgent service will start automatically.

2.2.3.2 Custom

If you are planning to install CertAgent with the same configuration in multiple systems, you can automate the process by customizing the configurations in a file, and optionally creating a post-installation script. Customized settings will be applied automatically and suppressed the prompts.

2.2.3.2.1 Configuration File

A sample configuration file is provided in the package:

```
custom-install/install-config.ini
```

A description for each variable is found in the table below. Update the values in the configuration file as appropriate for your installation.

NOTE: All values are required to set unless indicated as optional. Values with spaces are required to be specified within double quotes.

Field	Description	Example
CA_SERIAL_NUMBER	(Optional) CertAgent serial number; if specified, CertAgent's features will be enabled; if not specified, DHUMA_SERIAL_NUMBER must be specified	
DHUMA_SERIAL_NUMBER	(Optional) Dhuma serial number; if specified, Dhuma's features will be enabled; if not specified, CA_SERIAL_NUMBER must be specified	
CA_HOME	CertAgent installation directory	/usr/local/certagent7 "C:\Program Files\CertAgent7"
NIAP_MODE	1: NIAP compliance 2: non-NIAP compliance	1
JAVA_HOME	64-bit Java installation directory If NIAP_MODE=1, Java 8 is required If NIAP_MODE=0, Java 8, 11 or above is required	/usr/java/jre1.8.0_231 "C:\Program Files\Java\jre1.8.0_231"
APPEND_LD_LIBRARY_PATH	(Optional) Linux only If the directories containing the 64-bit HSM libraries and the Oracle Instant Client libraries (if OCI driver will be used) are not specified in the LD_LIBRARY_PATH variable. These directories can be specified in this variable which will be appended to the LD_LIBRARY_PATH variable	/opt/acala
HOST	Hostname or IP address of the system	ca1.infosecorp.com
ADMIN_PORT	This port will be used for HTTPS with client authentication for the administrative sites	8443
PUBLIC_PORT	This port will be used for HTTPS without client authentication for the public site	433
HTTP_PORT	By default, CRLs and OCSP responses can be retrieved via the public port.	80

	To open an HTTP port for OCSP and CRL retrieval, specify a port number (1-65535); Otherwise, specify 0	
DB_TYPE	Database type: 1: install and create a new HyperSQL database 2: use an existing PostgreSQL database 3: use an existing Oracle database (NIAP_MODE=0 only) 4: use an existing HyperSQL database (NIAP_MODE=0 only)	1
HSQL_PORT	(Optional) DB_TYPE=1 only This port will be used for accessing the HyperSQL database	9001
HSQL_SA_PASS	(Optional) DB_TYPE=1 and NIAP_MODE=0 only The HyperSQL server is managed by the default SA account. This password will be set for the SA account; If not specified, password will be prompted during installation	
HSQL_USER	(Optional) DB_TYPE=1 Database user name	certagent
HSQL_PASS	(Optional) DB_TYPE=1 and NIAP_MODE=0 only This password will be set for the specified database user name; If not specified, password will be prompted during installation	
JDBC_PATH	(Optional) DB_TYPE=2, 3, or 4 only The JDBC library path (*.jar) of the selected database	/home/admin/Downloads/postgresql-42.2.8.jar C:\users\admin\Downloads\postgresql-42.2.8.jar
DB_URL	(Optional) DB_TYPE=2, 3, or 4 only Format: PostgreSQL: jdbc:postgresql://<host>:<port>/<database> Oracle: jdbc:oracle:thin:@//<host>:<port>/<service name> HyperSQL: jdbc:hsqldb:hsqldb://<host>:<port>/<alias>	
DB_USER	(Optional) DB_TYPE=2, 3, or 4 only Database user name	
DB_PASS	(Optional) DB_TYPE=2, 3, or 4 and NIAP_MODE=0 only Password of the specified database user name; If not specified, password will be prompted during installation	

HSM_LIB	Path of the 64-bit HSM library	/opt/acala/libacala.so "C:\Program Files\Acala\acala_p11x64.dll"
HSM_LABEL	HSM label	"ISC Acala"
HSM PIN	(Optional) NIAP_MODE=0 only HSM PIN; If not specified, password will be prompted during installation	
DN_ENCODING	DN encoding for the initial certificates 1: PrintableString 2: UTF8String	1
DN_BASE	Subject DN base This value (\$DN_BASE) can be appended to the SYSTEM_DN, ROOT_CA_DN, TLS_DN, ADMIN_DN, AUDITOR_DN and CA_OPS_DN values	"O=ISC, C=US"
SYSTEM_DN	Subject DN of the RSA-3072 system certificate \$DN_BASE variable can be appended to the DN	"CN=CertAgent 7.0.9 System Key, \$DN_BASE"
SYSTEM_VALIDITY	Validity period of the RSA-3072 system certificate Specify number of days, months or years	"5 years"
SYSTEM_KEY_TYPE	Key type of the system certificate 1: RSA 3072 2: NIST P-384	1
ROOT_CA_DN	Subject DN of the Root CA certificate \$DN_BASE variable can be appended to the DN	"CN=CertAgent 7.0.9 ROOT CA, \$DN_BASE"
ROOT_CA_VALIDITY	Validity period of the Root CA certificate Specify number of days, months or years	"5 years"
ROOT_KEY_TYPE	Key type of the Root CA certificate 1: RSA 3072 2: NIST P-384	1
TLS_DN	Subject DN of the TLS certificate \$DN_BASE and \$HOST variables can be used in the DN	"CN=\$HOST, \$DN_BASE"
TLS_VALIDITY	Validity period of the TLS certificate Specify number of days, months or years	"5 years"
TLS_KEY_TYPE	Key type of the TLS certificate 1: RSA 3072 2: NIST P-384	1

ADMIN_DN	Subject DN of the administrator's certificate \$DN_BASE variable can be appended to the DN	"CN=CertAgent 7.0.9 Administrator, \$DN_BASE"
AUDITOR_DN	Subject DN of the auditor's certificate \$DN_BASE variable can be appended to the DN	"CN=CertAgent 7.0.9 Auditor, \$DN_BASE"
CA_OPS_DN	Subject DN of the CA operations staff's certificate \$DN_BASE variable can be appended to the DN	"CN=CertAgent 7.0.9 CA Operation Staff, \$DN_BASE"
ROLE_KEY_TYPE	Key type of the administrator's, auditor's and CA operations staff's certificates 1: RSA 3072 2: NIST P-384	1
ROLE_VALIDITY	Validity period of the administrator's, auditor's and CA operations staff's certificates Specify number of days, months or years	"90 days"
OCSP_SIGNER_DN	Subject DN of the OCSP signer certificate \$DN_BASE variable can be appended to the DN	"CN=CertAgent 7.0.9 OCSP Signer, \$DN_BASE"
OCSP_SIGNER_VALIDITY	Validity period of the OCSP signer certificate Specify number of days, months or years	"5 years"
OCSP_SIGNER_KEY_TYPE	Key type of the OCSP signer certificate 1: RSA 3072 2: NIST P-384	1
KEY_PASS	(Optional) NIAP_MODE=0 only Password of the role credentials and trust keystore; If not specified, password will be prompted during installation	
CA_NAME	CA account name	ca7
CA_DISPLAY_NAME	Display name of the CA account	"CertAgent 7"
POST_SCRIPT	(Optional) Full path of the script which runs if CertAgent installed successfully; See next section for details	

2.2.3.2.2 Post-installation Script

If the POST_SCRIPT value is specified in the configuration, it will be run upon successful installation.

A sample post-installation script is provided in the package:

```
custom-install/sample-post-install.sh or .bat
```

This script:

- prompts to set the system PIN
- creates a CA account
- add the administrator, CA operations staff, and auditor certificates to the ACL with the corresponded permissions
- updates the account's configuration

2.2.3.2.3 Installation

Run the following command to install CertAgent with a configuration file:

./install.sh <configuration file>	(UNIX)
certagent.<version>.x64.exe <configuration file>	(Windows)

2.2.4 Distribution Files

The contents of the current distribution are as follows:

Directory	Files	Description
<ca home>	certagent.sh (UNIX) isc-certagent7 (UNIX) tomcat.sh (UNIX) isc-certagent7-tomcat (UNIX) isc-certagent7-hsqldb (UNIX) certagent.bat (Windows) tomcat.bat (Windows) hsqldb.bat (Windows) uninstall.exe (Windows)	CertAgent server scripts Apache Tomcat scripts (if installed) HyperSQL script (if installed) Uninstall program (Windows)
<ca home>/bin	*.kyp *.so.* (UNIX) *.dll (Windows)	CertAgent program libraries
<ca home>/conf	certagent.xml certagent.dtd certagent.<yyyy.mm.dd_hh.MM.ss>.xml	CertAgent XML configuration file, backup configuration files and DTD
<ca home>/conf/.system	*.der	system certificate directory
<ca home>/conf/logs	log log.<yyyy-mm-dd>	local debug text file directory
<ca home>/hsqldb	*	HyperSQL database directory (if installed)
<ca home>/keystore	*	keystore directory containing all credentials generated by the installer

<ca home>/lib	*.jar *.txt log4j.properties	Java program libraries, licenses, and log4j configuration file for additional audit trail output
<ca home>/tomcat	*	Apache Tomcat 8
<ca home>/tools	RAMISample.java reportgenerator.sh (UNIX) reportgenerator.bat (Windows)	RAMI sample program Report generator tool
<ca home>/tools/caccli	caccli.sh (UNIX) caccli.bat (Windows) *.txt	CACLI program Sample configuration files
<ca home>/tools/dbaccess	doc/* certagentdbaccess.jar DBAccessSample.java	DBAccess documents DBAccess Java library DBAccess sample program
<ca home>/webapps	*.war	Web application archives

2.3 Updating an Existing CertAgent 7.x Installation

CertAgent provides local administrators the ability to check for updates on demand via the update tool. At the local console a local administrator initiates the installation of an update package using the update tool. Once initiated, CertAgent verifies the digital signature on the package and will stop the process if the signature or the certificate used is not valid.

The supplied Update tool program can be used to check if an update is required, validate and install the update package:

```
<ca home>/update/update-tool.sh (UNIX)
<ca home>\update/update-tool.bat (Windows)
```

2.3.1 Checking for Update

To display the CertAgent version and check for an update:

1. Run the update tool with '-check' option.
2. It will display the current version and connect to ISC's web page (<https://www.infosecorp.com/inc/products.xml>) for the latest version number and release date.

```
C:\Program Files\CertAgent7\update> update-tool -check
CertAgent 7.0.9 Update Tool
Copyright(c) 1991-2020 Information Security Corp. All rights reserved.

*****
Checking for update...
*****
Installed version: 7.0.9
Your version is up-to-date.
```

If a newer version is available, it is delivered in a zipped archive via ISC's website..

2.3.2 Verifying the Update Package

Use the following command to verify the signed update package:

```
<ca home>/update/update-tool.sh -verify <p7m file> (UNIX)
<ca home>\update\update-tool.bat -verify <p7m file> (Windows)
```

CertAgent will verify the signature, obtain the signer certificate information, perform a path validation check, and verify the version from the update package. Results will be recorded to the audit trail. If the certificate is valid with proper extensions (code signing purpose in extended key usage and digital signature purpose in key usage) and its root certificate is in the trust anchor database, the signer certificate information and package information will be displayed. If the package or the signer certificate is invalid, an error message will be displayed.

Example:

```

C:\Program Files\CertAgent7\update> update-tool -verify certagent.7.0.9-
update.win.x64.p7m
CertAgent 7.0.8 Update Tool
Copyright(c) 1991-2020 Information Security Corp. All rights reserved.

*****
Verifying the update package...
*****

* Verifying the signature of the package...
Signer certificate:
  Serial: 11E022A8A5E3F378C7C8AD97597D69912358C523
  Issuer: CN=Information Security Corporation CA 5, L=Oak Park, O=ISC, ST=IL, C=US
  Subject: CN=Information Security Corporation Code Signing Certificate, L=Oak Park,
O=ISC, ST=IL, C=US
  NotBefore: 05/29/17 19:00:00 CDT
  NotAfter: 05/29/37 19:00:00 CDT
Signature verified.

* Verifying signer certificate...
Verified signer certificate with path validation

* Verifying package information...
Version: 7.0.9
Verified.

Update package verified.
EXIT

```

If the same or an older version of the update package is specified, a warning message will appear in the Verify package information section.

Example:

```

* Verifying package information...
Version: 7.0.9
Verified.
Warning: Version 7.0.9 has already been installed

```

2.3.3 Installing the Update

To install the update package:

1. Once the signed package has been verified, run the following command to install the update package:

```

<ca home>/update/update-tool.sh -install <p7m file> (UNIX)
<ca home>\update\update-tool.bat -install <p7m file> (Windows)

```

CertAgent will verify the package as described in the previous section. Program files will be extracted.

Example:

```
C:\Program Files\CertAgent7\update> update-tool -install certagent.7.0.9-
update.win.x64.p7m
CertAgent 7.0.8 Update Tool
Copyright(c) 1991-2020 Information Security Corp. All rights reserved.

*****
Verifying and Installing the update package...
*****
* Verifying update package; please wait...

* Verifying the signature of the package...
Signer certificate:
  Serial: 11E022A8A5E3F378C7C8AD97597D69912358C523
  Issuer: CN=Information Security Corporation CA 5, L=Oak Park, O=ISC, ST=IL,
C=US
  Subject: CN=Information Security Corporation Code Signing Certificate, L=Oak
Park, O=ISC, ST=IL, C=US
  NotBefore: 05/29/17 19:00:00 CDT
  NotAfter: 05/29/37 19:00:00 CDT
Signature verified.

* Verifying signer certificate...
Verified signer certificate with path validation

* Verifying package information...
Version: 7.0.9
Verified.

* Extracting program files; please wait...
Files extracted.
```

2. Review the package and signer information. Enter **yes** when prompted:

```
Update package verified.
Do you want to install the update now? (yes/no): yes
```


- The update script in the package will be executed. The update process will begin and its result will be recorded to the specified local file and the server's audit trail.

```
* Stopping CertAgent service...
Stopping CertAgent Server Controller.
CertAgent Server Controller stopped.

* Backing up CertAgent program files to C:\Program
Files\CertAgent7\update\backup_v7.0.8...

* Updating CertAgent files...
<list of files>

* Updating Tomcat files...
<list of files>

* Starting CertAgent service...
Starting up CertAgent Server Controller.
CertAgent Server Controller started.
CertAgent 7.0.9 update completed.

Result saved to: C:\Program Files\CertAgent7\update\update-7.0.9-
<YYYY.MM.DD_hh.mm.ss>.log

Please run the 'certagent.bat setpin' command to set the system PIN.
EXIT
```

2.4 Advanced Configuration

2.4.1 Customizing the Web Page Headers and Footers

Custom headers and footers can be added to the system admin, account admin, and public sites by adding appropriate Java system properties to the Tomcat's start up script.

Property Name	Value
isc.ca.admin.header	complete pathname of HTML file; if specified, the file is inserted into the header of all pages on the system admin site
isc.ca.admin.footer	complete pathname of HTML file; if specified, the file is inserted into the footer of all pages on the system admin site
isc.ca.account.header	complete pathname of HTML file; if specified, the file is inserted into the header of all pages on the account admin site
isc.ca.account.footer	complete pathname of HTML file; if specified, the file is inserted into the footer of all pages on the account admin site
isc.ca.public.header	complete pathname of HTML file; if specified, the file is inserted into the header of all pages on the public site
isc.ca.public.footer	complete pathname of HTML file; if specified, the file is inserted into the footer of all pages on the public site

To customize the header and footer files:

1. Create a new HTML file in an editor (e.g., `<ca home>/public-header.html`).
2. Add the desired HTML and CSS code.

For example, if you want a centered heading on a lime green background that looks as follows,



(UNCLASSIFIED//FOUO)

insert the following HTML into your file:

```
<div style="background-color:limegreen"><center>(UNCLASSIFIED//FOUO)</center></div>
```

3. Save your changes and exit the editor.
4. Repeat these steps as desired to create headers and footers for each of the types of pages on the website.
5. Add the following highlighted text into Tomcat's startup script as appropriate for your installation.

On UNIX: `<ca home>/tomcat.sh`:

```
<...>
# CUSTOM_OPT=
ADMIN_HEADER=$CA_HOME/admin-header.html
ADMIN_FOOTER=$CA_HOME/admin-footer.html
ACCT_HEADER=$CA_HOME/ca-header.html
ACCT_FOOTER=
PUBLIC_HEADER=$CA_HOME/public-header.html
PUBLIC_FOOTER=
CUSTOM_OPT=-Disc.ca.admin.header=$ADMIN_HEADER -
Disc.ca.admin.footer=$ADMIN_FOOTER -Disc.ca.account.header=$ACCT_HEADER -
Disc.ca.account.footer=$ACCT_FOOTER -Disc.ca.public.header=$PUBLIC_HEADER -
Disc.ca.public.footer=$PUBLIC_FOOTER
<...>
```

On Windows: <ca home>\tomcat.bat:

```
<...>
@REM set CUSTOM_OPT=
@REM custom header and footer settings
set ADMIN_HEADER=%CA_HOME%\admin-header.html
set ADMIN_FOOTER=%CA_HOME%\admin-footer.html
set ACCT_HEADER=%CA_HOME%\ca-header.html
set ACCT_FOOTER=
set PUBLIC_HEADER=%CA_HOME%\public-header.html
set PUBLIC_FOOTER=
CUSTOM_OPT=-Disc.ca.admin.header="%ADMIN_HEADER%" -
Disc.ca.admin.footer="%ADMIN_FOOTER%" -Disc.ca.account.header="%ACCT_HEADER%"
-Disc.ca.account.footer="%ACCT_FOOTER%" -
Disc.ca.public.header="%PUBLIC_HEADER%" -
Disc.ca.public.footer="%PUBLIC_FOOTER%"
<...>
```

6. Modify the ADMIN_HEADER, ADMIN_FOOTER, ACCT_HEADER, ACCT_FOOTER, PUBLIC_HEADER and PUBLIC_FOOTER values as appropriate for your installation.
7. Save your changes and restart Tomcat.

2.4.2 Configuring the Session Time-out

The default session time-out value for administrative and CA logins is 30 minutes. To change this value, append the following option to the CATALINA_OPTS variable to the Tomcat's startup script (<ca home>/tomcat.sh or <ca home>\tomcat.bat):

```
-Disc.ca.web.session.timeout=<time-out value in minutes>
```

2.4.3 Disabling NIAP Conformance Setting on Startup

If NIAP conformance options were enabled and one of the following situations occurs:

- integrity verification failed, or
- none of the administrators can login to the Admin site due to path validation failure;

the administrator should disable the NIAP conformance options on startup and correct the problem.

To disable the NIAP conformance option:

1. Shut down the CertAgent service and restart it in maintenance mode:

On UNIX, run the following commands:

```
<ca home>/certagent.sh stop  
<ca home>/certagent.sh start-maintenance  
<ca home>/certagent.sh setpin
```

On Windows, open Services program, select “CertAgent Server Controller”, click the stop button.

If HyperSQL database is used, start its service manually:

```
<ca home>\hsqldb.bat start
```

Then, run the following commands:

```
<ca home>\certagent.bat start-maintenance  
<ca home>\certagent.bat setpin
```

2. Once the problem has been corrected, stop the service manually and restart CertAgent normally:

On UNIX, run the following commands:

```
<ca home>/certagent.sh stop  
<ca home>/certagent.sh start  
<ca home>/certagent.sh setpin
```

On Windows, run the following command:

```
<ca home>\certagent.bat stop
```

If HyperSQL database is used, stop its service manually:

```
<ca home>\hsqldb.bat stop
```

Open Services program, select “CertAgent Server Controller”, click the start button.

Then, run the following command to set the system PIN:

```
<ca home>\certagent.bat setpin
```

2.4.4 Replacing Administrator Credentials

The administrator credentials generated by the installer should be considered temporary and only used to facilitate initial system setup. Once CertAgent is configured with operational CA accounts, these credentials should be replaced with properly issued credentials before making the system operational.

2.4.4.1 *Configuring a Profile to Issue Client Authentication Certificates*

The following extensions are required for client authentication certificates:

- (Critical) Basic Constraints: end entity
- (Critical) Key Usage: digital signature
- Extended Key Usage: client authentication

Before issuing a client authentication certificate, a profile in the desired CA account must be created and configured with the above extensions. For details on creating profiles and managing certificate extensions, see the Certificate Authority Guide (ca_cahelp.pdf) or click the Help menu item from the CA Account site.

2.4.4.2 *Generating New Credential*

To generate a new credential using Internet Explorer:

1. Launch Internet Explorer, and go to the public site, main page.
2. Select the desired CA account and select **Enroll**.
3. Complete the form. Make sure 'Mark keys as exportable' option is checked.
4. Click **Submit**.
5. Your new key pair will be generated locally and your signed certificate request will be submitted to the selected CA account. If this process is successful, a results page will be displayed.
6. Once your certificate has been issued by a CA account's operations staff, go to the Retrieval page of the public site, and enter the request ID.
7. Your certificate properties and retrieval links will be displayed. Click '**Download the trust anchor for this certificate path**' link, and save it to a file (e.g., <ca home>/keystore/newcaroot.der).
8. Import this root certificate into your IE's trusted root certification authority store.
9. Click '**Install this certificate path into CAPI/CNG**' link. Click **Yes** to confirm.
10. Open your Internet Explorer certificate store, Personal tab and select your certificate.
11. Click **Export** to export your certificate and private key. Be sure to select 'Yes, export the private key' option and 'include all certificates in the certificate path if possible' option, enter a new password (e.g., <new key password>) for the PKCS#12 file. Save your credentials to a PKCS#12 file (e.g., <ca home>/keystore/newkey.pfx).

12. For an administrator certificate, click Export again to export it to a certificate file. Be sure to select 'No, do not export the private key' and 'DER encoded binary X.509 (.CER) options. Save the certificate to a file (e.g., <ca home>/keystore/newkey.cer).

To generate a new credential using openssl:

1. Generate a key pair and a certificate request:

- a. Run the following command to generate a RSA-3072 key pair. Private key and request will be saved to admin.key and admin.csr respectively. The request will be in binary format and contain the subject "CN=Admin". Replace the output filename and subject as appropriate for your need.

```
openssl req -newkey rsa:3072 -sha384 -keyout admin.key -outform DER -out  
admin.csr -subj "/CN=Admin"
```

- b. Enter the password to protect the private key when prompted.

2. Submit the request to a CA account:

- a. Launch your browser, and go to the public site, main page.
- b. Select the desired CA account and select **Upload**.
- c. Select the appropriate profile, browse to the certificate request file (e.g., admin.csr), and click **Submit**.
- d. If this process is successful, a results page will be displayed. Please make a note of your request ID as you may need it to retrieve your certificate once it has been issued.

3. Retrieve the certificate and its chain:

- a. Once your certificate has been issued by a CA account's operations staff, go to the Retrieval page of the public site, and enter the request ID.
- b. Your certificate properties and retrieval links will be displayed. Click '**Download this certificate path to a local binary PKCS#7 (.p7b) file**' link, and save it to a file (e.g., admin.p7b).

4. Create a PKCS12 file:

- a. Run the following commands to convert the downloaded PKCS#7 file and create a PKCS#12 file:

```
openssl pkcs7 -print_certs -inform DER -outform PEM -in admin.p7b -out  
admin.p7b.pem  
  
openssl pkcs12 -export -out admin.p12 -inkey admin.key -in admin.p7b.pem
```

- b. Enter the private key password and export password when prompted.

5. Import your new credential (e.g., admin.p12) into your browser.

2.4.4.3 Replacing the Credential

Once you have generated a new administrator credential, the root certificate of the new administrator certificate must be imported to the server's trust keystore and the administrator certificate must be imported to an appropriate ACL. For NIAP compliant CertAgent, the root certificate and each issuer's CRL must be imported to CertAgent's trust anchor list and CRL store respectively.

2.4.4.3.1 Importing Root Certificate into Tomcat's Trust Keystore

To import the root certificate of the new administrator's certificate into Tomcat's trust keystore:

1. Open the Tomcat configuration file in an editor:

```
<ca home>/tomcat/conf/server.xml
```

2. Locate the trust keystore file and its password from the truststoreFile (e.g., /usr/local/certagent7/keystore/ca-root.ks) and truststorePass attributes in the Connector element.

3. Run the following command to import the root certificate into the trust keystore file:

```
keytool -importcert -alias <alias> -keystore <truststoreFile> -storepass  
<truststorePass> -storetype JKS -file <root cert path>
```

4. On Windows, restart the CertAgent service. On UNIX, restart the Tomcat service.

2.4.4.3.2 Importing Root Certificate and CRL into CertAgent's Trust Anchor List and CRL Store

If 'NIAP conformance: Enable strict certificate and path validations' option is enabled, the root certificate of the newly issued administrator certificate and its CRL must be imported into CertAgent's trust anchor list and CRL store.

For details, see the section entitled *Certificate and Path Validations* in the CertAgent Administrator Guide (ca_adminhelp.pdf).

2.4.4.3.3 Importing Administrator Certificate into ACL

Once you have generated a new administrator credential, you can add the new administrator certificate to the desired ACLs.

For details, see the sections entitled *Managing the Server Administration Access Control List* and *Managing an Existing CA Account* in the CertAgent Administrator Guide (ca_adminhelp.pdf).

2.4.5 Replacing TLS Credentials

The TLS and administrator credentials generated by the installer should be considered temporary and only used to facilitate initial system setup. Once CertAgent is configured with operational CA accounts, these credentials should be replaced with properly issued credentials before making the system operational.

For NIAP compliant CertAgent, HSM-based TLS credentials must be used. Follow the sections below to generate the HSM- or Software-based SSL credentials as appropriate for your installation.

2.4.5.1 HSM-based Credentials

2.4.5.1.1 Generating a Key Pair and a Certificate Request

For Acala, run the following command to generate a key pair with the specified key type `<type>` (e.g., RSA3072, ECCp384, or ECCp521) and output a PKCS#10 certificate request `<filename>.p10` with the distinguished name `<dn>` including a common name that is the IP address, FQDN, or resolvable hostname of your website.

```
acala_cli --gen-p10 <dn> --gen-key-type <type> --filename <filename>
```

For example:

```
acala_cli --gen-p10 "CN=ca.infoseccorp.com, O=ISC, C=US" --gen-key-type RSA3072 --  
filename newtls
```

For other HSM, please consult the HSM vendor documentation.

2.4.5.1.2 Submitting the Certificate Request

To submit the certificate request to a CA account:

1. Launch Internet Explorer and go to the public site main page.
2. Select the desired CA account and select **Upload**.
3. Click **Browse...** to select the PKCS#10 certificate request file you wish to upload.
4. Complete the rest of the form and click **Submit**.

Please make note of your request ID as you will need it to retrieve your certificate once it has been issued.

To retrieve your certificate:

1. Launch Internet Explorer and go to the public site main page.
2. Select the desired CA account and select **Retrieve**.
3. Enter the request ID you received when you submitted your certificate request and click **Retrieve**.
4. If your certificate has been issued, click 'Download this certificate path to a local binary PKCS#7 (.p7b) file' link.

2.4.5.1.3 Importing the Issued Certificate and Chain

For Acala, run the following command to import the issued certificate and chain from a PKCS#7 file:

```
acala_cli --import --filename <PKCS#7 file path>
```

For example:

```
acala_cli --import --filename newtls.p7b
```

For other HSM, please consult the HSM vendor documentation.

NOTE: If your TLS certificate is issued by an intermediate CA, all intermediate CA certificates and optionally the root certificate must be imported into the HSM. During the TLS handshake, Tomcat will send the TLS certificate and its chain to the client for certificate validation.

2.4.5.1.4 Updating the Configuration File

If the subject DN of the TLS certificate is not the same as the current one, update the CertAgent configuration to use the new TLS credential.

1. Open the following CertAgent configuration file in an editor:

```
<ca home>/acalashim/acalashim.xml
```

2. Locate the HSM_CERT_FILTER value and replace it with the subject DN of the new SSL certificate.

```
...
HSM_CERT_FILTER=CN=192.168.0.82, O=ISC, C=US
...
```

NOTE: If the DN contains a state or province name component, use ST (e.g. ST=IL) instead of S (e.g. S=IL).

3. Save the file and close your editor.

If the subject DN of the new TLS certificate is the same as the current one, current TLS credential must be deleted. Consult the HSM vendor documentation to delete an existing credential.

Restart the CertAgent service.

2.4.5.2 Software-based Credentials

To generate new software-based SSL credential, follow the steps in section 2.4.4.2 *Generating New Credential* but set the common name (CN) to the CertAgent system's host name or IP address.

Once you have generated a new SSL credential (e.g., `newssl.pfx`), you can replace the existing SSL credential with the new one.

1. Open the Tomcat configuration file in an editor:

```
<ca home>/tomcat/conf/server.xml
```

2. Locate the `keystoreFile`, and `keystorePass` attributes in two of the `Connector` elements and replace them with the new values.

```
<Connector port="8443" ...  
keystoreFile="/usr/local/certagent7/keystore/ca-ssl.p12"  
keystorePass="Password123456!"  
... />  
  
<Connector port="443" ...  
keystoreFile="/usr/local/certagent7/keystore/ca-ssl.p12"  
keystorePass="Password123456!"  
... />
```

3. Save the `server.xml` file and close your editor.
4. On Windows, restart the CertAgent service. On UNIX, restart the Tomcat service.

2.4.6 High Availability and Load Balancing

2.4.6.1 Overview

CertAgent high availability architecture is designed around leveraging a centralized, highly available database (Oracle, PostgreSQL) and highly available HSMs. In this architecture, most data is stored and shared among the CertAgent instances in the cluster through the database. Naturally, basic configuration information such as the database connection and local admin ACL are stored local to each CertAgent server instance.

Depending on the HSM vendor, their high availability solution varies. At the minimum, a high availability configuration may be obtained by replicating the HSM using the vendor's backup and restore procedures and then configuring the various CertAgent instances to use separate HSMs.

In the event that a particular component fails, the CertAgent instance relying on that component will be unable to issue certificates or CRLs until such time as the failure is resolved. Other CertAgent instances will continue to operate without issue.

In the case of a database node failure, there is no loss of functionality as long as the database connections are capable of automated failover.

In the case of an HSM failure where the HSM vendor does not support automated failover, the CertAgent instance will be unable to issue certificates or CRLs until the HSM failure is resolved. In this situation, the failing CertAgent instance is capable of emailing notifications to administrators alerting them of the issue.

Depending on the geographic distribution of the components and the HSM's ability to support high availability, there are two options to build the high availability system. **Figure 1** below shows the scenario where all components are centrally located and the HSM supports high availability with automatic failover. **Figure 2** shows the scenario where the HSMs do not support automatic failover or the components are geographically distributed.

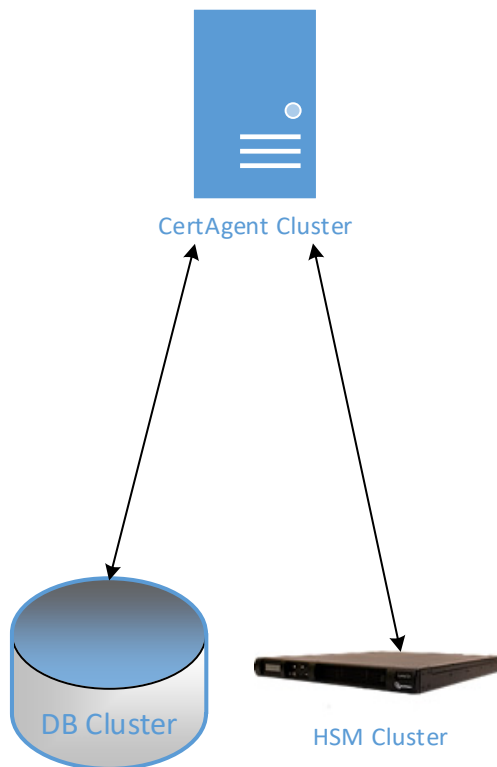


Figure 1

CertAgent with High Availability HSM

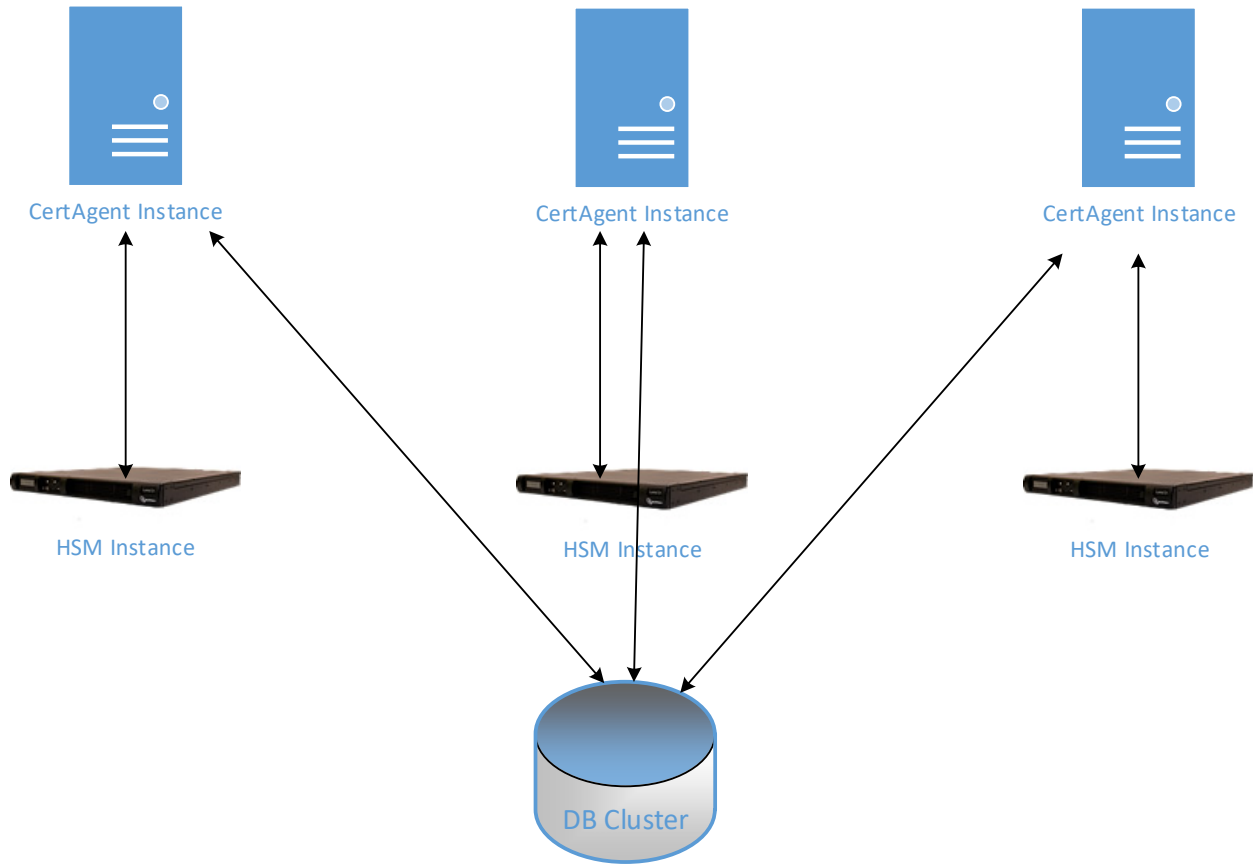


Figure 2

CertAgent Individual HSM Instances

In either case, due to database requirements, all CertAgent instances are required to communicate with a single database cluster. In most cases, this cluster is in a single geographic location due to the need of the database to maintain its integrity with minimal performance degradation.

2.4.6.2 Per-instance Data

The following items are stored local to each CertAgent instance:

- certagent.xml and DTD file
 - HSM library, HSM label, and the location of the system credential
 - Database info (driver, URL, user name, encrypted password, connection pool configuration)
- ACL for admins who can manage the system credentials and configure the database access information
- history of the system certificate (NOTE: the current system certificate is stored in the database)
- server log file that records server startup and shut down events
- admin log that records events before database is available

2.4.6.3 *Serialization*

Within an X.509 PKI there is a requirement that the serial number for each certificate be unique. There are two ways to guarantee this: maintain a monotonic counter or randomly generate serial numbers of sufficient size to statistically guarantee uniqueness. CertAgent uses a monotonic counter managed by the underlying database by using an SQL Sequence to generate the serial number which is the primary key of the table. As long as all instances of CertAgent use the same database, serial numbers are guaranteed to be unique.

2.4.6.4 *Task Management*

CertAgent has maintenance tasks that run periodically. In order to support this in a high availability or load balanced manner, CertAgent maintains a task queue. Each node will repeatedly remove a task from the queue, execute it, and then mark it complete.

2.4.6.5 *Shared Credentials*

In order for a CertAgent cluster to function, the system credential must be identical on all systems. This allows all instances of CertAgent in the cluster to decrypt configuration information stored in the database.

2.4.6.6 *Installing a CertAgent Cluster*

Assuming that CertAgent has been successfully deployed on the first host (Server A), you may configure it for high availability and load balancing as follows:

2.4.6.6.1 *UNIX*

1. Deploy the same JDK/JRE, JDBC driver or Instant Client, and same HSM library to one or more additional servers in the same location as the first host.
2. Then on each new server:
 - a. Obtain a copy from Server A of each of the following and install them on the new server:
 - the CertAgent home directory: /usr/local/certagent7 (exclude hsqldb directory, hsqldb.sh, and isc-certagent7-hsqldb.service)
 - the CertAgent service: /etc/systemd/system/isc-certagent7.service
 - the Tomcat service if installed: /etc/systemd/system/isc-certagent7-tomcat.service

NOTE: All CertAgent clusters will use the same database cluster. If HyperSQL was installed on Server A during the installation, the hsqldb directory, hsqldb.sh, and isc-certagent7-hsqldb files should not be copied to the new server.

- b. Install CertAgent and Tomcat services with the following commands:

```
systemctl daemon-reload
systemctl enable isc-certagent7 -now
systemctl start isc-certagent7
systemctl enable isc-certagent7-tomcat -now
systemctl start isc-certagent7-tomcat
```

- c. Set the permissions on all the scripts and services:

```
find /usr/local/certagent7/ -name '*.sh' -exec chmod 700 {} \;
find /etc/systemd/system/ -name 'isc-certagent7*' -exec chmod 600 {} \;
```

- d. Update the `HOST` variable in the script: `<ca home>/setenv.sh`.
- e. Verify the SSL credential files. Make sure all files specified in the Connector element in the Tomcat configuration file (`<ca home>/tomcat/conf/server.xml`) exists in the new server.
- f. Synchronize HSM data files. If a Thales HSM is used, make sure the `kmdata` folder is kept in sync between servers. If Acala is used, make sure the `acala.p15` file is kept in sync between servers.

3. Configure your DNS servers to use round robin DNS to map all CertAgent host systems to a single domain name.

2.4.6.6.2 Windows

1. Deploy the same JDBC driver or Instant Client, and same HSM library to one or more additional servers in the same location as the first host.
2. If JRE was not installed by the installer in the first host, install the same JRE to one or more additional servers in the same location as the first host.
3. Then on each new server:
 - a. Run the CertAgent installer.
 - b. Stop the CertAgent service.
 - c. Obtain a copy from Server A of each of the following and replace the existing ones on the new server with them:
 - the CertAgent conf directory: `C:\Program Files\CertAgent7\conf`
 - the CertAgent keystore directory: `C:\Program Files\CertAgent7\keystore`
 - d. Update the `HOST` variable in the script: `C:\Program Files\CertAgent7\setenv.bat`.
 - e. Verify the SSL credential files. Make sure all files specified in the Connector element in the Tomcat configuration file (`C:\Program Files\CertAgent7\tomcat\conf\server.xml`) exists in the new server.
 - f. Synchronize HSM data files. If a Thales HSM is used, make sure the `kmdata` folder is kept in sync between servers. If Acala is used, make sure the `acala.p15` file is kept in sync between servers.

4. Configure your DNS servers to use round robin DNS to map all CertAgent host systems to a single domain name.

2.4.6.7 Maintaining a CertAgent Cluster

1. When the servers start up, successively log into each CertAgent host (using actual IP addresses rather than common domain name) as a local administrator and enter the common system PIN.
2. Administrative tasks (such as issuing certificates and viewing audit trails) may be performed on any server in the cluster and changes will automatically propagate to the other servers.

NOTE: Updates to system credentials or local admin ACLs must be performed by an authorized local administrator who logs in to each server directly using its IP address and manually applying the desired changes.

3. All CertAgent hosts in the cluster must use the same system credentials. If the system credentials are changed on one of the hosts, a local admin must update them on all other hosts.

2.4.7 Configuring HTTP Port

By default, CRLs and OCSP responses can be retrieved via the public HTTPS port. If you also want to open an HTTP port for OCSP and CRL retrieval but did not configure it during installation, follow the steps below to enable the port manually.

1. Open the Tomcat configuration file in an editor:

```
<ca home>/tomcat/conf/server.xml
```

2. Locate the `<Service name="Catalina">` element and insert the highlighted line to open an HTTP port (e.g., 80).

```
<Service name="Catalina">
  <Connector port="80"/>
```

3. Save the server.xml file and close your editor.
4. Open the CertAgent configuration file in an editor:

```
<ca home>/setenv.sh
```

(UNIX)

```
<ca home>\setenv.bat
```

(Windows)

5. Set the HTTP port to the HTTP_PORT variable.

HTTP_PORT=80	(UNIX)
SET HTTP_PORT=80	(Windows)

6. Save the file and close your editor.
7. Restart CertAgent service.

2.4.8 Updating Java

2.4.8.1 CertAgent Version 7.0.9

This section describes how to configure CertAgent to use a newer version of Java once it has been installed in CertAgent version 7.0.9.

1. Stop the CertAgent service.
2. Open the CertAgent configuration file in an editor:

<ca home>/setenv.sh	(UNIX)
<ca home>\setenv.bat	(Windows)

3. Update the JRE_HOME variable to the new Java installation directory in the highlighted text below:

JRE_HOME=/usr/java/jre1.8.0_251; export JRE_HOME	(UNIX)
SET JRE_HOME=C:\Program Files\Java\jre1.8.0_251	(Windows)

4. If you are updating the Java from version 8 to 11 or above and CertAgent is installed with the NIAP-compliant option, update the JAVA_SECURITY variable in the highlighted text below:

JAVA_SECURITY=\$CA_HOME/conf/ca.java.security.11	(UNIX)
SET JAVA_SECURITY=%CA_HOME%\conf\ca.java.security.11	(Windows)

5. Save the changes.
6. Restart CertAgent service.

2.4.8.2 CertAgent Version 7.0.8

This section describes how to configure CertAgent to use a newer version of Java 8 once it has been installed in CertAgent version 7.0.8.

1. Stop the CertAgent service.
2. Update the JRE_HOME variable to the new Java installation directory in the highlighted text below in <ca home>/setenv.sh:


```
JRE_HOME=/usr/java/jre1.8.0_251; export JRE_HOME
```

or <ca home>\setenv.bat:

```
SET JRE_HOME=C:\Program Files\Java\jre1.8.0_251
```

NOTE: If CertAgent has been installed with the NIAP-compliant option, specify the installation directory of Java 8 only. Otherwise, specify the installation directory of Java 8, 11, or above.

3. Save the changes.
4. Restart CertAgent service.

2.4.8.3 CertAgent Version 7.0.7 or Below

This section describes how to configure CertAgent to use a newer version of Java 8 once it has been installed in CertAgent version 7.0.7 or below.

1. Stop the CertAgent service.
2. Update the Java installation directory in the variables and files as described below:

OS	File	Variable	Change
UNIX	certagent.sh hsqldb.sh /tools/reportgenerator.sh /tools/cacli/cacli.sh /update/update-tool.sh	JAVA_PATH	From <JAVA HOME>/bin/java to <NEW JAVA HOME>/bin/java
	tomcat.sh	JAVA_HOME	From <JAVA HOME> to <NEW JAVA HOME>
Windows	certagent.bat hsqldb.bat \\tools\\reportgenerator.bat \\tools\\cacli\\cacli.bat \\update\\update-tool.bat	JAVA_PATH	From <JAVA HOME>\\bin\\java to <NEW JAVA HOME >\\bin\\java
	tomcat.bat	JRE_HOME	From <JAVA HOME> to <NEW JAVA HOME >

3. If CertAgent has been installed with the NIAP-compliant option, copy the security properties file `java.security` from the existing <JAVA HOME>/lib/security directory to the <NEW JAVA HOME>/lib/security directory.

If the new Java version is Oracle JDK/JRE 8u152 or below, copy the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy files (`local_policy.jar` and

US_export_policy.jar) from the existing <JAVA_HOME>/lib/security directory to the <NEW JAVA_HOME>/lib/security directory.

- Restart the CertAgent service.

2.4.9 Configuring EST Port

CertAgent supports the following EST operations, authentications, and URLs:

Operation	Authentication	URL
Distribution of CA certificates	None	https://<host>:<public port>/.well-known/est/[<ca>]/cacerts
Enrollment of clients	Certificate-based	https://<host>:<admin port>/.well-known/est/[<ca>]/simpleenroll
	Basic	https://<host>:<public port>/.well-known/est/[<ca>]/simpleenroll
Re-enrollment of clients	Certificate-based	https://<host>:<admin port>/.well-known/est/[<ca>]/simplereenroll
	Basic	https://<host>:<public port>/.well-known/est/[<ca>]/simplereenroll

NOTE: By default, the admin port (e.g., 8443) is used to accept EST requests with client authentication. The public port (e.g., 443) is used to accept EST requests with no or basic authentication. If your EST client supports multiple ports in above URL formats, no configuration is required.

If your EST client (e.g., libEST) supports only one port for no, basic client, and client authentications, use the following the steps to configure the public port manually:

- Open the Tomcat configuration file in an editor:

```
<ca home>/tomcat/conf/server.xml
```

- Locate the `Connect` element for the Public Site and change the highlighted value from “false” to “want”.

```
<!-- CertAgent Public Site -->
<Connector port="443" ... clientAuth="want" ... />
```

- Save the `server.xml` file and close your editor.
- Restart CertAgent service.

The TLS connection via the public port now requests a client certificate for authentication but does not fail if one is not presented. CertAgent now supports the following URL format for any EST operations and authentications:

```
https://<host>:<public port>/.well-known/est/[<ca>]/<operation>
```

When accessing the Public site, the browser prompts for a certificate. Selecting a certificate or clicking Cancel will grant access to the Public site.

2.4.10 Section 508 Compliance

CertAgent's Administrative, CA Account, and Public sites are 508 compliant. To better assist users with accessibility needs, CertAgent should be accessed using a Section 508 compatible browser and have any necessary compatible assistive technologies (e.g. screen reader) installed and configured per those products' guidance.

3 Managing the CertAgent Server

By default, the CertAgent service starts and stops automatically upon system start-up and shut-down. For NIAP-compliant CertAgent, Tomcat starts automatically upon system PIN entry and stops upon CertAgent service shut-down rather than upon system start-up and shut-down. If HyperSQL database server was installed as part of the CertAgent package, the corresponding service starts and stops automatically as well.

3.1 Using the System Services

On UNIX, run the following commands to manually start or stop the CertAgent, Tomcat, HyperSQL database server services:

```
systemctl [start | stop] isc-certagent7
systemctl [start | stop] isc-certagent7-tomcat
systemctl [start | stop] isc-certagent7-hsqldb
```

On Windows, open Services program, select “CertAgent Server Controller”, click the start or stop button to manually start or stop the CertAgent, Tomcat and HyperSQL database services. Alternatively, you can run the following commands as an administrator:

```
C:\Program Files\CertAgent7\certagent.bat [start | stop]
C:\Program Files\CertAgent7\tomcat.bat [start | stop]
C:\Program Files\CertAgent7\hsqldb.bat [start | stop]
```

NOTE: When shutting down the services, how long this takes depends heavily upon the number of CA accounts and database sizes on your system. In any case, please wait until a “CertAgent services stopped” message has been appended to the server log in <ca home>/conf/server.log and a “Stopped” message has been appended to the HyperSQL database log in <ca home>/hsqldb/hsqldb-stop.txt before restarting the services.

On UNIX, you can check the status of each service by running the following commands:

```
systemctl status isc-certagent7
systemctl status isc-certagent7-tomcat
systemctl status isc-certagent7-hsqldb
```

3.2 Using CertAgent Script

3.2.1 Starting the Server

```
<ca home>/certagent.sh start (UNIX)
<ca home>\certagent.bat start (Windows)
```

3.2.2 Starting the Server in SSL Debug Mode

If the server is started in this mode, SSL debug messages will be saved to `<ca home>/tomcat/logs/catalina.<YYYY-MM-DD>.log` file.

```
<ca home>/certagent.sh start-ssldebug (UNIX)
<ca home>\certagent.bat start-ssldebug (Windows)
```

3.2.3 Starting the Server in Maintenance Mode

When fatal error occurred (e.g., failure of integrity is detected), CertAgent will display an error message and shut itself down in an orderly manner. An administrator must start CertAgent in maintenance mode by running the following commands:

```
<ca home>/certagent.sh start-maintenance (UNIX)
<ca home>\certagent.bat start-maintenance (Windows)
```

When CertAgent is running in a maintenance mode, only the Admin Site will be accessible by authorized users and the following NIAP conformance options will be disabled:

- Enable data integrity on the Trust Anchor list
- Enable data integrity on ACLs
- Run integrity tests on server startup
- Enable strict certificate and path validations
- Enable restrictions on security roles

An administrator should login to the Admin Site to resolve the issue. Once the issue has been fixed, an administrator should enable all the NIAP conformance options and log out. Then, restart the service.

3.2.4 Stopping the Server

```
<ca home>/certagent.sh stop (UNIX)
<ca home>\certagent.bat stop (Windows)
```

3.2.5 Checking the Server Status

```
<ca home>/certagent.sh status (UNIX)
<ca home>\certagent.bat status (Windows)
```

3.2.6 Viewing the Server Log

```
<ca home>/certagent.sh log (UNIX)
<ca home>\certagent.bat log (Windows)
```

3.2.7 Viewing the Version Number

```
<ca home>/certagent.sh version (UNIX)
<ca home>\certagent.bat version (Windows)
```

4 Setting the System PIN

Sensitive data (database password, email password, LDAP password, and HSM PIN) are encrypted with the system certificate and stored in the database and configuration file. An administrator must enter the PIN of the HSM in which the system credential resided on each time the system is booted via the PIN entry page, PIN entry API, or CertAgent script. The PIN entry page and API must be access from the localhost or one or more additional authorized IP addresses via HTTPS without client authentication.

4.1 Using the CertAgent Script

For NIAP-compliant CertAgent, system PIN must be entered via the CertAgent script.

To have the script prompt for the system PIN upon start-up, run the following command:

```
certagent.sh setpin (UNIX)
certagent.bat setpin (Windows)
```

If a dialog box prompt is preferred, enable the '-gui' option in the setpin section of the script:

```
$JAVA_PATH $CA_OPTS -classpath $CLS_PATH com.infoseccorp.ca.CertAgent setpin
-gui (UNIX)

"%JAVA_PATH%" %CA_OPTS% -classpath "%CLSPATH%" com.infoseccorp.ca.CertAgent setpin
-gui (Windows)
```

If your HSM uses an external authentication method (e.g., PED), append the '-extauth' option to the setpin command:

```
$JAVA_PATH $CA_OPTS -classpath $CLS_PATH com.infoseccorp.ca.CertAgent setpin
-extauth (UNIX)

"%JAVA_PATH%" %CA_OPTS% -classpath "%CLSPATH%" com.infoseccorp.ca.CertAgent setpin
-extauth (Windows)
```

4.2 Using the PIN Entry Page²

The System PIN Entry site is secured using HTTPS without client authentication. Access to this site is limited to localhost and one or more additional authorized IP addresses.

To access the PIN Entry page, launch Internet Explorer (or any other browser) and enter the following URL in its address bar:

```
https://<host>:<public port>/certagentadmin/admin/pin.jsp
```

² Non-NIAP-compliant CertAgent only

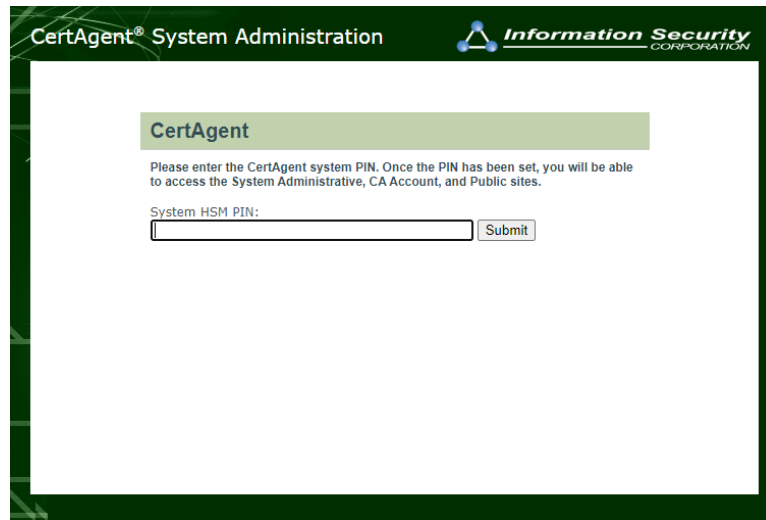
Be sure to replace `<host>` and `<public port>` with the appropriate system IP address or 127.0.0.1 and public site access port (default: 443) of your CertAgent webserver.

For Internet Explorer users, if the warning message 'There is a problem with this website's security certificate.' appears, ignore this warning and click '**Continue to this website (not recommended)**'.

For Internet Explorer users on Windows Server 2012, if the dialog with message 'Content from the website listed below is being blocked by the Internet Explorer Enhanced Security Configuration. https://127.0.0.1' appears, click **Add...** . Then, click **Add** again to add https://127.0.0.1 to the zone and **Close** to close the dialog.

For Firefox users, if the warning message 'Your connection is not secure' appears, click **Advanced** and click **Add Exception....** In the Add Security Exception dialog, click **Confirm Security Exception**.

The following page will appear:



Enter the HSM PIN and click **Submit**.

4.2.1 Configuring the Authorized IP Addresses

By default, the authorized IP addresses configured by the installer are: 127.0.0.1, 0:0:0:0:0:0:0:1, and the IP address of the CertAgent system which allow the administrator to access the PIN entry page from the CertAgent system using the one of the following URLs:

```
https://127.0.0.1:<public port>/certagentadmin/admin/pin.jsp
```

```
https://localhost:<public port>/certagentadmin/admin/pin.jsp
```

```
https://<server IP>:<public port>/certagentadmin/admin/pin.jsp
```

To update the list of authorized IP addresses:

1. Open the CertAgent Configuration file in an editor:

```
<ca home>/conf/certagent.xml
```

2. Update the attribute addresses value in the pin-entry element with a comma delimited list of authorized IP addresses.

```
<...>  
<pin-entry addresses="127.0.0.1,0:0:0:0:0:0:0:1,192.168.0.81"/>  
<...>
```

3. Save the `certagent.xml` file and close your editor.
4. Restart the CertAgent service.

4.3 Using the PIN Entry API³

The PIN Entry API:

URL: `https://<hostname>:<public port>/certagentadmin/admin/SetSysPIN`

Allows HSM PIN entry over a TLS-secured connection (without client authentication) from the localhost and one or more additional authorized IP addresses.

To establish a TLS connection with the CertAgent PIN entry API, the client must trust the trust anchor for the server's TLS certificate (*i.e.*, the root certificate for path validation of the server's SSL credentials). For Java application, the client's Java trust keystore must contain a trust anchor for the server's TLS certificate.

To enter the HSM PIN, POST '`pin=<HSM PIN>`' to the API. If your HSM PIN contains any special characters, be sure to encode it using HTML URL encoding. For Java application, use `URLEncoder.encode(<HSM PIN>, "UTF-8")` method to encode your HSM PIN.

After POSTing its request, the client process should read the HTTP return code and response written to the output stream of the open connection by the server. The following are the possible responses:

HTTP Code	Response and Description
200	The PIN has been set successfully; "System PIN set successfully" will be returned in the response
400	Failed to set the system PIN; "Cannot set system PIN. <error>" will be returned in the response
405	Failed to set the system PIN because method GET was used; "HTTP method GET is not supported; Please use POST" will be returned in the response

³ Non-NIAP-compliant CertAgent only

5 Using the CertAgent Web Interface

CertAgent's web-based administrative interface may be accessed by authorized users.

A CA account (ca7) and three profiles (ca7clientauth, ca7tls and ca7ocsp) are created during the installation. An initial (temporary) set of role certificates generated by the installer is automatically added to the ACLs during installation.

The following table describes the available roles, their credential file name, and the ACLs in which their certificates are located.

Role	Credential	ACL
Administrator	ca-admin.p12	Admin Site and CA Account (ca7)
Auditor	ca-audit.p12	Admin Site and CA Account (ca7)
CA Operations Staff	ca-operations-staff.p12	CA Account (ca7)

You should import these temporary credentials (<ca home>/keystore/*.p12 with password <p12 pass>) into your web browser's certificate store in order to gain access to the CertAgent sites.

5.1 Importing Administrator Credentials into Browsers

If you selected the NIAP compliance option during installation, AES-256 will be used to encrypt your private key. The PKCS#12 files generated by the installer can only be imported to compatible browsers (e.g., Firefox 56+).

5.1.1 Firefox

To import the administrator's credentials into Firefox:

1. Click the **Menu** button and select **Preferences** on UNIX or **Options** on Windows.
2. From the left-side menu, select **Privacy & Security**.
3. Click **View Certificates**.
4. In the Certificate Manager dialog, select the 'Your Certificates' tab and click **Import**.
5. Browse to the PKCS#12 file (e.g., <ca home>/keystore/ca-admin.p12) and click **Open**.
6. Enter the password that was used to encrypt the private key and click **OK**.
7. Firefox will alert you when the certificate has been installed successfully.
8. Select the 'Authorities' tab, select the root certificate (e.g., CertAgent <version> Root CA XXXX) which listed under the organization you have entered during the installation.
9. Click **Edit Trust**, click both checkboxes in the Edit CA certificate trust settings dialog and click **OK**.
10. Click **OK** to close the Certificate Manager dialog.

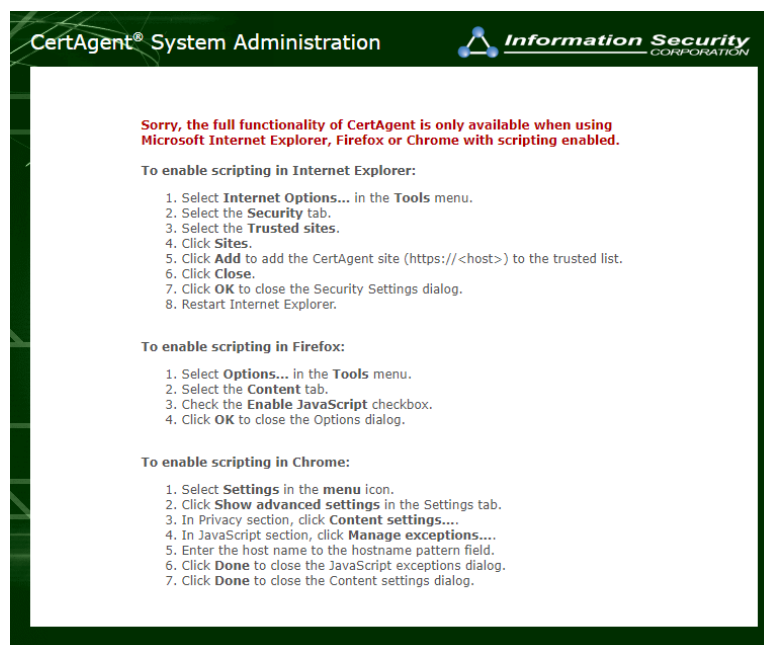
5.1.2 Internet Explorer

To import the administrator's credentials into Internet Explorer 11:

1. Select the Tools, Internet Options from the menu bar.
2. Select the 'Content' tab and click **Certificates**.
3. Select the 'Personal' tab and click **Import**.
4. In the Certificate Import Wizard:
 - a. Click **Next**.
 - b. Click **Browse..**, locate the PKCS#12 file (e.g., <ca home>/keystore/ca-admin.der) and click **Open**.
 - c. Click **Next**, enter the password that was used to encrypt the private key and click **Next**.
 - d. Select 'Automatically select the certificate store based on the type of certificate' option, browse the store to 'Personal' and click **Next**. Then, click **Finish**.
 - e. When the Security Warning dialog appears with the Root CA information (e.g., CertAgent 7.0.8 Root CA), click **Yes** to trust this certificate.
 - f. It will alert you when the certificate has been installed successfully.

5.2 Enable Scripting in Browsers

The full functionality of CertAgent is only available when using a browser with scripting enabled. If script is disabled in your browser, the following dialog will appear when accessing the CertAgent site:



To enable scripting in Internet Explorer:

1. Select **Internet Options...** in the Tools menu.
2. Select the Security tab.
3. Select the Trusted sites.
4. Click **Sites**.
5. Click **Add** to add the CertAgent site (https://<host>) to the trusted list.
6. Click **Close**.
7. Click **OK** to close the Security Settings dialog.
8. Restart Internet Explorer.

To enable scripting in Firefox:

1. Enter “about:config” in the URL bar.
2. Enter “javascript.enabled” in the Search bar and press ENTER.
3. Double click on the value to change it to ‘true’.

To enable scripting in Chrome:

1. Select **Settings** in the **menu** icon.
2. Click **Show advanced settings** in the Settings tab.
3. In Privacy section, click **Content settings....**
4. In JavaScript section, click **Manage exceptions....**
5. Enter the host name to the hostname pattern field.
6. Click **Done** to close the JavaScript exceptions dialog.
7. Click **Done** to close the Content settings dialog.

5.3 Using the Administrative Site

To access the CertAgent system administration login page, launch Internet Explorer and enter the following URL into its address bar:

```
https://<hostname or IP address>:<admin port>/certagentadmin/admin/login.jsp
```

Be sure to replace <hostname or IP address> and <admin port> with the appropriate system name or IP address and TLS port of your CertAgent webserver. Select your certificate (e.g., CertAgent

<version> Administrator) in the security dialog to authenticate yourself to the webserver, and then click **OK**.

For more information on server management and how to use the administrative site, please refer to the online help system:

```
https://<hostname or IP address>:<admin port>/certagentadmin/admin/help.html
```

5.4 Using the CA Account Site

To access the CertAgent account administration login page for a CA account, launch Internet Explorer and enter the following URL into its address bar:

```
https://<hostname or IP address>:<admin port>/certagentadmin/ca/login.jsp
```

Be sure to replace <hostname or IP address> and <admin port> with the appropriate system name or IP address and TLS port of your CertAgent webserver. Select your certificate (*e.g.*, CertAgent <version> Administrator) in the security dialog to authenticate yourself to the webserver, and then click **OK**.

For more information on CA account management and usage, please refer to the on-line help pages:

```
https://<hostname or IP address>:<admin port>/certagentadmin/ca/help.html
```

5.5 Using the Public Site

To access the CertAgent main public page for end-users, launch Internet Explorer (or any other browser) and enter the following URL into its address bar:

```
https://<hostname or IP address>:<public port>/certagent/main.jsp
```

Be sure to replace <hostname or IP address> and <public port> with the appropriate system name or IP address and TLS port of your CertAgent webserver.

For more information on usage of the public site, please refer to the on-line help pages:

```
https://<hostname or IP address>:<public port>/certagent/help.html
```

6 Additional Management Tools

6.1 The CertAgent Command Line Tool

All system administrative tasks can be performed via the administrative and CA web interfaces. Some of these tasks can also be managed via the supplied CertAgent command line program:

<code><ca home>/tools/caccli/caccli.sh</code>	(UNIX)
<code><ca home>\tools\caccli\caccli.bat</code>	(Windows)

6.1.1 Command Line Syntax

To display the usage of the command line tool, run the following command:

```
Syntax:
-h

Example:
#> ./caccli.sh -h
```

Usage will be displayed:

CertAgent Installation Guide

```
CertAgent 7.0.9 command line tool
Copyright(c) 1991-2020 Information Security Corp. All rights reserved.

Usage:
* Print this help, then exit
  -h

* Create CA account
  -createacct -ca <ca name> [-displayname <display name>]

* Create profile
  -createprofile -ca <ca name> -profile <profile name> [-displayname <profile
display name>]

* Assign existing credentials to a CA account
  -assign -ca <ca name> [<HSM option>] -cert <CA certificate> [-chain <CA's chain
certificates>]

* Generate key pair and assign self-signed certificate to a CA account
  -genroot -ca <ca name> -dn <dn> [<HSM option>] [-t <kypname>] [-H <hash>]
  [-y <validity>] [-file <config file>]

* Generate key pair; assign certificate request to a CA account (for manual
submission of certificate request to external CA)
  -gencrq -ca <ca name> -dn <dn> [<HSM option>] -o <request output file> [-f <output
format>] [-t <kypname>] [-H <hash>]

* Generate key pair; assign certificate request to a CA account and submit to
superior CA on same system
  -gensubcrq -ca <ca name> -dn <dn> [<HSM option>] -issuer <issuer CA account>
  [-email <email>] [-t <kypname>] [-H <hash>]

* Install certificate issued by a superior CA on the same system
  -install -ca <ca name>

* Install certificate issued by an external CA
  -installext -ca <ca name> -cert <issued certificate with chain>

* Display configuration settings for a CA account
  -showconf -ca <ca name> [-credential] [-rami] [-enrollment] [-est]
  [-certprop] [-certext] [-crqfilter] [-revocationpolicy] [-crl] [-ocsp]
  [-ldap] [-mail] [-public]

* Display configuration settings for a profile
  -showconf -profile <profile name> [-rami] [-enrollment] [-certprop]
  [-certext] [-crqfilter] [-mail]

* Update preferences for an account or profile
  -updateacct (-ca <ca name> | -profile <profile name>) -file <config file>

* Display all CA accounts, or all profiles of a specified CA account
  -showacct [-ca <ca name>]

* Import CA account and its profiles configurations
  -importacct -ca <ca name> -file <in file>

* Export CA account and its profiles configurations to a file
  -exportacct -ca <ca name> -file <out file>

* Display the slots and labels on an HSM
  -showslots -hsmlib <library>

* Display the types of keys that can be generated
```

```

-showkeytypes [<HSM options>]

* Display the types of hash function available for specified key type and size
-showhash [-t <kyptype>]

* Display the ACL for a CA account, profile, or Admin Site
-showacl (-ca <ca name> | -profile <profile name> | -admin)
[-acl <permissions>]

* Add a certificate to the ACL for a CA account, profile, or Admin Site
-addacl (-ca <ca name> | -profile <profile name> | -admin) -acl <permission>
-cert <cert file>

* Update the permissions of the ACL for a CA account, profile, or Admin Site (lists
certs and prompts for id to update)
-updateacl (-ca <ca name> | -profile <profile name> | -admin)

* Remove a certificate from the ACL for a CA account, profile, or Admin Site (lists
certs and prompts for id to remove)
-removeacl (-ca <ca name> | -profile <profile name> | -admin)

* Delete a particular profile
-deleteacct -profile <profile name>

* Enable a disabled CA account
-enableacct -ca <ca name>

* Disable an active CA account
-disableacct -ca <ca name>

* Import issued CRL to a CA account
-importcrl -ca <ca name> -file <crl file>

* Import issued certificates to a CA account
-importcert -ca <ca name> -file <cert file or directory contains cert files>

* Export certificates match the specified not before date range
-exportcert -ca <ca name> -o <output directory> -date
<notBeforeDate MM/dd/yyyy> [<notAfterDate MM/dd/yyyy>]
[-fn <filename format>]

* Submit a certificate request
-submit (-ca <ca name> | -profile <profile name>) -file <request file>
[-email <email>]

* Display trust anchors
-showtrust

* Add a trust anchor
-addtrust -file <cert file>

* Remove a trust anchor
-removetrust

* Display CRLs for path validation
-showcrl

* Add a CRL for path validation
-addcrl -file <crl file>

* Remove a CRL for path validation
-removecrl

```

```

* Display system information
-getinfo

* Display unique subject DN statistics
-statistics

HSM/PKCS#11 options (if -L is used, either -l or -s must be present):
-L, -hsmlib <lib>      use specified vendor-supplied library for HSM
communications
-l, -hsmlabel <label>  use specified HSM label
-s, -hsmslot <slot>    use specified HSM slot
-p, -hsmpin <PIN>      use specified HSM password (optional)

Options:
-y <validity>          certificate validity period in years (default=5)
-gmt0                  set the times in validity period to 00:00:00 GMT
-t <kypname>           key type and size (default=rsa-3072)
                        use -showkeytypes option to list available values
-H <hash>              hash function (default=6 (SHA-384))
                        use -showhash option to list available values for a particular
key type
    4      SHA-1
    5      SHA-256
    *6     SHA-384
    7      SHA-512
    8      SHA-224
-f <output format>     certificate request output format (default=0)
    *0     ASN.1 DER-encoded
    1     PEM-encoded
-fn <filename format>  exported filename format (default=0)
    *0     serial number of the certificate
    1     common name of the certificate
-email <email>         list of comma-delimited email addresses for reply
                        during certificate request submission
-acl <permissions>     ACL permissions
    an XOR'ed combination of the following values:
        1      admin
        2      audit
        4      certify
        8      revoke
        16     RAMI
        32     DBAccess

```

NOTE: Depending on the NIAP compliance and Dhuma installation options, the set of tasks available are appropriately limited.

6.1.2 Creating a CA Account

To create a new CA account, specify a CA name (alphanumeric characters only). Optionally, specify a display name. Otherwise, the CA name will be used as the display name.


```
Syntax:
-createacct -ca <ca name> [-displayname <display name>]
```

```
Example:
#> ./cacli.sh -createacct -ca testca -displayname "Test Root CA"
```

6.1.3 Creating a Profile

To create a profile under an existing CA account, specify the master CA account name (alphanumeric characters only), a profile name (alphanumeric characters only), and its display name.

```
Syntax:
-createprofile -ca <ca name> -profile <profile name> [-displayname <profile display name>]
```

```
Example:
#> ./cacli.sh -createprofile -ca testca -profile testprofile -displayname "Test Profile"
```

6.1.4 Generating Credentials for a CA Account

To generate a new key pair and self-signed certificate as well as install them as the credentials for an existing CA account, you will need to specify the CA account name, subject DN of the certificate, and other optional attributes.

If optional parameters are not specified, the tool will create a 3072-bit RSA key pair and a self-signed certificate and use SHA-384 as the hash function for signing. The credential will be generated on the same HSM in which the system credential resides. The self-signed certificate will have a five year validity period, a basic constraint extension with the CA bit asserted, a key usage extension with certificate signing, CRL signing and digital signature purposes asserted, a subject key identifier and an authority key identifier extensions.

To customize the extensions in the certificate, create a configuration file with the desired extensions and specify it on the command line using the `-file` option. A sample extension configuration file can be found in `<ca home>/tools/cakey-conf-sample.txt`.

```
Syntax:
-genroot -ca <ca name> -dn <dn> [<HSM option>] [-t <kypname>] [-H <hash>] [-y
<validity>] [-gmt0] [-file <config file>]

Example 1 (create certificate with default settings):
#> ./cacli.sh -genroot -ca "testca" -dn "CN=Test Root CA, O=ISC, C=US"

Example 2 (create certificate with customized settings):
#> ./cacli.sh -genroot -ca "testca" -dn "CN=Test Root CA, O=ISC, C=US" -t rsa-3072 -
y 10 -file /usr/local/certagent7/tools/cacli/cakeyconfig.txt -L
/usr/safenet/luna/libCryptoki2.so -s 1

Example 3 (non-NIAP only; create certificate with the times in validity period
rounded to 00:00:00 GMT)
#> ./cacli.sh -genroot -ca "testca" -dn "CN=Test Root CA, O=ISC, C=US" -gmt0
```

6.1.5 *Generating a Certificate Request for a CA Account*

To generate a new key pair and certificate request for an existing CA account and submit the request to a superior CA account on the same system, specify the CA account name, subject DN for the certificate request, and the superior CA account name. Optionally, specify the key type and size, message digest, your e-mail address, and HSM options. Otherwise, an RSA-3072 key pair will be generated on the same HSM in which the system credential resides and an RSA-3072 bit with SHA-384 certificate request will be created.

```
Syntax:
-gensubcrq -ca <ca name> -dn <dn> [<HSM option>] -issuer <issuer CA account>
[-email <email>] [-t <kypname>] [-H <hash>]

Example:
#> ./cacli.sh -gensubcrq -ca subca -dn "CN=Test Sub CA, O=ISC, C=US" -issuer testca
-email subca@infoseccorp.com -L /usr/safenet/lunaclient/lib/libCryptoki2_64.so -s 1
```

To generate a new key pair and certificate request for an existing CA account and submit the request manually to an external certificate authority, specify the CA account name, subject DN and full output path for the certificate request. Optionally, specify the key type and size, output format, message digest and HSM options; otherwise, an RSA-3072 key pair will be generated on the same HSM in which the system credential resides and a 3072 bit ASN.1 DER-encoded RSA-with-SHA384 certificate request will be generated.

```
Syntax:
-gencrq -ca <ca name> -dn <dn> [<HSM option>] -o <request output file> [-f <output
format>] [-t <kypname>] [-H <hash>]
```

Example 1:

```
#> ./caccli.sh -gencrq -ca subca2 -dn "CN=Test Sub CA2, O=ISC, C=US" -o
/usr/subca2.crq
```

Example 2:

```
#> ./caccli.sh -gencrq -ca subca2 -dn "CN=Test Sub CA2, O=ISC, C=US" -o
/usr/subca2.crq -L /usr/safenet/lunaclient/lib/libCryptoki2_64.so -s 1 -f 1 -t us-p-
256 -H 5
```

6.1.6 Installing a CA Certificate

To install a certificate issued by a superior CA on the same system, specify the name of the CA account with which the certificate is to be associated.

NOTE: If the key pair was generated on an HSM, the [HSM option] is not required as HSM settings will be retrieved from the configuration.

```
Syntax:
-install -ca <ca name>
```

Example:

```
#> ./caccli.sh -install -ca subca
```

To install a certificate issued by an external CA, specify the name of the CA account with which the certificate is to be associated together with the full pathname of the PKCS#7 file containing the entire certificate chain:

```
Syntax:
-installext -ca <ca name> -cert <issued certificate with chain>
```

Example:

```
#> ./caccli.sh -installext -ca subca2 -cert /usr/subca2.p7b
```

6.1.7 Selecting Existing Credentials for a CA Account

To assign a credential on an HSM to an existing CA account, specify the CA account name, and the CA certificate. If the CA certificate is not self-signed, specify its chain in a PKCS#7 file as well. If the CA credential does not reside on the same HSM as the system credential, specify the HSM option as well.

Syntax:
`-assign -ca <ca name> [<HSM option>] -cert <CA certificate> [-chain <CA's chain certificates>]`

Example 1 (select credential for a root account):
`#> ./cacli.sh -assign -ca testca -cert /usr/hsmroot.der`

Example 2 (select credential for a subordinate account):
`#> ./cacli.sh -assign -ca subca -cert /usr/hsmsub.der -chain /usr/issuer.p7b -L /usr/safenet/lunaclient/lib/libCryptoki2_64.so -s 1`

6.1.8 Listing Account Configuration Settings

To list the configuration settings for a CA account, specify the account name and optionally one of more of the following category options:

<code>-credential</code>	account credential
<code>-rami</code>	registration authority management interface configuration
<code>-enrollment</code>	enrollment configuration
<code>-est</code>	Enrollment over Secure Transport (EST) configuration
<code>-certprop</code>	certificate issuance - property configuration
<code>-certext</code>	certificate issuance - extension configuration
<code>-crqfilter</code>	request filter configuration
<code>-revocationpolicy</code>	revocation policy configuration
<code>-crl</code>	CRL processing configuration
<code>-ocsp</code>	OCSP configuration
<code>-ldap</code>	LDAP configuration
<code>-mail</code>	e-mail configuration
<code>-public</code>	public site configuration

If no category option is specified, all configuration settings will be displayed.

Syntax:
`-showconf -ca <ca name> [-credential] [-rami] [-enrollment] [-est] [-certprop] [-certext] [-crqfilter] [-revocationpolicy] [-crl] [-ldap] [-mail] [-public]`

Example:
`#> ./cacli.sh -showconf -ca testca -certprop -certext`

6.1.9 Listing Profile Configuration Settings

To list the configuration settings for a profile, specify the profile name and optionally one of more of the following category options:

<code>-rami</code>	registration authority management interface configuration
--------------------	---

-enrollment	enrollment configuration
-est	Enrollment over Secure Transport (EST) configuration
-certprop	certificate issuance – property configuration
-certtext	certificate issuance – extension configuration
-crqfilter	request filter configuration
-certprop	certificate issuance – property configuration
-mail	e-mail configuration

If no category option is specified, all configuration settings will be displayed.

Syntax:
`-showconf -profile <profile name> [-rami] [-enrollment] [-est] [-certprop] [-certtext] [-crqfilter] [-mail]`

Example:
`#> ./cacli.sh -showconf -subca testprofile -enrollment`

6.1.10 Updating Account Configuration Settings

To update an account or profile configuration, specified the desired account and configuration file.

Sample configuration file for CA account can be found in `<ca home>/tools/default-ca-conf.txt`.

Sample configuration file for profile can be found in `<ca home>/tools/default-profile-conf.txt`.

Syntax:
`-updateacct (-ca <ca name> | -profile <profile name>) -file <config file>`

Example 1 (update CA account):
`#> ./cacli.sh -updateacct -ca testca -file /usr/testca-conf.txt`

Example 2 (update profile):
`#> ./cacli.sh -updateacct -profile testprofile -file /usr/testsubca-conf.txt`

6.1.11 Listing Accounts and Profiles

To list all CA accounts or profiles of a particular CA account, use the following option.

Syntax:
`-showacct [-ca <ca name>]`

Example 1 (list all CA accounts):
`#> ./cacli.sh -showacct`

Example 2 (list all profiles of a CA account):
`#> ./cacli.sh -showacct-ca testca`

6.1.12 Listing Supported Key Generation Options

To list the key types and sizes that can be generated by the HSM, use the following command. If the HSM option is not specified, the HSM in which the system credential resides will be applied.

```
Syntax:  
-showkeytypes [<HSM option>]
```

Example 1 (list all available key types and size of the HSM):

```
#> ./cacli.sh -showkeytypes -L /usr/safenet/lunaclient/lib/libCryptoki2_64.so -s 1 -  
p hsm pin
```

6.1.13 Listing Available Hash Functions for a Specific Key Type and Size

To list the available message digest algorithms that may be used with a particular key type and size (according to current NIST guidelines), use the following command.

```
Syntax:  
-showhash [-t <kypname>]
```

Example 1 (listing available hash functions for use with default rsa-3072 keys):

```
#> ./cacli.sh -showhash
```

Example 2 (listing available hash functions for use with us-p-256):

```
#> ./cacli.sh -showhash -t us-p-256
```

6.1.14 Listing the Slots and Labels on an HSM

To list the available slots and labels on an HSM, use the following command.

```
Syntax:  
-showslots -hsmlib <library>
```

Example:

```
#> ./cacli.sh -showslots -hsmlib /usr/safenet/lunaclient/lib/libCryptoki2_64.so
```

6.1.15 Viewing an Account's ACL

To list the access control list of a particular account, use the following command. Optionally, specify the permissions. Otherwise, a certificate with any permission will be returned.

```

Syntax:
-showacl (-ca <ca name> | -profile <profile name> | -admin) [-acl <permissions>]

-acl <permissions>      ACL permissions
    an XOR'ed combination of the following values:
        1      admin
        2      audit
        4      certify
        8      revoke
        16     RAMI
        32     DBAccess

Example 1 (list the ACLs for a CA account):
#> ./cacli.sh -showacl -ca testca

Example 2 (list the ACL for a profile):
#> ./cacli.sh -showacl -profile testprofile

Example 3 (list the ACL for the Admin Site with an admin permission)
#> ./cacli.sh -showacl -admin -acl 1

```

6.1.16 Adding to an Account's ACL

Once an account has been created, only authorized entities can access it via the administrative web pages or through its RA Management Interface (assuming that RAMI is enabled for the account). The system administrator authorizes a user to access a specific account simply by adding their certificate to the appropriate access control list (ACL). The required parameters for this command include the account name and full path to the user's X.509 certificate or PKCS#7 file.

Syntax:
-addacl (-ca <ca name> | -profile <profile name> | -admin) -acl <permission> -cert <certfile>

ACL permissions <permission>:
an XOR'ed combination of the following values:

1	admin
2	audit
4	certify
8	revoke
16	RAMI
32	DBAccess

Example 1 (add authorized user's certificate to CA account's ACL with 'admin' permission):

```
#> ./cacli.sh -addacl -ca testca -acl 1 -cert /usr/user1.der
```

Example 2 (add authorized user's certificate to CA account's ACL with 'certify', 'revoke' and 'RAMI' permissions):

```
#> ./cacli.sh -addacl -ca testca -acl 28 -cert /usr/user1.der
```

Example 3 (add authorized user's certificate to profile's ACL with 'certify' and 'revoke' permissions):

```
#> ./cacli.sh -addacl -profile testprofile -acl 12 -cert /usr/user1.der
```

Example 4 (add authorized user's certificate to Admin Site's ACL with 'audit' permission):

```
#> ./cacli.sh -addacl -admin -acl 2 -cert /usr/user1.der
```

NOTE: CA account supports all permissions. Profile supports 'certify', 'revoke' and 'RAMI' permissions only.

6.1.17 Updating the Permission of a Certificate from an Account ACL

To update account rights to a previously authorized entity, simply update their certificate from the appropriate ACL using the following command.

Syntax:
-updateacl (-ca <ca name> | -profile <profile name> | -admin)

Example 1 (update the permission of authorized user's certificate in CA account's ACL):

```
#> ./cacli.sh -updateacl -ca testca
```

Example 2 (update the permission of authorized user's certificate from profile's ACL):

```
#> ./cacli.sh -updateacl -profile testca
```

Example 3 (update the permission of authorized user's certificate from Admin Site's ACL):

```
#> ./cacli.sh -updateacl -admin
```

You will be presented with a list of all certificates in the appropriate ACL for the specified CA account and then be prompted to enter the IDs of those certificates that you wish to update from the list.

6.1.18 Removing a Certificate from an Account's ACL

To deny account rights to a previously authorized entity, simply remove their certificate from the appropriate ACL using the following command.

```
Syntax:
-removeacl (-ca <ca name> | -profile <profile name> | -admin)

Example 1 (remove authorized user's certificate from CA account's ACL):
#> ./cacli.sh -removeacl -ca testca

Example 2 (remove authorized user's certificate from profile's ACL):
#> ./cacli.sh -removeacl -profile testprofile

Example 3 (remove authorized user's certificate from Admin Site's ACL):
#> ./cacli.sh -removeacl -admin
```

You will be presented with a list of all certificates in the appropriate ACL for the specified CA account and then be prompted to enter the ID of the certificate that you wish to remove from the list or a '*' to remove all certificates.

6.1.19 Importing Certificates

If you are migrating over from another certificate authority product, you may wish to import your existing certificates into the CertAgent database. To import a certificate or all certificates located in a directory, use the following command.

```
Syntax:
-importcert -ca <ca name> -file <cert file or directory contains cert files>

Example 1 (import a certificate):
#> ./cacli.sh -importcert -ca subca -file /usr/cert1.der

Example 2 (import all certificate in a directory):
#> ./cacli.sh -importcert -ca subca -file /usr/cert_directory
```

6.1.20 Importing a CRL

If you are migrating over from another certificate authority product, you may wish to import previously issued CRLs. To import the latest CRL, use the following command.

```
Syntax:
-importcrl -ca <ca name> -file <crl file>

Example 1 (import a CRL):
#> ./cacli.sh -importcrl -ca subca -file /usr/subca.crl
```

6.1.21 Exporting Certificates

To export certificates, match the specified not before date range, using the following command.

```
Syntax:
-exportcert -ca <ca name> -o <output directory> -date <notBeforeDate MM/dd/yyyy>
[<notAfterDate MM/dd/yyyy>] [-fn <filename format>]

<filename format>:
  0: serial number of the certificate
  1: common name of the certificate

Example 1 (export certificates that were issued between 01/01/2019 and today):
#> ./cacli.sh -exportcert -ca subca -o /usr/outDir -date "01/01/2019"

Example 2 (export certificates that were issued between 01/01/2019 and 05/31/2019
and use the common name of the certificate as the file name):
#> ./cacli.sh -exportcert -ca subca -o /usr/outDir -date "01/01/2019" "05/31/2019 -
fn 1"
```

6.1.22 Submitting Certificate Requests

To submit a certificate request to a CA account or its profile, using the following command.

```
Syntax:
-submit (-ca <ca name> | -profile <profile name>) -file <request file> [-email
<email>]

Example 1 (submit a request to a CA account):
#> ./cacli.sh -submit -ca testca -file /usr/user1.p10

Example 2 (submit a request to a profile):
#> ./cacli.sh -submit -profile testprofile -file /usr/user1.p10
```

6.1.23 Deleting a Profile

To delete an existing profile along with its configuration, use the following command.

NOTE: Once a profile has been deleted, all its certificate requests and certificates will be assigned to the master CA account.

```
Syntax:
-deleteacct -profile <profile name>

Example:
#> ./cacli.sh -deleteacct -profile testprofile
```

6.1.24 Disabling an Account

If an account is no longer in service, it can be disabled using the following command:

NOTE: Once an account has been disabled, all issued certificates, CRLs, profiles, configuration and audit trails stored in the database will be retained. Only the certificates in the ACL will be removed. Access to the disabled account will be denied via all interfaces.

```
Syntax:
-disableacct -ca <ca name>

Example:
#> ./cacli.sh -disableacct -ca subca
```

6.1.25 Listing Trust Anchors

If CertAgent is in NIAP mode, CertAgent maintains a list of trust anchors for path validation. To view the trust anchor list, run the following command.

```
Syntax:
-showtrust

Example:
#> ./cacli.sh -showtrust
```

You will be presented with a list of all trust anchors with their subject DN, serial number and not after date information.

6.1.26 Adding a Trust Anchor

If CertAgent is in NIAP mode, CertAgent maintains a list of trust anchors for path validation. To add a trust anchor to the list, run the following command:

```
Syntax:
-addtrust -file <cert file>

Example:
#> ./cacli.sh -addtrust -file /usr/trust.der
```

6.1.27 Removing a Trust Anchor

If CertAgent is in NIAP mode, CertAgent maintains a list of trust anchors for path validation. To remove an existing trust anchor from the list, run the following command:

```
Syntax:
-removetrust

Example:
#> ./cacli.sh -removetrust
```

You will be presented with a list of trust anchors and then be prompted to enter the ID of the certificate that you wish to remove from the list or a '*' to remove all the certificates.

6.1.28 Listing CRLs

If CertAgent is in NIAP mode, CertAgent maintains a list of CRLs for path validation. To view the CRL list, run the following command.

```
Syntax:
-showcrl

Example:
#> ./cacli.sh -showcrl
```

You will be presented with a list of all CRLs with their issuer name and next update date information.

6.1.29 Adding a CRL

If CertAgent is in NIAP mode, CertAgent maintains a list of CRLs for path validation. To add a CRL to the list, run the following command:

```
Syntax:
-addcrl -file <crl file>

Example:
#> ./cacli.sh -addcrl -file /usr/subca.crl
```

6.1.30 Removing a CRL

If CertAgent is in NIAP mode, CertAgent maintains a list of CRLs for path validation. To remove an existing CRL from the list, run the following command:

```
Syntax:
-removecrl

Example:
#> ./cacli.sh -removecrl
```

You will be presented with a list of CRLs and then be prompted to enter the ID of the CRL that you wish to remove from the list or a '*' to remove all the CRLs.

6.1.31 Displaying System Information

To display the version number of CertAgent components and system information, run the following command:

```
Syntax:
-getinfo

Example:
#> ./cacli.sh -getinfo
```

The following key-value pair will be displayed:

Parameter	Format and Description
CertAgent	String Version number of CertAgent
CDK	String Version number of ISC CDK library
SA	String Version number of ISC SA library
JNI	String Version number of the JNI library
FIPS	Boolean True if the ISC CDK library is in FIPS mode
OS	String Operation system CertAgent is running on: "Windows" or "Linux"
Time	String Current date and time of the system in mm/dd/yy hh:mm:ss zZ format Example: 12/03/17 16:04:02 CDT-0600
NIAP	Boolean True if CertAgent is in NIAP mode
Error	String Optional; Set to error message if CDK is in error state

6.1.32 Displaying Unique Subject DN Statistics

To display the unique subject DN statistics, run the following command:

```
Syntax:
-statistics

Example:
#> ./cacli.sh -statistics
```

You will be presented with a list of CA accounts and their statistics.

6.1.33 Exporting CA Account and Its Profiles Configurations

NOTE: This option is only available if CertAgent is installed with non-NIAP-compliant option.

To export the configuration of a CA account and all the profiles to a file, run the following command:

```
Syntax:
-exportacct -ca <ca name> -file <out file>

Example:
#> ./caccli.sh -exportacct -ca ca7 -file /tmp/exported.cfg
```

NOTE: If you would like to export a specific profile or CA account only, use the Export function in the CA Account web interface instead.

6.1.33.1 Exported File Format

The following sections describe the format of the exported configuration file. Depending on the target CertAgent system and CA account, the value of the profile ID, encrypted password for email and LDAP configurations, and account-specific URLs may need to be updated manually before importing the configuration file.

6.1.33.1.1 CA Account and Profile

The 'CA Account and Profile' section defines the CA account's display name, number of profiles, profile IDs and their display names.

The following sample defines the CA's display name (CertAgent 7), the number of profile (2), the first profile ID (ca7tls) with display name (TLS) and the second profile ID (ca7clientauth) with display name (Client Auth).

```
#####
# CA Account and Profile
#####
ca.include.master=1
ca.displayname=CertAgent 7
ca.profile.count=2
0-ca.id=ca7tls
0-ca.displayname=TLS
1-ca.id=ca7clientauth
1-ca.displayname=Client Auth
```

CA account ID and profile ID must be unique within a CertAgent system. If you are planning to import the same profiles to another CA account of the same system, update the value of each profile ID (*-ca.id) manually.

6.1.33.1.2 CA Account Configuration

The 'CA Account Configuration' section defines the settings of the CA account.

For example:

```
#####
# CA Account Configuration
#####
#----- Enrollment - Configuration
crq.key.type=0x1
crq.key.size=3072
crq.key.size.enforce=0
crq.key.size.max=4096
crq.key.ecc=256|384|
...
```

6.1.33.1.3 Profile Configuration

The 'Profile Configuration' section defines the settings of a profile based on the profile index starting from zero. The index of the profile is defined in the 'CA Account and Profile' section.

For example:

```
#####
# Profile Configuration #0: TLS
#####
#----- Enrollment - Configuration
0-crq.key.type=0x1
0-crq.key.size=3072
0-crq.key.size.enforce=0
0-crq.key.size.max=4096
0-crq.key.ecc=256|384|
...
```

6.1.33.1.4 Email Settings

If the email setting is enabled, the authentication password will be encrypted with the system certificate and saved to the 'email.authEPass' or '<index>-email.authEPass' value of the configuration file.

For example:

```
#----- Email Settings
email.enable=1
...
email.authEPass=MIICzQYJKoZIhvcNAQcDoIICvjCCAROCAQAxggJ1M...
...
#----- Email Settings
0-email.enable=1
...
0-email.authEPass=MIICzQYJKoZIhvcNAQcDoIICvjCCAROCAQAxggJ1M...
...
```

If you are planning to import the configuration file to another CertAgent system, the encrypted password will not be able to decrypt. In this case, you will be prompted to enter the password after uploading the configuration file. To avoid decrypting the invalid password, comment out the encrypted password entry by inserting a '#' character at the beginning of the entry. You will then be prompted to

enter the password later. Alternatively, to avoid the prompt, you can specify the password in the 'email.authPass' or '<index>-email.authPass' entry and comment out the encrypted password 'email.authEPass' or '<index>-email.authEPass' entry.

For example:

```
#----- Email Settings
email.enable=1
...
#email.authEPass=MIICzQYJKoZIhvcNAQcDoIICvjCCAROCAQAxggJ1M...
email.authPass=<password>
...
#----- Email Settings
0-email.enable=1
...
#0-email.authEPass=MIICzQYJKoZIhvcNAQcDoIICvjCCAROCAQAxggJ1M...
0-email.authPass=<password>
...
```

6.1.33.1.5 LDAP Repository Settings

If LDAP repository setting is configured, the bind or keystore password will be encrypted with the system certificate and saved to the 'xldap.bind.epass.<index>' value of the configuration file.

For example:

```
#----- LDAP Repositories
xldap.enable=1
...
xldap.bind.epass.0=MIICzQYJKoZIhvcNAQcDoIICvjCCAROCAQAxggJ1M...
...
```

If you are planning to import the configuration file to another CertAgent system, the encrypted password will not be able to decrypt. In this case, you will be prompted to enter the password after uploading the configuration file. To avoid decrypting the invalid password, comment out the encrypted password entry by inserting a '#' character at the beginning of the entry. You will then be prompted to enter the password later. Alternatively, to avoid the prompt, you can specify the password in the 'xldap.bind.pass.<index>' entry and comment out the encrypted password 'xldap.bind.epass.<index>' entry.

For example:


```
#----- LDAP Repositories
xldap.enable=1
...
#xldap.bind.epass.0=MIICzQYJKoZIhvcNAQcDoIICvjCCAROCAQAxggJlM...
xldap.bind.pass.0=<password>
...
```

6.1.33.1.6 Account-Specific URLs

For the extensions containing the account-specific URLs, the configuration file may need to update manually before importing it to another CertAgent system or CA account.

For example, Authority Information Access and CRL Distribution Points extensions:

```
#----- Certificate Issuance - Extensions
cert.ext.aia=0
cert.ext.aia.values=3\!http://ca1.infoseccorp.com/certagent/ocsp/ca7|2\!http://ca1
.infoseccorp.com/certagent/getinfo.jsp?ca=ca7&type=CA_BIN|
...
cert.ext.cdp=0
cert.ext.cdp.values=http://ca1.infoseccorp.com/certagent/getinfo.jsp?ca=ca7&type=
CRL_BIN|
```

The URLs above contain the server's hostname and account information:

- `http://<hostname>/certagent/ocsp/<ca>`
- `http://<hostname>/certagent/getinfo.jsp?ca=<ca>&type=CA_BIN`
- `http://<hostname>/certagent/getinfo.jsp?ca=<ca>&type=CRL_BIN`

If you are planning to import the configuration to another CertAgent system or CA account, the hostname (<hostname>) and/or CA account (<ca>) values must be updated manually.

6.1.34 Importing CA Account and Its Profiles Configurations

NOTE: This option is only available if CertAgent is installed with non-NIAP-compliant option.

To import the configuration of a CA account and its profiles to an existing CA account, run the following command:

```
Syntax:
-importacct -ca <ca name> -file <in file>

Example:
#> ./cacli.sh -importacct -ca testca -file /tmp/exported.cfg
```

NOTE: If the profile name already exists in the specified CA account, it will update the configuration. If the profile name doesn't exist, it will create the profile and apply the configuration automatically. If the profile name already exists in the other CA account, this configuration will not be applied.

6.2 The Certificate Report Generator

A shell script is provided for generating certificate status reports.

```
<ca home>/tools/reportgenerator.sh (UNIX)
<ca home>\tools\reportgenerator.bat (Windows)
```

6.2.1 Command Line Syntax

```
Usage:
* Print this help, then exit
-h

* Generate certificate report
-ca <ca name> [options] -out <output path>

* Count number of matching certificates
-count -ca <ca name> [options]

Required switches:

-ca      CA account name

-out     output file path

Options:

-assign  assigned account name
         if omitted, all certificates will be returned; otherwise, only
certificates assigned to the specified account will be returned

-status  status of certificates:
         5  *any
         0  valid
         1  pending revocation/on hold
         2  revoked
         3  expired

-search  <search attribute>=<search string>
         certificate search filter comprised of one or more search attributes:
         SERIAL  serial number
         REQID   request ID
         CN      common name
         T       title
```

```

    O      organization
    OU     organizational unit
    L      locality
    ST     state
    MAIL   e-mail address
for 'fuzzy' searches, place one or more asterisks in search string

-date    <date type> <from date> <to date>
search for certificates within the date range:
date type:
    0     revocation date
    1     not before date
    2     not after date
from and to dates must be formatted as: mm/dd/yyyy

-sort    <attribute> <order>
sort the certificates by:
attribute:
    0     serial number
    1     request ID
    2     DN
    3     status
    4     assigned account
    5     revocation date
    6     not before date
    7     not after date
order:
    0     ascending
    1     descending

-format  output format:
    1     *text
    2     CSV

```

6.2.2 Sample commands

Example 1 (list certificates of a particular CA account):

```
#> ./reportgenerator.sh -ca testca -out /usr/report1.txt
```

Example 2 (list all certificates that will be expired on particular dates):

```
#> ./reportgenerator.sh -ca testca -date 2 01/01/2020 12/01/2020 -out
/usr/report2.txt
```

Example 3 (list all certificates that belong to a particular organization):

```
#> ./reportgenerator.sh -ca testca -search O=ISC -out /usr/report3.txt
```

Example 4 (list all expired certificates):

```
#> ./reportgenerator.sh -ca testca -status 3 -out /usr/report4.txt
```

Example 5 (list all valid certificates sorted by expiration date in ascending order and output to file in CSV format):

```
#> ./reportgenerator.sh -ca testca -status 0 -sort 7 0 -format 2 -out
/usr/report5.txt
```

7 The CertAgent RA Management Interface

The CertAgent Registration Authority Management Interface (RAMI):

URL: `https://<hostname>:<TLS port>/certagentadmin/ca/rami`

Allows a remote or automated client process (acting on behalf of an authorized user) to:

- submit a certificate request for immediate processing and obtain an issued certificate,
- revoke a certificate,
- reinstate a certificate with a status of on-hold or pending revocation,
- issue a CRL,
- retrieve the CA accounts information,
- query certificate request information,
- query certificate information,
- and retrieve an issued certificate

over a TLS-secured connection (with client authentication). Once an HTTPS connection has been established, the process simply POSTs to the server all required parameters and values for the desired operation. Response status and operation details will be returned to the client process as formatted text that may be parsed for further action by that client. In this way, an authorized *registration authority* (RA) may submit pre-screened certificate requests in an automated fashion.

This chapter documents the procedure to be followed by a Java-based RA. Similar considerations would apply if your application is written in a different language. A sample Java program illustrating a typical interaction with the RA Management Interface is provided in the package:

`<ca home>/tools/RAMISample.java`

Run the following command as appropriate for your system to compile and run the sample program:

```
javac -classpath ./usr/local/certagent7/lib/gson-2.8.6.jar RAMISample.java
java -classpath ./usr/local/certagent7/lib/gson-2.8.6.jar RAMISample

javac -classpath ".;C:\Program Files\CertAgent7\lib\gson-2.8.6.jar" RAMISample.java
java -classpath ".;C:\Program Files\CertAgent7\lib\gson-2.8.6.jar" RAMISample
```

Alternatively, RAMI requests can be submitted via the cURL command line tool.

1. Run the following command to convert the client PKCS#12 file to PEM format:

```
openssl pkcs12 -in <p12 file> -out <client pem file> -nodes
```

2. Run the following command to convert the trust anchor certificate to PEM format:

```
openssl x509 -inform der -in <root cert> -out <root pem file>
```

3. Run the curl command to submit the post data.

```
curl <url> --cert <client pem file> -v -o <out p7 file> --cacert <root pem file> --tlsv1.2 [--data-urlencode "<name>=<value>"]* [--data-urlencode <name>@<file>]
```

7.1 Establishing a TLS Session with Client Authentication

For a Java application to establish a TLS connection with the CertAgent RA Management Interface, the following requirements must be satisfied:

- client credentials (certificate and private key) must be available on an attached HSM, in a local PKCS#12 file, or in a Java keystore file.
- the client must possess the passwords for its own private key and the keystore in which it resides, or the HSM PIN if the credentials are stored on an HSM.
- the client's Java trust keystore must contain a trust anchor for the server's TLS certificate (*i.e.*, the root certificate for path validation of the server's SSL credentials).
- the server's Java trust keystore must contain the trust anchor for the user's TLS certificate.
- the client's certificate must be added to the ACL of the appropriate CA account with 'RAMI' permission.

NOTE: Java does not support PKCS#12 that uses AES-256 to encrypt the private key. If you are planning to use CertAgent temporary credentials (e.g., `ca-operations-staff.p12`) as the client credentials and NIAP compliance option was selected during the installation, run the following OpenSSL commands to create a new PKCS#12 that uses DES3 to encrypt the private key. Use the new credentials in your Java program.

```
openssl pkcs12 -in ca-operations-staff.p12 -out ca-op-staff.pem
openssl pkcs12 -export -in ca-op-staff.pem -out ca-operations-staff-des3.p12
```

7.2 Submitting a Certificate Request

As it is assumed the RA can perform certain actions more easily than the CertAgent system, if in fact they are required at all, certificate requests submitted via RAMI are handled somewhat differently than requests manually submitted to the standard web interface (even if the corresponding features are enabled in the GUI for the CA account to which the requests are submitted). In particular:

- requests lacking a subject e-mail address are not rejected even when "Class 1 assurance" is enabled.
- neither the CA nor the subject of the request are sent e-mail notifications after the request is received or processed.

7.2.1 Request

To submit a certificate request and obtain a certificate, POST the following data to the RA Management Interface.

NOTE: Unsigned PKCS#10 certificate requests are acceptable if submitted via the RAMI. After issuing the certificates, these requests will be signed in CMS by the CA's private key and stored in the database.

7.2.1.1 Required Parameters

Parameter	Format and Description
action	"enrollKey"
ca	CA login name
request	a URL-encoded and base64-encoded PKCS#10

7.2.1.2 Optional Parameters

7.2.1.2.1 Contact Email Addresses

Parameter	Format and Description
userEmail	a list of comma-delimited e-mail addresses of the user

7.2.1.2.2 Validity Period

Parameter	Format and Description
cert.validity.num	validity period if set, the cert.validity.unit setting is required
cert.validity.unit	validity period units: years: 1; months: 2; days: 3
cert.validity.nb	not before date number of milliseconds since 01/01/1970 00:00:00 GMT if the cert.validity.na is set and this value is not set, the issuance time will automatically be set to the not before date if set, this value cannot precede the current time
cert.validity.na	not after date number of milliseconds since 01/01/1970 00:00:00 GMT if set, the cert.validity.num setting is ignored and the specified cert.validity.nb setting will be used
cert.validity.gmt0	set the times to 00:00:00 GMT enable: 1; disable: 0 this setting applies to CertAgent installed with non-NIAP-compliant mode only; if set, the cert.validity.num setting is required; the certificate's not before date will

	rounded up to the 00:00:00 GMT of next day. The not after date will be set to 00:00:00 GMT of the specified validity period
--	---

NOTE: If the validity period is specified, the duration must be shorter than or equal to the default validity period defined in the CA account or profile. Otherwise, the request will be rejected.

7.2.1.2.3 Issued Certificate Format

Parameter	Format and Description
<code>response.cert.format</code>	<p>issued certificate format in the response</p> <p>if not set or set to 1, the issued certificate and chain in base64-encoded PKCS#7 format will be included in the 'base64CertChain' value of the response</p> <p>if set to 2, only the issued certificate in base64-encoded X.509 format will be included in the 'base64Cert' value of the response</p> <p>if set to 3, both issued certificate and chain in base64-encoded PKCS#7 format and issued certificate in X.509 format will be included in the 'base64CertChain' and 'base64Cert' values respectively in the response</p>

7.2.1.3 Optional Override Parameters⁴

The POST parameters in this section are optional; they are applied to the certificate being issued only if the “allow POST to override defaults” setting or the Post Rule is enabled by the specified CA. In NIAP compliant mode, these options are not available.

In processing these parameters with POST Rule, the following rules are applied:

- if a parameter is not supplied, the default value of that attribute or extension is applied.
- if a parameter is supplied and complied to the specified rule, the parameter value will be appended to or overridden the default extension based on the rule
- if a parameter is supplied with an empty value, the request will be rejected.
- if a parameter is supplied, its *.append option will be ignored

NOTE: POST Rule is available for extended key usage, key usage, and subject alternative name extensions.

In processing these parameters without a POST Rule and the “allow POST to override default” option enabled, the following rules are applied:

- if a parameter is not supplied, the default value of that attribute or extension is applied.
- if a parameter is supplied with an empty value, the corresponding attribute or extension is removed from the certificate.

⁴ Non-NIAP-compliant CertAgent only

- if an *.append option is available for the parameter, the parameter value is appended to the default value if the append option is set to “1” (though duplicate values are ignored); otherwise, the POST value always replaces the default value.
- if a parameter value conflicts with its default setting in the GUI (e.g., the CA has marked an extension critical, but the POST sets it to non-critical), the explicit POST value will be used.

Name	Parameter	Format and Description
Subject DN	cert.rdn	<p>subject DN of the issued certificate</p> <p><rdn>[, <rdn>]+</p> <p><rdn>: <attribute type>=<value></p> <ul style="list-style-type: none"> • EM=<e-mail Address> • CN=<common name> • UID=<user ID> • T=<title> • OU=<organizational unit> • L=<locality / city> • O=<organization> • ST=<state / province> • C=<2 characters country code> • SERIALNUMBER=<serial number> • DC=<domain component> • STREET=<street> • DNQ=<DN Qualifier> • SURNAME=<surname> • GIVENNAME=<given name> • INITIALS=<initials> • GENERATION=<generation> • PSEUDONYM=<pseudonym> • PHONE=<phone number> • POSTALCODE=<postal code> • UNIQUEIDENTIFIER=<hex-encoded unique identifier> <p>example: CN=RAMI Test User, O=ISC, C=US</p>
DN Encoding	dn.encoding	<p>DN encoding</p> <ul style="list-style-type: none"> • 0: PrintableString • 1: UTF8
Hash Algorithm	cert.md	<p>hash algorithm to be used to sign the certificate</p> <p>SHA1 SHA224 SHA256 SHA384 SHA512</p>
Basic Constraints	cert.ext.bc	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.bc.ca	certificate authority: 1; end entity: 0
	cert.ext.bc.pathlen	path length: none: -1; else 0 to 5 (this value is ignored if cert.ext.bc.ca=0)
Key Usage	cert.ext.ku	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.ku.append	(optional) if set to 1, the cert.ext.ku.value setting is appended to the default setting with duplicates ignored
	cert.ext.ku.value	<p>an XOR'ed combination of the following values:</p> <ul style="list-style-type: none"> • digital signature: 0x80

		<ul style="list-style-type: none"> • non-repudiation: 0x40 • key encipherment: 0x20 • data encipherment: 0x10 • key agreement: 0x08 • certificate signing: 0x04 • CRL signing: 0x02 • encipher-only: 0x01 • decipher-only: 0x100
Extended Key Usage	cert.ext.eku	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.eku.append	(optional) if set to 1, the cert.ext.eku.value and cert.ext.eku.oid settings are appended to the default setting with duplicates ignored
	cert.ext.eku.value	extended key usage value: an XOR'ed combination of the following values: <ul style="list-style-type: none"> • server authentication: 0x01 • client authentication: 0x02 • code signing: 0x04 • e-mail protection: 0x08 • time stamping: 0x10 • Microsoft EFS: 0x20 • accept any: 0x40 • PIV Card Authorization: 0x80 • Microsoft Smart Card Logon: 0x0100 • OCSP signing: 0x0200 • data validation and certification server: 0x0400 • IPSEC End System: 0x0800 • IPSEC Tunnel: 0x1000 • IPSEC User: 0x2000 • SCVP Responder: 0x4000 • Extensible Authentication Protocol over PPP: 0x8000 • Extensible Authentication Protocol over LAN: 0x010000 • SCVP Server: 0x020000 • SCVP Client: 0x040000 • IPSEC IKE: 0x080000 • CMC Registration Authority: 0x10000
	cert.ext.eku.oid	a list of customize OIDs delimited by example: 1.3.6.1.4.1.311.10.3.3 2.16.840.1.113730.4.1
Authority Key Identifier	cert.ext.aki	exclude: -1; otherwise an XOR'ed combination of the following values: <ul style="list-style-type: none"> • critical: 0x01 • key ID: 0x02 • CA issuer DN: 0x04 • issuer serial number: 0x08
Subject Key Identifier	cert.ext.ski	exclude: -1; include as non-critical: 0; include as critical: 1
Issuer Alternative Name	cert.ext.ian	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.ian.append	(optional) If set to 1, all cert.ext.ian.* settings are appended to the default setting with duplicates ignored
	cert.ext.ian.em	a list of RFC822 names delimited by

		example: test@infoseccorp.com
	cert.ext.ian.on	<p>a list of other names <othername>, delimited by </p> <p><othername>: <oid>!<value type>!<value></p> <p><value type>: Octet String: 0; UTF8 String: 1</p> <p>if <value type>=0, value can be a text string or hex-encoded string starting with '0x'; if <value type>=1, value can be a text string or UTF8 string starting with '\u'</p> <p>example:</p> <p>1.3.6.1.4.1.311.25.1!0!0xe41224e2fe7a724d84add5024147f0f4 1.3.6.1.4.1.311.20.2.3!1!test </p>
	cert.ext.ian.dns	<p>a list of DNS names delimited by </p> <p>example: ca.infoseccorp.com</p>
	cert.ext.ian.dn	<p>a list of directory names delimited by </p> <p>example: CN=Directory Name, C=US CN=Directory Name2, C=US</p>
	cert.ext.ian.url	<p>a list of URLs delimited by </p> <p>example: http://ca.infoseccorp.com</p>
	cert.ext.ian.ip	<p>a list of IP addresses delimited by </p> <p>example: 192.168.0.20</p>
	cert.ext.ian.edipartyname	<p>a list of EDI party names <EDIPartyname>, delimited by </p> <p><EDIPartyname>: [<name assigner>]!<party name></p> <p>example: nameAssigner!partyName!partyName2</p>
	cert.ext.ian.regid	a list of registered IDs, delimited by
	cert.ext.ian.x400address	a base64-encoded, ASN.1 DER-encoded PDU
Subject Alternative Name	cert.ext.san	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.san.append	(optional) if set to 1, all cert.ext.san.* settings are appended to the default setting with duplicates ignored
	cert.ext.san.em	<p>a list of RFC822 names delimited by </p> <p>example: test@domain.com test2@domain.com</p>
	cert.ext.san.on	<p>a list of other names <othername> delimited by </p> <p><othername>: <oid>!<value type>!<value></p> <p><value type>: Octet String: 0; UTF8 String: 1</p> <p>if <value type>=0, value can be a text string or hex-encoded string starting with '0x'; if <value type>=1, value can be a text string or UTF8 string starting with '\u'</p> <p>example:</p> <p>1.3.6.1.4.1.311.25.1!0!0xe41224e2fe7a724d84add5024147f0f4 1.3.6.1.4.1.311.20.2.3!1!test </p>
	cert.ext.san.dns	<p>a list of DNS names delimited by </p> <p>example: www.infoseccorp.com</p>
	cert.ext.san.dn	a list of directory names delimited by

		example: CN=Directory Name, C=US CN=Directory Name2, C=US
	cert.ext.san.url	a list of URLs, delimited by example: http://www.infoseccorp.com
	cert.ext.san.ip	a list of IP addresses, delimited by example: 192.168.0.20
	cert.ext.san.edipartyname	a list of EDI party names <EDIPartyName>, delimited by <EDIPartyName>: [<name assigner>]!<party name> example: nameAssigner!partyName!partyName2
	cert.ext.san.regid	a list of registered IDs, delimited by
	cert.ext.san.x400address	a base64-encoded, ASN.1 DER-encoded PDU
Authority Information Access	cert.ext.aia	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.aia.append	(optional) if set to 1, the cert.ext.aia.values setting is appended to the default setting with duplicates ignored
	cert.ext.aia.values	a list of authority information access values delimited by [<type>!<value>]* <type>: <OID type> or <2: CA Issuer> or <3: CA OCSP> <OID type>: 1^<OID> example: 1^1.2.3.4!http://ca.infoseccorp.com/accessmethod 2!http://www.infoseccorp.com/caissuer 3!http://ca.infoseccorp.com/ocsp
CRL Distribution Points	cert.ext.cdp	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.cdp.append	(optional) if set to 1, the cert.ext.cdp.values setting is appended to the default setting with duplicates ignored
	cert.ext.cdp.values	a list of CRL distribution points delimited by [<crl distribution points>]* example: https://ca.infoseccorp.com:8443/certagent/public/getcrl.jsp?issuer=ca
Subject Directory Attributes	cert.ext.sda	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.sda.rfcCitizenship	country of citizenship (RFC 3739) (exactly 2 characters)
	cert.ext.sda.dodCitizenship	country of citizenship (US DOD) (exactly 2 characters)
	cert.ext.sda.nationality	nationality (US DOD)
	cert.ext.sda.employeetype	employee type (RFC 2798)
S/MIME Capabilities	cert.ext.smime	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.smime.append	(optional) if set to 1, the cert.ext.smime.* settings are appended to the default setting with duplicates ignored

	<code>cert.ext.smime.symmetric</code>	<p>symmetric key algorithms an XOR'ed combination of the following values:</p> <ul style="list-style-type: none"> • aes128-cbc: 0x01 • aes256-cbc: 0x02 • aes128-gcm: 0x04 • aes256-gcm: 0x08 • aes128-wrap: 0x10 • aes256-wrap: 0x20
	<code>cert.ext.smime.hash</code>	<p>hash algorithms an XOR'ed combination of the following values:</p> <ul style="list-style-type: none"> • sha-256: 0x01 • sha-384: 0x02 • sha-512: 0x04
	<code>cert.ext.smime.rsa</code>	<p>asymmetric key algorithms (applies to RSA certificates only) an XOR'ed combination of the following values:</p> <ul style="list-style-type: none"> • sha256WithRSAEncryption: 0x01 • sha384WithRSAEncryption: 0x02 • sha512WithRSAEncryption: 0x04 • RSA Encryption: 0x08 • RSAES OAEP: 0x10 • PKCS1 RSA PSS: 0x20
	<code>cert.ext.smime.ecc</code>	<p>asymmetric key algorithms (applies to ECC certificates only) an XOR'ed combination of the following values:</p> <ul style="list-style-type: none"> • ecdsaWithSHA256: 0x01 • ecdsaWithSHA384: 0x02 • ecdsaWithSHA512: 0x04 • dhSinglePass-stdDH-sha256kdf-scheme (aes256-wrap): 0x08 • dhSinglePass-stdDH-sha384kdf-scheme (aes256-wrap): 0x10 • dhSinglePass-stdDH-sha512kdf-scheme (aes256-wrap): 0x20
	<code>cert.ext.smime.custom</code>	<p>a list of customize algorithms <algorithm> delimited by <algorithm>: <oid>[!<parameter>] example: 2.16.840.1.101.3.4.2.8 1.3.132.1.11.1-2.16.840.1.101.3.4.1.5 represents sha3-256 and dhSinglePass-stdDH-sha256kdf-scheme (aes192-wrap)</p>
Netscape Certificate Type	<code>cert.ext.nct</code>	exclude: -1; include as non-critical: 0; include as critical: 1
	<code>cert.ext.nct.append</code>	(optional) if set to 1, the <code>cert.ext.nct.value</code> setting is appended to the default setting with duplicates ignored
	<code>cert.ext.nct.value</code>	<p>an XOR'ed combination of the following values:</p> <ul style="list-style-type: none"> • SSL Client Certificate: 0x80 • SSL Server Certificate: 0x40 • S/MIME User Certificate: 0x20 • Object Signing Certificate: 0x10 • SSL CA Certificate: 0x04 • S/MIME CA Certificate: 0x02 • Object Signing CA Certificate: 0x01 <p>example: 0xA0</p>
QC Statement	<code>cert.ext.qcs</code>	exclude: -1; include as non-critical: 0; include as critical: 1

	cert.ext.qcs.limitValue	statement OID: id-etsi-qcs-QcLimitValue <3 characters currency code> <amount> <exponent> example: USD 10000 0
	cert.ext.qcs.qcCompliance	statement OID: id-etsi-qcs-QcCompliance exclude: 0; include: 1
	cert.ext.qcs.telAgencyAuthCertClause	statement OID: NES Telecommunication Agency Authentic Certificate Clause
	cert.ext.qcs.retentionPeriod	statement OID: id-etsi-qcs-QcRetentionPeriod retention period; number of years
Certificate Policies	cert.ext.cp	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.cp.append	(optional) if set to 1, the cert.ext.cp.values setting is appended to the default setting with duplicates ignored
	cert.ext.cp.values	a list of certificate policies delimited by [<OID>[!<CPS policy qualifier>][!<User notice policy qualifier>]]]* <CPS policy qualifier>: 0^<URL> <User notice policy qualifier>: 1^<user notice text> CPS and user notice policy qualifiers are optional example: 2.16.840.1.101.2.1.11.5 1.2.3.4.5!0^http://www.infosec corp.com/cps 1.2.3.4.6!1^certificate user notice 1.2.3.4.7!0^http://www.infosec corp.com/cps2!1^certificate user notice
Policy Constraints	cert.ext.pc	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.pc.explicit.num	the number of additional certificates that may appear in the path before an explicit policy is required for the entire path: inhibit policy mapping example: 2
	cert.ext.pc.inhibit.num	the number of additional certificates that may appear in the path before policy mapping is no longer permitted example: 2
Inhibit Any Policy	cert.ext.iap	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.iap.num	the number of additional certificates that may appear in the path before any policy is no longer permitted example: 2
Policy Mapping (CA Certificate only)	cert.ext.pm	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.pm.append	(optional) if set to 1, the cert.ext.pm.values setting is appended to the default setting (though duplicates are ignored)
	cert.ext.pm.values	a list of mapping pairs: issuer domain policy equivalent to subject domain policy; delimited by (<issuer domain policy OID> <subject domain policy OID>)* example: 1.3.6.1.4.1.311.21.53 1.2.3.4.87 1.3.6.1.4.1.311.21.54 1.2.3.4.89
	cert.ext.nc	exclude: -1; include as non-critical: 0; include as critical: 1

Name Constraints (CA Certificate only)	cert.ext.nc.append	(optional) if set to 1, cert.ext.nc.include.values and cert.ext.nc.exclude.values settings are appended to the default setting; if POSTed DN or URI values match the default settings, the POSTed minimum and maximum values replace the default values
	cert.ext.nc.include.values	included subtree; minimum; maximum (optional) (<DN or URI>![<minimum>![<maximum>]])* example: infoseccorp.com!0! isc.com!1!3
	cert.ext.nc.exclude.values	excluded subtree; minimum; maximum (optional) (DN or URI>![<minimum>![<maximum>]])* example: O=ABC, C=US!0!
OCSP no Revocation Checking	cert.ext.ocspNoCheck	exclude: -1; include as non-critical: 0; include as critical: 1
Certificate Template Name	cert.ext.ctn	exclude: -1; include as non-critical: 0; include as critical: 1
	cert.ext.ctn.value	a certificate template name
Custom Extensions	cert.ext.custom	an opaque base64-encoded PDU containing custom extensions

7.2.1.4 Sample

To submit a certificate request to the CA account “testca” and specified the subject DN “CN=User 1, O=ISC, C=US”, post the following data using the Java program:

```
action=enrollKey&ca=testca&request=MIICrzCCAzcCAQAwajELMAkGA1UEBhMCVVMxIjAgBgNVBAoTG
UluZm9ybWFOaW9uIFNlY3VyaXR5IENvcnAxZDZANBgNVBAwTB1Rlc3RlcjEmMCQGA1UEAxMdQ2VydEFnZW50I
EtlesSBFbnJvbGxtZW50IFRlc3QwggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDI%2BovE1NM3wzk
SLDQUVEQkISQ923J1N6lY3nI2DnrYfJcYGAY8NhtMwQvv2t2TSCL5vEHbD166t0mi7Ka8dW2QK93USDEdE3j
LdOoQ5RZb%2BoC9644%2FvJJrt%2BAs%2FQ0yU9Ba6YAvqgvedGt7jJNgFRTbI%2Bu6Y%2Bxe49y2yC52k7G
qN31eVxhspqkgOerDWfp9muYfAs6SMrT7rEkNUsNsXj6UQjKjjmBi3He7UDrA7qADuLHwrXGqlb7h xv0u0rY
pXX3NmnoF8UzfiP0e06%2BZToqF1uowRSazWZ43QN72yZhFNK9fmFErXglFLWferui5joLJBHhWs%2F1%2F
fZjMZtQ4xZ5AgMBAAGgADANBgkqhkiG9w0BAQUFAAOCAQEApRpnjFad3CR6VVzZbDkZI6osTnzpOyi4W46Ja
7SO5Trf%2BEFOXchHakRVHDh4RqhClJf%2BKr%2F9jR4stQeEib4S%2BirTuyve3c57111xfEQug3KZ347RG
a4N%2BxUQv%2BFX2cRqzUgMTI2v3dwDlasntsIP5d7xLkhX7QotN%2FCuwtPFWV5hdhgSzzvFrqlYgvlWYU
oDt8Ch2yXORfkjhxOGXrBIFiyPrkMees%2B7BzkJRiACp6affD%2BjQz5a3rDkIZE7zSOBLvIHAgeKU7X4Ir
%2FLg4R7AMz6X%2Bzwq%2F310moPWDxJgItONZWEVzoTYECPx04%2BLQdMuln%2Bljx%2Bqy7idYyggBw2w%
3D%3D&cert.rdn=CN%3DUser+1%2C+O%3DISC%2C+C%3DUS
```

Alternatively, use the curl command to post the data and save the response to a file “response.txt”:

```
# convert the binary certificate request to base64-encoded format
openssl base64 -in <binary request file> -out <base64 request file>

# post the action, ca, and request values
curl <url> --cert <client pem file> -v --cacert <root pem file> --tlsv1.2 --data-
urlencode "action=enrollKey" --data-urlencode "ca=testca" --data-urlencode
"cert.rdn=CN=User 1, O=ISC, C=US" --data-urlencode request@<base64 request file> -o
response.txt
```

7.2.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.2.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	text/html Valid information was posted. Request status, detail, and other information will be returned. See the next section for details
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

7.2.2.2 Response Format

Parameter	Format and Description
requestStatus	A code indicating the current status of the request: 0 (success): processing the request was successful and a certificate was issued 1 (error): an error occurred while processing the request 2 (issued and error): an error occurred but the request was successfully processed and a certificate was issued
detail	The detailed information regarding the processing of the client's request
requestID	The request ID of the certificate request
serialNo	The serial number of the issued certificate
base64CertChain	(Optional) The issued certificate and chain in base64-encoded PKCS#7 format Specified if 'response.cert.format' parameter is not set or set to 1 or 3
base64Cert	(Optional) The issued certificate in base64-encoded X.509 format Specified if 'response.cert.format' parameter is set to 2 or 3

7.3.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.3.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	text/html Valid information was posted. Request status and details will be returned. See next section for details
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

7.3.2.2 Response Format

Parameter	Format and Description
requestStatus	0 (success): certificate was revoked 1 (error): an error occurred during processing of the revocation action 2 (revoked with error): an error occurred but the certificate was successfully revoked anyway
detail	The detailed information regarding the processing of the client's request

7.3.2.3 Sample

The following are two possible responses illustrating success and failure, respectively:

```
requestStatus=0
detail=Certificate status changed to Revoked - Key Compromise.
```

```
requestStatus=1
detail=Certificate has already been revoked.
```

7.4 Reinstating a Certificate

7.4.1 Request

To reinstate an on-hold or pending revocation certificate, POST the following information to the CertAgent RAMI.

Parameter	Format and Description
action	"reinstateCert"
ca	CA login name

serial	Serial number of the certificate to be reinstated
--------	---

7.4.1.1 Sample

To reinstate an on-hold certificate issued by the CA account “testca” and with serial number “63489B00000000000000000000000000”, post the following data using the Java program:

```
action=reinstateCert&ca=testca&serial=63489B00000000000000000000000000000002
```

Alternatively, use the curl command:

[illegible]

7.4.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.4.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	text/html Valid information was posted. Request status and detail will be returned. See the next section for details
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

7.4.2.2 Response Format

Parameter	Format and Description
requestStatus	0 (success): Certificate has been reinstated. 1 (error): Error occurred during the processing.
detail	The detailed information regarding the processing of the client's request

7.4.2.3 Sample

The following are two possible responses illustrating success and failure, respectively:

```
requestStatus=0
```

```
detail=Certificate status changed to Valid
```

```
requestStatus=1  
detail=Certificate is valid.
```

If the information posted was invalid (e.g., invalid CA account was specified), HTTP status code 400 (bad request) and error message will be returned in the response.

7.5 Issuing a CRL

7.5.1 Request

To issue a CRL, POST the following information to the CertAgent RAMI.

Parameter	Format and Description
action	"issueCRL"
ca	CA login name

7.5.1.1 Sample

To issue a CRL by the CA account "testca", post the following data using the Java program:

```
action=issueCRL&ca=testca
```

Alternatively, use the curl command:

```
curl <url> --cert <client pem file> -v --cacert <root pem file> --tlsv1.2 --data-  
urlencode "action=issueCRL" --data-urlencode "ca=testca"
```

7.5.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.5.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	text/html Valid information was posted. Request status and detail will be returned. See the next section for details
400 (Bad Request)	text/html

	Invalid information was posted (e.g., invalid CA account was specified); error message will be returned
--	---

7.5.2.2 Response Format

Parameter	Format and Description
requestStatus	0 (success): CRL has been issued. 1 (error): Error occurred during the processing.
detail	The detailed information regarding the processing of the client's request

If the information posted was invalid (e.g., invalid CA account was specified), HTTP status code 400 (bad request) and error message will be returned in the response.

7.5.2.3 Sample

The following are two possible responses illustrating success and failure, respectively:

```
requestStatus=0
detail=CRL issued
```

```
requestStatus=1
detail=Invalid CA certificate: Certificate expired or not yet valid
```

7.6 Listing CA Account Names

7.6.1 Request

To query the CA accounts in which the user is authorized to manage, POST the following information to the CertAgent RAMI.

Parameter	Format and Description
action	"getAuthCANames"

7.6.1.1 Sample

Post the following data using the Java program:

```
action=getAuthCANames
```

Alternatively, use the curl command to post the data and save the response to a file "names.txt":

```
curl <url> --cert <client pem file> -v --cacert <root pem file> --tlsv1.2 --data-  
urlencode "action=getAuthCANames" -o names.txt
```

7.6.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.6.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	application/json Valid information was posted. A JSON array object containing JSON objects will be returned. See the next section for details. If no accounts can be managed by the user, an empty response body will be returned

7.6.2.2 Response Format

The JSON object format is as follows:

Parameter	Format and Description
accountID	String CA account name
displayName	String Display name of the CA account
masterID	(Optional) String Master CA account name; if specified, this is a profile account; otherwise, a master account
ramiRight	Boolean True if user is an authorized RA with the RAMI right False if user is an authorized user but without the RAMI right
ramiSetting	(Optional) Integer An XOR'ed combination of the following values: <ul style="list-style-type: none"> • Allow key enrollment: 1 • Allow POST to override default settings: 2 • Allow certificate revocation and reinstatement: 4 • Allow CRL issuance: 8 • Allow certificate request and certificate queries: 16

7.6.2.3 Sample

```
[
  {
    "accountID":"testca",
    "displayName":"Test CA Master Profile",
    "ramiRight":true,
    "ramiSetting":31
  },
  {
    "accountID":"userprofile",
    "displayName":"User Certificate Profile",
    "masterID":"testca",
    "ramiRight":true,
    "ramiSetting":31
  },
  {
    "accountID":"npeprofile",
    "displayName":"NPE Certificate Profile",
    "masterID":"testca",
    "ramiRight":true,
    "ramiSetting":31
  },
  {
    "accountID":"norami",
    "displayName":"Test CA no RAMI access",
    "ramiRight":false
  }
]
```

7.7 Listing Certificate Requests

7.7.1 Request

To query the certificate request information, POST the following information to the CertAgent RAMI.

Parameter	Format and Description
action	"listRequests"
ca	CA login name
type	(optional) if specified, only the information of the specified type of requests will be returned; otherwise, information of all the requests will be returned 0: pending 1: processed 2: rejected

7.7.1.1 Sample

To get the information of all pending requests of the CA account "testca", post the following data using the Java program:

```
action=listRequests&ca=testca&type=0
```

Alternatively, use the curl command to post the data and save the response to a file “requests.txt”:

```
curl <url> --cert <client pem file> -v --cacert <root pem file> --tlsv1.2 --data-  
urlencode "action=listRequests" --data-urlencode "ca=testca" --data-urlencode  
"type=0" -o requests.txt
```

7.7.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.7.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	application/json Valid information was posted. A JSON array object containing JSON objects will be returned. See the next section for details. If no matching requests found, an empty response body will be returned
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

7.7.2.2 Response Format

The JSON object format is as follows:

Parameter	Format and Description
requestID	String Certificate Request ID
type	Number Status of the requests: 0: pending 1: processed 2: rejected
subjectDN	String Subject DN of the certificate request
keyType	String Key type and size Example: RSA2048
lastModDate	String Last modified date in MM/dd/yy HH:mm:ss zZ format Example: 08/03/16 16:04:02 CDT-0500

7.7.2.3 Sample

```
[
  {
    "keyType": "RSA2048",
    "lastModDate": "08\03\16 16:04:02 CDT-0500",
    "requestID": "020019B1E7C665A298FEFCFF779BAFC7B8A637E8",
    "subjectDN": "CN=Test 1, O=Information Security Corp, C=US",
    "type": 0
  },
  {
    "keyType": "RSA2048",
    "lastModDate": "08\02\16 11:50:47 CDT",
    "requestID": "0847BB439403051D5BC5EEC177F73EFE6EA19390",
    "subjectDN": "CN=Test 2, O=Information Security Corp, C=US",
    "type": 0
  }
]
```

7.8 Listing Certificates

7.8.1 Request

To query the certificate information, POST the following information to the CertAgent RAML.

Parameter	Format and Description
action	"listCerts"
ca	CA login name
type	(optional) if specified, only the information of the specified type of certificates will be returned; otherwise, the information of all certificates will be returned 0: valid user certificate 1: pending revocation user certificate 2: revoked user certificate 3: expired user certificate

7.8.1.1 Sample

To get the information of all certificates of the CA account "testca", post the following data using the Java program:

```
action=listCerts&ca=testca
```

Alternatively, use the curl command to post the data and save the output to a file "certinfo.txt":

```
curl <url> --cert <client pem file> -v --cacert <root pem file> --tlsv1.2 --data-urlencode "action=listCerts" --data-urlencode "ca=testca" -o certinfo.txt
```

7.8.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.8.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	application/json Valid information was posted. A JSON array object containing JSON objects will be returned. See the next section for details. If no matching certificates found, an empty response body will be returned
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

7.8.2.2 Response Format

The JSON object format is as follows:

Parameter	Format and Description
requestID	String Certificate request ID
serial	String Serial number of the certificate
subjectDN	String Subject DN of the certificate
notBeforeDate	String notBefore date in MM/dd/yy HH:mm:ss zZ format Example: 08/03/16 16:04:02 CDT-0500
notAfterDate	String notAfter date in MM/dd/yy HH:mm:ss zZ format Example: 08/03/16 16:04:02 CDT-0500
type	Integer Certificate type/status: 0: valid 1: pending revocation 2: revoked 3: expired
status	Integer Certificate status 0: valid 2: revoked 6: on hold
reason	(optional) Integer reason code for revoked or on-hold status

	<p>If status=2:</p> <p>0: unspecified</p> <p>1: key compromise</p> <p>2: CA compromise</p> <p>3: affiliation changed</p> <p>4: superseded</p> <p>5: cessation of operation</p> <p>8: remove from crl</p> <p>9: privilege withdrawn</p> <p>10: AA compromise</p> <p>If status=6:</p> <p>1: none</p> <p>2: call issuer</p> <p>3: reject</p> <p>4: pickup token</p>
<code>revocationDate</code>	<p>(optional) String</p> <p>Revocation date in MM/dd/yy HH:mm:ss zZ format</p> <p>Example: 08/03/16 16:04:02 CDT-0500</p>

NOTE: Not all type/status/reason code combinations can occur. At present, only the following combinations will be used:

Type	Status	Reason	Description
0	0	0	Valid
1	2	0-5, 8-10	Revoked certificate that has not (yet) appeared on a CRL
1	6	1- 4	On hold certificate that has not (yet) appeared on a CRL
2	2	0-5, 8-10	Revoked certificate that has appeared on at least one CRL
2	6	1- 4	On hold certificate that has appeared on at least one CRL
3	0	0	Expired certificate that was valid prior to expiration
3	2	0-5, 8-10	Expired certificate that was revoked prior to its expiration
3	6	1- 4	Expired certificate that was on hold prior to its expiration

7.8.2.3 Sample

Sample response containing the information of one valid and one revoked certificate:

```
[
  {
    "notAfterDate": "08\02\18 11:50:47 CDT-0500",
    "notBeforeDate": "08\02\16 11:50:47 CDT-0500",
    "requestID": "0847BB439403051D5BC5EEC177F73EFE6EA19390",
    "status": 0,
    "subjectDN": "CN=Valid Cert, O=Information Security Corp, C=US",
    "type": 0,
    "serial": "4A286B000000000000000000000000000000000005"
  },
  {
    "notAfterDate": "07\29\18 14:35:49 CDT-0500",
    "notBeforeDate": "07\29\16 14:35:49 CDT-0500",
    "requestID": "57AF09755561E9584095091F440E186CF45552DF",
    "reason": 1,
    "status": 2,
    "subjectDN": "CN=Revoked Cert, O=ISC, C=US",
    "revocationDate": "08\04\16 15:43:52 CDT-0500",
    "type": 1,
    "serial": "6F6A6D000000000000000000000000000000000001"
  }
]
```

7.9 Retrieving a Certificate

7.9.1 Request

To retrieve an issued certificate, POST the following information to the CertAgent RAML.

Parameter	Format and Description
action	"getCert"
ca	CA login name
serial	Serial number of the certificate

7.9.1.1 Sample

To retrieve the certificate with serial number "57CB860000000000000000000000000001" which was issued by the CA account "testca", post the following data:

```
action=getCert&ca=testca&serial=57CB860000000000000000000000000001
```

Alternatively, use the curl command to post the data and save the certificate to a file “cert.der”:

[illegible]

7.9.2 Response

After POSTing its request, the client process should read the response information written to the output stream of the open connection by the server.

7.9.2.1 Status Code and Content Type

Code	Content-Type and Description
200 (OK)	application/pkix-cert Valid information was posted. A binary X.509 DER-encoded certificate will be returned
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

8 The Database Access Service

CertAgent's Database Access Service (DBAccess) can be used by remote clients to:

- execute SQL Select queries (as well as Create and Drop Index commands) against the integrated certificate database on behalf of those CA accounts
- retrieve the CA account names
- retrieve the subject DN of the specified CA's current certificate
- retrieve the audit trail records of the admin site and CA accounts
- update the contact email addresses that are associated with the certificate records

Requests are transmitted to the server over a TLS-secured connection (with client authentication) by clients who are authenticated against an ACL maintained by the administrators; authentication must succeed or the requests are not processed.

The DBAccess library that encapsulates this functionality and exports an API that is available to authorized client processes can be found, together with sample Java code illustrating its use, in the `<ca home>/tools/dbaccess` folder.

The following table describes the method names, required permissions, ACL, and description for the supported requests:

Method Name	Permission	ACL	Description
executeQuery	DBAccess	CA account	Query the certificate table
executeUpdate	DBAccess	CA account	Create or drop index on the certificate table
getIndexInfo	DBAccess	CA account	Query the index information of the certificate table
replaceContactEmails	DBAccess	CA account	Replace a contact email associated with a new one for all certificate records
replaceContactEmailsBySerial	DBAccess	CA account	Replace a contact email associated with the certificate record that matches the serial number
queryCAAuditTrail	audit	CA account	Retrieve the audit trail records of the CA account
getCADN	admin	Admin	Retrieve the subject DN of the specified CA's current certificate
getCAName	admin	Admin	Retrieve the CA account names
queryAdminAuditTrail	audit	Admin	Retrieve the audit trail records of the Admin site

For details on assigning a client certificate to an ACL with appropriate permission, see the sections entitled *Managing the Server Administration Access Control List* and *Managing an Existing CA Account* in the CertAgent Administrator Guide (ca_adminhelp.pdf).

8.1 Developing a Java Client

To aid in the development and deployment of your Java client, you may copy the DBAccess folder (containing the Java library and sample program) from the `<ca_home>/tools/dbaccess` folder on the CertAgent server to the client system. You may cut and paste the supplied sample code from your application and then link it to the DBAccess library. See the supplied API documentation for usage details.

For a Java client application to successfully use the DBAccess API, the following requirements must be satisfied:

- client credentials (certificate and private key) must be available on an attached HSM, in a local PKCS#12 file, or in a Java keystore file
- the client must possess the passwords for its own private key and the keystore in which it resides, or the HSM PIN if the credentials are stored on an HSM
- the client certificate must be installed into the ACLs with appropriate permission for which the client wishes to submit the request
- the client's Java trust keystore must contain a trust anchor for the server's TLS certificate (*i.e.*, the root certificate for path validation of the server's SSL credentials)
- the client must provide the host address and the TLS admin port for the server

NOTE: Java does not support PKCS#12 that uses AES-256 to encrypt the private key. If you are planning to use CertAgent temporary credentials (e.g., `ca-admin.p12`) as the client credentials and AES-256 cipher was selected during the installation, run the following OpenSSL commands to create a new PKCS#12 that uses DES3 to encrypt the private key. Use the new credentials in your Java program.

```
openssl pkcs12 -in ca-admin.p12 -out ca-admin.pem
openssl pkcs12 -export -in ca-admin.pem -out ca-admin-des3.p12
```

8.2 Using the DBAccess API

A sample Java program illustrating use of the DBAccess service is provided in the package:

```
<ca_home>/tools/dbaccess/DBAccessSample.java.
```

The following outline explains what is going on the sample program and how to build it:

1. The client first loads its credentials into a Java `KeyStore` object.

For credentials located in a PKCS#12 file, a `KeyStore` object is instantiated and initialized with the full pathname of the PKCS#12 file and its password:

```
KeyStore ks = KeyStore.getInstance("PKCS12");
String caUserP12 = keystoreDir + "ca-admin.p12";
String caUserP12Password = "Password123456!";
ks.load(new FileInputStream(caUserP12), caUserP12Password.toCharArray());
```

2. Specify an alias for your private key.

```
String alias = "1";
```

If you don't know the alias of your key entry, the following code can be used to print all the aliases in the `KeyStore` object.

```
Enumeration<String> en = ks.aliases();
while (en.hasMoreElements())
{
    System.out.println(en.nextElement());
}
```

3. If a custom trust keystore is used, assign the trust keystore filename and its password to the following Java system properties.

```
// trust key store
String trustKeyStore = keystoreDir + "ca-root.ks";
// trust key store password
String trustKeyStorePassword = "password";

System.setProperty("javax.net.ssl.trustStore", trustKeyStore);
System.setProperty("javax.net.ssl.trustStorePassword", trustKeyStorePassword);
```

4. To obtain a list of CA account names, the client instantiates a `DBAccess` object with the server's hostname (or IP address), its admin port, a CA account name set to null, the `KeyStore` object constructed above and the key alias and password required for the client's private key. Use the `getCAName(false)` method to obtain the CA account names. To obtain a list of CA account names and their profiles, use the `getCAName(true)` method.

NOTE: The client is required to have an 'admin' permission for the Admin site.

```
DBAccess db = DBAccess(host, port, null, ks, alias, keyPassword);
String[] caNames = dbaccess.getCANames(false);
```

5. To obtain the subject DN of the CA certificate, use the `getCADN` method.

NOTE: The client is required to have an 'admin' permission for the Admin site.


```
String caDN = adminDBAccess.getCADN(caName);
if (caDN != null)
{
    System.out.println("    Found DN: " + caDN);
}
```

6. To query the audit database of the Admin site to return the specified number of latest audit records, use the `queryAdminAuditTrail` method with the desired number of records, fields, and sorting order to be returned.

NOTE: The client is required to have an 'audit' permission for the Admin site.

For information on the CertAgent Audit database schema, see the section entitled *Audit Table*.

```
int rows = 10;
boolean sortAscending = true;
boolean includeTime = true;
boolean includeType = true;
boolean includeServerIP = false;
boolean includeClientIP = false;
boolean includeLevel = false;
boolean includeClientID = true;
boolean includeEvent = true;
DBAccessResultSet rrs0 = adminDBAccess.queryAdminAuditTrail(rows, includeTime,
includeType, includeServerIP, includeClientIP, includeLevel, includeClientID,
includeEvent, sortAscending);

while (rrs0.next())
{
    System.out.println("-----");
    System.out.println("Time: " + rrs0.getTimestamp(1));
    System.out.println("Type: " + rrs0.getInt(2));
    System.out.println("Client ID: " + rrs0.getString(3));
    System.out.println("Event: " + rrs0.getString(4));
}
```

7. To query the Certificate database, the client either instantiates a `DBAccess` object with the server's hostname (or IP address), its admin port, a CA account name, the `KeyStore` object constructed above and the key alias and password required for the client's private key:

```
DBAccess caDBAccess = DBAccess(host, port, ca, ks, alias, keyPassword);
```

or instantiates one without specifying the CA account name in the constructor and use `setCA(String ca)` method:

```
DBAccess caDBAccess = DBAccess(host, port, null, ks, alias, keyPassword);
db.setCA(ca);
```

8. Optionally, use the `validate` method to validate the settings.

```
db.validate();
```

9. Each CA account has its own certificate table. Use `DBSchema.getTable_name()` method to retrieve the table name of a particular CA account. The client queries the specified certificate database using a SQL SELECT statement with the `DBAccess::executeQuery()` method; the result set will be returned in a `DBAccessResultSet` object. For current information on the Certificate database schema, see the section entitled *Certificate Table*. An exception is thrown at runtime if the client is not authorized to query the database of the specified CA account, the SQL statement is invalid, or the CertAgent server is not running.

NOTE: The client is required to have 'DBAccess' permission for the CA account.

```
DBAccessResultSet rrs = caDBAccess.executeQuery("SELECT SERIAL, DN, REPLYEM,
NOTBEFOREDATE, NOTAFTERDATE FROM " + DBSchema.getTable_name(ca) + " WHERE
TYPE=0 ORDER BY NOTBEFOREDATE");
```

10. The client obtains the entire result set by repeatedly calling the `DBAccessResultSet::next()` method until its return value indicates that no additional results are available. For details on the `DBAccessResult` class, please consult the DBAccess API documentation.

```
while (rrs.next())
{
    System.out.println("-----");
    System.out.println("Serial: " + rrs.getString(1));
    System.out.println("DN: " + rrs.getString(2));
    System.out.println("Reply email: " + rrs.getString(3));
    System.out.println("Not before date: " + rrs.getTimestamp(4));
    System.out.println("Not after date: " + rrs.getTimestamp(5));
}
```

To obtain the number of data rows returned from the query, call the `DBAccessResultSet::getSize()` method.

11. To count the number of rows in the table with a particular property, use `COUNT(*)`. For example, to count number of expired certificates, execute the following query:

```
DBAccessResultSet rrs = caDBAccess.executeQuery("SELECT COUNT(*) FROM " +
DBSchema.getTable_name(ca) + " WHERE TYPE=3");
```

The returned value type for `COUNT` is integer.

```

if (rrs.next())
{
    int count = rrs.getInt(1);
    System.out.println("Result: " + count + " found");
}

```

12. By default, the database is indexed by SERIAL and DN with index name CA_CERTI_<ca name>. To create a new index, invoke `DBAccess::executeUpdate()` with a CREATE INDEX statement as in the following example:

```

caDBAccess.executeUpdate("CREATE INDEX test on ENTRY
(SERIAL,DN,NOTBEFOREDATE,NOTAFTERDATE)");

```

To query the index names, use the `getIndexInfo(true)` method:

```

DBAccessResultSet set = caDBAccess.getIndexInfo(true);
while (set.next())
{
    System.out.println("Name: " + set.getString(1));
}

```

To query the index details, use the `getIndexInfo(false)` method:

```

DBAccessResultSet set2 = caDBAccess.getIndexInfo(false);
while (set2.next())
{
    System.out.println("Name: " + set2.getString(1));
    System.out.println("Column Name: " + set2.getString(2));
    System.out.println("Ascending/Descending: " + set2.getString(3));
}

```

To drop an existing index, use a DROP INDEX statement as in:

```

caDBAccess.executeUpdate("DROP INDEX test");

```

For current information on the CertAgent database schema, see the section entitled *CertAgent Database Schema*. An exception is thrown at runtime if the client is not authorized to update the database of the specified CA account, the SQL statement is invalid, or the CertAgent server is not running.

13. To update the email addresses associated with the certificate records used for certificate renewal notification, use one of the methods: `replaceContactEmails(String oldAddress, String newAddressList)` and `replaceContactEmailsBySerial(String serial, String newAddressList)`.

NOTE: The client is required to have a 'DBAccess' permission for the CA account.

To replace an existing email address in the contact email address list with a new one for all certificate records:

```
String oldAddress = "old@xyz.com";
String newAddressList = "new@xyz.com";
int count = caDBAccess.replaceContactEmails(oldAddress, newAddressList);
```

The number of updated records will be returned.

To replace an existing email address with more than one email address, assign a comma delimited list (e.g., "new@xyz.com,new2@xyz.com") to the `newAddressList` parameter. To remove an existing email address, assign `null` to the `newAddressList` parameter.

To replace any empty contact email address list with a new list for all certificate records, assign `null` to the `oldAddress` parameter.

To update the email address list of the certificate matches to the serial number, use the `replaceContactEmailsBySerial` method.

```
caDBAccess.replaceContactEmailsBySerial(serial, newAddressList);
```

An exception is thrown at runtime if the client is not authorized to update the database of the specified CA account, or the serial number does not exist.

14. To query the audit database of a CA account to return the specified number of latest audit records, use the `queryCAAuditTrail` method with the desired number of records, fields, and sorting order to be returned.

NOTE: The client is required to have an 'audit' permission for the CA account. For information on the CertAgent Audit database schema, see the sections entitled *Audit Table* and *CertAgent Database Schema*.

```

int rows2 = 10;
boolean sortAscending2 = false;
boolean includeTime2 = true;
boolean includeType2 = true;
boolean includeServerIP2 = false;
boolean includeClientIP2 = false;
boolean includeLevel2 = false;
boolean includeProfile2 = false;
boolean includeClientID2 = true;
boolean includeEvent2 = true;
DBAccessResultSet rrs3 = caDBAccess.queryCAAuditTrail(rows2, includeTime2,
includeType2, includeServerIP2, includeClientIP2, includeLevel2,
includeProfile2, includeClientID2, includeEvent2, sortAscending2);

while (rrs3.next())
{
    System.out.println("-----");
    System.out.println("Time: " + rrs0.getTimestamp(1));
    System.out.println("Type: " + rrs0.getInt(2));
    System.out.println("Client ID: " + rrs0.getString(3));
    System.out.println("Event: " + rrs0.getString(4));
}

```

15. The client application may be compiled using Java 1.8 or above by running the following command:

```

javac -classpath ./certagentdbaccess.jar:. DBAccessSample.java      (UNIX)
javac -classpath .\certagentdbaccess.jar;. DBAccessSample.java      (Windows)

```

16. Finally, execute the client program using the following command:

```

java -classpath ./certagentdbaccess.jar:. DBAccessSample           (UNIX)
java -classpath .\certagentdbaccess.jar;. DBAccessSample           (Windows)

```

8.3 Supported SQL Syntax

At present the `DBAccess::executeQuery()` method only accepts SQL SELECT statements in the following form:

```

SELECT ((<columns> | *) | COUNT(*)) FROM <table name> [WHERE <expression>]
[GROUP BY <column>] [ORDER BY <column> [ASC | DESC]]

```

and only CREATE INDEX and DROP INDEX statements in the following forms:

```

CREATE INDEX <index> ON <table name> (<column>[,<column>])
DROP INDEX <index>

```

8.3.1 CertAgent Database Schema

8.3.1.1 Certificate Table

The following table describes the existing columns in the integrated CertAgent certificate table.

Column	Format and Description
Serial	String serial number of the certificate; primary key of the table
ReqID	String certificate request ID
Type	int certificate type/status: 0: valid user certificate 1: pending revocation user certificate 2: revoked user certificate 3: expired user certificate For example: To return only valid user certificates, specify "Type=0" in the WHERE clause
DN	String subject DN of the certificate
Status	int certificate status 0: valid 2: revoked 6: on hold
Reason	int reason code for revoked or on-hold status If status=0: 0: valid If status=2: 0: unspecified 1: key compromise 2: CA compromise 3: affiliation changed 4: superseded 5: cessation of operation 8: remove from crl 9: privilege withdrawn 10: AA compromise If status=6: 1: none 2: call issuer 3: reject 4: pickup token

RevocationDate	Timestamp revocation date NOTE: RevocationDate is null for a valid certificate, changed to the revocation date when the certificate is revoked or put on hold, and reset to null when the status of an on-hold certificate is changed back to valid
NotBeforeDate	Timestamp notBefore date
NotAfterDate	Timestamp notAfter date
KeyID	int ID of the certificate issuer
Cert	byte[] byte array containing the ASN.1 DER-encoded certificate
Pick	int certificate retrieval status:0: not yet retrieved 1: retrieved NOTE: retrieval status no longer supported since version 6.4.1, value 1 will always be returned from a query
ReplyEM	String list of user's contact email addresses delimited by comma null if no email addresses are provided
Profile	String account name of assigned profile null if certificate is assigned to the master account NOTE: If profile exists, use "profile is null" in the WHERE clause to return certificates assigned to master account
DayToExpire	int number of days before expiration on which to send renewal notification; only set if the "notify user if certificate will expire" option is enabled default is -1

NOTE: Not all type/status/reason code combinations can occur. At present, only the following combinations will be used:

Type	Status	Reason	Description
0	0	0	Valid
1	2	0-5, 8-10	Revoked certificate that has not (yet) appeared on a CRL
1	6	1- 4	On hold certificate that has not (yet) appeared on a CRL
2	2	0-5, 8-10	Revoked certificate that has appeared on at least one CRL
2	6	1- 4	On hold certificate that has appeared on at least one CRL
3	0	0	Expired certificate that was valid prior to expiration
3	2	0-5, 8-10	Expired certificate that was revoked prior to its expiration
3	6	1- 4	Expired certificate that was on hold prior to its expiration

8.3.1.2 Audit Table

The following table describes the existing columns in the integrated CertAgent Audit table for the Admin Site.

Column	Format and Description
TYPE	int Type of the event: 1: credentials 2: PIN 4: ACL 8: audit 16: login 32: database 64: job 128: CA account 256: email 512: NIAP 1024: DBAccess 2048: System 4096: TLS session
SERVER	String IP address of the CertAgent system
CLIENT	String IP address of the client system, CACLI, or NULL (for the events that are triggered by the system)
LDATE	Timestamp Timestamp of the event
LLEVEL	Int Level of the event: 1: error 3: information
EVENT	String Recorded event
ClientID	String The identity of the client: Subject DN of an authorized user's certificate, CACLI, Startup Script, or NULL (for the events that are triggered by the system)

The following table describes the existing columns in the integrated CertAgent Audit table for the CA account.

Column	Format and Description
TYPE	int 1: request 2: certificate 3: CRL 4: OCSP 5: user 6: login 7: credential 8: RAMI 9: DBAccess 10: config 11: EST
SERVER	String IP address of the CertAgent system
CLIENT	String IP address of the client system, EST user name, CACLI, or NULL (for the events that are triggered by the system)
LDATE	Timestamp Timestamp of the event
LLEVEL	Int Level of the event: 1: error 3: information
EVENT	String Recorded event
CLIENTID	String The identity of the client: Subject DN of an authorized user's certificate, EST user name, CACLI, or NULL (for the events that are triggered by the system)

8.3.2 Sample SQL statements

Return all columns of rows in which subject DN contains “bob”:

```
SELECT * FROM CA_CERT_A WHERE DN like '%bob%'
```

Count the number of valid certificates:

```
SELECT COUNT(*) FROM CA_CERT_A WHERE TYPE=0
```

CertAgent Installation Guide

Return the specified columns of rows in which valid certificates will be expired on 2015 ordered by notAfter date:

```
SELECT SERIAL,DN,USERCERTIFICATE,NOTAFTERDATE FROM CA_CERT_A WHERE TYPE=0 AND  
(NOTAFTERDATE BETWEEN to_timestamp('01-01-2015','MM-DD-YYYY') AND to_timestamp('01-  
01-2016','MM-DD-YYYY')) ORDER BY NOTAFTERDATE
```

Return the number of certificates and expiration date group by month and year:

```
SELECT to_char(NOTAFTERDATE,'YYYY-MM') as Y,COUNT(*) FROM CA_CERT_A GROUP BY  
to_char(NOTAFTERDATE,'YYYY-MM')
```

Return the specified columns of rows in which certificates are on hold:

```
SELECT SERIAL,DN,REVOCATIONDATE FROM CA_CERT_A WHERE TYPE=1 AND STATUS=6
```

Return the specified columns of rows in which valid certificates have not been retrieved by the users:

```
SELECT SERIAL,DN,REPLYEM FROM CA_CERT_A WHERE TYPE=0 AND PICK=0
```

Create a new index:

```
CREATE INDEX test on ENTRY (SERIAL,DN,NOTBEFOREDATE,NOTAFTERDATE)
```

Drop an index:

```
DROP INDEX test
```

9 Retrieving System Information and CA Resources

CertAgent's `getinfo.jsp` page:

URL: `https://<hostname>:<port>/certagent/getinfo.jsp`

Allows a remote or automated client process to:

- retrieve CertAgent version, system time, and version information.
- retrieve CA certificate and optional chain.
- retrieve a CRL.

The page can be access via the default port for the public site (HTTPS without client authentication) or an HTTP port (if configured manually).

9.1 *Retrieving CertAgent Version and System Information*

9.1.1 *Request*

To retrieve the CertAgent version and system information, submit a GET or POST request to the following URL:

`https://<hostname>:<port>/certagent/getinfo.jsp?type=SYSTEM`

9.1.2 Response

An HTTP status code 200 (OK), a JSON object containing the following information of Content-Type application/json will be returned.

Parameter	Format and Description
CertAgent	String Version number of CertAgent
CDK	String Version number of ISC CDK library
SA	String Version number of ISC SA library
JNI	String Version number of the JNI library
FIPS	Boolean True if the ISC CDK library is in FIPS mode
OS	String Operating system CertAgent is running on: "Windows" or "Linux"
time	String Current date and time of the system in mm/dd/yy hh:mm:ss zZ format Example: 12/03/17 16:04:02 CDT-0600
NIAP	Boolean True if CertAgent is operating in NIAP mode
maintenance	Boolean True if CertAgent is operating in maintenance mode
error	String Optional; Error message if CDK is in error state

9.1.2.1 Sample

```
{
  "CertAgent": "7.0.9",
  "JNI": "7.0.9",
  "SA": "7.7.2.0",
  "CDK": "8.0.0.8",
  "FIPS": true,
  "OS": "Windows",
  "time": "10/29/20 10:30:24 CDT-0500",
  "NIAP": false,
  "maintenance": false
}
```

9.2 Retrieving CA Resources

9.2.1 Request

To retrieve the CA's certificates or CRL, submit a GET or POST request to the following URL:

```
https://<hostname>:<port>/certagent/getinfo.jsp?ca=<ca name>&type=<type>
```

Parameter	Format and Description
ca	CA login name
type	"CA_BIN" "CA_PEM" "CA_P7_BIN" "CA_P7_PEM" "CRL_BIN" "CRL_PEM"

9.2.2 Response

9.2.2.1 Status Code and Content Type

Code	Content Type and Description
200 (OK)	Valid information was posted. See the next section for details
400 (Bad Request)	text/html Invalid information was posted (e.g., invalid CA account was specified); error message will be returned

9.2.2.2 Response Format

Type Value	Response Content Type and Description
CA_BIN	application/pkix-cert CA certificate in binary format
CA_PEM	application/pkix-cert CA certificate in PEM-encoded format
CA_P7_BIN	application/x-pkcs7-mime CA certificate and its chain in binary PKCS#7 format
CA_P7_PEM	application/x-pkcs7-mime CA certificate and its chain in PEM-encoded PKCS#7 format
CRL_BIN	application/pkix-crl CRL in binary format
CRL_PEM	application/pkix-crl CRL in PEM-encoded format

10 Backup and Recovery

10.1 System Files

CertAgent uses a directory tree rooted at `<ca home>/conf` to store system configuration settings, and audit trails. If HyperSQL database has been installed, it uses a directory tree rooted at `<ca home>/hsqldb`. The contents of the tree are described in the following table.

Directory/File	Description
<code><ca home>/conf/.system</code>	system certificates directory
<code><ca home>/conf/logs</code>	admin log directory
<code><ca home>/conf/certagent.xml</code>	current XML configuration file
<code><ca home>/conf/certagent.< yyyy.MM.dd_HH.mm.ss>.xml</code>	backup copies of the configuration files
<code><ca home>/conf/certagent.dtd</code>	DTD file for certagent.xml
<code><ca home>/conf/server.log</code>	server log file
<code><ca home>/hsqldb</code>	HyperSQL database directory (if installed)
<code>C:\Windows\System32\CASRVCTL.exe</code> <code>C:\Windows\System32\casrvctlS.ini</code> <code>C:\Windows\System32\casrvctlK.ini</code>	Windows service and configuration files on Windows system only

To perform a complete or partial backup of your CertAgent system:

1. Stop the CertAgent service according to the instructions found in *Using the System Services*. Please make sure all the services are completely stopped before proceeding to the next step.
2. You may now back up the entire `certagent7`, or `conf` and `hsqldb` directory trees.

The following shows a typical back up process using a UNIX command shell:

```
Backup the entire certagent7 directory
# cd /usr/local
# tar -czvf certagent7-backup.tar.gz certagent7

Backup the conf and hsqldb directories only
# cd /usr/local/certagent7
# tar -czvf certagent7-conf-backup.tar.gz conf hsqldb
```

For Windows system, copy the `C:\Program Files\CertAgent7` or `C:\Program Files\CertAgent7\conf` and `\hsqldb` directories, `C:\Windows\System32\CASRVCTL.exe`, `C:\Windows\System32\casrvctlS.ini` and `C:\Windows\System32\casrvctlK.ini` to a temporary backup directory.

Then, move the backup tar.gz file or backup directory to a backup media that can be accessed in the event of a system failure.

3. Once your backup is complete, you may restart the CertAgent service.

To restore a corrupted CertAgent system from a backup copy:

1. Stop the CertAgent service according to the instructions found in *Using the System Services*. Please make sure all the services are completely stopped before proceeding to the next step.
2. Consider using your standard backup procedures to make another complete backup of the current CertAgent system as additional insurance.
3. Recover any damaged files from a known good backup.
4. Restart the CertAgent service.

10.2 Database

The current CertAgent release requires an Oracle, a PostgreSQL, or a HyperSQL database for storage of all CA account configuration settings, certificate requests, certificates, CRLs, audit trails, and access control lists.

The following database tables and sequences are created by CertAgent:

Table	Sequence	Description
CA_KEY	CA_KEYS	stores all system and CA credentials; sequence applies to Oracle database only
CA_ACCT		stores CA accounts and profiles
CA_ACL	CA_ACLS	stores ACLs for system and account admin sites; sequence applies to Oracle database only
CA_ADMIN_AUDIT		stores audit trails for system admin site
CA_ADMIN_CONFIG		stores configuration settings for system admin site
CA_JOB	CA_JOBS	stores jobs for all CA accounts; sequence applies to Oracle database only
CA_JOB_CONFIG		stores job configuration settings
CA_TRUST	CA_TRUSTS	stores trust anchors certificates for path validation; sequence applies to Oracle database only
CA_CRL	CA_CRLS	stores CRLs for path validation; sequence applies to Oracle database only
CA_CRQ_<ca>		stores certificate requests for a CA account
CA_CERT2_<ca>	CA_CERT2S_<ca>	stores certificates for a CA account; sequence applies to Oracle database only
CA_CRL_<ca>		stores CRLs for a CA account
CA_AUDIT_<ca>		stores audit trails for a CA account
CA_CONFIG_<ca>		stores configuration settings for a CA account

CA_SEQ_<ca>		stores the next values of the serial number and CRL number for CRL number extension; applies to HyperSQL and PostgreSQL databases only
CA_EST_<ca>	CA_ESTS_<ca>	stores the authorized EST user name and password (PBKDFv2/SHA-256 generated check value)
CA_OCSP_<ca>		stores cached responses for a CA account; created if Dhuma is installed
CA_DHUMA		stores Dhuma accounts and info; created if Dhuma is installed
CA_DHUMACRL	CA_DHUMA_CRL_SEQ	stores CRLs for Dhuma accounts; created if Dhuma is installed; sequence applies to Oracle database only
CA_DHUMA_<name>		stores the certificate revocation information and cached responses for a Dhuma account; created if Dhuma is installed
	CA_SERIAL_<ca>	stores the next serial number
	CA_CRL_NUM_<ca>	stores the next CRL number for CRL number extension

The following database tables and indices are created by CertAgent:

Table	Index	Description
CA_CERT_<ca>	CA_CERTI_<ca>	creates index on serial number and DN columns

10.2.1 HyperSQL

To perform a backup of the HyperSQL 2.5.1 CertAgent database:

1. Stop the CertAgent service according to the instructions found in *Using the System Services*. Please make sure all the services are completely stopped before proceeding to the next step.
2. Run the following command as appropriate for your system to back up the CertAgent database:

```
java -cp /usr/local/certagent7/hsqldb/hsqldb.jar
org.hsqldb.lib.tar.DbBackupMain --save /usr/local/certagent7/db-backup.tar
/usr/local/certagent7/hsqldb/db/certagent

java -cp "C:\Program Files\CertAgent7\hsqldb\hsqldb.jar"
org.hsqldb.lib.tar.DbBackupMain --save "C:\Program Files\CertAgent7\db-
backup.tar" "C:\Program Files\CertAgent7\hsqldb\db\certagent"
```

3. Move the backup tar file to a backup media that can be accessed in the event of a system failure.
4. Once your backup is complete, you may restart the CertAgent service.

To restore the database from a backup copy:

1. Stop the CertAgent service according to the instructions found in *Using the System Services*. Please make sure all the services are completely stopped before proceeding to the next step.

2. Rename your existing database directory: `<ca home>/hsqldb/db`
3. Run the following command as appropriate for your system to restore the CertAgent database to `<ca home>/hsqldb/db`.

```
java -cp /usr/local/certagent7/hsqldb/hsqldb.jar
org.hsqldb.lib.tar.DbBackupMain --extract /usr/local/certagent7/db-backup.tar
/usr/local/certagent7/hsqldb/db

java -cp "C:\Program Files\CertAgent7\hsqldb\hsqldb.jar"
org.hsqldb.lib.tar.DbBackupMain --extract "C:\Program Files\CertAgent7\db-
backup.tar" "C:\Program Files\CertAgent7\hsqldb\db"
```

4. Restart the CertAgent service.

Additional information can be found from the HyperSQL database webpage:

http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

10.2.2 PostgreSQL

For information on PostgreSQL 9.4 backup and recovery operations, please consult the database vendor documentation:

<http://www.postgresql.org/docs/9.4/static/backup.html>

10.2.3 Oracle

For information on Oracle 11g backup and recovery operations, please consult the database vendor documentation:

https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm

10.3 HSM

CertAgent's system and CA credentials are stored in an HSM. For information on HSM backup and recovery operations, please consult the HSM vendor.