



Cloud Service Automation

Software Version: 4.80

For Microsoft Windows and Linux operating systems

Configuration Guide

Document Release Date: June 2018

Software Release Date: January 2017



Hewlett Packard
Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© 2010-2016 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

The OpenStack® Word Mark and the Square O Design, together or apart, are trademarks or registered trademarks marks of OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hpe.com/>.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support site at: <https://softwaresupport.hpe.com/>.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: <https://softwaresupport.hpe.com/web/softwaresupport/access-levels>.

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is <https://softwaresupport.hpe.com/km/KM01702731>.

Contents

Configuration Overview	14
Check for updates	14
Getting Started	15
Prepare LDAP for CSA	15
Configure the CSA Truststore Properties	17
Location of the CSA Truststore	17
Request Software Licenses	18
Request a Software License	19
Request a Software License for a Clustered Environment	20
Request a Software License for a System with an Updated IP Address ..	20
Enable TLS on Your Web Browser	20
Chrome, Windows	21
Chrome, Ubuntu	22
Chrome, Red Hat Enterprise Linux	22
Microsoft Internet Explorer	22
Firefox	23
Configure the Provider Organization	23
Add a Software License	24
Configure a Proxy for Resource Providers Outside the Internal Network	24
Update the CSA Service Startup Type on Windows	27
Location of the JRE Installed with CSA on Windows	27
Secure Connections	28
Configure Secure Connections for Client Browsers	29
Configure CSA to Use a Trusted Certificate Authority-Signed or Subordinate Certificate Authority-Signed Certificate	32
Step 1: Create a Keystore and Self-Signed Certificate	33
Step 2: Create a Certificate Signing Request	34
Step 3: Submit the Certificate Signing Request to a Certificate Authority	34
Step 4: Import the Certificate Authority's Root Certificate	34
Step 5: Import the Certificate Authority-Signed Certificate	35

Step 6: Configure the Marketplace Portal	37
Step 7: Configure the Web Server	38
Step 8: Configure Client Browsers	38
Step 9: Test Secure Connections	39
Configure CSA to Use a Certificate Authority-Signed Certificate and a Certificate Authority-Provided Keystore	39
Step 1: Import the Certificate Authority's Root Certificate	40
Step 2: Convert the Certificate Authority-Provided Keystore	41
Step 3: Determine the Alias for the Certificate from the JKS Keystore	42
Step 4: Configure the Marketplace Portal	42
Step 5: Configure the Web Server	43
Step 6: Configure Client Browsers	44
Step 7: Test Secure Connections	44
Configure CSA to Use an Internal Certificate Authority-Signed Certificate	45
Step 1: Import the Certificate Authority's Root Certificate	45
Step 2: Import the Internal Certificate Authority-Signed Certificate	46
Step 3: Configure the Marketplace Portal	48
Step 4: Configure the Web Server	49
Step 5: Configure Client Browsers	49
Step 6: Test Secure Connections	50
Configure CSA to Use a Self-Signed Certificate	50
Step 1: Create a Keystore and Self-Signed Certificate	51
Step 2: Export the Self-Signed Certificate	52
Step 3: Import the Self-Signed Certificate as a Trusted Certificate	53
Step 4: Configure the Marketplace Portal	53
Step 5: Configure the Web Server	55
Step 6: Configure Client Browsers (Optional)	55
Step 7: Test Secure Connections	56
Configure CSA to create a new self-signed certificate for global search	56
Masking Passwords in standalone.xml Using the JBoss vault Script	59
Configure Secure Connections for LDAP	65
Configure Secure Connections for SMTP	66
Configure Secure Connections for an Oracle Database	67
Configure Secure Connections for Microsoft SQL Server	70

Configure Secure Connections for Operations Orchestration Load Balancer	71
Configure Secure Internal Communication	74
Authentication Checking	75
Operations Orchestration	77
Configure OO for Topology Designs	77
Configure an Internal User	78
Deploy Content Packs	79
Configure OO Properties in the csa.properties File	79
Configure a Secure Connection between CSA and OO	81
Run the Cloud Content Capsule Installer	82
Update and Redeploy the Service Manager Base Content Pack	82
Configure Single Sign-On between CSA and OO	85
Configure and Enable HP Single Sign-On	85
Configure LDAP Users for Single Sign-On	86
Obscure Passwords in OO Flows (Optional)	86
Configure OO for Sequential Designs	87
Add a JRE to the System Path	88
Configure Internal Users	89
Deploy Content Packs Required by CSA	90
Set Up System Accounts for the CSA Content Pack	91
Set Up System Properties for the CSA Content Pack	92
Configure a Secure Connection between CSA and Operations Orchestration	92
Run the Cloud Content Capsule Installer	93
Update and Redeploy the Service Manager Base Content Pack	94
Configure HP Single Sign-On between CSA and Operations Orchestration	96
Configure and Enable HP Single Sign-On	97
Configure LDAP Users for Single Sign-On	97
Obscure Passwords in Operations Orchestration Flows (Optional)	98
Configure OO Version 9.07	98
Add a JRE to the System Path	99
Install CSA Flows	100
Set Remote Action Services	101

Configure System Accounts Settings	101
Configure System Properties Settings	102
Configure General System Configuration Settings in Operations Orchestration Central	102
Configure a Secure Connection between CSA and Operations Orchestration	103
Obscure Passwords in Operations Orchestration Flows (Optional) ...	104
Check RAS Timeout Settings (Optional)	105
Change Operations Orchestration REST API Timeout (Optional)	105
Operations Orchestration Root Content Configuration	106
Import Operations Orchestration Flows	107
Cloud Service Management Console	108
Customize the Cloud Service Management Console dashboard	108
Enable the Cloud Analytics Side-Bar Menu	109
Enable the <code>[[[Undefined variable CSAVariables.tabMigrateToCloud]]]</code> Secondary Tiles	111
Prerequisites	112
Configure the Cloud Optimizer Side-Bar Menu	113
Prerequisites	114
Configure the Cloud Optimizer Menu in the Cloud Service Management Console	114
Configure the Cloud Optimizer Health Status for CSA	115
Configure SSL	116
Configure REST API-based Communication to Integrate CSA and Cloud Optimizer	116
Configure the SSL Certificate	117
Configure Cloud Optimizer Notification-based Communication	119
Basic Cloud Optimizer Kafka Configuration	119
SSL Configuration Changes on Cloud Optimizer	121
SSL Configuration Changes on CSA	122
Provider Configuration Changes in CSA	123
Create a Cloud Optimizer Provider	124
Configure the Cloud Optimizer Provider Properties	124
Enable Other Predefined dashboard Tiles	125
Create a Side-Bar Menu	126

Add a Launcher Sub-Menu	130
Modify a Side-Bar Menu	133
Disable a Side-Bar Menu	133
Side-Bar menu Configuration File Syntax	133
Dashboard Widget Configuration	135
Configuration Steps:	135
Add a Custom Widget	140
Customize the Cloud Service Management Console Font	141
Customize the Cloud Service Management Console Title	142
Rename or Delete the Sample Consumer Organization	143
Configure HTML Email Notifications	145
Configure the Notification Properties	145
Configure the Default Notification Templates	146
Default HTML Templates	147
Using the Notification Templates	148
Notification Tokens	148
Common Tokens	149
Approval Tokens	150
Subscription Tokens	150
Configure HTML Notification for Pause on Failure	151
Set options in the UI	151
Pause-on-failure notifications	152
Templates	152
Tokens	153
Customize the Default Notification Templates	153
Unsupported HTML Notification Functionality	155
HTML Template Configuration/Troubleshooting Notes	156
Configure Security Warning Messages for Cloud Service Management Console	157
Enable Verification of an Imported Service Design, Service Offering, or Catalog Content Archive	158
Prerequisites	159
Examples Used in this Section	159
Enable Verification	160
Create a Signed Content Archive	161

Locating or Creating a Keystore and Certificate	161
Signing the Content Archive	162
Re-Sign a Content Archive	163
Common CSA Tasks	165
Launch the Cloud Service Management Console	165
Launch the Marketplace Portal	165
Launch the default Marketplace Portal	165
Launch an organization-specific Marketplace Portal	166
Launch the default remote instance of a Marketplace Portal	167
Launch an organization-specific remote instance of a Marketplace Portal	168
Start CSA	168
Restart CSA	170
Stop CSA	171
Encrypt a password	172
Clear the web browser cache	172
Uninstall CSA	173
Uninstall CSA on Windows	173
The Marketplace Portal	176
Configure Global Search	176
Enable Global Search	177
Disable Global Search	177
Configure the Showback Report Menu	178
Configure the Link to HPE IT Business Analytics	179
Configure HP SSO	179
Encrypt a Marketplace Portal Password	180
Configure Security Warning Messages for Marketplace Portal	181
Configure Uploaded or Downloaded File Security Warning Messages	181
Configure Attach Documents Security Warning Message	182
The OO Designer	183
Integrate CSA with HPE UCMDB	186
Using HPE CSA (CI Type based) Palette	186
Configure CSA for HPE UCMDB Integration	187
CSA Component Types aligned with HPE UCMDB's CI Types	188

Designer Visible	191
CI Type Component Relationship	192
Creating a Custom CSA Component Type	193
User Administration	198
Allow Non-Administrator Users to Start and Stop the CSA, Marketplace Portal, or Global Search Service on Windows	198
Allow the CSA, Marketplace Portal, and Global Search Services to be Run as a Non-Administrator User on Windows	201
Create Non-Administrator Users	201
Configure the Services	202
Configure File System Permissions for the Non-Administrator Users	203
Change CSA Built-In User Accounts	205
Cloud Service Management Console User Accounts	206
Marketplace Portal User Account	215
LDAP Account Lockout Mechanism for the Cloud Service Management Console and Marketplace Portal	216
Configure IPv6	217
Launch the Cloud Service Management Console	218
Common Access Card	219
Stop CSA	220
Update JBoss Configuration to Set Up Client Authentication	220
Configure the Identity Management component	222
Configure Certificate Revocation	227
Configure CSA to Use a Certificate Revocation List	228
Configure CSA to Use a Certificate Revocation List Distribution Point	229
Configure CSA to Use the Online Certificate Status Protocol	229
Restart CSA	229
Single Sign-On	230
Integrate with HP Single Sign-On	230
Disable HP Single Sign-On (HPSSO)	231
Configure the Cloud Service Management Console	231
Configure the Identity Management component	232
Workflow Designer Configuration - SSO	238
Integrate CSA with a Single Sign-On Solution	239
Verify the CSA Provider Organization's LDAP Server Configuration	240

Verify the CSA Consumer Organization's LDAP Server Configuration ..	241
Configure the Custom SSO Server to Work with CSA	242
Stop CSA	242
Configure the Cloud Service Management Console	242
Configure the Marketplace Portal	242
Configure Proxy Mapping	243
Start CSA	243
Verify the Single Sign-On Integration	243
Integrate CSA with CA SiteMinder	243
Configure the CSA Provider Organization's LDAP Server	245
Configure the CSA Consumer Organization's LDAP Server	245
Configure the SiteMinder Policy Server for CSA Integration	246
Configure the SiteMinder Web Agent for CSA Integration	247
Configure CSA for SiteMinder Integration	248
Stop CSA	248
Configure the Cloud Service Management Console	248
Configure the Marketplace Portal	249
Configure the Identity Management component	250
Start CSA	253
Launch the Marketplace Portal	253
Customize the Marketplace Portal Landing Page (Optional)	254
Customize the Logout Page (Optional)	254
Configure the Marketplace Portal to Use the Fully-Qualified Domain Name of the SiteMinder Web Agent (Optional)	256
Request Flow	257
Configure SAML	258
SAML Configuration on a CSA Fresh Install	259
ADFS Group Claim Configuration	261
Rule creation method #1:	261
Rule creation method #2:	262
Exporting the ADFS Certificate and Importing the Certificate in Identity Management component	263
Adding SAML Configuration for the Organization	264
SAML Configuration on a CSA Upgrade	265
Database Administration	266

Restart the Database	266
Configure the CSA Reporting Database User	266
Create the CSAReportingDBUser	267
Edit the CSAReportingDBUser Password	268
Update the CSA Database System	269
Update the CSA Database User or Password	271
Import Large Archives	273
Import Large Archives Using the CSA Content Archive Tool	273
Import Large Archives from the Cloud Service Management Console or through the REST API	274
Purge Service Subscriptions and Audit Data	277
About Service Subscriptions	277
About Audit Data	278
Deleting Service Subscriptions and Audit Data	279
Upgrade or Install a Fresh CSA Database Schema	290
Upgrading or Installing the Database Schema	291
Configure CSA to Mitigate Frequently Dropped Database Connections ...	297
Appendix: Cloud Service Management Console Properties	300
Appendix: Marketplace Portal Attributes	349
Appendix: OO Settings	359
Appendix: Identity Management component Configuration	362
External Configuration	362
Configure Seeded Authentication	363
Configure the Java Relying Party Library	364
IdentityServiceConfig	365
IdentityAuthenticationProvider	365
HeaderAuthenticationProvider	365
Internal Configuration	366
JwtTokenFactory	366
ConvergedLdapAuthConfig	367
ConvergedActiveDirectoryAuthenticationProvider and ConvergedLdapAuthenticationProvider	368
SeededAuthenticationProvider	368
IdentityAuthenticationProvider	369

MultiTenantAuthenticationProvider	369
IdentityServiceImpl	370
IdentityController	371
KeystoneAuthenticationProvider	371
KeystoneSecondaryAuthenticationProvider	372
RestTemplateFactoryImpl	372
TrustFactory	373
Token Store Cleanup Service	374
Appendix: Operations Orchestration Manual Configuration for Designs	375
Manually Configure Operations Orchestration for Topology Designs	375
Configure a Secure Connection between CSA and OO	376
Configure an Internal User	377
Deploy Content Packs	378
Update and Redeploy the Service Manager Base Content Pack	379
Configure OO Properties in the csa.properties File	381
Configure Single Sign-On between CSA and OO	383
Configure and Enable HP Single Sign-On	383
Configure LDAP Users for Single Sign-On	384
Obscure Passwords in OO Flows (Optional)	384
Manually Configure Operations Orchestration for Sequential Designs	385
Add a JRE to the System Path	386
Install the CSA Content Pack	387
Configure Internal Users	387
Deploy Content Packs	389
Update and Redeploy the Service Manager Base Content Pack	390
Set Up System Accounts for the Content Packs	392
Set Up System Properties for the Content Packs	393
Configure a Secure Connection between CSA and OO	393
Configure HP Single Sign-On between CSA and OO	394
Configure and Enable HP Single Sign-On	395
Configure LDAP Users for Single Sign-On	396
Obscure Passwords in OO Flows (Optional)	396
Appendix: Hubot Notifications Integration with CSA	398

Send documentation feedback399

Configuration Overview

The Configuration Overview provides information about how to set up the Cloud Service Management Console and Cloud Service Automation (CSA) to enable users to log in and use the Cloud Service Management Console and Marketplace Portal. Some tasks must be completed before you can start using CSA.

The user who sets up CSA should have knowledge of or work with someone who has knowledge of LDAP, TLS, Operations Orchestration, and the resource providers that will be integrated with CSA.

Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com> (this site requires a Passport ID). Select **Dashboards > Manuals**.

Check for updates

Documentation may be updated periodically. To check for recent updates or to verify you are using the most recent edition of a document, click the "go" link to download the guide from the [HPE Software Support portal](#). HPE Passport login is required.

Document	Link
Configuration Guide	go
Cluster Configuration Guide Using a Load Balancer	go
Cluster Configuration Guide Using an Apache Web Server	go
FIPS 140-2 Compliance Configuration Guide	go
FIPS 140-2 Compliance Statement	go

Getting Started

This chapter provides information for common setup tasks that need to be completed for CSA.

Caution: If you are configuring CSA to be compliant with FIPS 140-2 on Windows, you **MUST** configure FIPS 140-2 compliance before configuring anything else. Do **NOT** configure any other feature of CSA and do not use any of the CSA tools until you have configured CSA to be compliant with FIPS 140-2. See Cloud Service Automation FIPS 140-2 Compliance Configuration Guide for more information.

Tasks include:

- ["Prepare LDAP for CSA" below](#) (required)
- ["Configure the CSA Truststore Properties" on page 17](#) (required)
- ["Request Software Licenses" on page 18](#) (required)
- ["Enable TLS on Your Web Browser" on page 20](#) (required)
- ["Configure the Provider Organization" on page 23](#) (required)
- ["Add a Software License" on page 24](#) (required)
- ["Configure a Proxy for Resource Providers Outside the Internal Network" on page 24](#) (optional)
- ["Update the CSA Service Startup Type on Windows" on page 27](#) (optional)
- ["Location of the JRE Installed with CSA on Windows" on page 27](#) (required)

Prepare LDAP for CSA

CSA supports limited authentication and has a fixed set of user names (and associated passwords) that can be used to log in. This basic form of authentication can be used for initial setup and experimentation with the product, but in a production environment, authentication should be configured to occur against a directory service.

CSA can be configured to authenticate against a Lightweight Directory Access Protocol (LDAP) server. Users can then log in with a pre-existing user name (such as an enterprise email address) and password combination. LDAP authenticates the login credentials by verifying that the user name and password match an existing user in the LDAP directory.

In CSA, LDAP is used to:

- Authenticate a user's login to the Cloud Service Management Console and Marketplace Portal
- Authenticate a user's access to information
- Authorize a user's access to information
- Retrieve information about a user's manager for approvals
- Retrieve information about a user's group membership for approvals

These functions are configured when you configure LDAP and access control for an organization.

Before you configure LDAP for the Cloud Service Management Console or Marketplace Portal, you should be familiar with your enterprise LDAP server and LDAP configuration tasks.

Note: The user object configured in LDAP that is used to log in to CSA and by which users can be identified should be configured to contain the following attribute types:

- **User Email - Required.** This attribute type designates the email address of the user to which to send email notifications. Common LDAP attribute names for email include **mail**, **email**, and **userPrincipalName**. If the value for this attribute in the user object in LDAP is empty or not valid, the user for whom the value is empty or not valid does not receive email notifications.
- **Manager Identifier - Required.** This attribute type identifies the manager of the user. A common LDAP attribute name for a user's manager is **manager**. If the value for this attribute in the user object in LDAP is empty or not valid, approval policies that use the User Context Template will fail.
- **Manager Identifier Value - Required.** This attribute type describes the value of the manager identifier. A common value for the manager identifier in LDAP is the **dn** (distinguished name) of the manager's user object. If the manager's user object cannot be located based on the values for manager identifier and manager identifier value, approval policies that use the User Context Template will fail.

The group object configured in LDAP must contain the following attribute type:

- **Group Membership - Required.** This attribute type identifies a user as belonging to the group. Common LDAP attribute names that convey group membership include **member** and **uniqueMember**.

The attribute names configured in your LDAP directory for these attribute types are used when configuring an organization's LDAP in the Cloud Service Management Console.

Note: Do not create users in your LDAP directory that match the built-in users provided by CSA: `csaCatalogAggregationTransportUser`, `csaReportingUser`, `ooInboundUser`, and

codarintegrationUse. Creating the same users in LDAP may allow the CSA built-in users unintended access to the Cloud Service Management Console or give the LDAP users unintended privileges.

Configure the CSA Truststore Properties

You must configure information about the CSA's keystore. Do the following:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file in a text editor.
2. Enter values for the `csaTruststore` and `csaTruststorePassword` properties.

Property	Description
<code>csaTruststore</code>	Required. The CSA keystore that stores trusted Certificate Authority certificates. Note: On Windows, use only forward slashes (/) as your path separators.
<code>csaTruststorePassword</code>	Required. The encrypted password of the CSA keystore (see "Encrypt a password" on page 172 for instructions on encrypting passwords). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.

For more information about these properties, see ["Appendix: Cloud Service Management Console Properties" on page 300](#).

3. Save and exit the file.
4. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

Location of the CSA Truststore

The location of the CSA truststore depends on the JRE you are using with CSA and where the JRE has been installed.

The following are examples of where the CSA truststore may be located.

- If you are using the JRE that is installed with CSA (OpenJDK JRE), the truststore is located in the following location:

CSA_HOME/openjre/lib/security/cacerts

For example:

Windows: C:\Program Files\HPE\CSA\openjre\lib\security\cacerts

Linux: /usr/local/hpe/csa/openjre/lib/security/cacerts.

- If you are using an Oracle JRE, the truststore may be found in the following location:

JAVA_HOME/lib/security/cacerts

For example:

Windows: C:\Program Files\Java\jre7\lib\security\cacerts

Linux: /usr/local/bin/jre1.7.0_71/lib/security/cacerts

Request Software Licenses

CSA version 4.80 requires a software license. CSA licensing is based on the number of operating system instances (OSIs) being used in current, active subscriptions.

After the initial installation of CSA version 4.80, when you log in to the Cloud Service Management Console, a temporary 90-day trial license is activated. Once the trial license expires, you are limited to 25 OSIs. If you created more than 25 OSIs during the trial period, you cannot create any additional OSIs. You can add more licenses at any time to increase your OSI capacity.

The following topics are covered in this section:

- Request a software license
- Request a software license for a clustered environment
- Request a software license for a system with an updated IP address

For information about how to view, add, or delete a license, see the *Cloud Service Management Console Help*.

Request a Software License

If you received an Electronic Delivery Receipt, use the link to the licensing portal located in the receipt and follow the online instructions to request a software license. Otherwise, to access the licensing portal, go to <http://www.hpe.com/software/licensing>, enter your Entitlement Order Number, and follow the online instructions to request a software license.

See the [Software License Activation Quick Start Guide](#) for more information about requesting a software license.

IP Address Limitations

When you request a software license, you must supply the IP address (IPv4 or Ipv6) of the system on which CSA is installed.

Do NOT use the following IP addresses when requesting a software license:

- Loopback address - 127.0.0.1 (IPv4) or ::1 (IPv6)

Request a Software License for a Clustered Environment

If you are configuring CSA in a clustered environment, use the IP address of the load balancer - either the `APACHE_IP_ADDR` or the `LOAD_BALANCER_IP_ADDR` (see the examples given in the *Cloud Service Automation Cluster Configuration Guide Using a Load Balancer*. The license should be installed on only one node in the clustered environment.

Request a Software License for a System with an Updated IP Address

If you change the IP address of the system on which CSA is running, you must request a new software license.

If you immediately add the new license without restarting CSA, the license will not be accepted. You must restart CSA before adding the new license. See ["Restart CSA" on page 170](#) for instructions. For more information about managing software licenses, see the *Cloud Service Management Console Help*.

Enable TLS on Your Web Browser

The Cloud Service Management Console is configured to require https (http over a secure connection) for client browsers. Specifically, the Cloud Service Management Console is configured to use the TLS protocol. You must enable TLSv 1.0 as the required minimum protocol for the browser, and, if applicable, disable the SSL protocols.

Enable your Web browser to use the TLS protocol.

Chrome, Windows

1. Exit or kill all Chrome sessions.
2. If you added a shortcut to launch Chrome from the Taskbar, remove it: right-click the shortcut on the Taskbar and select **Unpin this program from taskbar**.
3. For every shortcut you use to launch Chrome, do the following:
 - a. Right-click on the shortcut and select **Properties**.
 - b. Select the **Shortcut** tab.
 - c. At the end of the Target field, enter the following after the last quotation mark (and include a space after the last quotation mark but before the following content):
--ssl-version-min=tls1
 - d. Click **OK**.
 - e. If asked for administrator privileges, click **Continue**.
4. If you deleted the shortcut from the Taskbar, right-click on any updated shortcut and select **Pin to Taskbar**.
5. If Chrome is your default browser, edit the registry:
 - a. Click on the **Start** icon, enter **regedit** in the Search programs and files box, and press **Enter**.
 - b. From the Registry Editor, select **HKEY_CLASSES_ROOT > http > shell > open > command**.
 - c. Double-click (**Default**).
 - d. Adding the following at the end of the Value data field (and include a space before the following content):
--ssl-version-min=tls1
 - e. Click **OK**.
 - f. Close the Registry Editor dialog.

Caution: Depending on how you launch Chrome, your browser session still may allow SSLv3 connections.

Chrome, Ubuntu

1. Exit or kill all Chrome sessions.
2. Edit the `/usr/share/applications/google-chrome.desktop` file.
3. For every line that starts with `Exec`, add the following argument:

`--ssl-version-min=tls1`

4. Save and exit the file.

Chrome, Red Hat Enterprise Linux

1. Exit or kill all Chrome sessions.
2. When invoking the browser from the command line, add the following argument:

`--ssl-version-min=tls1`

Microsoft Internet Explorer

1. Open the **Tools** menu (click on the tools icon or type Alt - x) and select **Internet options**.
2. Select the **Advanced** tab.
3. Scroll down to the bottom of the **Settings** section.
4. If TLS is not enabled, select the checkboxes next to **Use TLS 1.0**, **Use TLS 1.1**, and **Use TLS 1.2**.
5. Disable SSL 2.0 and SSL 3.0, if enabled (recommended). Unselect the checkbox next to **Use SSL 2.0** and/or **Use SSL 3.0**.
6. Click **OK**.

Firefox


1. Launch the Firefox browser.
2. In the Location Bar (address bar), enter **about:config** and press **Enter**.
3. In the Search box, enter **security.tls** and press **Enter**.
4. Double-click **security.tls.version.min**.
5. Set the value to **1** and click **OK**.

Configure the Provider Organization

1. Launch the Cloud Service Management Console by typing the following URL in a supported web browser: `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which the Cloud Service Management Console resides.

Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`

2. Log in to the Cloud Service Management Console as a CSA Administrator (see the *Cloud Service Automation Concepts Guide* and *Cloud Service Management Console Help* for more information about the CSA Administrator role).
3. Click the **Organizations** menu.

In the left-navigation frame, the provider organization icon () appears to the right of the

provider organization that is automatically set up (CSA-Provider). You may modify the provider organization, as needed. However, you cannot delete it. There can be only one provider organization.

4. In the left-navigation frame, select the provider organization.
5. Configure the provider organization by selecting and entering information into each section of the organization's navigation frame (General Information, LDAP, Access Control, Email Notifications, and Catalogs). See the *Cloud Service Management Console Help* for more information about the fields in each section (available in a printable PDF format). This document is available on the HPE Software Support Web site at: <https://softwaresupport.hpe.com>. (This site requires a Passport ID). Select **Dashboards > Manuals**.

Add a Software License

CSA version 4.80 requires a software license. CSA licensing is based on the number of operating system instances (OSIs) being used in current, active subscriptions.

After the initial installation of CSA version 4.80, when you log in to the Cloud Service Management Console, a temporary 90-day trial license is activated. Once the trial license expires, you are limited to 25 OSIs. If you created more than 25 OSIs during the trial period, you cannot create any additional OSIs. You can add more licenses at any time to increase your OSI capacity.

Before you can add a software license, you must request a license using the licensing portal. See ["Request Software Licenses" on page 18](#) for more information.

To add a software license, log in to the Cloud Service Management Console as the CSA Administrator. From the **Options** menu, select **Licensing**. For more detailed information about adding a license, see the *Cloud Service Management Console Help*.

For information about how to view or delete a license, see the *Cloud Service Management Console Help*.

Configure a Proxy for Resource Providers Outside the Internal Network

If you are using a network proxy server to communicate with a resource provider outside of the internal network (the resource provider's service access point is located outside of the internal network), configure CSA and Operations Orchestration to use this proxy server.

If you are using a network proxy server to communicate with a resource provider outside of the internal network, proxy configuration is required in the following situations:

- CSA - Validating the accessibility of a resource provider's URL. When a resource provider is created or modified, accessibility of the provider URL is validated with an HTTP or HTTPS GET call.
- Operations Orchestration - Contacting a resource provider. When an Operations Orchestration workflow provisioning step is executed, Operations Orchestration attempts to contact the resource provider.

If you do not configure the proxy server, you may see a Provider Validation Failed message when creating or updating a resource provider whose service access point is located outside of the internal network. Or, provisioning of a design fails when Operations Orchestration is unable to communicate with a resource provider that is located outside of the internal network.

To configure the proxy server for CSA and Operations Orchestration, do the following:

1. On the system running CSA, open the following file in a text editor:

Windows: CSA_HOME\jboss-as\bin\standalone.conf.bat

Linux: CSA_HOME/jboss-as/bin/standalone.conf

2. After the last uncommented line that sets the JAVA_OPTS property, add the following lines:

- o **Windows:**

```
rem # HTTP Proxy Settings
set "JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=<proxy.company.com>
-Dhttp.proxyPort=<proxy_port>"
```

```
rem # HTTPS Proxy Settings
set "JAVA_OPTS=%JAVA_OPTS% -Dhttps.proxyHost=<proxy.company.com>
-Dhttps.proxyPort=<proxy_port>"
```

```
rem # HTTP/HTTPS hosts not handled by the proxy
set "JAVA_OPTS=%JAVA_OPTS% -
Dhttp.nonProxyHosts=mycsaserver^^^|localhost^^^|127.*^^^|10.* "
```

where <proxy.company.com> is the fully-qualified domain name of the proxy server, <proxy_port> is the port used to communicate with the proxy server, and ^^| is the separator used when defining more than one non-proxy host.

- o **Linux:**

```
# HTTP Proxy Settings
JAVA_OPTS= "$JAVA_OPTS -Dhttp.proxyHost=<proxy.company.com>
-Dhttp.proxyPort=<proxy_port>"
```

```
# HTTPS Proxy Settings
JAVA_OPTS= "$JAVA_OPTS -Dhttps.proxyHost=<proxy.company.com>
-Dhttps.proxyPort=<proxy_port>"
```

```
# HTTP/HTTPS hosts not handled by the proxy
```

```
JAVA_OPTS= "$JAVA_OPTS -  
Dhttp.nonProxyHosts=mycsaserver\|localhost\|127.*\|10.* "
```

where *<proxy.company.com>* is the fully-qualified domain name of the proxy server,
<proxy_port> is the port used to communicate with the proxy server, and \| is the separator
used when defining more than one non-proxy host.

3. Save and exit the file.
4. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

5. If you have integrated with Operations Orchestration version 9.07, do the following:
 - a. Log in to Operations Orchestration Studio.
 - b. Open the **Configuration** folder.
 - c. Right-click the **System Properties** folder and select **New**.
 - d. In the dialog, enter **CSA_Proxy_Host** and click **OK**.
 - e. Set the **Property Value** to the fully-qualified domain name of the proxy server and click **OK**.
 - f. Right-click the **System Properties** folder and select **New**.
 - g. In the dialog, enter **CSA_Proxy_Port** and click **OK**.
 - h. Set the **Property Value** to the port used to communicate with the proxy server and click **OK**.
6. If you have integrated with Operations Orchestration version 10.50, do the following:
 - a. Log in to Operations Orchestration Central.
 - b. Click the **Content Management** button.
 - c. Select **Configuration Items > System Properties**.
 - d. Click the **Add** icon.
 - e. Enter the following information if it is not already configured:

Field	Description
Name	CSA_Proxy_Host
Override Value	The fully-qualified domain name of the proxy server.
Name	CSA_Proxy_Port
Override Value	The port used to communicate with the proxy server.

- f. Click **Save**.

Update the CSA Service Startup Type on Windows

If you have services or applications installed on the same system as CSA on Windows that CSA requires to be available when CSA is started (such as the database), update the CSA service startup to be delayed. This allows those services time to start before CSA starts if the system is rebooted.

To delay the start of the CSA on system reboot, do the following:

1. On the server that hosts Cloud Service Automation, navigate to **Start > Administrative Tools > Services**.
2. In the Service dialog, right-click on the CSA service and select **Properties**.
3. In the Properties dialog, locate the **Startup type** field and change the value to **Automatic (Delayed Start)**.
4. Click **OK**.

Location of the JRE Installed with CSA on Windows

Note: The IA-openjre directory is only installed on Windows.

The JRE installed with CSA (OpenJDK JRE) is located in the following location:

CSA_HOME\openjre

For example: C:\Program Files\HPE\CSA\openjre

An additional JRE is installed with CSA in the CSA_HOME\IA-openjre directory. This JRE is used exclusively by the CSA installer. This JRE should NOT be used for any other purposes.

Secure Connections

This chapter provides general information about configuring secure connections between CSA and some commonly used components of CSA and securing internal communication. You should consult your security expert for more detailed information about configuring secure connections in your environment.

Note: By default, CSA supports secure connections using TLSv1.2 and to enable support for TLSv1.2, you must configure the load balancer. CSA configuration can be manually changed to support TLSv1.1 or TLSv1.0, to work with older load balancers or other HTTPS clients that do not support TLSv1.2. However it is not recommended to enable TLSv1.1 or TLSv1.0 for security reasons.

Information includes:

- ["Configure Secure Connections for Client Browsers" on the next page](#) (required when the CSA self-signed certificate expires)
- ["Configure Secure Connections for LDAP" on page 65](#) (required if the LDAP server requires a secure connection)
- ["Configure Secure Connections for SMTP" on page 66](#) (required if the SMTP server requires a secure connection)
- ["Configure Secure Connections for an Oracle Database" on page 67](#) (required if the Oracle database requires a secure connection)
- ["Configure Secure Connections for Microsoft SQL Server" on page 70](#) (required if Microsoft SQL Server requires a secure connection)
- ["Configure Secure Connections for Operations Orchestration Load Balancer" on page 71](#) (required if you are running the OO LB server and it requires a secure connection)
- ["Configure Secure Internal Communication" on page 74](#) (recommended)

The function of http over a secure connection is configured by the `com.hp.csa.service.ssl.certificate.validation` property in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file and the `strictSSL` attribute in the `CSA_HOME/portal/conf/mpp.json` file. That is, http over a secure connection can be configured to encrypt the connection only or http over a secure connection can be configured to encrypt the connection, validate the certificate's expiration date, verify the certificate's hostname, and authenticate the certificate. See the *Secure Connections* section in ["Appendix: Cloud Service](#)

[Management Console Properties](#) on page 300 for more information about the `com.hp.csa.service.ssl.certificate.validation` property and the *Provider Attributes* and *Identity Management component Attributes* sections in ["Appendix: Marketplace Portal Attributes"](#) on page 349 for more information about the `strictSSL` attribute.

Configure Secure Connections for Client Browsers

The Cloud Service Management Console is configured to require https (http over a secure connection) for client browsers. For a secure connection to be established, a certificate must first be installed on the CSA (CSA) server.

A self-signed certificate is created and configured when CSA is installed and is configured with the fully-qualified domain name that was entered during the installation. This self-signed certificate is used when an https browser requests are issued for the Cloud Service Management Console and expires 120 days after CSA is installed.

When client browsers connect to the Cloud Service Management Console in this default configuration, the client browser will usually issue warnings that the certificate was not issued by a trusted authority. The end user can choose to continue to the web site or close the browser.

Although the self-signed certificate can be used in production, it is recommended that you replace this certificate. You can configure a trusted third-party Certificate Authority-signed or subordinate Certificate Authority-signed certificate (see ["Configure CSA to Use a Trusted Certificate Authority-Signed or Subordinate Certificate Authority-Signed Certificate"](#) on page 32) or configure an internal Certificate Authority-signed certificate (see ["Configure CSA to Use an Internal Certificate Authority-Signed Certificate"](#) on page 45). If the self-signed certificate expires before you are ready to move to production, you can replace the expired self-signed certificate by configuring a new self-signed certificate (see ["Configure CSA to Use a Self-Signed Certificate"](#) on page 50).

The following sections describe some common scenarios of configuring secure connections for CSA and the Marketplace Portal:

- ["Configure CSA to Use a Trusted Certificate Authority-Signed or Subordinate Certificate Authority-Signed Certificate"](#) on page 32
- ["Configure CSA to Use a Certificate Authority-Signed Certificate and a Certificate Authority-Provided Keystore"](#) on page 39

- ["Configure CSA to Use an Internal Certificate Authority-Signed Certificate" on page 45](#)
- ["Configure CSA to Use a Self-Signed Certificate" on page 50](#)

Note: Certificate chains require additional configuration and general information about importing a chain of certificates is provided in this section. However, you should consult your security expert for more detailed information when using certificate chains in your environment.

Wildcard certificates do not require special configuration.

If one of these scenarios does not match your situation, follow these general guidelines:

1. Obtain a root certificate and signed certificate and/or keystore. The root certificate is used to authenticate the signed certificate. The keystore stores the signed certificate. If you are generating a self-signed certificate, the self-signed certificate is used as the root certificate. If you need to create a certificate signing request to obtain this information, look for the steps to "Create a Keystore and Self-Signed Certificate" and "Create a Certificate Signing Request" for more detailed information.
2. Import the root certificate into the JRE's truststore. Look for the step to "Import the Certificate Authority's Root Certificate" for detailed instructions about how to import the root certificate into the JRE's keystore.
3. Complete one of the following steps, based on if you have a signed certificate only, a keystore only, or if you have both a signed certificate and keystore.
 - If you have a signed certificate only, do the following:
 - i. Create and import the certificate into a JKS keystore. Look for the step to "Import the Internal Certificate Authority-Signed Certificate" for more detailed information about how to create and import the certificate into a JKS keystore.

If the signed certificate contains a chain of certificates, you must copy the root certificate and each intermediate certificate in the chain to a separate certificate file and import each certificate file into the keystore in the following order (each certificate must have a unique alias):

- root certificate
- intermediate or subordinate certificate(s) in hierarchical order
- primary or end-user certificate

Use the signed certificate as the primary certificate. You will use the alias of the primary certificate when you configure the Web server. Work with your security expert to determine if the signed certificate contains a chain of certificates and to copy each certificate to a separate file.

- ii. Configure the Marketplace Portal. This step includes converting the JKS keystore into a PKCS#12 keystore used by the Marketplace Portal. Look for the step to "Configure the Marketplace Portal" for more detailed information.
 - o If you have a keystore only, do the following:
 - i. Determine the type of keystore you have. You must have two keystore types: JKS and PKCS#12 (CSA and the Marketplace Portal use two different types of keystores). Convert the existing keystore into the type that you need. Look for the step to "Convert the Certificate Authority-Provided Keystore" for more detailed information about how to generate both of the required keystores.
 - ii. Export the certificate from the keystore. You will need to provide the name and location of the certificate file when configuring the Marketplace Portal. Look for the step to "Export the Self-Signed Certificate" for more detailed information about how to export a certificate from a keystore.
 - iii. Configure the Marketplace Portal. You can skip the steps to convert the keystore to PKCS#12 format as you have already completed these steps. Look for the step to "Configure the Marketplace Portal" for more detailed information.
 - o If you have both the signed certificate and keystore, do the following:
 - i. Determine the type of keystore you have. You must have two keystore types: JKS and PKCS#12 (CSA and the Marketplace Portal use two different types of keystores). Convert the existing keystore into the type that you need. Look for the step to "Convert the Certificate Authority-Provided Keystore" for more detailed information about how to generate both of the required keystores.
 - ii. Configure the Marketplace Portal. You can skip the steps to convert the keystore to PKCS#12 format as you have already completed these steps. Look for the step to "Configure the Marketplace Portal" for more detailed information.
 4. Configure the Web server. This step configures CSA to use the JKS keystore. Look for the step to "Configure the Web Server" for more detailed information.
 5. Configure client browsers. This step is optional and tests whether or not the browser on a client system is configured to trust certificates signed by your Certificate Authority. Look for the step to "Configure Client Browsers" for more detailed information.
 6. Test secure connections to the Cloud Service Management Console. Test the connection to the Cloud Service Management Console. Look for the step to "Test Secure Connections" for more detailed information.

Note: If you have configured CSA to be compliant with FIPS 140-2, you must substitute the CSA

server truststore (for example, `csa_server_truststore.p12`) for the Java truststore (`cacerts`) and substitute the CSA server truststore password for the Java truststore password (`changeit`) in the examples. See the *Cloud Service Automation FIPS 140-2 Compliance Configuration Guide* for more information about the CSA server truststore and password.

Configure CSA to Use a Trusted Certificate Authority-Signed or Subordinate Certificate Authority-Signed Certificate

This section describes the process you should follow to obtain, install, and configure a trusted third-party Certificate Authority-signed or subordinate Certificate Authority-signed certificate for use by CSA. The process by which you acquire a certificate depends on your organization. If you are obtaining a certificate from a trusted third-party Certificate Authority, such as Verisign, perform the following general steps, which are described in detail below. If you are generating and/or obtaining a certificate from an internal Certificate Authority, such as a corporate Certificate Authority, you should perform the general steps in "[Configure CSA to Use an Internal Certificate Authority-Signed Certificate](#)" on page 45.

1. Create a keystore and a self-signed certificate
2. Create a certificate signing request
3. Submit the certificate signing request to a Certificate Authority
4. Import the Certificate Authority's root certificate
5. Import the Certificate Authority-signed certificate
6. Configure the Marketplace Portal
7. Configure the Web server
8. Configure client browsers
9. Test the secure connection

Note: In the following instructions, `CSA_HOME` is the directory in which CSA is installed (for example, on Windows the directory is `C:\Program Files\HPE\CSA` and on Linux the directory is `/usr/local/hpe/csa`) and the `keytool` utility is included with the JRE.

Also, the following instructions are applicable for subordinate Certificate Authorities. Wherever the

Certificate Authority is mentioned, the subordinate Certificate Authority is implied. For example, if the content states to submit the certificate to a Certificate Authority, you may also submit the certificate to a subordinate Certificate Authority.

Step 1: Create a Keystore and Self-Signed Certificate

Create a self-signed certificate to send with your request to a Certificate Authority by doing the following:

1. Open a command prompt and change directories to `CSA_HOME`.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -genkeypair -alias csa_ca_signed  
-validity 365 -keyalg rsa -keysize 2048 -keystore  
.\jboss-as\standalone\configuration\.keystore_ca_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -genkeypair -alias csa_ca_signed  
-validity 365 -keyalg rsa -keysize 2048 -keystore  
./jboss-as/standalone/configuration/.keystore_ca_signed
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed

You can use different values for `-alias`, `-validity`, `-keysize` and `-keystore`. These instructions assume that you will use the `-alias` and `-keystore` values recommended here; you will have to adjust the commands accordingly if you use different values.

3. Enter a keystore password.

This password is used to control access to the keystore. This password must be the same as the password you enter for the key later in this procedure.

4. When you are prompted for your first and last name, enter the fully qualified domain name of the CSA server.
5. Follow the prompts to enter the remaining organization and location values.
6. Enter the keystore password you supplied earlier to use as the key password.

Although `keytool` allows you to enter different passwords for the keystore and the key, the two passwords must be the same to work with CSA.

Step 2: Create a Certificate Signing Request

To enable a Certificate Authority to sign the self-signed certificate, you will need to create a Certificate Signing Request using the following procedure:

1. Open a command prompt and change directories to `CSA_HOME`.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -certreq -alias csa_ca_signed  
-file C:\csacsr.txt -keystore .\jboss-as\standalone\configuration\.keystore_ca_  
signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -certreq -alias csa_ca_signed  
-file /tmp/csacsr.txt -keystore ./jboss-as/standalone/configuration/.keystore_  
ca_signed
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed

3. When you are prompted for a password, enter the password you supplied for the keystore and key when you created the keystore and self-signed certificate in step 1.

Step 3: Submit the Certificate Signing Request to a Certificate Authority

Submit the Certificate Signing Request to the Certified Authority following the procedure used by your organization or the third-party provider. After the submission has been processed, you will receive a Certificate Authority-signed certificate and a root certificate for the Certificate Authority.

In our example, we will assume the Certificate Authority's root certificate is named `csaca.cer`, the Certificate Authority-signed certificate is named `csa_ca_signed.cer`, and that both are located in `C:\` on Windows or in `/tmp` on Linux. .

Step 4: Import the Certificate Authority's Root Certificate

This step configures the JRE so it trusts the Certificate Authority that has signed your certificate. The JRE ships with a list of common, trusted Certificate Authority certificates that are stored in a keystore

named `cacerts`. If the Certificate Authority used to sign your certificate is well known, it is likely that this root certificate is already present in the `cacerts` keystore. It is recommended that you perform the following steps even if you suspect that the certificate is already installed. The `keytool` command will detect if the certificate is already present, and you can exit the import process if the certificate exists.

1. Open a command prompt.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias csaca -file C:\csaca.cer -  
trustcacerts -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias csaca -file /tmp/csaca.cer -  
trustcacerts -keystore CSA_JRE_HOME/lib/security/cacerts
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed

3. When prompted for the keystore password, enter `changeit`.
4. Enter `yes` when prompted to trust the certificate.

Step 5: Import the Certificate Authority-Signed Certificate

1. The Certificate Authority-signed certificate (`csa_ca_signed.cer`) contains a chain of certificates and you must copy the root and any intermediate certificates in the chain to separate files. Work with your security expert to copy each certificate to a separate file.
2. Open a command prompt and change directories to `CSA_HOME`.
3. Import the certificate file(s):

You must import each separate file in the following order (each certificate must have a unique alias):

- root certificate
- intermediate or subordinate certificate(s) in hierarchical order
- primary or end-user certificate

For example, if the Certificate Authority-signed certificate contains three certificates (root, intermediate, and primary) and you copied the root certificate to C:\root.cer on Windows or /tmp/root.cer on Linux, and the intermediate certificate to C:\intermediate.cer on Windows or /tmp/intermediate.cer on Linux, (you will use the Certificate Authority-signed certificate as the primary certificate), run the following commands in the following order to import each certificate:

Windows:

```
"CSA_JR_HOME\bin\keytool" -importcert -alias csa_ca_signed_root -file
C:\root.cer -trustcacerts -keystore
.\jboss-as\standalone\configuration\keystore_ca_signed

"CSA_JRE_HOME\bin\keytool" -importcert -alias csa_ca_signed_intermediate -file
C:\intermediate.cer -trustcacerts -keystore
.\jboss-as\standalone\configuration\keystore_ca_signed

"CSA_JRE_HOME\bin\keytool" -importcert -alias csa_ca_signed -file C:\csa_ca_
signed.cer -trustcacerts -keystore
.\jboss-as\standalone\configuration\keystore_ca_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias csa_ca_signed_root -file
/tmp/root.cer -trustcacerts -keystore
./jboss-as/standalone/configuration/keystore_ca_signed

CSA_JRE_HOME/bin/keytool -importcert -alias csa_ca_signed_intermediate -file
/tmp/intermediate.cer -trustcacerts -keystore
./jboss-as/standalone/configuration/keystore_ca_signed

CSA_JRE_HOME/bin/keytool -importcert -alias csa_ca_signed -file /tmp/csa_ca_
signed.cer -trustcacerts -keystore
./jboss-as/standalone/configuration/keystore_ca_signed
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed

Use the alias of the primary certificate (csa_ca_signed) and keystore name (CSA_HOME/jboss-as/standalone/configuration/keystore_ca_signed) when you configure the web server.

4. When prompted, enter the password for the key and keystore.

Use this password when you configure the web server.

Step 6: Configure the Marketplace Portal

This step converts the CSA keystore to a PKCS#12 archive and configures the Marketplace Portal to use the Certificate Authority-signed certificate.

1. Open a command prompt and navigate to `CSA_HOME`.
2. Convert the CSA keystore to a PKCS#12 archive. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importkeystore -srckeystore  
.\jboss-as\standalone\configuration\.keystore_ca_signed -deststoretype PKCS12 -  
destkeystore .\portal\conf\.mppkeystore_ca_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importkeystore -srckeystore  
./jboss-as/standalone/configuration/.keystore_ca_signed -deststoretype PKCS12 -  
destkeystore ./portal/conf/.mppkeystore_ca_signed
```

3. When prompted, enter the password for the PKCS#12 archive. You will need this password when you configure the passphrase attribute later in this section.
4. When prompted, enter the password for the CSA keystore (changeit).
5. Open the `CSA_HOME/portal/conf/mpp.json` file in a text editor.
6. Update the `ca` attribute value for the provider. Enter the path to the certificate file that you imported in ["Step 5: Import the Certificate Authority-Signed Certificate" on page 35](#). For example, `C:\csa_ca_signed.cer` on Windows or `/tmp/csa_ca_signed.cer` on Linux. If you imported a chain of certificates, use the certificate file of the primary certificate.
7. Update the `ca` attribute value for the `idmProvider`. Enter the path to the certificate file that you imported in ["Step 5: Import the Certificate Authority-Signed Certificate" on page 35](#). For example, `C:\csa_ca_signed.cer` on Windows or `/tmp/csa_ca_signed.cer` on Linux. If you imported a chain of certificates, use the certificate file of the primary certificate.
8. Update the `pxf` attribute value. Enter the name of the PKS#12 archive you created earlier. For example, `..\conf\.mppkeystore_ca_signed`.
9. Update the `passphrase` attribute value. Enter the encrypted password used to access the `.mppkeystore_ca_signed` archive (see ["Encrypt a Marketplace Portal Password" on page 180](#) for instructions). An encrypted password is preceded by `ENC` without any separating spaces and is

enclosed in parentheses.

10. Save and exit the file.

Step 7: Configure the Web Server

1. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.
2. Locate the following entry:

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore"  
keystore-password="changeit"/>
```

3. Set the path attribute to the keystore you used in ["Step 5: Import the Certificate Authority-Signed Certificate" on page 35](#), set the password attribute to the value that corresponds to the password you selected for the keystore, and add the `alias` attribute and set it to the alias you used in ["Step 5: Import the Certificate Authority-Signed Certificate" on page 35](#).

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore_ca_signed"  
keystore-password="keystorePassword" alias="csa_ca_signed"/>
```

Note: If you imported a chain of certificates, use the alias of the primary certificate.

Note: This example stores the password in clear text. If you want to use an encrypted password, see ["Masking Passwords in standalone.xml Using the JBoss vault Script" on page 59](#) for information about creating a password vault for JBoss.

4. Restart the CSA service.
See ["Restart CSA" on page 170](#) for instructions.
5. After the service has started, review the log files in the `CSA_HOME/jboss-as/standalone/log/` directory and verify that no TLS or keystore errors are present.

Step 8: Configure Client Browsers

The client browser must be configured to trust certificates that are signed by the Certificate Authority. In most situations, this step will already have occurred. Client browsers are likely to already trust well-known third-party Certificate Authorities, or will have previously accessed and trusted Web sites that use internal Certificate Authority root certificates.

To test whether or not the browser on a client system is configured to trust certificates signed by your Certificate Authority, open a supported Web browser and navigate to `https://<csahostname>:8444/csa`. If you do not see a certificate warning, then the browser is configured properly.

If client browsers need to be configured to trust certificates signed by your Certificate Authority, then you will need to make the root certificate available to clients so it can be installed in the browser. The process of installing the root certificate will vary based on the browser.

- **Microsoft Internet Explorer** and **Chrome**: From Windows Explorer, double-click on the `.cer` file to begin the import process. Install the certificate in the Trusted Root Certification Authorities store. For information about how to import the certificate, see the browser's online documentation.
- **Firefox**: To begin the import process, select **Tools > Options**, select **Advanced**, select the **Encryption** tab, and click **View Certificates**. Import the root certificate into the **Authorities** tab. For information about how to import the certificate, see the browser's online documentation.

Step 9: Test Secure Connections

To test the connection to the Cloud Service Management Console, on a client system, open a supported Web browser and navigate to `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system that was used when the certificate was created. If the client browser is configured to accept the Certificate Authority's root certificate and the Web application opens without a certificate warning, then you have successfully configured CSA to use a Certificate Authority-signed certificate. If a certificate warning is displayed, review steps 1-8 to be sure they were followed as documented.

Configure CSA to Use a Certificate Authority-Signed Certificate and a Certificate Authority-Provided Keystore

This section describes the process you should follow to install and configure a root certificate, Certificate Authority-signed certificate, and Certificate Authority-provided keystore for use by CSA. In this example, the Certificate Authority provides you with a root certificate, signed certificate, and a keystore containing the signed certificate. A Certificate Authority may provide you with a keystore if you are using a wildcard certificate.

Perform the following general steps, which are described in detail below:

1. Import the Certificate Authority's root certificate
2. Convert the Certificate Authority-provided keystore
3. Determine the alias for the certificate from the JKS keystore
4. Configure the Marketplace Portal
5. Configure the Web server
6. Configure client browsers
7. Test the secure connection

Note: In the following instructions,

CSA_HOME is the directory in which CSA is installed (for example, C:\Program Files\HPE\CSA on Windows or /usr/local/hpe/csa on Linux) and the keytool utility is included with the JRE.

In this example, it is assumed that you are given a Certificate Authority-signed certificate (referred to as `csa_ca_signed.cer`), a Certificate Authority's root certificate (referred to as `ca_root.cer`), and a keystore provided by the Certificate Authority that contains the Certificate Authority-signed certificate (referred to as `.keystore_caprovided`). All files are located in C:\ on Windows and /tmp on Linux.

Step 1: Import the Certificate Authority's Root Certificate

This step configures CSA's JRE so it trusts the Certificate Authority that has signed the certificate by importing the Certificate Authority's root certificate into a keystore named `cacerts` that is shipped with the JRE. The JRE ships with a list of common, trusted Certificate Authority certificates that are stored in this keystore. If the Certificate Authority used to sign the certificate is well known, it is likely that this root certificate is already present in this keystore. It is recommended that you perform the following steps even if you suspect that the certificate is already installed. The `keytool` command will detect if the root certificate is already present, and you can exit the import process if the certificate exists.

1. Open a command prompt.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias csaca -file C:\ca_root.cer -trustcacerts -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

Linux:


```
CSA_JRE_HOME/bin/keytool -importcert -alias csaca -file /tmp/ca_root.cer -  
trustcacerts -keystore CSA_JRE_HOME/lib/security/cacerts
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed

3. When prompted for the keystore password, enter changeit.
4. Enter yes when prompted to trust the certificate.

Step 2: Convert the Certificate Authority-Provided Keystore

The keystore used by CSA must be in JKS format. The keystore used by the Marketplace Portal must be in PKCS#12 format. You will need to provide both types of keystores. This section provides the tasks to convert a JKS keystore to a PKCS#12 keystore and a PKCS#12 keystore to a JKS keystore. If your Certificate Authority provided you a keystore in another format, ask your Certificate Authority how to convert it to either the JKS or PKCS#12 format. Then, complete the tasks in this step to create both required keystore formats.

1. Determine the format of the Certificate Authority-provided keystore. If you do not know the format, ask the Certificate Authority for this information. If your Certificate Authority provided you a keystore in a format other than JKS or PKCS#12, ask your Certificate Authority how to convert it to either the JKS or PKCS#12 format.
2. Open a command prompt and change directories to CSA_HOME.
3. To convert a JKS keystore to a PKCS#12 keystore, run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importkeystore -srckeystore C:\.keystore_caprovided  
-deststoretype PKCS12 -destkeystore C:\.keystore_mpp
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importkeystore -srckeystore /tmp/.keystore_caprovided  
-deststoretype PKCS12 -destkeystore /tmp/.keystore_mpp
```

To convert a PKCS#12 keystore to a JKS keystore, run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importkeystore -srckeystore C:\.keystore_caprovided  
-deststoretype JKS -destkeystore C:\.keystore_csa
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importkeystore -srckeystore /tmp/.keystore_caprovided  
-deststoretype JKS -destkeystore /tmp/.keystore_csa
```

4. When prompted, enter the password for the destination and source keystores. For simplicity, use the same passwords.

Use this password when you configure the Marketplace Portal and the Web server.

Step 3: Determine the Alias for the Certificate from the JKS Keystore

Determine the alias for the certificate from the JKS keystore. You will need this alias when you configure the Web server.

If the Certificate Authority provided a JKS keystore, run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -list -keystore C:\.keystore_caprovided
```

Linux:

```
CSA_JRE_HOME/bin/keytool -list -keystore /tmp/.keystore_caprovided
```

If you converted the Certificate Authority-provided keystore to JKS, run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -list -keystore C:\.keystore_csa
```

Linux:

```
CSA_JRE_HOME/bin/keytool -list -keystore /tmp/.keystore_csa
```

If there is more than one entry displayed, contact the Certificate Authority and ask which alias to use for the certificate. If a certificate chain is being used, typically you would use the alias of the primary certificate.

Step 4: Configure the Marketplace Portal

This step configures the Marketplace Portal to use the root certificate and the PKCS#12 keystore.

1. Open the `CSA_HOME/portal/conf/mpp.json` file in a text editor.
2. Update the `ca` attribute value for the provider. Enter the path to the root certificate file. For example, `C:\ca_root.cer` on Windows or `/tmp/ca_root.cer` on Linux.
3. Update the `ca` attribute value for the `idmProvider`. Enter the path to the root certificate file. For example, `C:\ca_root.cer` on Windows or `/tmp/ca_root.cer` on Linux.
4. Update the `pkcs12` attribute value. Enter the name of the PKCS#12 keystore you created earlier. For example, if the Certificate Authority provided a PKCS#12 keystore, `C:\.keystore_caprovided` on Windows or `/.keystore_caprovided` on Linux. If you converted the Certificate Authority-provided keystore to PKCS#12, `C:\.keystore_mpp` on Windows or `/.keystore_mpp` on Linux.
5. Update the `password` attribute value. Enter the encrypted password used to access the PKCS#12 keystore (see ["Encrypt a Marketplace Portal Password" on page 180](#) for instructions). An encrypted password is preceded by `ENC` without any separating spaces and is enclosed in parentheses. This is the password from step 2 (Convert the Certificate Authority-Provided Keystore).
6. Save and exit the file.

Step 5: Configure the Web Server

1. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.
2. Locate the following entry:

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore"
keystore-password="changeit"/>
```
3. Set the `path` attribute to the JKS keystore, set the `keystore-password` to the value that corresponds to the password you selected for the JKS keystore, and add the `alias` and set it to the alias you determined in step 3 (Determine the Alias for the Certificate from the JKS Keystore).

For example, if the Certificate Authority provided a JKS keystore, update the entry to:

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore_
caprovided" keystore-password="keystorePassword" alias="<alias_from_step3>"/>
```

For example, if you converted the Certificate Authority-provided keystore to JKS, update the entry to:

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore_csa"
keystore-password="keystorePassword" alias="<alias_from_step3>"/>
```

Note: This example stores the password in clear text. If you want to use an encrypted password, see ["Masking Passwords in standalone.xml Using the JBoss vault Script"](#) on page 59 for information about creating a password vault for JBoss.

4. Restart the CSA service.

See ["Restart CSA"](#) on page 170 for instructions.

5. After the service has started, review the log files in the `CSA_HOME/jboss-as/standalone/log/` directory and verify that no TLS or keystore errors are present.

Step 6: Configure Client Browsers

The client browser must be configured to trust certificates that are signed by the Certificate Authority. In most situations, this step will already have occurred. Client browsers are likely to already trust well-known third-party Certificate Authorities, or will have previously accessed and trusted Web sites that use internal Certificate Authority root certificates.

To test whether or not the browser on a client system is configured to trust certificates signed by your Certificate Authority, open a supported Web browser and navigate to `https://<csahostname>:8444/csa`. If you do not see a certificate warning, then the browser is configured properly.

If client browsers need to be configured to trust certificates signed by your Certificate Authority, then you will need to make the root certificate available to clients so it can be installed in the browser. The process of installing the root certificate will vary based on the browser.

- **Microsoft Internet Explorer and Chrome:** From Windows Explorer, double-click on the `.cer` file to begin the import process. Install the certificate in the Trusted Root Certification Authorities store. For information about how to import the certificate, refer to the browser's online documentation.
- **Firefox:** To begin the import process, select **Tools > Options**, select **Advanced**, select the **Encryption** tab, and click **View Certificates**. Import the root certificate into the Authorities tab. For information about how to import the certificate, refer to the browser's online documentation.

Step 7: Test Secure Connections

To test the connection to the Cloud Service Management Console, on a client system, open a supported Web browser and navigate to `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system that was used when the certificate was created. If the client browser is configured to accept the Certificate Authority's root certificate and the Web application

opens without a certificate warning, then you have successfully configured CSA to use a Certificate Authority-signed certificate. If a certificate warning is displayed, review steps 1-6 to be sure they were followed as documented.

Configure CSA to Use an Internal Certificate Authority-Signed Certificate

This section describes the process you should follow to install and configure an internal root and internal Certificate Authority-signed certificate for use by CSA. An internal certificate is one that is generated by an internal Certificate Authority, such as a corporate or government Certificate Authority. For an internal Certificate Authority, you do not have to generate a self-signed certificate nor create a certificate signing request. The internal Certificate Authority should provide you with a root certificate and signed certificate.

Perform the following general steps, which are described in detail below:

1. Import the internal Certificate Authority's root certificate
2. Import the internal Certificate Authority-signed certificate
3. Configure the Marketplace Portal
4. Configure the Web server
5. Configure client browsers
6. Test the secure connection

Note: In the following instructions, `CSA_HOME` is the directory in which CSA is installed (for example, `C:\Program Files\HPE\CSA on Windows` or `/usr/local/hpe/csa` on Linux) and the `keytool` utility is included with the JRE.

In this example, we will assume you are given an internal Certificate Authority-signed certificate (referred to as `csa_internalca_signed.cer`), an internal Certificate Authority's root certificate (referred to as `csainternalca.cer`), and both certificates are located in `C:\` on Windows or `/tmp` on Linux.

Step 1: Import the Certificate Authority's Root Certificate

This step configures the JRE so it trusts the internal Certificate Authority that has signed your certificate by importing the internal Certificate Authority into a keystore named `cacerts` that is shipped

with the JRE.

1. Open a command prompt.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias csainternalca -file  
C:\csainternalca.cer -trustcacerts -keystore "CSA_JRE_  
HOME\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias csainternalca -file  
/tmp/csainternalca.cer -trustcacerts -keystore CSA_JRE_  
HOME/lib/security/cacerts
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed.

3. When prompted for the keystore password, enter changeit.
4. Enter yes when prompted to trust the certificate.

Step 2: Import the Internal Certificate Authority-Signed Certificate

1. The internal Certificate Authority-signed certificate (`csa_internalca_signed.cer`) contains a chain of certificates and you must copy the root and any intermediate certificates in the chain to separate files. Work with your security expert to copy each certificate to a separate file.
2. Open a command prompt and change directories to CSA_HOME.
3. Import the certificate file(s):

You must import each separate file in the following order (each certificate must have a unique alias):

- root certificate
- intermediate or subordinate certificate(s) in hierarchical order
- primary or end-user certificate

For example, if the internal Certificate Authority-signed certificate contains three certificates (root, intermediate, and primary) and you copied the root certificate to `C:\root.cer` on Windows or

/tmp/root.cer and the intermediate certificate to C:\intermediate.cer on Windows or /tmp/intermediate.cer on Linux (you will use the internal Certificate Authority-signed certificate file as the primary certificate), run the following commands in the following order to import each certificate:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias csa_internalca_signed_root -file
C:\root.cer -trustcacerts -keystore
.\jboss-as\standalone\configuration\keystore_internalca_signed

"CSA_JRE_HOME\bin\keytool" -importcert -alias csa_internalca_signed_
intermediate -file C:\intermediate.cer -trustcacerts -keystore
.\jboss-as\standalone\configuration\keystore_internalca_signed

"CSA_JRE_HOME\bin\keytool" -importcert -alias csa_internalca_signed -file
C:\csa_internalca_signed.cer -trustcacerts -keystore
.\jboss-as\standalone\configuration\keystore_internalca_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias csa_internalca_signed_root -file
/tmp/root.cer -trustcacerts -keystore
./jboss-as/standalone/configuration/.keystore_internalca_signed

CSA_JRE_HOME/bin/keytool -importcert -alias csa_internalca_signed_intermediate
-file /tmp/intermediate.cer -trustcacerts -keystore
./jboss-as/standalone/configuration/.keystore_internalca_signed

CSA_JRE_HOME/bin/keytool -importcert -alias csa_internalca_signed -file
/tmp/csa_internalca_signed.cer -trustcacerts -keystore
./jboss-as/standalone/configuration/.keystore_internalca_signed
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed.

Use the alias of the primary certificate (csa_internalca_signed) and keystore name (CSA_HOME/jboss-as/standalone/configuration/.keystore_internalca_signed) when you configure the Web server.

4. When prompted, enter the password for the key and keystore.

Use this password when you configure the Web server.

Step 3: Configure the Marketplace Portal

This step converts the CSA keystore to a PKCS#12 archive and configures the Marketplace Portal to use the internal Certificate Authority root certificate.

1. Open a command prompt and navigate to `CSA_HOME`.
2. Convert the CSA keystore to a PKCS#12 archive. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importkeystore -srckeystore  
.\jboss-as\standalone\configuration\keystore_internalca_signed -deststoretype  
PKCS12 -destkeystore .\portal\conf\.mppkeystore_internalca_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importkeystore -srckeystore  
./jboss-as/standalone/configuration/keystore_internalca_signed -deststoretype  
PKCS12 -destkeystore ./portal/conf/.mppkeystore_internalca_signed
```

3. When prompted, enter the password for the PKCS#12 archive. You will need this password when you configure the passphrase attribute later in this section.
4. When prompted, enter the password for the CSA keystore (changeit).
5. Open the `CSA_HOME/portal/conf/mpp.json` file in a text editor.
6. Update the `ca` attribute value for the provider. Enter the path to the certificate file that you imported in step 2. For example, `C:\csa_internalca_signed.cer` on Windows or `/tmp/csa_internalca_signed.cer` on Linux. If you imported a chain of certificates, use the certificate file of the primary certificate.
7. Update the `ca` attribute value for the `idmProvider`. Enter the path to the certificate file that you imported in step 2. For example, `C:\csa_internalca_signed.cer` on Windows or `/tmp/csa_internalca_signed.cer` on Linux. If you imported a chain of certificates, use the certificate file of the primary certificate.
8. Update the `pxf` attribute value. Enter the name of the PKS#12 archive you created earlier. For example, `..\conf\.mppkeystore_internalca_signed`.
9. Update the `passphrase` attribute value. Enter the encrypted password used to access the `.mppkeystore_internalca_signed` archive (see ["Encrypt a Marketplace Portal Password" on page 180](#) for instructions). An encrypted password is preceded by `ENC` without any separating

spaces and is enclosed in parentheses.

10. Save and exit the file.

Step 4: Configure the Web Server

1. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.
2. Locate the following entry:

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore"  
keystore-password="changeit"/>
```

3. Set the `path` attribute to the keystore you used in step 2, set the `keystore-password` attribute to the value that corresponds to the password you selected for the keystore, and add the `alias` attribute and set it to the alias you used in step 2.

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore_  
internalca_signed" keystore-password="keystorePassword" alias="csa_internalca_  
signed" />
```

Note: If you imported a chain of certificates, use the alias of the primary certificate.

Note: This example stores the password in clear text. If you want to use an encrypted password, see ["Masking Passwords in standalone.xml Using the JBoss vault Script" on page 59](#) for information about creating a password vault for JBoss.

4. Restart the CSA service. See ["Restart CSA" on page 170](#) for instructions.
5. After the service has started, review the log files in the `CSA_HOME/jboss-as/standalone/log/` directory and verify that no TLS or keystore errors are present.

Step 5: Configure Client Browsers

The client browser must be configured to trust certificates that are signed by the Certificate Authority. In most situations, this step will already have occurred. Client browsers are likely to already trust well-known third-party Certificate Authorities, or will have previously accessed and trusted Web sites that use internal Certificate Authority root certificates.

To test whether or not the browser on a client system is configured to trust certificates signed by your Certificate Authority, open a supported Web browser and navigate to

`https://<csahostname>:8444/csa`. If you do not see a certificate warning, then the browser is configured properly.

If client browsers need to be configured to trust certificates signed by your Certificate Authority, then you will need to make the root certificate available to clients so it can be installed in the browser. The process of installing the root certificate will vary based on the browser.

- **Microsoft Internet Explorer** and **Chrome**: From Windows Explorer, double-click on the `.cer` file to begin the import process. Install the certificate in the Trusted Root Certification Authorities store. For information about how to import the certificate, refer to the browser's online documentation.
- **Firefox**: To begin the import process, select **Tools > Options**, select **Advanced**, select the **Encryption** tab, and click **View Certificates**. Import the root certificate into the Authorities tab. For information about how to import the certificate, refer to the browser's online documentation.

Step 6: Test Secure Connections

To test the connection to the Cloud Service Management Console, on a client system, open a supported Web browser and navigate to `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system that was used when the certificate was created. If the client browser is configured to accept the Certificate Authority's root certificate and the Web application opens without a certificate warning, then you have successfully configured CSA to use a Certificate Authority-signed certificate. If a certificate warning is displayed, review steps 1-5 to be sure they were followed as documented.

Configure CSA to Use a Self-Signed Certificate

This section describes the process you should follow to obtain, install, and configure a self-signed certificate for use by CSA.

In general, it is recommended that you replace CSA's self-signed certificate with a Certificate Authority-signed certificate. However, you may consider replacing CSA's self-signed with a self-signed certificate you create in the following situations:

- CSA's self-signed certificate has expired and you do not want to configure a Certificate Authority-signed certificate at this time.
- The hostname that you entered when you installed CSA has changed (the hostname you entered during installation is used to configure CSA's self-signed certificate).

- You entered an IP address instead of the fully-qualified domain name when CSA was installed.
- Obtaining a Certificate Authority-signed certificate is not an option in your environment.

You should perform the following general steps:

1. Create a keystore and a self-signed certificate.
2. Export the self-signed certificate.
3. Import the self-signed certificate as a trusted certificate.
4. Configure the Marketplace Portal.
5. Configure the web server.
6. Configure client browsers (optional).
7. Test the secure connection.

Note: In the following instructions, CSA_HOME is the directory in which CSA is installed (for example, on Windows the directory is C:\Program Files\HPE\CSA, and on Linux the directory is /usr/local/hpe/csa). The keytool utility is included with the JRE.

Step 1: Create a Keystore and Self-Signed Certificate

To create a self-signed certificate, complete the following steps:

1. Open a command prompt and change directories to CSA_HOME.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -genkeypair -alias csa_self_signed  
-validity 365 -keyalg rsa -keysize 2048  
-keystore .\jboss-as\standalone\configuration\  
.keystore_self_signed [-ext san=ip:<ip_address>]
```

Linux:

```
CSA_JRE_HOME/bin/keytool -genkeypair -alias csa_self_signed  
-validity 365 -keyalg rsa -keysize 2048  
-keystore ./jboss-as/standalone/configuration/  
.keystore_self_signed [-ext san=ip:<ip_address>]
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed and `-ext san=ip:<ip_address>` is the option to specify the IP address of the system on which CSA is installed. This option is required if you specified an IP address instead of the fully-qualified domain name when you installed CSA. If you specified the fully-qualified domain name during installation, you may omit this option.

You can use different values for `-alias`, `-validity`, `-keysize` and `-keystore`. These instructions assume that you will use the `-alias` and `-keystore` values recommended here; you will have to adjust the commands accordingly if you use different values.

3. Enter a keystore password.

This password is used to control access to the keystore. This password must be the same as the password you enter for the key later in this procedure.

4. When you are prompted for your first and last name, enter the fully qualified domain name of the CSA server.
5. Follow the prompts to enter the remaining organization and location values.
6. Enter the keystore password you supplied earlier to use as the key password.

Although `keytool` allows you to enter different passwords for the keystore and the key, the two passwords must be the same to work with CSA.

Step 2: Export the Self-Signed Certificate

Export the self-signed certificate using the following procedure:

1. Open a command prompt and change directories to `CSA_HOME`.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -export -alias csa_self_signed  
-file C:\csa_self_signed.cer  
-keystore .\jboss-as\standalone\configuration\  
.keystore_self_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -export -alias csa_self_signed  
-file /tmp/csa_self_signed.cer
```

```
-keystore ./jboss-as/standalone/configuration/  
.keystore_self_signed
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

NOTE: Please do not anytime delete the default `jboss.cer` certificate from the folder `CSA_HOME\jboss-as\standalone\configuration\`

3. When you are prompted for a password, enter the keystore password used in step 1.

Step 3: Import the Self-Signed Certificate as a Trusted Certificate

This step configures the JRE so it trusts the self-signed certificate.

1. Open a command prompt.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias csa_self_signed  
-file C:\csa_self_signed.cer -trustcacerts  
-keystore "CSA_JRE_HOME\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias csa_self_signed  
-file /tmp/csa_self_signed.cer -trustcacerts  
-keystore CSA_JRE_HOME/lib/security/cacerts
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

3. When prompted for the keystore password, enter `changeit`.
4. Enter `yes` when prompted to trust the certificate.

Step 4: Configure the Marketplace Portal

This step converts the CSA keystore to a PKCS#12 archive and configures the Marketplace Portal to use the self-signed certificate.

1. Open a command prompt and navigate to CSA_HOME.
2. Convert the CSA keystore to a PKCS#12 archive. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importkeystore -srckeystore  
. \jboss-as\standalone\configuration\.keystore_self_signed -deststoretype PKCS12  
-destkeystore .\portal\conf\.mppkeystore_self_signed
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importkeystore -srckeystore  
./jboss-as/standalone/configuration/.keystore_self_signed -deststoretype PKCS12  
-destkeystore ./portal/conf/.mppkeystore_self_signed
```

3. When prompted, enter the password for the PKCS#12 archive. You will need this password when you configure the passphrase attribute later in this section.
4. When prompted, enter the password for the CSA keystore (changeit).
5. Open the CSA_HOME/portal/conf/mpp.json file in a text editor.
6. Update the ca attribute value for the provider. Enter the path to the certificate file that you imported in step 2. For example, C:/csa_self_signed.cer on Windows or /tmp/csa_self_signed.cer on Linux.
7. Update the ca attribute value for the idmProvider. Enter the path to the certificate file that you imported in step 2. For example, C:/csa_self_signed.cer on Windows or /tmp/csa_self_signed.cer on Linux.
8. Update the pfx attribute value. Enter the name of the PKS#12 archive you created earlier. For example, ../conf/.mppkeystore_self_signed.
9. Update the passphrase attribute value. Enter the encrypted password used to access the .mppkeystore_self_signed archive (see ["Encrypt a Marketplace Portal Password" on page 180](#) for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.
10. Save and exit the file.

Step 5: Configure the Web Server

1. Open `CSA_HOME\jboss-as\standalone\configuration\standalone.xml` in a text editor.
2. Locate the following entry:

```
<keystore keystore-password="changeit" path="CSA_
HOME\jboss-as\standalone\configuration\.keystore"/>
```

3. Set the `path` attribute to the keystore you used in step 2, set the `keystore-password` attribute to the value that corresponds to the password you selected for the keystore, and add the `key-alias` attribute and set it to the alias you used in step 2.

```
<keystore path="CSA_HOME\jboss-as\standalone\configuration\.keystore_self_
signed" keystore-password="keystorePassword" alias="csa_self_signed"/>
```

Note: This example stores the password in clear text. If you want to use an encrypted password, see ["Masking Passwords in standalone.xml Using the JBoss vault Script" on page 59](#) for information about creating a password vault for JBoss.

4. Restart the CSA service. See ["Restart CSA" on page 170](#) for instructions.
5. After the service has started, review the log files in the `CSA_HOME/jboss-as/standalone/log/` directory and verify that no TLS or keystore errors are present.

Step 6: Configure Client Browsers (Optional)

Because the self-signed certificate is not signed by a Certificate Authority, when accessing the Cloud Service Management Console, warning messages are displayed in the browser (these messages do not affect normal operations of CSA). To avoid these warning messages, import the `csa_self_signed.cer` file or add an exception.

- **Microsoft Internet Explorer** and **Chrome:** From Windows Explorer, double-click on the `csa_self_signed.cer` file to begin the import process. Install the certificate in the Trusted Root Certification Authorities store. For information about how to import the certificate, refer to the browser's online documentation.
- **Firefox:** Add an exception by opening the browser and navigating to `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which CSA is running. When the **This Connection is Untrusted** page opens, select

I Understand the Risks, click the **Add Exception** button, verify the Server Location, and click **Confirm Security Exception**. For information about how to import the certificate, refer to the browser's online documentation.

Step 7: Test Secure Connections

To test the connection to the Cloud Service Management Console, on a client system, open a supported Web browser and navigate to `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system that was used when the certificate was created. If the client browser is configured to accept the self-signed certificate (that is, you have completed step 6) and the Web application opens without a certificate warning, then you have successfully configured CSA to use a self-signed certificate. If you did not complete step 6, verify that the only certificate warning relates to the certificate not being issued by a trusted authority. If any other certificate warning is displayed, review steps 1-6 to be sure they were followed as documented.

Configure CSA to create a new self-signed certificate for global search

This section describes the process you should follow to create a new self-signed certificate required for global search functionality. These steps are required when a certificate expires, a new certificate is generated, or a self signed certificate is replaced with a CA-signed certificate.

Note: In the following instructions, `CSA_HOME` is the directory in which CSA is installed (for example, on Windows the directory is `C:\Program Files\HPE\CSA` and on Linux the directory is `/usr/local/hpe/csa`) and the `keytool` utility is included with the JRE.

Also, the following instructions are applicable for subordinate Certificate Authorities. Wherever the Certificate Authority is mentioned, the subordinate Certificate Authority is implied. For example, if the content states to submit the certificate to a Certificate Authority, you may also submit the certificate to a subordinate Certificate Authority.

To create a new self-signed certificate to send with your request to a Certificate Authority, complete the following steps:

1. Open a command prompt and change directories to CSA_HOME.
2. Run the following command to generate a new certificate and keystore:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -genkeypair -alias CSA -validity 365 -keyalg rsa -  
keysize 2048 -keystore <KEYSTORE> -storetype PKCS12
```

Linux:

```
CSA_JRE_HOME/bin/keytool -genkeypair -alias CSA -validity 365 -keyalg rsa -  
keysize 2048 -keystore <KEYSTORE> -storetype PKCS12
```

where <KEYSTORE> refers to the keystore location.

Example:

Windows:

```
"CSA_JR_HOME\bin\keytool" -genkeypair -alias CSA -validity 365 -keyalg rsa -  
keysize 2048 -keystore .new_keystore -storetype PKCS12
```

Linux:

```
CSA_JRE_HOME/bin/keytool -genkeypair -alias CSA -validity 365 -keyalg rsa -  
keysize 2048 -keystore .new_keystore -storetype PKCS12
```

You can use different values for -alias, -validity, -keysize and -keystore. These instructions assume that you will use the -alias and -keystore values recommended here. You will need to adjust the commands accordingly if you use different values.

3. Export the newly generated certificate out of keystore.

Windows:

```
"CSA_JR_HOME\bin\keytool" -exportcert -keystore <KEYSTORE> -alias CSA -file  
<CERTIFICATE-FILE> -storetype PKCS12
```

Linux:

```
CSA_JRE_HOME/bin/keytool -exportcert -keystore <KEYSTORE> -alias CSA -file  
<CERTIFICATE-FILE> -storetype PKCS12
```

where <CERTIFICATE-FILE> refers to the filename for the exported certificate.

Example:

Windows:

```
"CSA_JR_HOME\bin\keytool" -exportcert -keystore <KEYSTORE> -alias CSA -file  
csasearchcertificate.cert -storetype PKCS12
```

Linux:

```
CSA_JRE_HOME/bin/keytool -exportcert -keystore <KEYSTORE> -alias CSA -file  
csasearchcertificate.cert -storetype PKCS12
```

4. Import the newly created certificate into the JRE truststore.

Windows:

```
"CSA_JR_HOME\bin\keytool" -importcert -keystore <CSA_JR_  
HOME>\lib\security\cacerts -file <CERTIFICATE-FILE> -alias <ALIAS-NAME>
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -keystore <CSA_JR_  
HOME>\lib\security\cacerts -file <CERTIFICATE-FILE> -alias <ALIAS-NAME>
```

Note: <ALIAS-NAME> must be used since it does not exist in the JRE cacert truststore. If a similar name exists then delete it first as follows:

Windows:

```
"CSA_JR_HOME\bin\keytool" -delete -alias <ALIAS_NAME> -keystore <CSA_JR_  
HOME>\lib\security\cacerts
```

Linux:

```
CSA_JRE_HOME/bin/keytool -delete -alias <ALIAS_NAME> -keystore <CSA_JR_  
HOME>\lib\security\cacerts
```

5. Import the Elasticsearch certificate into the newly created keystore.

Windows:

```
"CSA_JR_HOME\bin\keytool" -importcert -keystore <KEYSTORE> -storetype PKCS12 -  
file <ELASTIC-SEARCH-CERTIFICATE> -trustcacerts -alias ES
```

Example:

```
"CSA_JR_HOME\bin\keytool" -importcert -keystore <KEYSTORE> -storetype PKCS12 -  
file CSA-HOME\elasticsearch-1.6.1\config\es.crt
```

Linux:

```
CSA_JR_HOME/bin/keytool -importcert -keystore <KEYSTORE> -storetype PKCS12 -  
file <ELASTIC-SEARCH-CERTIFICATE> -trustcacerts -alias ES
```

Example:

```
CSA_JR_HOME/bin/keytool -importcert -keystore <KEYSTORE> -storetype PKCS12 -  
file CSA-HOME/elasticsearch-1.6.1/config/es.crt
```

where <ELASTIC-SEARCH-CERTIFICATE> is the location of the elasticsearch certificate file.

6. Open the `CSA_HOME/CSA/csa-search-service/app.json` file in a text editor.
7. Change the following string (there are two occurrences):

Change: "pfx" : ".keystore"

To: "pfx": "<KEYSTORE>"

8. Restart the CSA and HPE Search Service services. See ["Restart CSA" on page 170](#) for instructions.

Masking Passwords in standalone.xml Using the JBoss vault Script

JBoss provides a script that allows passwords in the `standalone.xml` file to be masked. The following tasks describe how to use the JBoss vault script and configure CSA to use the masked password.

1. Verify that the `JAVA_HOME` environment variable has been defined and that `JAVA_HOME` has been set to the directory in which the JRE that is used by CSA is installed (for example, on Windows: `C:\Program Files\HPE\CSA\openjre` and on Linux: `/usr/local/hpe/csa/openjre`).

Note: Do NOT enclose the value in quotation marks, even if the path name includes a space. The vault script will fail if the `JAVA_HOME` variable definition contains quotation marks.

To verify that `JAVA_HOME` has been defined, from a command prompt, type:

```
echo JAVA_HOME
```

2. Create a keystore used by vault. This vault keystore is used to store the CSA keystore password.

Note: This example saves the vault keystore and encrypted vault file in the `CSA_HOME/jboss-as/standalone/configuration/` directory (the contents of this directory are

automatically backed up during an upgrade). You may choose to store the vault keystore and encrypted vault file in any location. However, you must remember to use those locations in subsequent steps in this task and, if those locations are not automatically backed up during upgrade, to manually back up the files before upgrade.

- a. Open a command prompt.
- b. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -genkey -alias vault -validity 365 -keyalg rsa  
-keysize 2048 -keystore .\jboss-as\standalone\configuration\csa_  
vault.keystore
```

Linux:

```
CSA_JRE_HOME/bin/keytool -genkey -alias vault -validity 365 -keyalg rsa  
-keysize 2048 -keystore ./jboss-as/standalone/configuration/csa_  
vault.keystore
```

where

CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed

You can use different values for `-alias`, `-validity`, `-keysize` and `-keystore`. These instructions assume that you will use the `-alias` and `-keystore` values recommended here; you will have to adjust the commands accordingly if you use different values.

- c. Enter the vault keystore password (for example, csavault).

This password is used to control access to the vault keystore. This password must be the same as the password you enter for the key in step e of this task.
 - d. Follow the prompts to enter your first and last name, organization, and location values.
 - e. Enter the key password. Click **Enter** to use the vault keystore password you supplied earlier (for example, csavault).

Although `keytool` allows you to enter different passwords for the keystore and the key, the two passwords must be the same to work with CSA.
3. Run the vault script. The script will generate the masked password and the values to configure in the `standalone.xml` file to use the masked password.
 - a. On Linux from the command prompt, make the vault script executable. Type: `chmod 775 CSA_HOME/jboss-as/bin/vault.sh`

- b. From the command prompt, type:

Windows:

```
CSA_HOME\jboss-as\bin\vault
```

Linux:

```
CSA_HOME/jboss-as/bin/vault.sh
```

- c. Select **0** to start the interactive session.
d. Enter the following information, when prompted, to configure the vault keystore:

Prompt	Description
Directory to store encrypted files	Directory in which the vault encrypted file is stored (for example, CSA_HOME/jboss-as/standalone/configuration). Verify that a vault encrypted file (VAULT.dat on Windows or ENC.dat on Linux) does not already exist in this directory. If the file exists, select a different directory.
Keystore URL	The name and location of the vault keystore (for example, CSA_HOME/jboss-as/standalone/configuration/csa_vault.keystore).
Keystore password (twice)	The password to the vault keystore (for example, csavault).
8 character salt	A random number (for example, 12345678).
Iteration count as a number	The number of times the CSA keystore password is hashed (for example, 25).
Keystore alias	The alias used to identify the CSA keystore password in the vault keystore (for example, vault).

- e. Make a copy of the vault property block that is displayed. For example, copy:

Windows:

```
<vault>
  <vault-option name="KEYSTORE_URL" value="CSA_HOME\jboss-
as\standalone\configuration\csa_vault.keystore"/>
  <vault-option name="KEYSTORE_PASSWORD" value="MASK-2PtpNyQsI1E7t"/>
  <vault-option name="KEYSTORE_ALIAS" value="vault"/>
  <vault-option name="SALT" value="12345678"/>
```

```
<vault-option name="ITERATION_COUNT" value="25"/>
<vault-option name="ENC_FILE_DIR" value="CSA_HOME\jboss-
as\standalone\configuration\"/>
</vault>
```

Linux:

```
<vault>
  <vault-option name="KEYSTORE_URL" value="CSA_HOME/jboss-
as/standalone/configuration/csa_vault.keystore"/>
  <vault-option name="KEYSTORE_PASSWORD" value="MASK-2PtpNyQsI1E7t"/>
  <vault-option name="KEYSTORE_ALIAS" value="vault"/>
  <vault-option name="SALT" value="12345678"/>
  <vault-option name="ITERATION_COUNT" value="25"/>
  <vault-option name="ENC_FILE_DIR" value="CSA_HOME/jboss-
as/standalone/configuration/"/>
</vault>
```

You will need to add this content to the `standalone.xml` file (the exact location is described in a later step).

- f. Select **0** to store a secured attribute.
- g. Enter the following information, when prompted, to generate the vault entry to use for the CSA keystore password in the `standalone.xml` file:

Prompt	Description
Secured attribute value (twice)	Enter the CSA keystore password (for example, changeit).
Vault Block	Enter a name for the vault block (for example, csa_keystore).
Attribute Name	Enter the attribute being stored (for example, password).

Note the VAULT entry (for example, `VAULT::csa_keystore::password::1`). You will need this value when you configure the `standalone.xml` file.

- h. Enter **2** to exit the script.

Note: The vault script converts the format of the vault keystore (for example, `CSA_HOME/jboss-as/standalone/configuration/csa_vault.keystore`) to JCEKS.

- 4. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.

5. Locate the following entry for the CSA server keystore (this entry may have been modified):

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore"  
keystore-password="changeit"/>
```

6. Update the entry by changing the value of the `keystore-password` attribute to the vault entry you generated (for example, `VAULT::csa_keystore::password::1`).

For example:

Windows:

```
<keystore path="CSA_HOME\jboss-as\standalone\configuration\.keystore"  
keystore-password="{VAULT::csa_keystore::password::1}"/>
```

Linux:

```
<keystore path="CSA_HOME/jboss-as/standalone/configuration/.keystore"  
keystore-password="{VAULT::csa_keystore::password::1}"/>
```

Add the vault property block to `<server xmlns="urn:jboss:domain:1.3">` after the `system-properties` block. For example, using the example values, enter the following:

Windows:

```
<server xmlns="urn:jboss:domain:1.3">  
.  
.  
.  
<system-properties>  
.  
.  
.  
</system-properties>  
<vault>  
  <vault-option name="KEYSTORE_URL" value="CSA_HOME\jboss-  
as\standalone\configuration\csa_vault.keystore"/>  
  <vault-option name="KEYSTORE_PASSWORD" value="MASK-2PtpNyQsI1E7t"/>  
  <vault-option name="KEYSTORE_ALIAS" value="vault"/>  
  <vault-option name="SALT" value="12345678"/>  
  <vault-option name="ITERATION_COUNT" value="25"/>  
  <vault-option name="ENC_FILE_DIR" value="CSA_HOME\jboss-  
as\standalone\configuration\"/>  
</vault>
```

Linux:

```
<server xmlns="urn:jboss:domain:1.3">
.
.
.
<system-properties>
.
.
.
</system-properties>
<vault>
  <vault-option name="KEYSTORE_URL" value="CSA_HOME/jboss-
as/standalone/configuration/csa_vault.keystore"/>
  <vault-option name="KEYSTORE_PASSWORD" value="MASK-2PtpNyQsI1E7t"/>
  <vault-option name="KEYSTORE_ALIAS" value="vault"/>
  <vault-option name="SALT" value="12345678"/>
  <vault-option name="ITERATION_COUNT" value="25"/>
  <vault-option name="ENC_FILE_DIR" value="CSA_HOME/jboss-
as/standalone/configuration/" />
</vault>
```

Note: In a clustered environment, add the vault xml entries in host.xml as shown below.

For example, using the example value, enter the following:

```
Host.xml -
<?xml version='1.0' encoding='UTF-8'?>
<host name="master_node" xmlns="urn:jboss:domain:1.2">

<vault>
<vault-option name="KEYSTORE_URL" value="CSA_
HOME\jbossas\standalone\configuration\csa_vault.keystore"/>
<vault-option name="KEYSTORE_PASSWORD" value="MASK-2PtpNyQsI1E7t"/>
<vault-option name="KEYSTORE_ALIAS" value="vault"/>
<vault-option name="SALT" value="12345678"/>
<vault-option name="ITERATION_COUNT" value="25"/>
<vault-option name="ENC_FILE_DIR" value="CSA_HOME\jbossas\
standalone\configuration\"/>
</vault>

<management>
  <security-realms>
    <security-realm name="ManagementRealm">
      <authentication>
        <properties path="mgmt-users.properties" relative-
to="jboss.domain.config.dir"/>
      </authentication>
    </security-realm>
  </security-realms>
</management>
```



```
        <security-realm name="ApplicationRealm">
            <authentication>
                <properties path="application-users.properties" relative-
to="jboss.domain.config.dir" />
            </authentication>
        </security-realm>
    </security-realms>
    <management-interfaces>
        <native-interface security-realm="ManagementRealm">
            <socket interface="management"
port="${jboss.management.native.port:9999}"/>
        </native-interface>
        <http-interface security-realm="ManagementRealm">
            <socket interface="management"
port="${jboss.management.http.port:9990}"/>
        </http-interface>
    </management-interfaces>
</management>
```

Configure Secure Connections for LDAP

If the LDAP server requires a secure connection, follow these steps to import the LDAP server Certificate Authority's root certificate into the Java truststore of CSA. If necessary, contact your LDAP administrator to obtain the LDAP server certificate.

If the LDAP server does not require a secure connection, you can omit this task.

Note: If you have configured CSA to be compliant with FIPS 140-2, you must substitute the CSA server truststore (for example, `csa_server_truststore.p12`) for the Java truststore (`cacerts`) and substitute the CSA server truststore password for the Java truststore password (`changeit`) in the examples. See the *Cloud Service Automation FIPS 140-2 Compliance Configuration Guide* for more information about the CSA server truststore and password.

1. Open a command prompt and run the `keytool` utility with the following options to create a local trusted certificate entry for the LDAP server.

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -trustcacerts -alias ldap
-keystore "CSA_JRE_HOME\lib\security\cacerts"
-file <c:\certificate_name.cer> -storepass changeit
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -trustcacerts -alias ldap  
-keystore CSA_JRE_HOME/lib/security/cacerts  
-file </tmp/certfile_name.cer> -storepass changeit
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed and `<c:\certfile_name.cer>` on Windows or `</tmp/certfile_name.cer>` on Linux is the path and name of the Certificate Authority's root certificate for the LDAP server. The file extension may be `.crt` rather than `.cer`. You can also use a different value for `-alias`.

2. At the prompt to import the certificate, type **Yes**.
3. Press **Enter**.
4. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

Configure Secure Connections for SMTP

For each organization, if its SMTP server requires a secure connection, follow these steps to import the SMTP server Certificate Authority's root certificate into the Java truststore of CSA. If necessary, contact your SMTP server administrator to obtain the SMTP server certificate.

If the SMTP server does not require a secure connection, you can omit this task.

Note: If you have configured CSA to be compliant with FIPS 140-2, you must substitute the CSA server truststore (for example, `csa_server_truststore.p12`) for the Java truststore (`cacerts`) and substitute the CSA server truststore password for the Java truststore password (`changeit`) in the examples. See the *Cloud Service Automation FIPS 140-2 Compliance Configuration Guide* for more information about the CSA server truststore and password.

1. Open a command prompt and run the `keytool` utility with the following options to create a local trusted certificate entry for the SMTP server.

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -trustcacerts -alias smtp  
-keystore "CSA_JRE_HOME\lib\security\cacerts"  
-file <c:\certfile_name.cer> -storepass changeit
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -trustcacerts -alias smtp  
-keystore CSA_JRE_HOME/lib/security/cacerts  
-file </tmp/certfile_name.cer> -storepass changeit
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed and <c:\certfile_name.cer> on Windows </tmp/certfile_name.cer> on Linux is the path and name of the Certificate Authority's root certificate for the SMTP server. The file extension may be .crt rather than .cer. You can also use a different value for -alias.

2. At the prompt to import the certificate, type **Yes**.
3. Press **Enter**.
4. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

Configure Secure Connections for an Oracle Database

If the Oracle database server requires a secure connection, complete the following steps (if the Oracle database does not require a secure connection, you can omit these steps):

Note: If you have configured CSA to be compliant with FIPS 140-2, you cannot configure a secure connection for the Oracle database. If you configure a secure connection for the Oracle database, you cannot configure CSA to be compliant with FIPS 140-2.

1. Complete one of the following tasks:
 - o If you do not want to configure CSA to check the database DN, do the following:
 - i. Open CSA_HOME/jboss-as/standalone/configuration/standalone.xml in a text editor.
 - ii. Add the following to the Oracle datasource:

```
<connection-url>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL = TCPS)(HOST = <host>)(PORT = 1521)))(CONNECT_DATA =(SERVICE_NAME = ORCL)))</connection-url>
```

where <host> is the name of the system on which the Oracle database server is installed.

- iii. Save and close the file.
- iv. Import the Oracle database server Certificate Authority's root certificate into the Java truststore of CSA.
 - A. Copy the Oracle database server Certificate Authority's root certificate to the CSA system. If necessary, contact your database administrator to obtain the Oracle database server certificate.
 - B. On the CSA system, open a command prompt and run the `keytool` utility with the following options to create a local trusted certificate entry for the Oracle database server.

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -trustcacerts  
-alias oracledb  
-keystore "CSA_JRE_HOME\lib\security\cacerts"  
-file <c:\certfile_name.cer> -storepass changeit
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -trustcacerts  
-alias oracledb  
-keystore CSA_JRE_HOME/lib/security/cacerts  
-file </tmp/certfile_name.cer> -storepass changeit
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed and `<c:\certfile_name.cer>` on Windows or `</tmp/certfile_name.cer>` on Linux is the path and name of the Certificate Authority's root certificate for the Oracle database server. The file extension may be `.crt` rather than `.cer`. You can also use a different value for `-alias`.

- C. At the prompt to import the certificate, type **Yes**.
- D. Press **Enter**.
- E. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

- o If you want to configure CSA to check the database DN, do the following:
 - i. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.
 - ii. Add the following to the Oracle datasource:

```
<connection-url>jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCPS)(HOST = <host>)(PORT = 1521)))(CONNECT_DATA = (SERVICE_NAME = ORCL))(SECURITY=(SSL_SERVER_CERT_DN="CN=abc,OU=dbserver,O=xyz,L=Sunnyvale,ST=CA,C=US")))</connection-url>
```

where <host> is the name of the system on which the Oracle database server is installed and the values for SSL_SERVER_CERT_DN are for the DN of the Oracle database server.

- iii. Add the following to the system-properties element:

```
<property name="oracle.net.ssl_server_dn_match" value="true" />
```

- iv. Save and close the file.
- v. Import the Oracle database server Certificate Authority's root certificate into the Java truststore of CSA.
 - A. Copy the Oracle database server Certificate Authority's root certificate to the CSA system. If necessary, contact your database administrator to obtain the Oracle database server certificate.
 - B. On the CSA system, open a command prompt and run the keytool utility with the following options to create a local trusted certificate entry for the Oracle database server.

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -trustcacerts  
-alias oracledb  
-keystore "CSA_JRE_HOME\lib\security\cacerts"  
-file <c:\certfile_name.cer> -storepass changeit
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -trustcacerts  
-alias oracledb  
-keystore CSA_JRE_HOME/lib/security/cacerts  
-file </tmp/certfile_name.cer> -storepass changeit
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed and <c:\certfile_name.cer> on Windows or </tmp/certfile_name.cer> on Linux is the path and name of the Certificate Authority's root certificate for the Oracle database server. The file extension may be .crt rather than .cer. You can also use a different value for -alias.

- C. At the prompt to import the certificate, type **Yes**.
- D. Press **Enter**.

E. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

2. If client authentication is enabled on the Oracle database server, do the following:

- a. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.
- b. Add the following to the `system-properties` element:

```
<property name="javax.net.ssl.keyStore" value="<certificate_key_file>" />
<property name="javax.net.ssl.keyStorePassword" value="<certificate_key_
file_password>" />
<property name="javax.net.ssl.keyStoreType" value="<certificate_key_file_
type>" />
```

where `<certificate_key_file>` is the same keystore file defined by the `certificate-key-file` attribute in the `ssl` element (for example, `CSA_HOME/jboss-as/standalone/configuration/.keystore`), `<certificate_key_file_password>` is the password to the keystore file (for example, `changeit`), and `<certificate_key_file_type>` is the keystore type (for example, `JKS` or `PKCS12`).

- c. Save and close the file.
- d. Use Oracle's wallet manager to import CSA's certificate into the Oracle database server's wallet as a trusted certificate.

Configure Secure Connections for Microsoft SQL Server

If Microsoft SQL Server requires a secure connection, complete the following steps (if Microsoft SQL Server does not require a secure connection, you can omit these steps):

1. Open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor.
2. Locate the `connection-url` entry for the Microsoft SQL Server datasource and change `ssl=request` to `ssl=authenticate`.

For example:

```
<connection-url>
  jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=requestauthenticate
</connection-url>
```

3. Save and close the file.
4. Import the Microsoft SQL Server Certificate Authority's root certificate into the Java truststore of CSA.
 - a. Copy the Microsoft SQL Server Certificate Authority's root certificate to the CSA system. If necessary, contact your database administrator to obtain the Microsoft SQL Server certificate.
 - b. On the CSA system, open a command prompt and run the `keytool` utility with the following options to create a local trusted certificate entry for the Microsoft SQL Server.

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -trustcacerts  
-alias mssqlldb -keystore "CSA_JRE_HOME\lib\security\cacerts"  
-file <c:\certfile_name.cer> -storepass changeit
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -trustcacerts  
-alias mssqlldb -keystore CSA_JRE_HOME/lib/security/cacerts  
-file </tmp/certfile_name.cer> -storepass changeit
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed and `<c:\certfile_name.cer>` on Windows or `</tmp/certfile_name.cer>` on Linux is the path and name of the Certificate Authority's root certificate for the Oracle database server. The file extension may be `.crt` rather than `.cer`. You can also use a different value for `-alias`.

- c. At the prompt to import the certificate, type **Yes**.
- d. Press **Enter**.
- e. Restart CSA.

See "[Restart CSA](#)" on page 170 for instructions.

Configure Secure Connections for Operations Orchestration Load Balancer

If the Operations Orchestration Load Balancer (OO LB) server requires a secure connection, follow these steps to import the OO LB server Certificate Authority's root certificate into the Java truststore of CSA. If necessary, contact your OO LB administrator to obtain the OO LB server certificate.

Note: If you have configured CSA to be compliant with FIPS 140-2, you must substitute the CSA server truststore (for example, `csa_server_truststore.p12`) for the Java truststore (`cacerts`) and substitute the CSA server truststore password for the Java truststore password (`changeit`) in the examples. See the *Cloud Service Automation FIPS 140-2 Compliance Configuration Guide* for more information about the CSA server truststore and password.

For each system running CSA, import the root certificate of OO LB's Certificate Authority into CSA (you must first export OO LB's certificate from OO LB's truststore and then import it into CSA's truststore).

1. Open OO LB in a Web browser (using https).

Export the certificate from the Web browser.

If you are using a Chrome Web browser, do the following:

- a. In the address bar, click the lock icon with the red X over it and select **certificate information**.
- b. In the Certificate dialog, do the following:
 - i. Select the **Details** tab.
 - ii. Click **Copy to File**.
 - iii. In the Certificate Export Wizard, do the following:
 - A. Click **Next**.
 - B. Select **Base-64 encoded X.509 (.CER)** and click **Next**.
 - C. Click **Browse** and select a directory in which to save the certificate.
 - If you are running OO LB on the same system as CSA, select the `CSA_JRE_HOME/lib/security` directory (where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed), enter **paslb.cer** as the file name, and click **Save**.
 - If you are running OO LB on a system that is not running CSA, select a directory in which to store the certificate file, enter **paslb.cer** as the file name, and click **Save**.
 - D. Click **Next**.
 - E. Click **Finish**.
 - F. Click **OK**.
 - iv. Click **OK**.

- 2.

If you are using a Firefox Web browser, do the following:

- a. Click **Add Exception**.
- b. In the Add Security Exception dialog, click **View**.
- c. In the Certificate Viewer, do the following:
 - i. Select the **Details** tab.
 - ii. Click **Export**.
 - iii. Select a directory in which to save the certificate.
 - If you are running OO LB on the same system as CSA, select the `CSA_JRE_HOME/lib/security` directory (where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed), enter **paslb.cer** as the file name, select **X.509 Certificate (PEM)** as the Type, and click **Save**.
 - If you are running OO LB on a system that is not running CSA, select a directory in which to store the certificate file, enter **paslb.cer** as the file name, select **X.509 Certificate (PEM)** as the Type, and click **Save**.
 - iv. Click **Close**.
- d. Click **Cancel**.

If you are using a Windows IE Web browser, do the following:

- a. In the address bar, click **Certificate Error** and select **View certificates**.
- b. In the Certificate Export Wizard, do the following:
 - i. Select the **Details** tab.
 - ii. Click **Copy to File**.
 - iii. In the Certificate Export Wizard, do the following:
 - A. Click **Next**.
 - B. Select **Base-64 encoded X.509 (.CER)** and click **Next**.
 - C. Click **Browse** and select a directory in which to save the certificate.
 - If you are running OO LB on the same system as CSA, select the `CSA_JRE_HOME/lib/security` directory (where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed), enter **paslb.cer** as the file name, and click **Save**.
 - If you are running OO LB on a system that is not running CSA, select a directory in which to store the certificate file, enter **paslb.cer** as the file name, and click **Save**.
 - D. Click **Next**.

- E. Click **Finish**.
 - F. Click **OK**.
 - iv. Click **OK**.
 3. If you are running OO LB on a system that is not running CSA, copy the `paslb.cer` file to the `CSA_JRE_HOME/lib/security` directory on the system running CSA (where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed).
 4. On the system running CSA, open a command prompt and run the following commands:
Windows:

```
cd "CSA_JRE_HOME\lib\security"

..\..\bin\keytool -importcert -alias paslb -file paslb.cer
-keystore cacerts -storepass changeit
```

Linux:

```
cd CSA_JRE_HOME/lib/security

../../bin/keytool -importcert -alias paslb -file paslb.cer
-keystore cacerts -storepass changeit
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.
 5. When prompted to trust the certificate, enter `yes`.

Configure Secure Internal Communication

CSA global search services use HTTPS to communicate internally with CSA on TCP ports that are not normally used for communication with other systems, the Cloud Service Management Console, or the Marketplace Portal. To prevent access to these ports from an internal or external network, it is recommended that you configure network firewall rules for these ports. Consult your network administrator about configuring firewall rules.

TCP Ports Used for Internal Communication

CSA Service	TCP Port Used	Communication between Nodes in a Clustered Environment?
Search Service	9000	No

TCP Ports Used for Internal Communication, continued

CSA Service	TCP Port Used	Communication between Nodes in a Clustered Environment?
Elasticsearch	9201	No
Elasticsearch	9300	Yes

Authentication Checking

Authentication checking has been turned on in order to improve security of the HPE Search Service. A certificate has been generated to make authorization checking possible.

Note: The certificate will expire 120 days after the patch has been installed.

Steps to generate a new certificate valid for another 120 days is given below:

Points to Note:

- The keytool commands below will prompt for a password. The default password is 'changeit'.
 - The following commands will fail unless you use the same Java executable that is used to run CSA.
1. On the server open a shell and change directory to %CSA_HOME%\elasticsearch-1.6.1\config.
 2. Make a backup of the CSAKS.jks and CSATS.jks files.
 3. To see the current ES certificate:

```
keytool -list -v -keystore CSAKS.jks -alias ES
```

Take note of the line that begins with "Owner" specifically the value after "CN=". This should be the FQDN (fully qualified domain name) of the server.

4. Delete the existing expired certificate:

```
keytool -delete -alias ES -keystore CSAKS.jks
```

5. Generate a new certificate:

```
keytool -genkey -alias ES -keyalg RSA -dname "CN=, OU=HP, O=HP, L=Palo Alto, ST=CA, C=US" -validity 120 -keystore CSAKS.jks
```

Note: CN= should contain the FQDN (fully qualified domain name) of the server.

6. Import certificate to CSATS.jks file:

```
keytool -export -alias ES -keystore CSAKS.jks -file es.crt
```

```
keytool -delete -alias ES -keystore CSATS.jks
```

```
keytool -importcert -trustcacerts -alias ES -file es.crt -keystore CSATS.jks
```

Enter 'yes' to trust this certificate.

7. Import certificate in search service:

For Windows:

```
keytool -delete -alias ES -keystore ..\..\csa-search-service\keystore -  
storetype PKCS12
```

```
keytool -importcert -alias ES -file es.crt -keystore ..\..\csa-search-  
service\keystore -storetype PKCS12 -trustcacerts
```

For Linux:

```
keytool -delete -alias ES -keystore ../../csa-search-service/.keystore -  
storetype PKCS12
```

```
keytool -importcert -alias ES -file es.crt -keystore ../../csa-search-  
service/.keystore -storetype PKCS12 -trustcacerts
```

Enter 'yes' to trust certificate.

8. Restart Elasticsearch service and HPE Search service.

Operations Orchestration

The CSA solution includes a number of Operations Orchestration (OO) flows that perform CSA operations.

Note: If you followed the instructions in the *Cloud Service Automation Installation Guide* or *Cloud Service Automation Upgrade Guide* to configure Operations Orchestration, you should have already completed the tasks in this section.

In this release, you can install Operations Orchestration with CSA using the CSA installer or you can install Operations Orchestration externally. Only one instance of Operations Orchestration is required for both topology and sequential designs. If you have upgraded from an earlier version of CSA that uses multiple instances of Operations Orchestration for sequential designs, you can continue to use the multiple instances of Operations Orchestration for sequential designs. If you have upgraded from an earlier version of CSA that uses only a single instance of Operations Orchestration or are installing CSA for the first time, only one configured instance of Operations Orchestration is supported.

This chapter describes the following tasks:

- ["Configure OO for Topology Designs" below](#)
- ["Configure OO for Sequential Designs" on page 87](#)

Note: If you are configuring OO for both topology and sequential designs, complete the configuration for topology designs before the configuration for sequential designs.

Configure OO for Topology Designs

The following tasks are to configure OO for topology designs. Configure only one instance of OO for topology designs.

Note: If you followed the instructions in the *Cloud Service Automation Installation Guide* or *Cloud Service Automation Upgrade Guide* to configure Operations Orchestration, you should have already completed the tasks in this section.

Complete the following tasks to configure OO to integrate with CSA:

- [Configure an internal user](#)
- [Deploy Content Packs](#)
- [Configure OO Properties in the csa.properties File](#)
- [Configure a secure connection between CSA and OO](#)
- [Run the Cloud Content Capsule Installer](#)
- [Update and Redeploy the Service Manager Base Content Pack](#)
- [Configure Single Sign-On between CSA and OO](#)
- [Obscure passwords in OO Flows \(optional\)](#)

Note: In the following instructions, `CSA_HOME` is the directory in which CSA is installed and `OO_HOME` is where you installed Operations Orchestration.

Be sure all the latest patches for Operations Orchestration have been installed. See the *Cloud Service Automation System and Software Support Matrix* for more information.

Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com> (this site requires a Passport ID). Select **Dashboards > Manuals**.

Configure an Internal User

Internal users can be used to configure Operations Orchestration for CSA.

This user is used for provisioning topology designs.

1. Log in to OO Central.
2. Click the **System Configuration** button.
3. Select **Security > Internal Users**.
4. Click the + (Add) icon.
5. Enter the following information:

Field	Recommended Value
User Name	admin
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The admin user is used with HP Single Sign-On (HPSSO). When Operations Orchestration is launched from the Cloud Service Management Console, this user allows access to Operations Orchestration without having to log in. If you are using topology designs, the admin user can also be used for provisioning topology designs.

6. Click **Save**.

Deploy Content Packs

Note: You can also deploy content packs from the **Content Store**. Refer to online help for more details.

1. From OO Central, click **Content Management**.
2. Click the **Content Packs** tab.
3. Click the **Deploy New Content** icon.
4. In the Deploy New Content dialog, in the upper left corner, click the + (Add files for deployment) icon.
5. Deploy the Component Tool content packs. From Operations Orchestration Central, navigate to the `tools/oo/content-packs` directory. Add and deploy the content packs (after each successful deployment, to add and deploy the next content pack without closing the dialog, click the **Reset** icon in the upper left corner to clear the dialog and enable the + (Add files for deployment) icon):

The deployment may take a few minutes and the dialog will show a progress bar.

6. When the deployment succeeds, click **Close** to close the dialog.

Configure OO Properties in the csa.properties File

If you integrated with OO using the installer during the installation or upgrade process during the installation or upgrade process, you do not need to configure these properties (they are already configured). These properties are used to integrate with Operations Orchestration. In the subscription

event overview section of the **[[[Undefined variable CSAVariables.tabOperations]]]** area in the Cloud Service Management Console, selecting the Process ID opens Operations Orchestration to the detailed page of the selected process when these properties are configured.

Edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file and configure the following properties:

Property	Description
OOS_URL	<p>The URL used to access Operations Orchestration Central. This is the Operations Orchestration used for provisioning topology designs. For example, <code>https://<hostname>:8445</code>.</p> <p>This property is automatically set during installation. If you are using the embedded Operations Orchestration that is included with CSA, this property is set using the values entered for the Fully qualified domain name on Windows or the Fully Qualified Hostname on Linux and HPE OO Port fields during installation. If you are using a standalone/external Operations Orchestration, this property is set using the values entered for the HPE OO Hostname and HPE OO Port fields during installation.</p>
OOS_USERNAME	<p>The username used to log in to Operations Orchestration Central.</p> <p>This property is automatically set during installation using the value entered for the HPE OO User Name field during installation.</p>
OOS_PASSWORD	<p>The encrypted password used by the user defined in OOS_USERNAME to log in to Operations Orchestration Central.</p> <p>This property is automatically set during installation using the value entered for the HPE OO Password field during installation.</p>
embedded.oo.root.dir	<p>Location of the embedded Operations Orchestration when it is installed with CSA. This property is generated when embedded Operations Orchestration is installed during the CSA installation.</p> <p>This property is the only indicator of embedded Operations Orchestration, which is important mainly for uninstallation and upgrades. This property cannot be edited.</p>

Configure a Secure Connection between CSA and OO

If you integrated with OO using the installer during the installation or upgrade process, you do not need to configure a secure connection (it has already been configured).

Export Operations Orchestration's certificate from Operations Orchestration's truststore. If Operations Orchestration and CSA are not installed on the same system, copy the certificate to the CSA system and import the certificate into CSA's truststore. TLS must be configured between CSA and Operations Orchestration.

Do the following:

1. On the system running Operations Orchestration, open a command prompt and change to the directory where Operations Orchestration is installed.
2. Run the following command:

Windows:

```
.\java\bin\keytool -export -alias tomcat -file C:\oo.cer  
-keystore .\Central\var\security\key.store -storepass changeit
```

Linux:

```
./java/bin/keytool -export -alias tomcat -file /tmp/oo.cer  
-keystore ./Central/var/security/key.store -storepass changeit
```

where C:\oo.cer on Windows and /tmp/oo.cer on Linux are examples is an example of a filename and location used to store the exported root certificate (you can choose a different filename and location).

3. If Operations Orchestration is not running on the same system as CSA, copy oo.cer from the Operations Orchestration system to the system running CSA.
4. On the system running CSA, open a command prompt.
5. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias tomcat -file C:\oo.cer -  
trustcacerts -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias tomcat -file /tmp/oo.cer -  
trustcacerts -keystore CSA_JRE_HOME/lib/security/cacerts
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed

6. When prompted for the keystore password, enter changeit.
7. Enter yes when prompted to trust the certificate.

Run the Cloud Content Capsule Installer

The Cloud Content Capsule Installer is used to install and update content for CSA and Operations Orchestration.

1. Open a command prompt and navigate to the CSA_HOME/Tools/CSLContentInstaller directory.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\java"\bin\java -jar csl-content-installer.jar
```

Linux:

```
CSA_JRE_HOME/bin/java -jar csl-content-installer.jar
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed.

3. From the installer, enter the information to deploy content to Operations Orchestration and import service designs into CSA.

For more information about the Cloud Content Capsule Installer, see the *Cloud Service Automation Capsule Installer Guide*.

Update and Redeploy the Service Manager Base Content Pack

Update and redeploy the oo10-sm-cp-1.0.3.jar base content pack. If you deployed an earlier version of the Service Manager base content pack, you must do the following (if this is a fresh installation of Operations Orchestration and you did not deploy an earlier version of the Service Manager base content pack, you do not have to complete these steps):

1. Stop the Operations Orchestration services:

Windows:

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the HPE Operations Orchestration Central service and select **Stop**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Stop**.

Linux:

- a. On the server that hosts Operations Orchestration, run the following command:
`<HPE00installation>/central/bin/central stop`
For example, `/usr/local/hpe/csa/00/central/bin/central stop`
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: `<HPE00installation>/ras/bin/ras stop`.
For example, `/usr/local/hpe/csa/00/ras/bin/ras stop`

2. Clear the Operations Orchestration Central cache by deleting the following folder:

`<HPE00installation>/central/var/cache`

For example,

Windows: C:\Program Files\HPE\HPE Operations Orchestration\central\var\cache

Linux: /usr/local/hpe/csa/oo/central/var/cache

3. If RAS is installed, clear the RAS artifact cache by deleting the following folder (on all RAS systems, including localhost):

`<HPE00installation>/ras/var/cache`

For example,

Windows: C:\Program Files\HPE\HPE Operations Orchestration\ras\var\cache

Linux: /usr/local/hpe/csa/oo/ras/var/cache

4. Run the following SQL command against the Operations Orchestration database:

```
DELETE from OO_ARTIFACTS where NAME =  
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.pom' or NAME =  
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.jar'
```

5. Start the Operations Orchestration services:

Windows:

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click the HPE Operations Orchestration Central service and select **Start**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Start**.

Linux:

- a. On the server that hosts Operations Orchestration, run the following command:
`<HPE00installation>/central/bin/central start`
For example, `/usr/local/hpe/csa/00/central/bin/central start`
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: `<HPE00installation>/ras/bin/ras start`.
For example, `/usr/local/hpe/csa/00/ras/bin/ras start`

6. Redeploy the oo10-sm-cp-1.0.3.jar base content pack:

- a. Log in to Operations Orchestration Central and click **Content Management**.
- b. Click the **Content Packs** tab.
- c. Click the **Deploy New Content** icon.
- d. In the Deploy New Content dialog, in the upper left corner, click the **+** (Add files for deployment) icon.
- e. Navigate to the `CSA_HOME/oo/ooContentPack` directory and select **oo10-sm-cp-1.0.3.jar**.
- f. Click **Deploy**.
The deployment may take a few minutes and the dialog will show a progress bar.
- g. Click **Close**.

Configure Single Sign-On between CSA and OO

If HP Single Sign-On (HP SSO) was enabled during installation of CSA, SSO can be configured between CSA and Operations Orchestration. Configuring HP SSO allows you to launch Operations Orchestration from the Cloud Service Management Console without having to log in to Operations Orchestration.

CSA provides a login user (admin) and password (cloud) and, earlier in this guide, you configured an internal user for Operations Orchestration with the same user name and password. When Single Sign-On is configured between CSA and Operations Orchestration, this user can be used for single sign-on. That is, if you are logged in to CSA as the admin user, you can launch Operations Orchestration from the Cloud Service Management Console and not have to log in to Operations Orchestration.

You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and the embedded Operations Orchestration to use the same LDAP source or, if CSA and the embedded Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the embedded Operations Orchestration user must be assigned any role that allows flows to be viewed.

Note: To use HP SSO between CSA and Operations Orchestration, the systems on which CSA and Operations Orchestration are installed must be in the same domain.

Configure and Enable HP Single Sign-On

To configure and enable HP SSO on Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > SSO**.
4. Select the **Enable** checkbox.
5. Enter the **InitString**. The `initString` setting for CSA and Operations Orchestration must be configured to the same value. In CSA, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/idm.war/WEB-INF/hpsssoConfiguration.xml` file. The `initString` value represents a secret key and should be treated as such in your environment (this string is used to encrypt and decrypt the `LWSSO_COOKIE_KEY` cookie that is used

to authenticate the user for single sign-on).

6. Enter the **Domain**. This is the domain name of the network of the servers on which CSA and Operations Orchestration are installed.
7. Click **Save**.

Configure LDAP Users for Single Sign-On

To enable single sign-on for LDAP users, you must either configure CSA and Operations Orchestration to use the same LDAP source or, if CSA and Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the Operations Orchestration user must be assigned any role that allows flows to be viewed.

To configure LDAP for Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > LDAP**.
4. Enter the information to configure LDAP.
5. Click **Save**.

Obscure Passwords in OO Flows (Optional)

Some OO flows included with CSA may show passwords in clear text when viewed in OO Central. You can obscure these passwords by modifying the flow in OO Studio.

Note: You must have OO Studio installed. OO Studio is supported on Windows platforms only and is not part of the embedded OO that is included with CSA. See the OO documentation, such as the *OO System Requirements*, for more information about OO Studio.

To obscure passwords in OO flows:

1. Open OO Studio.
2. Locate the flow to update.
3. Right-click on the flow and select **References > What uses this?**.

A list of flows that use the flow is displayed (that is, the flow to update is a subflow of the flows displayed in the list).

4. Select a flow from the list of flows.
5. Locate the subflow (the flow to update).
6. Right-click on the subflow and select **Properties**.
7. Located the property to obscure (such as a password), enable it, but do not assign a value to it.
8. Save the flow.
9. Repeat this procedure for every flow from the list of flows.

Configure OO for Sequential Designs

The following tasks are to configure OO for sequential designs. If you are installing CSA for the first time, configure only one instance of OO. If you have upgraded from an earlier version of CSA that has multiple instances of OO configured for sequential designs, you can continue to use multiple instances of OO, including OO 9.07.

Note: If you followed the instructions in the *Cloud Service Automation Installation Guide* or *Cloud Service Automation Upgrade Guide* to configure Operations Orchestration, you should have already completed the tasks in this section.

Complete the following tasks to configure Operations Orchestration to integrate with CSA:

- [Add a JRE to the System Path](#)
- [Install the CSA Content Pack](#)
- [Configure Internal Users](#)
- [Deploy Content Packs Required by CSA](#)
- [Set Up System Accounts for the CSA Content Pack](#)
- [Set Up System Properties for the CSA Content Pack](#)
- [Configure a Secure Connection Between CSA and Operations Orchestration](#)
- [Run the Cloud Content Capsule Installer](#)
- [Update and Redeploy the Service Manager Base Content Pack](#)
- [Configure Single Sign-On between CSA and Operations Orchestration](#)
- [Obscure passwords in Operations Orchestration flows \(optional\)](#)

Note: In the following instructions,
CSA_HOME is the directory in which CSA is installed
and OO_HOME is where you installed Operations Orchestration.

Be sure all the latest patches for Operations Orchestration have been installed. See the *Cloud Service Automation System and Software Support Matrix* for more information.

Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com>
(this site requires a Passport ID). Select **Dashboards > Manuals**.

Add a JRE to the System Path

The CSA flows that are imported require that a JRE be included in the system path on the system running CSA.

To add a JRE to the system path on Windows, complete the following steps:

1. Open the **Environment Variables** dialog:
 - a. Right-click **Computer** and select **Properties**.
 - b. Select **Advanced System Settings**.
 - c. Click **Environment Variables**.
2. Select the **Path** system variable.
3. Click **Edit**.
4. At the end of the value for **Variable value**, add a semicolon (;) and the following path:

If Operations Orchestration and CSA are installed on the same system:

```
OO_HOME/java/bin
```

or

If Operations Orchestration and CSA are installed on different systems:

```
CSA_JRE_HOME/bin
```

5. Click **OK** and close all windows.

To add a JRE to the system path on Linux, complete the following steps:

Open a shell and enter one of the following commands:

- If Operations Orchestration and CSA are installed on the same system:

```
export PATH=$PATH:$OO_HOME/java/bin
```

- If Operations Orchestration and CSA are installed on different systems:

```
export PATH=$PATH:$CSA_JRE_HOME/bin
```

Note: By setting the system path, all applications (that require a JRE) use the JRE that is installed with Operations Orchestration or CSA (depending on the path you configured and if it is the only path or the first path set to a JRE in the system path). If you need to run another JRE with an application, you must type in the relative path to that JRE to run it (for example, when you configure TLS).

Configure Internal Users

Internal users can be used to configure Operations Orchestration for CSA.

1. From the system on which CSA is installed (the system on which the content packs are installed), log in to Operations Orchestration Central.
2. Click **System Configuration**.
3. Select **Security > Internal Users**.
4. Click the **+** (Add) button.
5. Enter the following information:

Field	Recommended Value
User Name	csaouser
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The csaouser user is used to import the Operations Orchestration flows. When importing flows, this user is configured in the Operations Orchestration input file.

6. Click **Save**.
7. Click the **+** (Add) button.
8. Enter the following information:

Field	Recommended Value
User Name	csaouser
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The admin user is used with HP Single Sign-On (HP SSO). When Operations Orchestration is launched from the Cloud Service Management Console, this user allows access to HPE Operations Orchestration without having to log in. If you are using topology designs, the admin user can also be used for provisioning topology designs.

9. Click **Save**.
10. Log out of Operations Orchestration Central and log back in as the csaouser.

Deploy Content Packs Required by CSA

If you are integrating CSA with an external OO, then in that external OO, you must deploy some of the content packs that are required by CSA.

Deploy the following content packs:

- oo10-base-cp.jar - 1.9.0
- oo10-cloud-cp.jar - 2.1.0
- oo10-hpe-solutions-cp.jar - 1.10.0
- oo10-virtualization-cp.jar - 1.9.1

These JAR files are available in the directory `%CSA_HOME%\oo\ooContentPack`. Perform the following steps to deploy the content packs.

1. From Operations Orchestration Central, click **Content Management**.
2. Click the **Content Packs** tab.
3. Click the **Deploy New Content** icon.
4. In the **Deploy New Content** dialog, in the upper left corner, click the **+** (Add files for deployment) icon.
5. Deploy the Component Tool content packs. From Operations Orchestration Central, navigate to the `tools/oo/content-packs` directory. Add and deploy the content packs (after each successful

deployment, to add and deploy the next content pack without closing the dialog, click the Reset icon in the upper left corner to clear the dialog and enable the + (Add files for deployment) icon.

The deployment may take a few minutes and the dialog will show a progress bar. When you have finished deploying all the content packs, click **Close** to close the dialog.

Set Up System Accounts for the CSA Content Pack

Set up system accounts for the content packs:

1. Log in to Operations Orchestration Central.
2. Click **Content Management**.
3. Select **Configuration Items > System Accounts**.
4. Click the **Add** icon.
5. Enter the following information if it is not already configured:

Field	Recommended Value
System Account Name	CSA_REST_CREDENTIALS
User Name	oolnboundUser
Password	cloud

Note: The **User Name** configured for the CSA_REST_CREDENTIALS System Account setting must match the **Override Value** configured for the CSA_00_USER System Property setting.

6. Click **Save**.
7. Click the **Add** icon.
8. Enter the following information if it is not already configured:

Field	Recommended Value
System Account Name	CSA_SERVICEMANAGER_CREDENTIALS
User Name	falcon
Password	<leave_blank>

9. Click **Save**.

Set Up System Properties for the CSA Content Pack

Set up the following system properties for the content packs:

1. Log in to Operations Orchestration Central.
2. Click **Content Management**.
3. Select **Configuration Items > System Properties**.
4. Click the **Add** icon.
5. Enter the following information if it is not already configured:

Field	Recommended Value
Name	CSA_REST_URI
Override Value	https://<csa_hostname>:8444/csa/rest

6. Click **Save**.

Configure a Secure Connection between CSA and Operations Orchestration

If you integrated with Operations Orchestration using the installer (during the installation or upgrade process), you do not need to configure a secure connection (it has already been configured).

Export Operations Orchestration's certificate from Operations Orchestration's truststore. If Operations Orchestration and CSA are not installed on the same system, copy the certificate to the CSA system and import the certificate into CSA's truststore. TLS must be configured between CSA and Operations Orchestration.

Do the following:

1. On the system running Operations Orchestration, open a command prompt and change to the directory where Operations Orchestration is installed.
2. Run the following command:

Windows:

```
.\java\bin\keytool -export -alias tomcat -file C:\oo.cer  
-keystore .\Central\var\security\key.store -storepass changeit
```

Linux:

```
./java/bin/keytool -export -alias tomcat -file /tmp/oo.cer  
-keystore ./Central/var/security/key.store -storepass changeit
```

where C:\oo.cer on Windows and /tmp/oo.cer on Linux are examples is an example of a filename and location used to store the exported root certificate (you can choose a different filename and location).

3. If Operations Orchestration is not running on the same system as CSA, copy oo.cer from the Operations Orchestration system to the system running CSA.
4. On the system running CSA, open a command prompt.
5. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias tomcat -file C:\oo.cer -  
trustcacerts -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias tomcat -file /tmp/oo.cer -  
trustcacerts -keystore CSA_JRE_HOME/lib/security/cacerts
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed

6. When prompted for the keystore password, enter changeit.
7. Enter yes when prompted to trust the certificate.

Run the Cloud Content Capsule Installer

The Cloud Content Capsule Installer is used to install and update content for CSA and Operations Orchestration.

1. Open a command prompt and navigate to the CSA_HOME/Tools/CSLContentInstaller directory.
2. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\java"\bin\java -jar csl-content-installer.jar
```

Linux:

```
CSA_JRE_HOME/bin/java -jar csl-content-installer.jar
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

3. From the installer, enter the information to deploy content to Operations Orchestration and import service designs into CSA.

For more information about the Cloud Content Capsule Installer, see the *Cloud Service Automation Capsule Installer Guide*.

Update and Redeploy the Service Manager Base Content Pack

Update and redeploy the `oo10-sm-cp-1.0.3.jar` base content pack. If you deployed an earlier version of the Service Manager base content pack, you must do the following (if this is a fresh installation of Operations Orchestration and you did not deploy an earlier version of the Service Manager base content pack, you do not have to complete these steps):

1. Stop the Operations Orchestration services:

Windows:

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the HPE Operations Orchestration Central service and select **Stop**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Stop**.

Linux:

- a. On the server that hosts Operations Orchestration, run the following command:

```
<HPE00installation>/central/bin/central stop
```

For example, `/usr/local/hpe/csa/00/central/bin/central stop`
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: `<HPE00installation>/ras/bin/ras stop`.

For example, `/usr/local/hpe/csa/00/ras/bin/ras stop`

2. Clear the Operations Orchestration Central cache by deleting the following folder:

```
<HPE00installation>/central/var/cache
```

For example,

Windows: C:\Program Files\HPE\HPE Operations Orchestration\central\var\cache

Linux: /usr/local/hpe/csa/oo/central/var/cache

3. If RAS is installed, clear the RAS artifact cache by deleting the following folder (on all RAS systems, including localhost):

`<HPE00installation>/ras/var/cache`

For example,

Windows: C:\Program Files\HPE\HPE Operations Orchestration\ras\var\cache

Linux: /usr/local/hpe/csa/oo/ras/var/cache

4. Run the following SQL command against the Operations Orchestration database:

```
DELETE from OO_ARTIFACTS where NAME =  
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.pom' or NAME =  
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.jar'
```

5. Start the Operations Orchestration services:

Windows:

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click the HPE Operations Orchestration Central service and select **Start**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Start**.

Linux:

- a. On the server that hosts Operations Orchestration, run the following command:
`<HPE00installation>/central/bin/central start`
For example, `/usr/local/hpe/csa/oo/central/bin/central start`
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: `<HPE00installation>/ras/bin/ras start`.

For example, `/usr/local/hpe/csa/00/ras/bin/ras start`

6. Redeploy the `oo10-sm-cp-1.0.3.jar` base content pack:
 - a. Log in to Operations Orchestration Central and click **Content Management**.
 - b. Click the **Content Packs** tab.
 - c. Click the **Deploy New Content** icon.
 - d. In the Deploy New Content dialog, in the upper left corner, click the **+** (Add files for deployment) icon.
 - e. Navigate to the `CSA_HOME/oo/ooContentPack` directory and select **oo10-sm-cp-1.0.3.jar**.
 - f. Click **Deploy**.

The deployment may take a few minutes and the dialog will show a progress bar.

- g. Click **Close**.

Configure HP Single Sign-On between CSA and Operations Orchestration

If HP Single Sign-On (HP SSO) was enabled during installation of CSA, SSO can be configured between CSA and Operations Orchestration. Configuring HP SSO allows you to launch Operations Orchestration from the Cloud Service Management Console without having to log in to Operations Orchestration.

CSA provides a login user (`admin`) and password (`cloud`) and, earlier in this guide, you configured an internal user for Operations Orchestration with the same user name and password. When Single Sign-On is configured between CSA and Operations Orchestration, this user can be used for single sign-on. That is, if you are logged in to CSA as the `admin` user, you can launch Operations Orchestration from the Cloud Service Management Console and not have to log in to Operations Orchestration.

You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and the embedded Operations Orchestration to use the same LDAP source or, if CSA and the embedded Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the embedded Operations Orchestration user must be assigned any role that allows flows to be viewed.

Note: To use HP SSO between CSA and Operations Orchestration, the systems on which CSA and Operations Orchestration are installed must be in the same domain.

Configure and Enable HP Single Sign-On

To configure and enable HP SSO on Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > SSO**.
4. Select the **Enable** checkbox.
5. Enter the **InitString**. The `initString` setting for CSA and Operations Orchestration must be configured to the same value. In CSA, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/idm.war/WEB-INF/hpssoConfiguration.xml` file. The `initString` value represents a secret key and should be treated as such in your environment (this string is used to encrypt and decrypt the `LWSSO_COOKIE_KEY` cookie that is used to authenticate the user for single sign-on).
6. Enter the **Domain**. This is the domain name of the network of the servers on which CSA and Operations Orchestration are installed.
7. Click **Save**.

Configure LDAP Users for Single Sign-On

To enable single sign-on for LDAP users, you must either configure CSA and Operations Orchestration to use the same LDAP source or, if CSA and Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the Operations Orchestration user must be assigned any role that allows flows to be viewed.

To configure LDAP for Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > LDAP**.
4. Enter the information to configure LDAP.
5. Click **Save**.

Obscure Passwords in Operations Orchestration Flows (Optional)

Some Operations Orchestration flows included with CSA may show passwords in clear text when viewed in Operations Orchestration Central. You can obscure these passwords by modifying the flow in Operations Orchestration Studio.

Note: You must have Operations Orchestration Studio installed. Operations Orchestration Studio is supported on Windows platforms only and is not part of the embedded Operations Orchestration that is included with CSA. See the Operations Orchestration documentation, such as the *Operations Orchestration System Requirements*, for more information about Operations Orchestration Studio.

To obscure passwords in Operations Orchestration flows:

1. Open Operations Orchestration Studio.
2. Locate the flow to update.
3. Right-click on the flow and select **References > What uses this?**.

A list of flows that use the flow is displayed (that is, the flow to update is a subflow of the flows displayed in the list).

4. Select a flow from the list of flows.
5. Locate the subflow (the flow to update).
6. Right-click on the subflow and select **Properties**.
7. Located the property to obscure (such as a password), enable it, but do not assign a value to it.
8. Save the flow.
9. Repeat this procedure for every flow from the list of flows.

Configure OO Version 9.07

Only if you have upgraded from an earlier version of CSA that uses OO 9.07 for sequential designs, you can continue to use OO 9.07. For a new installation of CSA, OO 9.07 is not supported.

Complete the following tasks to configure Operations Orchestration to integrate with CSA:

- [Add a JRE to the system path](#)
- [Install CSA Flows](#)
- [Set Remote Action Services](#)
- [Configure System Accounts Settings](#)
- [Configure System Properties Settings](#)
- [Configure_General_System_Configuration_Settings_in_Operations Orchestration](#)
- [Configure a Secure Connection Between CSA and Operations Orchestration](#)
- [Obscure_Passwords_in_Operations Orchestration_Flows](#)
- [Check_RAS_Timeout_Settings \(optional\)](#)
- [Change_OO_REST_API_Timeout \(optional\)](#)
- [Operations OrchestrationRoot Content Configuration](#)

Note: In the following instructions, `CSA_HOME` is the directory in which CSA is installed and `OO_HOME` is where you installed Operations Orchestration.

Be sure all the latest patches for Operations Orchestration have been installed. See the *Cloud Service Automation System and Software Support Matrix* for more information.

Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com> (this site requires a Passport ID). Select **Dashboards > Manuals**.

Add a JRE to the System Path

The CSA flows that are imported require that a JRE be included in the system path on the system running CSA.

To add a JRE to the system path on Windows, complete the following steps:

1. Open the **Environment Variables** dialog:
 - a. Right-click **Computer** and select **Properties**.
 - b. Select **Advanced System Settings**.
 - c. Click **Environment Variables**.
2. Select the **Path** system variable.

3. Click **Edit**.
4. At the end of the value for **Variable value**, add a semicolon (;) and the following path:

If Operations Orchestration and CSA are installed on the same system:

`OO_HOME/java/bin`

or

If Operations Orchestration and CSA are installed on different systems:

`CSA_JRE_HOME/bin`
5. Click **OK** and close all windows.

To add a JRE to the system path on Linux, complete the following steps:

Open a shell and enter one of the following commands:

- If Operations Orchestration and CSA are installed on the same system:

`export PATH=$PATH:$OO_HOME/java/bin`
- If Operations Orchestration and CSA are installed on different systems:

`export PATH=$PATH:$CSA_JRE_HOME/bin`

Note: By setting the system path, all applications (that require a JRE) use the JRE that is installed with Operations Orchestration or CSA (depending on the path you configured and if it is the only path or the first path set to a JRE in the system path). If you need to run another JRE with an application, you must type in the relative path to that JRE to run it (for example, when you configure TLS).

Install CSA Flows

The flows for CSA must be installed in the Operations Orchestration Flow Library.

To install CSA flows:

1. If CSA and Operations Orchestration are running on different systems, copy the `CSA_HOME\CSAKit-4.800 Flow Content\XCSA-4_80-ContentInstaller.jar` file from the CSA system to the Operations Orchestration system (where `CSA_HOME` is the directory in which CSA is installed).

2. On the system running Operations Orchestration, open a command prompt (Windows) or shell (Linux) and change to the directory where the `CSA-4_80-ContentInstaller.jar` is located.
3. Run the following command:

Windows:

```
"OO_HOME\jre1.6\bin\java" -jar CSA-4_80-ContentInstaller.jar -centralPassword  
<OOAdminPassword>
```

Linux:

```
OO_HOME/jre1.6/bin/java -jar CSA-4_80-ContentInstaller.jar -centralPassword  
<OOAdminPassword>
```

Set Remote Action Services

1. Log in to Operations Orchestration Studio.
2. Open the **Configuration > Remote Action Services** folder.
3. Double-click **RAS_Operator_Path**.
4. Set the **URL** to:

```
https://<FQDN>:9004/RAS/services/RCAgentService
```

where `<FQDN>` is the fully qualified domain name or IP address of the Operations Orchestration host. Do not use localhost in the URL. Using localhost does not work correctly even though it appears to work when you run Operations Orchestration Studio on the same machine as the RAS.

RAS must be run on the same system as Operations Orchestration Studio. Running Operations Orchestration Studio on another machine produces errors and turns flows red with a cryptic error message about result assignments to result variables that do not exist.

Configure System Accounts Settings

1. Log in to Operations Orchestration Studio.
2. Open the **Configuration > System Accounts** folder.
3. Double-click **CSA_REST_CREDENTIALS**.
4. Verify the Credentials are set to the following values:

- **User Name:** oolnboundUser
- **Password:** cloud

where **CSA_REST_CREDENTIALS** are the credentials for CSA REST authentication.

Note: The **User Name** configured for the CSA_REST_CREDENTIALS System Account setting must match the **Property Value** configured for the CSA_OO_USER System Property setting.

Configure System Properties Settings

1. Log in to Operations Orchestration Studio.
2. Open the **Configuration > System Properties** folder.
3. Double-click **CSA_REST_URI**.
4. Set the **Property Value** to:

```
https://<csa_hostname>:8444/csa/rest
```

5. Double-click **CSA_OO_USER**.
6. Verify the **Property Value** is set to:

```
oolnboundUser
```

Note: The **Property Value** configured for the CSA_OO_USER System Property setting must match the **User Name** configured for the CSA_REST_CREDENTIALS System Account setting.

The other settings can be optionally configured. For information about the settings, refer to the *Cloud Service Automation Configuration Guide*.

Configure General System Configuration Settings in Operations Orchestration Central

1. Log in to Operations Orchestration Central.
2. Open the **Administration > System Configuration > General** tab.
3. Set the **Save history base on flags** property to true.

Configure a Secure Connection between CSA and Operations Orchestration

Export Operations Orchestration's certificate from Operations Orchestration's truststore. If Operations Orchestration and CSA are not installed on the same system, copy the certificate to the CSA system and import the certificate into CSA's truststore. TLS must be configured between CSA and Operations Orchestration.

Do the following:

1. On the system running Operations Orchestration, open a command prompt and change to the directory where Operations Orchestration is installed.
2. Run the following command:

Windows:

```
.\jre1.6\bin\keytool -export -alias pas -file C:\oo.cer  
-keystore .\Central\conf\rc_keystore -storepass bran507025
```

Linux:

```
./jre1.6/bin/keytool -export -alias pas -file /tmp/oo.cer  
-keystore ./Central/conf/rc_keystore -storepass bran507025
```

where C:\oo.cer on Windows and /tmp/oo.cer on Linux are examples of filenames and locations used to store the exported root certificate (you can choose a different filename and location).

3. If Operations Orchestration is not running on the same system as CSA, copy oo.cer from the Operations Orchestration system to the system running CSA.
4. On the system running CSA, open a command prompt.
5. Run the following command:

Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias pas -file C:\oo.cer -trustcacerts  
-keystore "<csa_jre>\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -alias pas -file /tmp/oo.cer -trustcacerts  
-keystore $CSA_JRE_HOME/lib/security/cacerts
```

where CSA_JRE_HOME is the directory in which the JRE that is used by CSA is installed..

6. When prompted for the keystore password, enter `changeit`.
7. Enter `yes` when prompted to trust the certificate.

Obscure Passwords in Operations Orchestration Flows (Optional)

Some Operations Orchestration flows included with CSA may show passwords in clear text when viewed in Operations Orchestration Central. You can obscure these passwords by modifying the flow in Operations Orchestration Studio.

Note: You must have Operations Orchestration Studio installed. Operations Orchestration Studio is supported on Windows platforms only and is not part of the embedded Operations Orchestration that is included with CSA. See the Operations Orchestration documentation, such as the *Operations Orchestration System Requirements*, for more information about Operations Orchestration Studio.

To obscure passwords in Operations Orchestration flows:

1. Open Operations Orchestration Studio.
2. Locate the flow to update.
3. Right-click on the flow and select **References > What uses this?**.

A list of flows that use the flow is displayed (that is, the flow to update is a subflow of the flows displayed in the list).

4. Select a flow from the list of flows.
5. Locate the subflow (the flow to update).
6. Right-click on the subflow and select **Properties**.
7. Located the property to obscure (such as a password), enable it, but do not assign a value to it.
8. Save the flow.
9. Repeat this procedure for every flow from the list of flows.

Check RAS Timeout Settings (Optional)

Remote Access Server (RAS) operations are subject to a default timeout limit of 20 minutes on Operations Orchestration Central. You can change the time-out setting to support operations that are likely to take more than 20 minutes to complete.

If you expect to run large deployments, change the time-out setting according to **Changing the timeout limit for RAS operations** in the *Operations Orchestration Software Administrator's Guide*. You may also refer to *Operations Orchestration User's Guide* sections **Adding a RAS override** and **Best practices for runtime environment overrides**. Both documents are available on the Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com> (this site requires a Passport ID). Select **Dashboards > Manuals**.

Change Operations Orchestration REST API Timeout (Optional)

The calls CSA makes to the Operations Orchestration REST APIs are synchronous, and Operations Orchestration will time-out the connection after one hour by default. To extend this time-out, do the following:

1. Open the following file in a text editor:

```
OO_HOME\Central\conf\Central.properties
```

2. Add the following lines:

```
# the maximum flow timeout value in milliseconds, this is equivalent to 2 hrs  
dharma.headless2.continuation.timeout=7200000
```

3. Open the following file in a text editor:

```
OO_HOME\Central\WEB-INF\applicationContext.xml
```

4. Add the following property to the `dharma.RCDefaults` section:

```
<bean id="dharma.RCDefaults"  
class="com.iconclude.dharma.util.spring.RCDefaultsSpringFactory" lazy-  
init="false" singleton="true">  
  
.....
```

```
<prop  
key="dharma.headless2.continuation.timeout">${dharma.headless2.continuation.tim  
eout}</prop>
```

5. Restart the HPE Operations Orchestration Central service.

- **Configure parameters** — Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action.

Click **Auto-Configure** to automatically map any parameter that is not mapped to a corresponding property. If a matching property does not exist, it is created.

Click the gear icon next to a parameter and select **Delete Mapping** to delete a property mapping.

Click the gear icon next to a parameter and select **Auto-Select Property for Mapping** to automatically create a property mapping. If no matching property is found, it is created.

Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the name, display name, description, and mapping type of the parameter.

Select one of the following mapping types:

Not Mapped — The parameter is not mapped to a value.

Constant Value — Type a value for the parameter.

Property — Select a property.

Token — Select a token, which is a system value that is automatically resolved internally when the parameter is read. Tokens are not available for Boolean or Integer parameters. See [Tokens](#) for a description of the tokens.

Provider Property — Select a provider property from the list. Any property defined by providers that are applicable for this resource offering are listed. A token value is displayed for t The default options selected are - **Provider Type** and **All Resource Offerings**. So, the system displays all Resource Offerings sorted by Provider Type by default.

Operations Orchestration Root Content Configuration

The Operations Orchestration content folders and actions that should be displayed in CSA are configurable. Follow the below steps to configure:

- a. Open the `csa.properties` file present in the below path in text editor:

```
C:\Program Files\HPE\CSA\jboss-as\standalone\deployments\csa.war\WEB-INF\classes
```

- b. Define the properties in the `csa.properties` file for the content folders you want to display.

For example,

```
# Action Selection Wizard content roots

csa.oo.content.root.lifecycle.action=/Library/CSA Content Pack,/Library/CSA Content

csa.oo.content.root.external.approval=/Library/CSA Content Pack,/Library/CSA Content

csa.oo.content.root.resource.pool.sync=/Library/CSA Content Pack,/Library/CSA Content
```

Points to Note:

- o While configuring the folder paths, multiple folders can be added by separating the folders using a comma (,) as shown in the example above.
- o You need not restart CSA after changing these properties as these properties are runtime reloadable.

Import Operations Orchestration Flows

Operations Orchestration flows can be executed by CSA lifecycle actions, or used to synchronize resource pools or to submit delegated approvals. If you skipped installing CSA and Operations Orchestration content during the installation process, then run the Cloud Content Capsule Installer, which automatically installs the required Operations Orchestration flows and service designs. See the *CSA/Codar Content at a Glance Guide* for details.

After CSA installation, if you wish to add additional Operations Orchestration flows, use the Operations Orchestration import functionality to import the content pack that contains the necessary flows, into the Operations Orchestration flow library.

Cloud Service Management Console

This chapter provides information for tasks needed to optionally customize the Cloud Service Management Console.

Tasks include:

- ["Customize the Cloud Service Management Console dashboard" below](#)
- ["Customize the Cloud Service Management Console Font" on page 141](#)
- ["Customize the Cloud Service Management Console Title" on page 142](#)
- ["Rename or Delete the Sample Consumer Organization" on page 143](#)
- ["Configure HTML Email Notifications" on page 145](#)
- ["Configure Security Warning Messages for Cloud Service Management Console " on page 157](#)
- ["Enable Verification of an Imported Service Design, Service Offering, or Catalog Content Archive" on page 158](#)

Customize the Cloud Service Management Console dashboard

The Cloud Service Management Console is made up of predefined menu items that launch predefined pages. You can customize the dashboard by adding new menu items, modifying existing menu items, adding launcher sub menus, or disabling existing menus.

Topics in this section include:

- ["Enable the Cloud Analytics Side-Bar Menu" on the next page](#)
- ["Configure the Cloud Optimizer Side-Bar Menu" on page 113](#)
- ["Create a Side-Bar Menu " on page 126](#)
- ["Add a Launcher Sub-Menu" on page 130](#)
- ["Modify a Side-Bar Menu" on page 133](#)
- ["Disable a Side-Bar Menu" on page 133](#)
- ["Side-Bar menu Configuration File Syntax" on page 133](#)

The Cloud Service Management Console can be customized by a user who has access to the system on which CSA is running and permissions to modify and save files in the CSA installation directory.

A disabled predefined custom menu definition, disabled sample menu definitions, and a disabled sample sub menu definition are provided in CSA as examples of how to create a side-bar menu item and sub-menu items. Examples of how to use the sample menu definitions and sub-menu definition are provided in this section.

NOTE: The predefined custom menu definition is still in the `config.json` file in the `dashboard/launcher`.

There are many examples of enabled and disabled items in the launcher configuration (`config.json`).

Enable the Cloud Analytics Side-Bar Menu

HPE IT Business Analytics automatically gathers metrics from CSA to build key performance indicators. It provides scorecards and dashboards so that Resource Supply Managers and Service Business Managers have insight into how to measure and optimize the cost, risk, quality and value of IT services and processes.

In CSA, the Resource Supply Manager, Service Business Manager, and Administrator roles have access to the Cloud Analytics side-bar menu item in the dashboard. Clicking on this menu item displays the next level of menu items (when these sub-menus enabled), which are displayed based on user roles:

- Resource Supply Managers and Administrators see the **Resource Analytics** menu item which launches a report that measures the cost and usage of resource providers in CSA.
- Service Business Managers and Administrators see the **Service Analytics** menu item which launches a report that measures the revenue, cost, and profit margin for business services in CSA.
- Service Business Managers and Administrators see the **Showback Report** menu item which launches a showback report for an organization.
- Resource Supply Managers, Service Business Managers, and Administrators see the **Advanced Reporting** menu items which launches a standalone version of HPE IT Business Analytics in a separate window and allows for more advanced operations, such as running custom reports and drilling down into additional details about information provided in the report.

Prerequisites

- You must have HPE IT Business Analytics installed and properly configured in your CSA environment.
- To ensure seamless navigation between the products, make sure that the HP Single Sign-On (HP SSO) for HPE IT Business Analytics is configured to enable logging on to CSA.
- For SSO between CSA and HPE IT Business Analytics to work successfully, both products must be installed on machines that are in the same Domain. The value of Domain and Protected Domain parameters specified for HP SSO configuration must be the same.

Note: As HP SSO is enabled by default, if you disable HP SSO manually, seamless navigation between CSA/Marketplace Portal and HPE IT Business Analytics will no longer work.

- You must configure users for both CSA and HPE IT Business Analytics for single sign-on (each user must have the same name and password). You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and HPE IT Business Analytics to use the same LDAP source or, if CSA and HPE IT Business Analytics use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the appropriate role to access the menu that launch HPE IT Business Analytics and the HPE IT Business Analytics user must be assigned a role that allows it to perform the expected functions in HPE IT Business Analytics.
- When configuring HP SSO for HPE IT Business Analytics, the `initString` setting for the Cloud Service Management Console and HPE IT Business Analytics must be configured to the same value. If you are also configuring SSO between HPE IT Business Analytics and the Marketplace Portal, the `initString` setting must be configured to the same value for the Cloud Service Management Console, the Marketplace Portal, and HPE IT Business Analytics. For the Cloud Service Management Console, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/hpsssoConfiguration.xml` file. Use this setting to configure HPE IT Business Analytics (and the Marketplace Portal).

The `initString` value represents a secret key and should be treated as such in your environment. Change the default value of the `initString` setting for the Cloud Service Management Console.

- Review the *HPE IT Business Analytics Administrator Guide* for more information.

To enable HPE IT Business Analytics menu in the Cloud Service Management Console:

1. Make a backup of the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file (where

CSA_HOME is the directory in which CSA is installed
)

2. Edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file.
3. Search for a menu called `executive_scorecard`. You can search for the second occurrence of the following text: `"id": "executive_scorecard"`.
4. Enable the first four menus by changing `"enabled": false` to `"enabled": true`, and disable the fifth menu by changing `"enabled": true` to `"enabled": false`.
5. In the data section for each of the menus, change `<<CONFIGURE_HOST_NAME>>` to match the host name of your HPE IT Business Analytics installation.
6. Save and exit the file.
7. If you are logged in to the Cloud Service Management Console, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser to view the changes. Clearing the browser cache is optional (a troubleshooting step).

Note: The changes do not require you to restart CSA.

Enable the `CSAVariables.tabMigrateToCloud` Secondary Tiles

HPE Enterprise Maps manages a centralized Business Model that links to Cloud Service Automation. To bring the highest cost savings, improved agility, and quality using CSA, the right applications and services need to be selected. The HPE Enterprise Maps Cloud Assessment process will identify the most suitable applications and services, and register them in CSA. Management can continuously evaluate the actual Cloud transformation progress to ensure that the IT infrastructure capabilities and Cloud providers are optimally used to meet the Cloud transformation goals.

Cloud Transformation Process

HPE Enterprise Maps consolidates information about the existing application portfolio and sends out surveys to appropriate stakeholders using data from tools such as Universal CMDB, PPM, APM, or spreadsheets. Based on the collected information, HPE Enterprise Maps calculates scores showing suitability of the systems from business, technical and financial points of view. The results are visualized using a set of predefined reports.

For selected services and applications, HPE Enterprise Maps creates initial service designs in CSA using information consolidated in the first phase.

The transformation feature is available in the Cloud Service Management Console, and access is provided to the Administrator, Service Designer, and Service Business Manager roles.

Click the **[[[Undefined variable CSAVariables.tabMigrateToCloud]]]** tile to see the next level of tiles (when these secondary tiles are enabled):

- **Cloud Assessment** tile – starts and manages data collection and surveys.
- **Reports** tile – displays the cloud transformation dashboard.

Prerequisites

- You must have HPE Enterprise Maps installed and properly configured in your CSA environment.
- To ensure seamless navigation between the products, make sure that the HP SSO for HPE Enterprise Maps is configured to enable logging on to CSA.
- For HP SSO between CSA and HPE Enterprise Maps to work successfully, both products have to be installed on machines that are in the same Domain. The value of Domain and Protected Domain parameters specified for HP SSO configuration must be the same.

Note: As HP SSO is enabled by default, if you disable HP SSO manually, seamless navigation between CSA/Marketplace Portal and HPE Enterprise Maps will no longer work.

- You must configure users for both CSA and HPE Enterprise Maps for single sign-on (each user must have the same name and password). You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and HPE Enterprise Maps to use the same LDAP source or, if CSA and HPE Enterprise Maps use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the appropriate role to access the tiles that launch HPE Enterprise Maps and the HPE Enterprise Maps user must be assigned a role that allows it to perform the expected functions in HPE Enterprise Maps.
- When configuring HP SSO for HPE Enterprise Maps, the `initString` setting for CSA and HPE Enterprise Maps must be configured to the same value. In CSA, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/hpsssoConfiguration.xml` file. The `initString` value represents a secret key and should be treated as such in your environment.
- Review the *HPE Enterprise Maps Installation and Configuration Guide* for more information.

To enable `CSAVariables.tabMigrateToCloud` tiles in the Cloud Service Management Console:

1. Make a backup of the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file (where `CSA_HOME` is the directory in which CSA is installed).
2. Edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file.
3. Search for a tile called `enterprise_maps`. You can search for the second occurrence of the following text: `"id": "enterprise_maps"`.
4. Under the "tiles" node, enable the first two tiles by changing `"enabled": false` to `"enabled": true`, and disable the third tile by changing `"enabled": true` to `"enabled": false`.
5. In the data section for each of the tiles, change `<<EM_HOST_NAME>>` to match the host name of your HPE Enterprise Maps installation.
6. Save and exit the file.
7. If you are logged in to the Cloud Service Management Console, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser to view the changes.

Note: The changes do not require you to restart CSA.

Configure the Cloud Optimizer Side-Bar Menu

Cloud Optimizer is a web-based analysis and visualization tool that analyzes performance trends of elements in virtualized environments. When Cloud Optimizer is integrated with CSA, Cloud Optimizer provides the ability to:

- Monitor the performance
- Analyze the capacity, usage, and forecast trends of the virtualized infrastructure
- Show health status information for the CSA service subscription

The Cloud Service Management Console provides the Cloud Optimizer menu item that launches the product web page for Cloud Optimizer. To use Cloud Optimizer you need to configure the menu item to launch the Cloud Optimizer dashboard. To see the health status information in CSA, you must have a provider configured and enabled for Cloud Optimizer.

Cloud Optimizer supports the vcentre and Helion Open Stack providers.

The following roles can access the Cloud Optimizer menu in the Management Console: Administrator, Service Designer, Service Business Manager, Resource Supply Manager, and Service Operations Manager.

Prerequisites

- You must have Cloud Optimizer installed and properly configured in your CSA environment.
- You must configure the Cloud Optimizer Health Status.
- To ensure seamless navigation between the products, make sure that the HP SSO for Cloud Optimizer is configured to enable logging on to CSA.
- For HP SSO between CSA and Cloud Optimizer to work successfully, both products have to be installed on machines that are in the same Domain. The value of Domain and Protected Domain parameters specified for HP SSO configuration must be the same.
- When configuring HP SSO for Cloud Optimizer, the `initString` setting for CSA and Cloud Optimizer must be configured to the same value. In CSA, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/hpssoConfiguration.xml` file. The `initString` value represents a secret key and should be treated as such in your environment.
- You must configure users for both CSA and Cloud Optimizer for single sign-on (each user must have the same name and password). You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and Cloud Optimizer to use the same LDAP source or, if CSA and Cloud Optimizer use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the appropriate role to access the menu items that launch Cloud Optimizer and the Cloud Optimizer user must be assigned a role that allows it to perform the expected functions in Cloud Optimizer.
- Review the Cloud Optimizer online help for more information.

Configure the Cloud Optimizer Menu in the Cloud Service Management Console

To configure the Cloud Optimizer menu in the Cloud Service Management Console, complete the following steps:

1. Make a backup of the CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json file (where CSA_HOME is the directory in which CSA is installed).
2. Edit the CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json file.
3. Search for a menu item called `cloud_optimizer`. You can search for the occurrence of the following text: `"id": "cloud_optimizer"`.
4. In the data section, change the URL from the Cloud Optimizer product web page to the Cloud Optimizer dashboard URL. For example, change `"http://www8.mydomain.com/us/en/software-solutions/vpv-server-virtualization-management/"` to `"<VPV_FQDN>:8444/PV/?CTX=CSA"` where `<VPV_FQDN>` is the fully-qualified domain name of the Cloud Optimizer installation.
5. Save and exit the file.
6. If you are logged in to the Cloud Service Management Console, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser to view the changes. Clearing the browser cache is optional (a troubleshooting step).

Note: The changes do not require you to restart CSA.

Configure the Cloud Optimizer Health Status for CSA

CSA and Cloud Optimizer integration provides health status information for the CSA service subscription provisioned on vCenter and Helion Open Stack providers.

The following two modes of communication are used between CSA and Cloud Optimizer to update the health status on CSA:

- The first mode is through the REST API where CSA calls the REST APIs provided by Cloud Optimizer to retrieve the health of the service subscription. You can refresh the health status of a service subscription by navigating through the Operations side-bar menu in the Cloud Service Management Console.

- The second mode is through the Cloud Optimizer notification, which notifies the health status change of a VM to CSA. CSA internally determines the service subscription to which the VM belongs to update the health status on the subscription.

Cloud Optimizer supports the following predefined health statuses: CRITICAL, MAJOR, MINOR, WARNING, NORMAL, and UNKNOWN. Cloud Optimizer also supports the power statuses: SUSPENDED and POWERED_OFF. Cloud Optimizer monitors each VM and determines the actual health status. Cloud Optimizer then reports the health status to CSA through either of the two modes mentioned above.

CSA is responsible for computing the overall health status of a service subscription based on each VM's health status, which is provisioned as part of the subscription. The highest severity health status of any of the VMs belonging to a subscription, is the health status shown for that subscription.

Complete the following tasks to configure the Cloud Optimizer health status for CSA:

Note: The commands used to configure the Cloud Optimizer are run on the Cloud Optimizer machine, which only supports Linux.

- ["Configure SSL" below](#)
- ["Configure Cloud Optimizer Notification-based Communication" on page 119](#)
- ["Provider Configuration Changes in CSA" on page 123](#)

Configure SSL

CSA and Cloud Optimizer communication is through the SSL protocol. You must complete the following prerequisite tasks for the integration to work.

- ["Configure REST API-based Communication to Integrate CSA and Cloud Optimizer" below](#)
- ["Configure the SSL Certificate " on the next page](#)

Configure REST API-based Communication to Integrate CSA and Cloud Optimizer

Following are the required configuration steps to integrate CSA and Cloud Optimizer. See the *Cloud Optimizer Configuration Guide* and *Cloud Optimizer* online help on the Cloud Optimizer server, for more

information.

To configure REST API-based communication to integrate CSA and Cloud Optimizer, complete the following steps on the Cloud Optimizer server:

1. Configure the data source.

You need to add the data source such as vSphere, and configure the vCenter provider details, which are required to be monitored by Cloud Optimizer.

2. Configure LDAP.

Configure the LDAP details and verify that you can log in with the configured admin user account.

3. Configure the CSA URL as follows:

- a. Configure the CSA URL address , for example: `https://10.1.2.1:8444`, and add the user name and password. Save the details.
- b. By default the "CSA Component Property" name `hostName` is mapped to the **Name** property of the **Cloud Optimizer Server Property**.
- c. Click Edit and add new properties in the **Instance Mapping Rule**.
 - i. Add a new "CSA Component Property" as `ipAddress` and select **IP Address** from the **Cloud Optimizer Server Property property** list.
 - ii. Add a new "CSA Component Property" as `instanceId` and select **System ID** from the **Cloud Optimizer Server Property** property list. (This is not required for the vCenter Data source.)

4. Apply the Cloud Optimizer license. Contact the vendor for the license.

A valid license determines the total number of VMs a Cloud Optimizer can monitor.

Configure the SSL Certificate

Configure the SSL certificate by exporting the Cloud Optimizer certificate and importing it into CSA's truststore.

Export the Cloud Optimizer Certificate

To export the Cloud Optimizer certificate, complete the following steps:

Note: Use the `/opt/OV/nonOV/jre/b/bin/` path used during installation to import or export the Cloud Optimizer certificate.

1. Use the SSH protocol to go into the Cloud Optimizer system and run the following command:

```
/opt/OV/nonOV/jre/b/bin/keytool -export -alias ovtomcatb -file <local_path>/co-  
certificate.cer -keystore /var/opt/OV/certificates/tomcat/b/tomcat.keystore
```

2. When prompted for the keystore password, enter changeit.

Import the Cloud Optimizer Certificate into CSA

To import the Cloud Optimizer certificate into CSA, complete the following steps:

Note: Import the certificate in the JRE's truststore path used during the CSA installation. For example, on Linux, the path is `/usr/hpe/csa/jre/lib/security/cacerts`, and on Windows the path is `C:\Program Files\HPE\CSA\openjre\lib\security\cacerts`.

1. Go to the `/opt/OV/nonOV/jre/b/bin/` folder.
2. Copy the exported certificate file, `co-certificate.cer`, and place it in any folder in which CSA is installed. For example, in `/tmp/co-certificate.cer`.
3. Run the following command to import the certificate:

```
keytool.exe -importcert -alias ovtomcatb -file /tmp/co-certificate.cer -  
keystore <JRE PATH used by CSA>/lib/security/cacerts
```

4. While importing the certificate into the CSA truststore, you may need to specify a different alias name than the one used in the certificate.

For example:

```
keytool.exe -importcert -alias <new_alias_name> -file /tmp/co-certificate.cer -  
keystore <JRE PATH used by CSA>/lib/security/cacerts
```

You would change the certificate alias for these reasons:

- If you used your own certificate. In this case, you also need to use your certificate password.
- If you have configured multiple Cloud Optimizers in CSA, and all or many of the Cloud Optimizer certificates have the same alias. You cannot import multiple Cloud Optimizer certificates with the same alias. Each Cloud Optimizer alias must be unique.

Run the following commands to configure multiple Cloud Optimizers in CSA so that all have unique aliases:

```
/opt/OV/nonOV/jre/b/bin/keytool -changealias -alias ovtomcatb -destalias  
ovtomcatbtwo -keystore /var/opt/OV/certificates/tomcat/b/tomcat.keystore
```

```
/opt/OV/nonOV/jre/b/bin/keytool -export -alias ovtomcatbtwo -file "/home/co-  
certificate.cer" -keystore  
"/var/opt/OV/certificates/tomcat/b/tomcat.keystore"
```

5. When prompted for the keystore password, enter the default `changeit` (unless you applied your own certificate).
6. Restart the CSA service. See ["Restart CSA" on page 170](#) for instructions.

Configure Cloud Optimizer Notification-based Communication

Cloud Optimizer uses Kafka as a message broker service to notify the registered Kafka consumer client. In this case, the consumer is CSA.

Kafka notification service can be run either in secure or non-secure mode. The non-secure mode of communication is the default mode, which works without additional configuration for CSA. However, you do need to do a basic Kafka configuration on the Cloud Optimizer server.

Complete the following tasks to configure Cloud Optimizer notification-based communication:

- ["Basic Cloud Optimizer Kafka Configuration" below](#)
- ["SSL Configuration Changes on CSA" on page 122](#)
- ["Configure the Cloud Optimizer Side-Bar Menu" on page 113](#) ["Configure the Cloud Optimizer Side-Bar Menu" on page 113](#)

Basic Cloud Optimizer Kafka Configuration

You need to configure Kafka to expose the ports and enable Cloud Optimizer to produce and consume messages through the CSA (the remote client).

To configure Kafka on Cloud Optimizer, complete the following steps:

1. Expose port 9092 as follows:

Note: 9092 is the default port. However, you must use the port that the Kafka broker is configured to run.

- a. Run the following command:

```
iptables -I INPUT -s 0/0 -p tcp --dport 9092 -j ACCEPT
```

- b. Or stop the firewall service using the following commands:

```
Service iptables stop
```

```
Service ip6tables stop
```

2. Change the Kafka server `/opt/OV/nonOV/kafka/config/server.properties` file as follows:

- a. For non-secure communication, set the listener property as PLAINTEXT

```
listeners=PLAINTEXT://<host name of CO machine>:9092
```

For example:

```
listeners=PLAINTEXT://10.2.11.195:9092
```

- b. For secure communication, set the listener property to SSL.

```
listeners=SSL://<host name of CO machine>:9092
```

For example:

```
listeners=SSL://10.2.11.195:9092
```

3. Configure the SSL parameters in the `/opt/OV/nonOV/kafka/config/server.properties` file as specified in the Cloud Optimizer's SSL Configuration Guide.

4. Restart Kafka services.

- a. Before restarting, set the environment:

```
export PATH=$PATH:/opt/OV/nonOV/kafka/bin
```

```
export PATH=$PATH:/opt/OV/nonOV/jre/b/bin
```

- b. Restart Kafka services as follows:

- i. Go to the `/opt/OV/nonOV/kafka/bin` folder.

- ii. Run the following command:

```
./kafka-server-start.sh ../config/server.properties &
```

where the (&) symbol executes the service in the background.

5. Or Alternatively you can use the following commands to enable/disable Kafka Service

To enable Kafka :

```
# /opt/OV/bin/msgbus.sh -enable
```

Example output:

```
===== Current Messagebus Configuration  
=====
```



```
HPEKafka and HPEZookeeper are disabled.
```

```
=====
Enabling Msgbus
Registering HPEKafka and HPEZookeeper
Starting HPEZookeeper and HPEKafka
=====
```

To Disable Kafka:

```
# /opt/OV/bin/msgbus.sh -disable
Example output:
Disabling Msgbus
Stopping HPEZookeeper and HPEKafka
Unregistering HPEKafka and HPEZookeeper
===== New Messagebus Configuration =====
HPEKafka and HPEZookeeper are disabled.
=====
```

To check the status:

```
# /opt/OV/bin/msgbus.sh -status
Example output:
hpekafka    HPE Kafka Service                CORE,SERVER (1364)  Running
hpezookeeper HPE Zookeeper Service            CORE,SERVER (989)   Running
```

6. Export the Cloud Optimizer certificate and import it into CSA's truststore. See the "[Configure the SSL Certificate](#)" on page 117 for instructions.

SSL Configuration Changes on Cloud Optimizer

For a secure mode communication, you need to configure the Cloud Optimizer Kafka services to be secure using the Java keystore certificate. These certificates need to be exported from the Cloud Optimizer and imported into CSA.

For SSL configuration on the Cloud Optimizer Kafka service, see the Kafka documentation at <http://docs.confluent.io/2.0.0/kafka/ssl.html> (since this link is a third-party link and could change, this link may or may not remain active).

SSL Configuration Changes on CSA

Make the following SSL configuration changes on CSA:

- Enable SSL configuration on CSA
- Enable SSL-Based Authentication

Enable SSL configuration on CSA

To enable SSL configuration on CSA, complete the following steps:

1. Export the certificate from the keystore used by the Kafka broker server:

```
/opt/OV/nonOV/jre/b/bin/keytool -export -alias <alias_name> -file /home/kafka-broker.cer -keystore <Path of the broker's server key store file>
```

2. Import the Kafka broker's certificate into CSA as follows:

- a. Copy the above exported certificate file, kafka-broker.cer, and place it in any folder in which CSA is installed. For example, in /tmp/kafka-broker.cer.

- b. Import the copied certificate file using the following command:

```
keytool.exe -importcert -alias <alias_name> -file /tmp/kafka-broker.cer -keystore <JRE_PATH_used_by_CSA>/lib/security/cacerts
```

3. Restart the CSA service. See ["Restart CSA" on page 170](#) for instructions.

Enable SSL-Based Authentication

If SSL-based authentication is enabled on Cloud Optimizer's Kafka broker, complete the following steps:

1. Export the CSA certificate.

For example: On the default CSA installation, run the following command:

Windows:

```
keytool -export -alias csa -keystore c:\Program Files\HPE\CSA\jboss-as\standalone\configuration\.keystore -file c:\temp\csa.cer
```

Linux:

```
keytool -export -alias csa -keystore /usr/hpe/csa/jboss-as/standalone/configuration/.keystore -file /tmp/csa.cer
```

2. Import the CSA certificate on to the Cloud Optimizer Kafka server as follows:

- a. Copy the above exported CSA certificate file, `csa.cer`, into any folder on Cloud Optimizer
- b. Run the following command (only Linux is supported on the Cloud Optimizer Kafka server):

```
keytool.exe -importcert -alias <csa> -file /var/temp/csa.cer -keystore <Path  
to Kafka broker's server trust store file>
```

3. Restart the Kafka service.

- a. Before restarting, set the environment:

```
export PATH=$PATH:/opt/OV/nonOV/kafka/bin
```

```
export PATH=$PATH:/opt/OV/nonOV/jre/b/bin
```

- b. Restart Kafka services as follows:

- i. Go to the `/opt/OV/nonOV/kafka/bin` folder.

- ii. Run the following command:

```
./kafka-server-start.sh ../config/server.properties &
```

where the (&) symbol executes the service in the background.

Configuration changes for the CSA properties

If you want to enable the SSL-based communication between CSA and Cloud Optimizer for Kafka notifications, make the following changes:

1. Search for the text `Cloud Optimizer integration properties` in the `csa.properties` file in CSA.
2. Below the `Cloud Optimizer integration properties` text, there are notes that explain how to enable SSL between CSA and Cloud Optimizer, such as the following:

```
# Configuration to enable SSL communication between Kafka consumer client on CSA and  
# Kafka server on Cloud Optimizer  
# Property format - <Cloud Optimizer hostname/IP Address>_ssl.enabled  
# where the hostname/IP Address should match the value configured as access point  
# of the CO provider  
# The default value is disabled and when SSL is enabled, the Kafka consumer on CSA  
# uses the truststore file 'csaTruststore' and it requires the kafka server certificate  
# to be imported into 'csaTruststore' file.
```

For example; `10.2.13.17_ssl=enable`

Provider Configuration Changes in CSA

The following sections describe how to configure Cloud Optimizer providers in CSA.

Create a Cloud Optimizer Provider

To create a Cloud Optimizer provider, complete the following steps:

1. In the **Providers** menu, select **By Type** in the left pane.
2. Select **HPE Cloud Optimizer**.
3. In the right pane, select the **Providers** tab.
4. Click the gear icon and select **Create Resource Provider**.
5. Add the required fields:

Item	Description
Display Name	The name of the Cloud Optimizer provider.
Service Access Point	Specify the Cloud Optimizer access URL for connecting to the provider. Use /PV as the suffix, which is mandatory. For example: https://10.2.13.177:8444/PV
User ID	The user ID for the Cloud Optimizer Service Access Point.
Password	The password for the Cloud Optimizer Service Access Point. Re-type the password in the Confirm Password field.
Enabled	This value determines whether the provider will be selected when provisioning a new service. The setting is either Enabled (when checked) or Disabled (when not checked). When Disabled , the provider will not be selected when provisioning new services. Disabling a provider will have no effect on existing services that are using that provider.

See the *Cloud Service Management Console Help* for more information about configuring providers.

Configure the Cloud Optimizer Provider Properties

- **CONSUMER_GROUP_ID:**

If CSA is configured in a high availability cluster mode, then create a property with the name `CONSUMER_GROUP_ID` and configure it in the Cloud Optimizer provider. Set a value that can be any string.

For example: `CSA_HA_CONFIG`

- **BOOTSTRAP_SERVERS:**

If the Cloud Optimizer's Kafka bootstrap server port is configured with a non-default port such as 9092, then you must create a property with the name `BOOTSTRAP_SERVERS` and configure it in the Cloud Optimizer provider. Set the value to `<server:port>`, where the server is the host address and port is the new port on which the Kafka broker server is running.

If there is a cluster of configured Kafka bootstrap servers, then you can optionally specify a comma-separated list of host addresses `<server:port>`.

- Change the properties on the providers:

On the vCenter provider, create a new property with the name `COURL` and set the value of the access point of the Cloud Optimizer that is configured to monitor it.

For example: `https://10.2.13.177:8444/PV`

Enable Other Predefined dashboard Tiles

CSA provides several predefined but disabled dashboard tiles. You can enable these tiles by doing the following:

1. Make a backup of the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` dashboard configuration file (where `CSA_HOME` is the directory in which CSA is installed).
2. Edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file:

`CSA_HOME` is the directory in which CSA is installed

- a. Locate the tile definition to enable.
 - b. Set the `enabled` attribute to **true**.
 - c. Save and exit the file.
3. Log in to the Cloud Service Management Console to view the tile. If you are already logged in, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser.

To modify the tile, see ["Create a Side-Bar Menu " on the next page](#) for more information.

Create a Side-Bar Menu

The Cloud Service Management Console is made up of predefined menus that launch predefined pages. You can customize the dashboard by creating side-bar menus in the dashboard that launch custom pages.

Menus are defined in a configuration file and the menu definitions determine what is displayed in the Cloud Service Management Console. The default dashboard configuration file defines a primary dashboard that consists of enabled menu items and disabled menu items, a sub-menu (launched from the Design menu), and a disabled sample launcher sub-menu. Information about menu attributes and values defined in the configuration file is included in the steps below. See ["Add a Launcher Sub-Menu" on page 130](#) for more information about how to add a secondary dashboard.

To create a Cloud Service Management Console menu item, do the following:

1. Make a backup of the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` dashboard configuration file (where `CSA_HOME` is the directory in which CSA is installed).
2. Edit the `config.json` dashboard configuration file.

In the configuration file, the menu defined for a dashboard are configured sequentially. That is, the first menu definition configured in a dashboard definition is the first menu item displayed in the dashboard. The second menu definition is the second menu item displayed. For example, in the default dashboard configuration file, the first menu item definition configured in the primary dashboard is the Organizations menu. The Organizations menu is the first menu displayed in the Cloud Service Management Console dashboard. The second menu definition is the Resources menu and it is the second menu displayed in the Cloud Service Management Console dashboard.

Determine where you want the menu item to appear in the dashboard and find the location in the configuration file. For example, if you want a menu to appear between the Organizations and Resources menu items in the dashboard, find the location between the Organizations and Resources menu definitions. If you want the menu to appear as the last menu item, find the end of the last enabled menu definition.

- a. Copy the sample menu definition, whose `id` attribute is set to `blanktile`, and place it in the selected location. The following is an example menu definition (multiple menu definitions are separated by a comma):

```
{
  "id": "<tile_id>",
  "name": "<tile_name>",
  "description": "<tile_description>",
  "enabled": <true_or_false>,
  "style": "<tile_style>",
  "target": "<tile_target>",
  "data": "<tile_data>",
  "helptopic": "<tile_helptopic>",
  "roles": ["<role_1>", "<role_2>", ... , "<role_n>"],
  "permissions": ["<permission_1>", "<permission_2>" ..., "<permission_n>"]
}
```

- b. Update the attribute values in the menu definition as described in the table.

Attribute	Description
id	A unique identifier of the menu in this dashboard among all menus defined for this dashboard.
name	<p>The name of the attribute in the <code>messages.properties</code> or <code>messages_<locale>.properties</code> file that defines the name of the menu that is displayed on the dashboard (where <code><locale></code> identifies the language to which the title has been translated, for example, <code>en</code> for English or <code>ja</code> for Japanese).</p> <p>The file may appear in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/custom</code> or <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/messages/dashboard</code> directory. If the file exists in both directories, the value defined in <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/custom</code> takes precedence.</p>
description	<p>The name of the attribute in the <code>messages.properties</code> or <code>messages_<locale>.properties</code> file that defines the description of the menu that is displayed on the dashboard (where <code><locale></code> identifies the language to which the title has been translated, for example, <code>en</code> for English or <code>ja</code> for Japanese).</p> <p>The file may appear in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/custom</code> or <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/messages/dashboard</code> directory. If the file exists in both directories, the value defined in <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/custom</code> takes precedence.</p>

Attribute	Description
enabled	Enable or disable the menu in the dashboard. If set to true , the menu is displayed in the dashboard. If set to false , the menu is not displayed in the dashboard.
style	<p>The name of the attribute in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/css/base.css</code> file that defines the color of the menu's header that is displayed on the dashboard.</p> <p>If you are creating an assistance menu (that is, you set <code>target</code> to assistance), you must set this attribute to a pre-defined style named assistance.</p>
target	<p>The type of page launched when the menu is selected. Values include:</p> <ul style="list-style-type: none"> • iframe - An iframe or page is launched within the same dashboard or page. • page - A new page is launched outside of the dashboard or page. • dashboard - A sub-dashboard is launched within the same dashboard or page. • assistance - If the <code>data</code> attribute is defined, a new page is launched outside of the dashboard or page. If the <code>data</code> attribute is not defined, no page is launched and the menu simply contains content defined by the <code>description</code> attribute. The <code>style</code> attribute must be set to assistance.
data	<p>What is launched, based on the type of <code>target</code>.</p> <p>If iframe or page is the type of <code>target</code> selected, enter a URL or relative path (relative to the location of this file, <code>CSA_HOME/jboss-as/standalone/deployments/</code>) and filename of a Java server page to display. For example, enter <code>http://www.mydomain.com/csa/administration/index.jsp</code>.</p> <p>If dashboard is the type of <code>target</code> selected, enter the unique dashboard <code>id</code> attribute of the dashboard to display. For example, the Designs menu of the main dashboard launches a sub- or secondary dashboard. The <code>id</code> of the secondary dashboard is designs therefore you would set the value of this attribute to designs.</p> <p>If assistance is the type of <code>target</code> selected and if you enter a value for this attribute, a Learn More link is displayed in the assistance menu. Clicking the Learn More link launches a page with the content defined by this attribute. Enter a URL or relative path (relative to the location of this file, <code>CSA_HOME/jboss-as/standalone/deployments/</code>) and filename of a Java server page to display. For example, enter <code>http://www.mydomain.com/csa/administration/index.jsp</code>.</p>
helptopic	If the type of <code>target</code> selected is iframe , this is the name of the help topic that is displayed when the Assistance icon on the page is selected. If the type of

Attribute	Description
	target selected is page , or dashboard , or assistance , this attribute is ignored.
roles	<p>The role required by the user in order for the menu to display in the dashboard. One or more roles may be entered. However, only one role must match the user role in order for the user to see the menu. Roles must be enclosed in quotation marks and, if more than one role is entered, separated by a comma (for example, "CSA_ADMIN", "RESOURCE_SUPPLY_MANAGER"). If no roles are specified, the menu can be seen by all users.</p> <p>Values include:</p> <ul style="list-style-type: none"> • CONSUMER_SERVICE_ADMINISTRATOR - The Consumer Service Administrator configures and manages consumer organizations. • CSA_ADMIN - The Administrator has access to all functionality in the Cloud Service Management Console. • RESOURCE_SUPPLY_MANAGER - The Resource Supply Manager creates and manages cloud resources, such as resource providers and resource pools. • SERVICE_BUSINESS_MANAGER - The Service Business Manager creates and manages service offerings and service catalogs. • SERVICE_DESIGNER - The Service Designer designs, implements, and maintains service designs (also referred to as blueprints), component palettes, component types, component templates, and resource offerings. • SERVICE_OPERATIONS_MANAGER - The Service Operations Manager views and manages subscriptions and service instances. <p>See the "Role Descriptions" help topic in the Cloud Service Management Console for more information about these roles (navigate to Organizations > Access Control > Role Descriptions in the online help).</p>
permission	The user permissions required by the user in order for the menu items to display in the dashboard. If both permissions as well as roles are defined, the user must have one of the matching entries to be considered valid.

- c. Save and exit the file.
3. Log in to the Cloud Service Management Console to view the menu. If you are already logged in, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser. Clearing the browser cache is optional (a troubleshooting step).

Add a Launcher Sub-Menu

Menus in the Cloud Service Management Console can be configured to launch a sub-menu. For example, in the default configuration of the Cloud Service Management Console, the Design menu launches another dashboard from which you can select a designer to use. This menu is configured with the `target` attribute set to **dashboard** and the `data` attribute set to the `id` of the sub-menu (**designs**). A sample sub-menu, whose `id` attribute is set to `providerpanel`, is provided.

After a menu item in the main dashboard is configured to launch a secondary dashboard, a secondary dashboard definition must be added to the dashboard configuration file. For example, in the default configuration of the Cloud Service Management Console, a sub-menu with an `id` of **designs** is defined. Information about dashboard attributes and values defined in the configuration file is included in the steps below.

To add a sub-menu, do the following:

1. Make a backup of the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` dashboard configuration file (where `CSA_HOME` is the directory in which CSA is installed).
2. Edit the `config.json` file.
 - a. Determine where you want the sub-menu (the menu that launches the launcher sub-menu) to appear in the dashboard and find the location in the configuration file. For example, if you want the sub-menu to appear between the Organizations and Resources menu in the dashboard, find the location between the Organizations and Resources menu definitions. If you want the menu to appear as the last menu, find the end of the last enabled menu definition.

Copy the sample sub-menu definition, whose `id` attribute is set to `providerpanel` and `target` attribute is set to `dashboard`, and place it in the selected location.

Update the content of the sub-menu (see ["Create a Side-Bar Menu"](#) on page 126 for more information about updating the content).

- b. In the configuration file, sub-menus are defined after the main dashboard. Locate where the main or any secondary dashboard definition ends, and add a sub-menu definition within the global dashboard definition. For example, in the default dashboard configuration file, you could add another sub-menu after the predefined **designs** sub-menu.

Copy the sample sub-menu definition, whose `id` attribute is set to `providerpanel` and `type` attribute is set to `secondary`, and place it in the selected location. The following is an example sub-menu definition (multiple dashboard definitions are separated by a comma):

```
{
  "id": "<dashboard_id>",
  "name": "<dashboard_name>",
  "style": "<dashboard_style>",
  "type": "<dashboard_type>",
  "helptopic": "<dashboard_helptopic>",
  "roles": ["<role_1>", "<role_2>", ... , "<role_n>"],
  "permissions": ["<permission_1>", "<permission_2>" ..., "<permission_n>"],
  "tiles": [ { ... } ]
}
```

- c. Update the attribute values in the dashboard definition as described in the table. See ["Create a Side-Bar Menu"](#) on page 126 for more information about menu attributes.

Attribute	Description
id	A unique identifier of the dashboard among all defined dashboards.
name	The name of the attribute in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/messages/dashboard/messages.properties</code> file that defines the name displayed in the dashboard. If this is the primary dashboard, the name is displayed above the menus. If this is a secondary dashboard, the name is the label that is displayed next to the left-facing arrow icon or back button in the header.
style	The name of the attribute in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/css/base.css</code> file that defines the color of the secondary dashboard's back button. For the primary dashboard, leave this value empty.
type	The type of dashboard. Values include: <ul style="list-style-type: none"> • primary - The dashboard that is displayed after launching CSA and successfully logging into the Cloud Service Management Console. This dashboard does not contain a back button. Only one primary dashboard can be defined. • secondary - A sub-dashboard that is launched from a dashboard and contains a back button. Zero, one, or multiple secondary dashboards can be defined.
helptopi	The name of the help topic that is displayed when the Assistance icon on the

Attribute	Description
c	page is selected.
roles	<p>The role required by the user in order for the dashboard to display. One or more roles may be entered. However, only one role must match the user role in order for the user to see the menu. Roles must be enclosed in quotation marks and, if more than one role is entered, separated by a comma (for example, "CSA_ADMIN", "RESOURCE_SUPPLY_MANAGER"). If no roles are specified, the menu can be seen by all users.</p> <p>Values include:</p> <ul style="list-style-type: none"> • CONSUMER_SERVICE_ADMINISTRATOR - The Consumer Service Administrator configures and manages consumer organizations. • CSA_ADMIN - The Administrator has access to all functionality in the Cloud Service Management Console. • RESOURCE_SUPPLY_MANAGER - The Resource Supply Manager creates and manages cloud resources, such as resource providers and resource pools. • SERVICE_BUSINESS_MANAGER - The Service Business Manager creates and manages service offerings and service catalogs. • SERVICE_DESIGNER - The Service Designer designs, implements, and maintains service designs (also referred to as blueprints), component palettes, component types, component templates, and resource offerings. • SERVICE_OPERATIONS_MANAGER - The Service Operations Manager views and manages subscriptions and service instances. <p>See the "Role Descriptions" help topic in the Cloud Service Management Console for more information about these roles (navigate to Organizations > Access Control > Role Descriptions in the online help).</p>
permissions	The user permissions required by the user in order for the menu items to display in the dashboard. If both permissions as well as roles are defined, the user must have one of the matching entries to be considered valid.
tiles	Tile definition. At least one tile must be configured. See " Create a Side-Bar Menu " on page 126 for more information about tile attributes.

- d. Save and exit the file.
3. Log in to the Cloud Service Management Console to view the dashboard. If you are already logged in, clear the browser cache (see "[Clear the web browser cache](#)" on page 172 for information about how to clear the web browser cache) and refresh the browser. Clearing the browser cache is optional (a troubleshooting step).

Modify a Side-Bar Menu

To modify an existing side-bar menu, edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file (where `CSA_HOME` is the directory in which CSA is installed):

1. Locate the menu definition that you want to modify.
2. Update one or more attributes. For a description of the attributes, refer to ["Create a Side-Bar Menu" on page 126](#).
3. Save and exit the file.

Disable a Side-Bar Menu

To disable a side-bar menu, edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json` file (where `CSA_HOME` is the directory in which CSA is installed):

1. Locate the menu definition that you want to disable.
2. Set the `enabled` attribute to **false**.
3. Save and exit the file.

Side-Bar menu Configuration File Syntax

The following is an example of a dashboard configuration file configured with only one launcher sub-menu that has one generic menu and an assistance menu defined.

```
{
  "dashboards": [
    {
      "id": "<primary_id>",
      "name": "<primary_name>",
      "style": "",
      "type": "primary",
      "helptopic": "<primary_helptopic>",
```

```
    "roles": ["CONSUMER_SERVICE_ADMINISTRATOR", "SERVICE_BUSINESS_MANAGER",
"SERVICE_DESIGNER", "CSA_ADMIN", "RESOURCE_SUPPLY_MANAGER", "SERVICE_OPERATIONS_
MANAGER"],
    "tiles": [
      {
        "id": "<tile_id_1>",
        "name": "<tile_name>",
        "description": "<tile_description>",
        "enabled": <true_or_false>,
        "style": "<tile_style>",
        "target": "<tile_target>",
        "data": "<tile_data>",
        "helptopic": "<tile_helptopic>",
        "roles": ["<role_1>", "<role_2>", ... , "<role_n>"]
      },
      .
      .
      .
      {
        "id": "<tile_id_n>",
        "name": "<tile_name>",
        "description": "<tile_description>",
        "enabled": <true_or_false>,
        "style": "<tile_style>",
        "target": "<tile_target>",
        "data": "<tile_data>",
        "helptopic": "<tile_helptopic>",
        "roles": ["<role_1>", "<role_2>", ... , "<role_n>"]
      }
    ]
  }, {
    "id": "<secondary_id>",
    "name": "<secondary_name>",
    "style": "<secondary_style>",
    "type": "secondary",
    "helptopic": "<secondary_helptopic>",
    "roles": ["<role_1>", "<role_2>", ... , "<role_n>"],
    "tiles": [
      {
        "id": "<tile_id>",
        "name": "<tile_name>",
```

```
        "description": "<tile_description>",
        "enabled": <true_or_false>,
        "style": "<tile_style>",
        "target": "<tile_target>",
        "data": "<tile_data>",
        "helptopic": "<tile_helptopic>",
        "roles": ["<role_1>", "<role_2>", ... , "<role_n>"]
    }, {
        "id": "<assistance_tile_id>",
        "name": "<assistance_tile_name>",
        "description": "<assistance_tile_description>",
        "enabled": <true_or_false>,
        "style": "assistance",
        "target": "assistance",
        "data": "<optional_Learn_More_Link>",
        "helptopic": "<value_is_ignored>",
        "roles": ["<role_1>", "<role_2>", ... , "<role_n>"]
    }
]
}
]
```

Dashboard Widget Configuration

The CSA Service Management Console dashboard provides twelve out of the box widgets.

The widgets and dashboard differ to those available in the Marketplace Portal configuration and are more suited for the administrators versus consumers.

Similar to the side-bar menu configuration (see ["Side-Bar menu Configuration File Syntax" on page 133](#)), the Service Management Console can be configured by the CSA Administrator.

Configuration Steps:

1. Open `CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboards/dashboardconfig` folder, where the configuration files that define the dashboard layout are located.

2. After installation, check for `dashboardconfig` folder to have the following 4 files:
 - **dashboard-backup.zip** - This is a zip file consisting backup of the last sync of the dashboard file system config to the database.
 - **dashboardconfig.done** - This is a trigger file that indicates the last sync is in a done state.
 - **dashboards** - The list of available dashboards.

Within the **dashboards** folder are two files:

csa.json - The CSA Dashboard

codar.json - The CODAR dashboard

- **tiles** - The list of tiles (widgets) available.
3. The following is just an example for a dashboard widget configuration. Refer to one of the file system for a detailed information.

```
{
  "id": "csa",
  "name": "csa_default_dashboard",
  "displayName": "CSA Dashboard",
  "enabled": true,
  "backgroundUri": "/csa/html-lib/images/common/dashboard/dashboard-solid-background-1.jpg",
  "restrictions": {
    "roles": [],
    "permissions": [],
    "licenseFeatures": [20795],
    "productFeatures": []
  },
  "tiles": [
    {
      "id": "1", [
        "tileId": "primaryid_csa",
        "selectedSize": "2x2",
        "row": 0,
```



```

"col": 0
},
"id": "2",
"tileId": "organization_subscription_summary",
"selectedSize": "5x2",
"row": 2,
"col": 5
}]
}

```

Following table provides description of each attribute in the above dashboard configuration:

Attribute	Description
id	Unique ID of the dashboard.
name	Internal name for the dashboard
displayName	Display name if no translation is available.
enabled	If the dashboard is enabled and available to users - set to true, else set to false.
backgroundUri	Optional path to a background image for this dashboard, can be a relative path or full URL.
restrictions	List of criteria that must be met for this dashboard to be available.
roles	List of roles required to access this dashboard. Similar to sidebar configuration. For example: SERVICE_OPERATIONS_MANAGER
permissions	List of permissions required to access this dashboard. Similar to sidebar configuration. For example: SERVICE_BLUEPRINT_WRITE

licenseFeatures	License ID required for this dashboard to be available. 20795: CSA, 20797: CODAR
productFeatures	Product features that must be available for this dashboard. For example: MONITORING, COMPLIANCE
tiles	List of tiles/widgets to include in this dashboard.
tileid	TileID reference to the widget to be used.
selectedSize	Default size (2x2) - Default tile/widget size for this tile, must match a size from the tile/widget definition or the first one will be automatically selected instead.
row	"0" is the default starting position
column	"0" is the default starting position

The first tile in the list is referring to the `primaryid_csa` widget. Within the **tiles** folder, `primaryid_csa.json` file contains the widget configuration.

```
{
  "id": "primaryid_csa",
  "name": "primaryid_csa",
  "displayName": "CSA Product tour",
  "template": "primaryid_csa.html",
  "enabled": true,
  "selectedSize": "2x2",
  "supportedSizes": ["2x2", "3x2", "3x3", "4x3", "4x4"],
  "interaction":
  {
    "enabled": true,
    "uri": ""
  },
}
```

```
"restrictions":
{
"licenseFeatures": [20795],
"productFeatures": ["MONITORING", "COMPLIANCE"],
"roles": ["<role_1>", "<role_2>", ... , "<role_n>"],
"permissions": ["<permission_1>", "<permission_2>", ... , "<permission_n>"]
}
}
```

Following table provides description of each attribute in the above tile/widget configuration:

Attribute	Description
id	Tile/widget ID. This value is referred by tileid in the dashboard configuration.
name	Internal name for this tile/widget
displayName	Display name if no translation is available.
template	Co-located html template to use for this widget. This file must exist in the same folder as the json file.
enabled	If the tile/widget is enabled - set to true, else set to false.
selectedSize	Default size (2x2) of tile if no size is configured in the dashboard json file.
supportedSizes	Available sizes for this tile.
interaction	Interaction configuration
enabled	true/false True- user can interact with the contents of the tile. False- a transparent overlay is added to avoid the user being able to click into a tile.
uri	Optional URI to launch when a user

	clicks a tile/widget. When this URI is set, a clickable overlay is added to the tile overriding the native interaction within a tile/widget.
restrictions	List of criteria that must be met for this widget to be available, same settings as available in the dashboard.
roles	List of roles required to access this dashboard. Similar to sidebar configuration. For example: SERVICE_OPERATIONS_MANAGER
permissions	List of permissions required to access this dashboard. Similar to sidebar configuration. For example: SERVICE_BLUEPRINT_WRITE
licenseFeatures	License ID required for this dashboard to be available. 20795: CSA, 20797: CODAR
productFeatures	Product features that must be available for this dashboard. For example: MONITORING, COMPLIANCE

4. Within the `primaryid_csa.json` is a reference to `primaryid_csa.html` which contains the portion of this widget that will be rendered to the user in the CSA Service Management Console.

NOTE: HTML content (`primaryid_csa.html`) will be added to the dashboard widget once it is rendered for the user.

When using **iframes** and **https**, the SSL Certificate must be trusted by the user's browser or otherwise the widget will fail to render.

Add a Custom Widget

To add a custom widget, get the HTTP/S location of a URL you want to iFrame into the dashboard.

Other than the `primaryid_csa` widget, the other widgets in CSA are using internally developed code to render the graphical data. To develop a similar graphical widget outside of CSA and to present it to the users by an **iframe** widget, follow these steps:

1. Create a new widget under the **tiles** folder.
2. Create a new template under the **tiles** folder for the widget (use the `primaryid_csa.html/json` as a sample for this template creation).
3. Add a new entry in the `csa.json` file under the **dashboards** folder for the new widget.

NOTE: Ensure that the **id** value of the newly added widget is unique within the dashboard list.

4. After the `.json` configuration, synchronize the changes into the database. This sync must be done at the time the `json` or `templates` have changed.

Perform the following steps to re-synchronize the dashboard configuration:

- a. Update/add/remove dashboard configuration files in the directory:
`CSA_HOME\jboss-as\standalone\deployments\csa.war\dashboard\dashboardconfig`
- b. Rename `dashboardconfig.done` to `dashboardconfig.load`
- c. Restart CSA on the system you have made the filesystem changes. Restart is executed for following reasons:
 - i. After the CSA restart, `dashboardconfig.load` is renamed to `dashboardconfig.done`
 - ii. To ensure in an HA environment, only one CSA is syncing the dashboard changes back into the database.

After the above steps are performed, a new widget will be available to the users who meet the restriction criteria specified.

Customize the Cloud Service Management Console Font

The font used by the Cloud Service Management Console can be customized. You can change the font if you are a user who has access to the system on which CSA is running. To change the font, on the system running CSA, do the following:

1. Open the `CSA_HOME/CSA_HOME/standalone/deployments/csa.war/custom/custom.css` file in a text editor (where `CSA_HOME` is the directory in which CSA is installed).

2. At the end of the file, add the following:

```
html, body {  
font-family: <font_name>;  
}
```

where `<font_name>` is the font used by the Cloud Service Management Console.

For example, to change the font to Arial, add the following to the file:

```
html, body {  
font-family: Arial;  
}
```

3. Save and exit the file.
4. Log in to the Cloud Service Management Console to view the changes. If you are already logged in, clear the browser cache (see ["Clear the web browser cache"](#) on page 172 for information about how to clear the web browser cache) and refresh the browser.

Customize the Cloud Service Management Console Title

The Cloud Service Management Console title appears at the top of the Cloud Service Management Console next to the HPE logo. By default, the title is "HPE Cloud Service Automation."

You can change the title if you are a user who has access to the system on which CSA is running. To change the title, on the system running CSA, do the following:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/custom/messages.properties` file in a text editor (where `CSA_HOME` is the directory in which CSA is installed).
2. Add the following attribute and value:

```
csa_title=<title>
```

where *<title>* is the title that displays at the top of the Cloud Service Management Console.

For example, to change the title to "HPE Cloud," add the following to the file:

```
csa_title=HPE Cloud
```

Note: You cannot change the HPE logo.

If you are translating the title, create a file named `messages_<locale>.properties` instead (where *<locale>* identifies the language to which the title has been translated, for example, `en` for English or `ja` for Japanese).

3. Save and exit the file.
4. Log in to the Cloud Service Management Console to view the title. If you are already logged in, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser.

Rename or Delete the Sample Consumer Organization

The sample consumer organization can be used by the `sample consumer` user to experiment with the Marketplace Portal. Customize the sample consumer organization by renaming it. Delete this sample consumer organization (and disable the `sample consumer` user) if you no longer are using it or if you are moving the application to production.

To rename the sample consumer organization:

1. Log in to the Cloud Service Management Console and do the following:
 - a. Click the **Administration** tab.
 - b. Select the **CSA Consumer** organization.
 - c. In the navigation frame, select **General Information**.
 - d. Update the **Organization Display Name**.
 - e. Click **Save**.
 - f. Look for and remember the Organization Identifier assigned to this organization. This identifier is used to define the default organization accessed by the Marketplace Portal and is assigned the sample users who access the organization.

2. Define the default organization accessed by the Marketplace Portal (the organization that is accessed by the Marketplace Portal when the Marketplace Portal is launched from a URL that does not specify the organization). Edit the `CSA_HOME/portal/conf/mpp.json` file and update the `defaultOrganizationName` attribute's value to the Organization Identifier (where the Organization Identifier is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Organizations** menu of the Cloud Service Management Console)).
3. Assign the sample users (consumer and consumerAdmin) who access the organization. Edit the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/seededorgs.properties` file and replace the existing sample Organization Identifier (for example, CONSUMER) with the one that was assigned in step 1.

For more information about the sample users defined in the `consumer-users.properties` file, see ["Change CSA Built-In User Accounts" on page 205](#).

To delete the sample consumer organization and disable the sample consumer user:

1. Log in to the Cloud Service Management Console and delete the sample consumer organization in the **General Information** page of the **Administration** area.

Note: To delete an organization, it must not have any active catalogs.

2. Edit the `CSA_HOME/portal/conf/mpp.json` file. Update the `defaultOrganizationName` attribute's value if it is set to CONSUMER. Set the value to an existing consumer organization's Organization Identifier where the Organization Identifier is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Organizations** menu of the Cloud Service Management Console). The `defaultOrganizationName` attribute defines the organization that is accessed by the Marketplace Portal when the Marketplace Portal is launched from a URL that does not specify the organization.
3. Edit the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/consumer-users.properties` file. Update the Consumer property to disable this user account. For example, set Consumer to the following encrypted value: `cloud,SERVICE_CONSUMER,ROLE_REST,disabled`.

See ["Encrypt a password" on page 172](#) for instructions about how to encrypt this value.

Configure HTML Email Notifications

CSA provides default email notification templates that can be configured to send custom HTML email notifications, instead of the existing text email notifications.

This chapter provides the following information:

- ["Configure the Notification Properties" below](#)
- ["Configure the Default Notification Templates" on the next page](#)
- ["Customize the Default Notification Templates" on page 153](#)
- ["Configure HTML Notification for Pause on Failure" on page 151](#)
- ["HTML Template Configuration/Troubleshooting Notes" on page 156](#)

Configure the Notification Properties

HTML notifications are enabled by default in the `csa.properties` file. You can configure the properties to change the defaults, if you wish.

To configure the notification properties, complete the following steps:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file in a text editor.
2. If you wish, you can change the default values for the `csa.notification.type` and/or `csa.notification.cacheTemplates` properties, as follows:

Property	Description
<code>csa.notification.type</code>	<p>Defines the type of email notification: <code>html</code> or <code>plain text</code>.</p> <ul style="list-style-type: none">◦ <code>html</code> enables custom HTML notifications.◦ <code>text</code> enables the legacy text-based notification. <p>Note: If email templates are defined, but the value is set to <code>text</code>, the emails will be sent as plain text.</p> <p>Default: <code>html</code></p>
<code>csa.notification.cacheTemplates</code>	Once an email template is used to send an email, it is

Property	Description
	<p>cached by default. Caching the notification templates improves the I/O performance while sending the notifications. If any notification template used by CSA is changed, then the changes will not be seen in later notifications unless the CSA service is restarted.</p> <p>The value of <code>csa.notification.cacheTemplates</code> may be set to <code>false</code> during development of custom notifications so that a service restart is not required every time a notification template is changed.</p> <p>Default: true</p>

3. Save any changes and exit.
4. Restart CSA. See ["Restart CSA" on page 170](#) for instructions.

Configure the Default Notification Templates

The default email HTML templates are common to all organizations handled by the CSA instance. CSA automatically looks for these templates in the `csa.war/WEB-INF/classes/notifications` directory to send out the notification emails. You can configure the notification templates, but the default template names and the location of the templates must remain the same as shown in ["Default HTML Templates" on the next page](#).

The HTML email notification templates can be configured for the following types of notifications:

- **Subscription-based notifications:** such as new services, cancel services, expired services, about to expire services and new services that failed during provisioning.
- **Approvals-related notifications:** such as approvals required for new services, service modification, and cancellation of services. Approve or reject notifications can also be sent as HTML emails.
- **New service requests notifications:** you can create email templates on a per-design basis.

See ["HTML Template Configuration/Troubleshooting Notes" on page 156](#) for configuration requirements and troubleshooting HTML email notification errors.

Default HTML Templates

CSA includes the following default notification templates in the `csa.war/WEB-INF/classes/notifications` directory that are shipped with CSA. These templates are Apache Velocity templates. Each one can be configured using the appropriate notification tokens described in ["Notification Tokens" on the next page](#)

Caution: Be sure to back up the original templates before you configure them.

- Subscription-based notification templates:
 - `ORDER.html` — Used to send an email notification to the subscriber that the service request is now active.
 - `CANCEL.html` — Used to send an email to the subscriber that the subscription has been canceled.
 - `EXPIRY.html` — Used to send an email to the subscriber that the subscription has expired. The expiration date is specified in this template.
 - `TO_EXPIRE.html` — Used to send an email to the subscriber that the subscription is going to expire at the specified date in this template.
 - `FAIL.html` — Used to send an email to the subscriber that a subscription failed and to contact the administrator. The reason for the failure is not configured in this email.
- Approval notification templates:
 - `APPROVAL_REQD.html` — Used to send an email to one or more approvers that a subscriber submitted a service request. The service request can be a new service request, a request to modify an existing subscription, or a request to cancel an existing subscription.
 - `APPROVE.html` — Used to send an email to the subscriber that the subscription has been approved.
 - `REJECT.html` — Used to send an email to the subscriber that the subscription has been rejected.
- Service design templates:
 - To create email templates that are specific to service designs, make a copy of `ORDER.html` and rename it so that the name of the file is in the following format `<SERVICE DESIGN NAME>$ORDER.html`. For example, if a service design is called "Simple vCenter Compute", the file should be renamed as `Simple vCenter Compute$ORDER.html`.

Using the Notification Templates

If CSA has to send out HTML email notifications, the email templates must be designed using pre-defined notification tokens. These tokens are described in ["Notification Tokens" below](#). It is not possible to add custom tokens as these custom tokens will not be recognized by CSA. There are three kinds of tokens:

- Common tokens that can be used in any of the templates.
- Approval tokens that can only be used with the approval templates.
- Subscription tokens that can only be used with the subscription-based templates.

For token configuration requirements, and troubleshooting emails that do not display correctly (such as the token itself displays rather than the actual value, or the email displays as plain text instead of HTML output), see ["HTML Template Configuration/Troubleshooting Notes" on page 156](#)

When the templates are configured, CSA first looks for templates under the `csa.war\WEB-INF\classes\notifications\ORGANIZATION_ID` folder where `ORGANIZATION_ID` is the **Org Id** of the Organization defined in CSA. The Organization Identifier can be retrieved from the **General Information** tab on the Organizations page in the Cloud Service Management Console.

For Subscription and Approval notifications, CSA looks for the email template that corresponds to the notification in the folder for the Organization, and then in the default location (`csa.war/WEB-INF/classes/notifications/`) for each type of email notification.

For new service requests, CSA will first look for a design-specific template whose file name is of the form "`<SERVICE DESIGN NAME>$ORDER.html`" in the `csa.war/WEB-INF/classes/notifications/ORGANIZATION_ID` folder. If this file is not found, it will then look for a file called `ORDER.html` in the same folder. The identified template file will be used for all new service requests made by this organization.

If these two files are not found, then CSA will look for `ORDER.html` at `csa.war/WEB-INF/classes/notifications`, which is the default template for all organizations served by the CSA instance. If this file is also not found, then CSA will send legacy text notifications.

Notification Tokens

There are three types of tokens that are used in the appropriate HTML notification template: common, approval, and subscription. Most of the tokens used by HTML notifications are common. The following sections describe these tokens.

Common Tokens

The following table lists the common tokens and their descriptions, which can be used in all the notification templates.

Token	Description
Subscriber Tokens	
subscriber.name	The name of the subscriber, usually in the "FirstName LastName" format.
subscriber.userId	The user ID used by the subscriber to log in to the CSA portals.
Organization Tokens	
org.name	The name of the organization, defined as the Display Name of the Organization.
org.portalTitle	The title of the portal, defined as the title of the organization's Marketplace Portal.
org.csaUrl	The CSA host name, protocol, and port, usually as "https://CSAHOST:CSAPORT".
org.csaHost	The CSA host name and protocol, as "https://CSAHOST".
org.mppUrl	The path to the Marketplace Portal, usually in the "/org/ORG_NAME" format.
org.legalNotice	The link that points to the Organization's privacy agreement.
org.termsOfUse	The link that points to the Organization's terms of use.
org.welcomeMessage	The welcome message defined for the Marketplace Portal in the Organization.
org.footerMessage	The copyright statement defined for the Marketplace Portal in the Organization.
Service Tokens	
service.serviceld	The subscription ID created by CSA for the new service request raised by the subscriber.
service.serviceName	The name of the service as given by the subscriber.
service.serviceState	The state of the service, such as in an ACTIVE state, PENDING_APPROVAL state, and so on.
service.startDate	The date when the service will become available.

Token	Description
service.endDate	The date when the service will become unavailable — CSA will automatically cancel the service. The text "Not set" will be shown for recurring subscriptions, that is, for subscriptions that do not have end dates. This text is coded using a Velocity macro at the top of each template. Change the text "Not set" in the macro, if required.
service.offeringName	The name of the service offering used to create the service request.
service.initialPrice	The initial price of the service.
service.recurringPrice	The recurring price of the service.
service.recurrentPeriod	The period for which the recurring pricing will apply for the service. The text will be displayed in English by default. To localize this text, create a Velocity macro similar to the one at the beginning of each template which was created to handle the text for the service end date.
service.currencySymbol	The currency symbol used to define the initial and recurring pricing of the service.

Approval Tokens

The following tokens can only be used in these Approval templates described in "[Default HTML Templates](#)" on page 147: APPROVAL_REQD.html, APPROVE.html, and REJECT.html.

Token	Description
approver.name	The name of the user who is expected to approve or reject the service request.
approver.userId	The user ID used by the approver to log in to the CSA portals.
approver.comment	The reason given by an approver for rejecting a service request. Available only for REJECT notification templates.
approvalResult	A Java List of all approvers who have responded with an APPROVE or a DENY to a service request. Internally contains "approver" objects stored as Java Map objects.

Subscription Tokens

The following token can only be used in ORDER.html. This token is not available for other Subscription-based email templates such as CANCEL.html, EXPIRY.html, TO_EXPIRE.html, and FAIL.html.

Token	Description
componentDisplayName	Every visible component has this additional property created for HTML notifications. It contains the display name of the component. For example, used as <code>SERVER_GROUP.componentDisplayName</code> .

Configure HTML Notification for Pause on Failure

When CSA provisioning fails, the provisioning operation is normally marked as Failed. However, in some cases, you might want to pause the failed operation until it is evaluated by a predetermined group of users known as operators. Operators might decide to rectify the failure and force provisioning to continue, or allow the provisioning to fail.

The pause-on-failure notifications feature allows CSA to send notifications as follows:

- To subscribers when provisioning pauses for the first time.
- To operators when provisioning is paused due to a failure.
- To subscribers when operators make the decision to fail the provisioning operation.

Set options in the UI

To enable this feature in the UI, follow these steps for each consumer and provider organization:

1. In the CSA Management Console **Organizations** screen, select a consumer.
 - a. In the **Consumer Organization** section, select **Operations**.
 - b. In the **OperationsConfiguration** section:
 - i. In the **Provisioning Error Handling** subsection, choose **Pause Subscription on Provisioning Errors**. Setting this property enables CSA to pause if a provisioning has errors, but does not let the provisioning fail. The default is Fail Subscriptions on Provisioning Errors.
 - ii. In the **Paused Subscription** subsection, check **Notify Subscribers** and **Notify Operations**. Setting this property enables CSA to notify subscribers that the subscription has been paused during provisioning. The default is not enabled.

If, subsequently, the operator rectifies the paused provisioning by clicking the **Resume** button in the Operations Console a successful-service notification is sent to the subscriber. However, if, subsequently, the operator cancels the paused subscription

using the **Cancel** button in the Operations Console, a failed-service notification is sent to the subscriber instead.

- c. In the **Operator Users to Notify When Paused** section, click **AddOperatorUsers** and enter the operator information.

Setting this property enables CSA to notify operators when the subscription fails during provisioning.

Important: Email configuration for the provider organization is required, or CSA will be unable to send email notifications to operators when provisioning fails. In addition, you must select the Ensure Operator Notify option when provisioning error handling is set to Pause Subscription on Provisioning Errors. If this option is not set, operators cannot receive email notifications, even if the email configuration is defined for the provider organization.

Pause-on-failure notifications

Templates for operator notification are available only as default notification templates as the templates are specific to the provider organization. Templates for subscriber notification are available for consumer organizations as default notification templates, as well as organization-specific templates, but not as design-specific templates.

The OOTB Pause-on-failure notifications will be designed using the default ports for SMC and Marketplace portals. That is, the links to the login page will use ports 8444 and 8089 respectively. If the CSA and/or MPP instances use nondefault ports, then the OOTB templates must be modified accordingly.

Templates

The following table shows the property names and corresponding templates:

Enabled Property Name	Template Name
Subscriber Notify	PAUSED_SUBSCRIBER.html
Operator Notify	PAUSED_OPERATOR.html

OOTB templates can be modified. However, HPE recommends that you make backups of all the templates in case of errors in the modified templates. For more information about templates, see"

[Configure the Default Notification Templates" on page 146](#) and "[Customize the Default Notification Templates" below.](#)

Tokens

The following tokens have been identified for Pause-on-failure notifications. The new notification templates for PAUSED_OPERATOR and PAUSED_SUBSCRIBER will use these tokens, in addition to the ones that were identified for existing notification templates.

Token Name	Description
subscriber.pauseNotify	Set to TRUE if the Notify Subscribers checkbox is selected in the Operations tab of a consumer organization. Default value: FALSE
operator.pauseNotify	Set to TRUE if the Notify Operators checkbox is selected in the Operations tab of a consumer organization. Default value: FALSE

Customize the Default Notification Templates

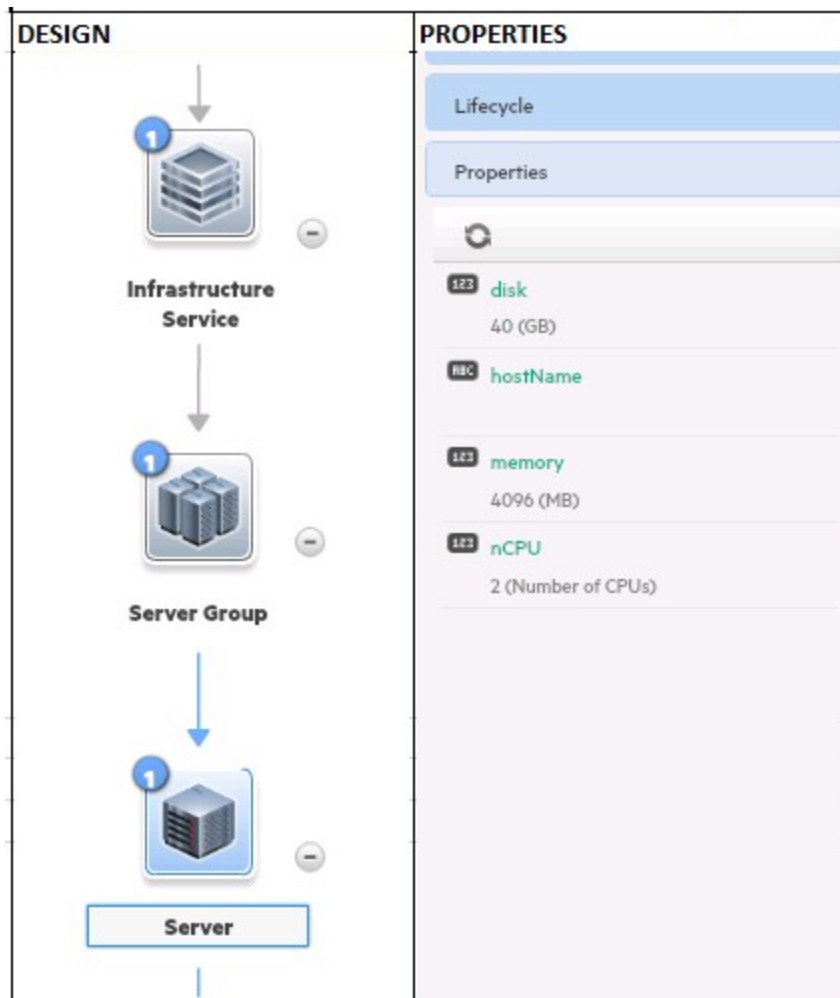
To customize the default notification templates, complete the following steps:

1. Customize the Organization by defining the application name. Select a logo for the organization. See "Notification Tokens" in "[Configure the Default Notification Templates" on page 146](#) to understand how CSA must be configured so that email templates can pull in data.
2. Go to `csa.war\WEB-INF\classes\notifications` and backup the default templates.
Use the original files for reference as notifications may stop working if the email templates are modified incorrectly.
3. From the Cloud Service Management Console, get the Organization Identifier of the organization for which notifications must be sent. Use this name as seen in the Provider Portal (case is sensitive) and create a directory using this name under `csa.war\WEB-INF\classes\notifications`. Copy the default email templates to this location. Modify the templates as required.
4. To create design-specific email templates, use the name of the Service design. Make a copy of `ORDER.html` in the same location, and rename it as follows:
`<SERVICE DESIGN NAME>$ORDER.html`, using \$ as a separator between the design name name and `ORDER.html`. Modify this template as required.

- For notifications mentioning new services, data on visible components and their non-encrypted visible properties can be retrieved as follows:

`${PARENT_COMPONENT_NAME-CHILD_COMPONENT_NAME.propertyName}`.

If a design is created in the Cloud Service Management Console as shown below, then the tokens will be created for the properties in the Server component. For example, a property called “serverCount” stored as a property in the component with the name (not display name!) SERVER_GROUP whose parent name is INFRASTRUCTURE_SERVICE, then the property is accessed in the email template as `${INFRASTRUCTURE_SERVICE-SERVER_GROUP.serverCount}`, where - is the separator between the two components.



Note: If the component does not have a parent component, then the property can be directly accessed using the component name as `${SERVER_GROUP.serverCount}`.

The approach described above should be used if only one component will be created using the combination. If multiple components will be created using this combination of parent and child components, or if a Clone Pattern will be used to create multiple components of a component type, then use the following approach.

The values are stored in a Java List object called PARENT_COMPONENT_NAME which contains individual Java Map objects for each child component. Use Apache Velocity's templating language to iterate through the Java List object and retrieve the value of the properties, as shown in the following sample code:

```
#foreach ($map in $SERVER_GROUP-SERVER)
  Hostname: $map.get('hostName')
  Flavor: $map.get('nCPU')CPU and $map.get('memory')Mb RAM
  Storage: $map.get('disk')Gb
#end
```

Unsupported HTML Notification Functionality

The following HTML notification functionality is not supported:

- It is not possible to create email templates specific for each version of a service design. For example, if a service design called "Simple vCenter Compute" has two or more versions, only a single email template can be created to be used by all the versions of this service design.
- Custom HTML notifications are not available for transfer of services, service instance upgrades or for services that have paused on failure. Legacy text notifications will be sent.
- Subject lines cannot be customized.
- Mails that are triggered by Notification API calls will be sent as legacy text notifications.
- Operations Orchestration flows that send notifications will not be sent in HTML format.
- HTML notifications will not be sent for approvals that may be configured for Public actions.
- Notifications will not be sent for Request Failure emails.
- Subscriber inputs will not be displayed in Approval Required email notifications.

HTML Template Configuration/Troubleshooting

Notes

- Be sure the HTML template file names are correct. The names must be the same as the default HTML template file names, with the exception of the design-specific email templates, which also follow a specific naming convention. Only the contents of the HTML files can be changed.
- The name of the directory under `csa.war/WEB-INF/classes/notifications` must be the same as the Organization Identifier, and not the Organization Display Name.
- Two or more versions of a service design will use the same template if a design-specific template is defined. If the components differ widely across these versions, it is not possible to create different design-specific templates for each version.
- The notification tokens are case-sensitive. For example: `${INFRASTRUCTURE_SERVICE-SERVER_GROUP.serverCount}` is not the same as `${infrastructure_service-Server_group.serverCount}`.
- If tokens are seen as-is in email notifications, check to see if the tokens are entered correctly. Verify that the components containing these properties, and the properties, are visible. Properties with encrypted data cannot be accessed.
- Only visible components and visible properties will be available.
 - The parent component need not be visible. So if the parent component is hidden, but the child component is visible, then all placeholder tokens corresponding to their visible properties of the child component will be resolved.
 - However, if the parent component is visible, but the child component is hidden, then the property values of the child component cannot be accessed using placeholder tokens.
- The tokens must be entered exactly as shown in the ["Notification Tokens" on page 148](#) section. If the token is entered incorrectly, the email notification will show the token name rather than the name that the token represents. For example, if you use `Subscriber.name` instead of `subscriber.name`, the email content will show **Subscriber.name** instead of the person's name.
- Be sure that the following characters are not used in the names of custom components:
`${[(.)]}` and whitespace

If CSA finds these characters in the name of the components, then they will be removed. For example, if a component name is `SERVER{TEST COMPONENT.NAME}`, then CSA will use the name **SERVERTESTCOMPONENTNAME**.

- To debug any possible issues in collecting data required for HTML notifications, enable debug logging for the following in `log4j2.xml`. This file can be found under `csa.war/WEB-INF/classes`:

```
com.hp.csa.service.notification.NotificationMailServiceImpl
```

```
com.hp.csa.service.notification.templates
```

Debug log entries will be seen in the `csa.log` file.

Note: For Clustered CSA nodes, email templates, `csa.properties` and `log4j2.xml` must be identical across all nodes so that the behavior is consistent in a cluster. The HTML templates and any modifications to the templates must be applied to each CSA node in the cluster.

Legacy text notifications will be sent under these conditions:

- If there is an error while creating emails in HTML. Look up `csa.log` and correct any parsing issues.
- If notifications are sent for Service Instance Upgrade, Pause On Failure and Transfer Subscription.

Configure Security Warning Messages for Cloud Service Management Console

You can enable or disable the security warning messages for files that are uploaded or downloaded.

The default upload message is:

Please make sure the files you upload are safe. Uploading malicious files will have legal consequences.

The default download message is:

Files you download may be potentially unsafe, it is advised to have a local antivirus software to prevent common threats.

To configure the security warning messages for Cloud Service Management Console:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/offerings/config.json` file in a text editor.
2. Find the `enableSecurityWarning` parameter value and set it to the desired value:

- `true` to enable the warning message.
- `false` to disable the warning message.

Enable Verification of an Imported Service Design, Service Offering, or Catalog Content Archive

Service design, service offering, and catalog content archives provide the ability to preserve these artifacts so they can be used to replicate them on another system or to restore them. CSA provides the ability to import archives of service designs, service offerings, catalogs, and their supported artifacts using the Cloud Service Management Console, Content Archive Tool, or REST APIs. By default, all service design, service offering, or catalog content archives are imported directly, without verification, into CSA.

Note: Service designs and catalogs can be imported using the Cloud Service Management Console, Content Archive Tool, or REST APIs. Service offerings can be imported using the Content Archive Tool or REST APIs.

For security reasons, you may want to verify the authenticity of a service design, service offering, or catalog content archive before importing it into CSA. When verification is enabled, CSA does the following:

- Verifies the digital signature of the content archive
- Validates the date of the certificate used to sign the content archive
- Verifies that the content in the content archive has not been modified after it was signed

If the content archive fails one of these validation or verification checks, the content archive will not be imported into CSA.

Caution: Verification cannot be enabled for importing a service design, service offering, or catalog content archive using the REST APIs. A service design, service offering, or catalog content archive imported using the REST APIs will always be imported directly. Verification can only be enabled for the Cloud Service Management Console or the Content Archive Tool.

Enabling the verification of imported service design, service offering, and catalog content archives requires that all imported service design, service offering, and catalog content archives be signed. Verification ensures the authenticity of the data within the service design, service offering, or catalog content archive has not been modified after it is signed. The following sections explain how to enable the verification of imported service design, service offering, and catalog content archives and how to sign these content archives so that they may be imported.

Prerequisites

Enabling verification requires that all imported service design, service offering, and catalog content archives are digitally signed using any JAR signing tool. CSA does not provide a JAR signing tool. A JAR signing tool is typically provided as part of a JDK, but CSA does not include a JDK.

Install a JDK and/or JAR signing tool on the same system that has the content archive that will be signed and the keystore used to sign the content archive. Refer to ["Create a Signed Content Archive" on page 161](#) for more information about creating a keystore and signing the content archive.

Examples Used in this Section

The examples in the following sections use the following information. You may want to customize some of the information to something more suitable for your needs (for example, the name and location of the keystore file or the alias of the certificate in the keystore). If you customize any of the information, be sure to substitute these customizations in all of the examples.

Item	Value(s) Used in Examples
JDK installation	Windows: C:\Program Files\Java\jdk Linux: /usr/bin/javac (CSA does not include a JDK.)
JRE that is used by CSA	CSA_JRE_HOME (The location where the JRE used by CSA is installed is referred to as CSA_JRE_HOME.) This JRE may be the OpenJDK JRE that is installed with CSA or a self-installed Oracle JRE (see the <i>Cloud Service Automation System and Software Support Matrix</i> for information about supported versions of the Oracle JRE).
keytool	The keytool is available in both the JRE that is used by CSA and the JDK installation. Either keytool may be used.

Item	Value(s) Used in Examples
	JRE: CSA_JRE_HOME/bin/keytool JDK: <ul style="list-style-type: none"> • Windows: C:\Program Files\Java\jdk\bin\keytool • Linux: /usr/bin/javac/bin/keytool
Content archive	Windows: C:\tmp\SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip Linux: /tmp/SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip
Keystore	Windows: C:\tmp\.keystore_archive_signing Linux: /tmp/.keystore_archive_signing
Alias used to access the certificate in the keystore	csa_archive
Keystore password	<keystore_password>
Key password	<key_password>

Enable Verification

To enable CSA to verify a service design, service offering, or catalog content archive when imported using the Cloud Service Management Console or the Content Archive Tool, set the following property to **true** in the CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties file (where CSA_HOME is the directory in which CSA is installed):

- csa.security.enable

If a property value is updated in the csa.properties configuration file, CSA must be restarted. Refer ["Restart CSA" on page 170](#) for information about how to restart CSA.

Note: Verifying service designs and catalogs before they are imported is done using the Cloud Service Management Console or the Content Archive Tool. Verifying service offerings before they are imported is done using the Content Archive Tool.

Create a Signed Content Archive

If verification of a service design, service offering, or catalog content archive is enabled, the content archive must be signed by a JAR signing tool before it can be imported into CSA.

If verification of a service design, service offering, or catalog content archive is enabled, it is recommended that you sign the service design, service offering, or catalog content archive immediately after exporting it.

To create a signed content archive, do the following:

- Locate or create a keystore and certificate used to sign the content archive
- Sign the content archive

Locating or Creating a Keystore and Certificate

Before you can sign the content archive, you must have an unexpired certificate that you can use. This certificate must be stored in a keystore that you can access and you must know the alias to access the certificate. The certificate can be signed by a certificate authority or it can be self-signed.

If you do not have a keystore or certificate to use, you can create a keystore and a self-signed certificate to sign the content archive.

Creating a Keystore and Self-Signed Certificate

The example shown in this section creates a keystore named `.keystore_archive_signing`, in which a self-signed certificate can be accessed using the alias `csa_archive`. The self-signed certificate is valid for 365 days and is generated using the RSA key algorithm and a 2048 bit key size.

1. Open a command prompt and change the directory to `CSA_JRE_HOME/bin`. For example, if you are using the JRE installed with CSA, go to `C:\Program Files\HPE\CSA\openjre\bin` on Windows or `/usr/local/hpe/csa/openjre/bin` on Linux.

2. Run the following command:

```
keytool -genkeypair -keystore /tmp/.keystore_archive_signing -alias  
csa_archive -validity 365 -keyalg rsa -keysize 2048
```

3. Enter a keystore password (`<keystore_password>`). This password is used to control access to the keystore. You will need this password when signing a content archive.

4. Follow the prompts to enter your name, organization, and location values.
5. Enter the key password (<key_password>). This password is used to control access to the alias. You will need this password when signing a content archive.

You have completed creating a keystore and self-signed certificate and can now sign your content archives.

Signing the Content Archive

To sign a content archive, the JAR signing tool, content archive to sign, and keystore must be located on the same system.

The example shown in this section signs the content archive `SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip` with the certificate stored in the keystore `.keystore_archive_signing` which is accessed using the password `<keystore_password>`. The certificate is accessed using the alias `csa_archive` and the password `<key_password>`.

1. Open a command prompt and change to the JDK's bin directory. For example, go to `C:\Program Files\Java\jdk\bin` on Windows or `/usr/bin/javac/bin` on Linux.
2. Run the following command:

Windows:

```
jarsigner -keystore C:\tmp\.keystore_archive_signing  
-storepass <keystore_password> -keypass <key_password>  
C:\tmp\SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip csa_archive
```

Linux:

```
jarsigner -keystore /tmp/.keystore_archive_signing  
-storepass <keystore_password> -keypass <key_password>  
/tmp/SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip csa_archive
```

Optionally, you may specify `-sigFile` with a value that will be used to name the signature files that are added to the signed content archive. If not specified, it will use the first eight letters of the alias (`csa_arch`) to name the signature files.

3. Optionally, verify the signed content archive by running the following command:

Windows:

```
jarsigner -verify C:\tmp\SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip
```

Linux:

```
jarsigner -verify /tmp/SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d.zip
```

The content archive is signed and can be imported into CSA.

Re-Sign a Content Archive

If the certificate you used to sign a content archive has expired, you can re-sign the content archive using a new certificate. The example in this section assumes that the JAR signing tool, content archive to re-sign, and keystore are located on the same system.

1. On the system, open a command prompt and create a directory in which to extract the files from the content archive and go to that directory. For example, run the following commands:

Windows:

```
mkdir C:\tmp\contentarchive  
cd C:\tmp\contentarchive
```

Linux:

```
mkdir /tmp/contentarchive  
cd /tmp/contentarchive
```

2. Extract the files from the content archive. For example, run the following command:

Windows:

```
"C:\Program Files\Java\jdk\bin\jar" -xvf C:\tmp\SERVICE_OFFERING_  
2c9f4ab8b896014ac3520ca7016d.zip
```

Linux:

```
/usr/bin/javac/bin/jar -xvf /tmp/SERVICE_OFFERING_  
2c9f4ab8b896014ac3520ca7016d.zip
```

3. Remove the expired signature files. For example, run the following command:

Windows:

```
rmdir /q /s META-INF
```

Linux:

```
rm -rf META-INF
```

4. Create a new content archive.

Windows:

```
"C:\Program Files\Java\jdk\bin\jar" -cvf C:\tmp\SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d_NEW.zip *
```

Linux:

```
/usr/bin/javac/bin/jar -cvf /tmp/SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d_NEW.zip *
```

5. Change to the JDK's bin directory. For example, go to:

Windows:

```
C:\Program Files\Java\jdk\bin
```

Linux:

```
/usr/bin/javac/bin
```

6. If you have access to the keystore, remove the expired certificate by running the following command:

Windows:

```
keytool -delete -keystore C:\tmp\.keystore_archive_signing -alias csa_archive -storepass <keystore_password>
```

Linux:

```
keytool -delete -keystore /tmp/.keystore_archive_signing -alias csa_archive -storepass <keystore_password>
```

7. If you are using a certificate generated for you, get the keystore, keystore password, and alias to access the certificate. If you are using a self-signed certificate, follow the instructions in ["Locating or Creating a Keystore and Certificate" on page 161](#) to generate a new self-signed certificate.
8. Re-sign the content archive. Follow the instructions in ["Signing the Content Archive" on page 162](#) to re-sign the content archive (use the new content archive name, SERVICE_OFFERING_2c9f4ab8b896014ac3520ca7016d_NEW.zip).

Common CSA Tasks

This chapter provides information about how to perform common CSA tasks.

Tasks include:

- "Launch the Cloud Service Management Console" below
- "Launch the Marketplace Portal" below
- "Start CSA" on page 168
- "Stop CSA" on page 171
- "Restart CSA" on page 170
- "Encrypt a password" on page 172
- "Clear the web browser cache" on page 172
- "Uninstall CSA" on page 173

Launch the Cloud Service Management Console

Launch the Cloud Service Management Console by typing the following URL in a supported web browser: `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which the Cloud Service Management Console resides.

Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`

Launch the Marketplace Portal

Launch the default Marketplace Portal

Launch the default Marketplace Portal by typing one of the following URLs in a supported Web browser:

- `https://<csahostname>:8444/mpp`
- `https://<csahostname>:8089`

where `<csahostname>` is the fully-qualified domain name of the system on which the Marketplace Portal instance resides and that was used when CSA was installed.

For example: `https://csa_system.abc.com:8444/mpp`

The organization associated with the default Marketplace Portal is defined in the `CSA_HOME/portal/conf/mpp.json` file. By default, this is the sample organization that is installed with CSA (CONSUMER). To modify the organization associated with the default Marketplace Portal, modify the `defaultOrganizationName` property value by setting it to the `<organization_identifier>` of the desired organization, where `<organization_identifier>` is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Organizations** menu of the Cloud Service Management Console).

Launch an organization-specific Marketplace Portal

Launch an organization's Marketplace Portal by typing the following URL in a supported Web browser:

`https://<csahostname>:8089/org/<organization_identifier>`

where:

- `<csahostname>` is the fully-qualified domain name of the system on which the Marketplace Portal instance resides and that was used when CSA was installed.
- `<organization_identifier>` is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Organisations** menu of the Cloud Service Management Console).

Example:

`https://csa_system.xyz.com:8089/org/ORGANIZATIONA`

Caution: Do not launch more than one organization-specific Marketplace Portal from the same browser session. For example, if you launch ORGANIZATION_A's Marketplace Portal in a browser, do not open a tab or another window from that browser and launch ORGANIZATION_B's Marketplace Portal. Otherwise, the user who has logged in to the Marketplace Portal launched for ORGANIZATION_A will start to see data for ORGANIZATION_B.

Instead, start a new browser session to launch another organization's Marketplace Portal.

Launch the default remote instance of a Marketplace Portal

Launch the default remote instance of the Marketplace Portal by typing one of the following URLs in a supported Web browser:

- `https://<csahostname>:8444/mpp`
- `https://<mpphostname>:8089`

where:

- `<csahostname>` is the fully-qualified domain name of the system on which CSA is installed and the URL in the `CSA_HOME/jboss-as/standalone/deployments/mpp.war/index.html` file (on the system on which CSA is installed) has been updated to `https://<mpphostname>:8089`.
- `<mpphostname>` is the fully-qualified domain name of the system on which the Marketplace Portal instance resides.

Examples:

- `https://csa_system.abc.com:8444/mpp`
- `https://mpp_system.abc.com:8089`

The organization associated with the default Marketplace Portal is defined in the `CSA_HOME/portal/conf/mpp.json` file (on the system on which the Marketplace Portal instance resides). By default, this is the sample organization that is installed with CSA (CONSUMER). To modify the organization associated with the default Marketplace Portal, modify the `defaultOrganizationName` property value by setting it to the `<organization_identifier>` of the desired organization, where `<organization_identifier>` is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Organizations** menu of the Cloud Service Management Console).

Launch an organization-specific remote instance of a Marketplace Portal

Launch an organization's remote instance of the Marketplace Portal by typing the following URL in a supported Web browser:

```
https://<mpphostname>:8089/org/<organization_identifier>
```

where:

- *<mpphostname>* is the fully-qualified domain name of the system on which the Marketplace Portal instance resides.
- *<organization_identifier>* is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Organizations** menu of the Cloud Service Management Console)

Example:

```
https://mpp_system.xyz.com:8089/org/ORGANIZATION_A
```

Caution: Do not launch more than one organization-specific Marketplace Portal from the same browser session. For example, if you launch ORGANIZATION_A's Marketplace Portal in a browser, do not open a tab or another window from that browser and launch ORGANIZATION_B's Marketplace Portal. Otherwise, the user who has logged in to the Marketplace Portal launched for ORGANIZATION_A will start to see data for ORGANIZATION_B.

Instead, start a new browser session to launch another organization's Marketplace Portal.

Start CSA

To start CSA on Windows, complete the following steps:

1. If you have configured CSA to be FIPS 140-2 compliant, create a CSA encryption keystore password file. The name and location of this file must match the value configured for the keystorePasswordField property in the CSA_HOME\jboss-

as\standalone\deployments\csa.war\WEB-INF\classes\csa.properties file.

The password file must contain only the following content: `keystorePassword=<CSA encryption keystore password>`

where `<CSA encryption keystore password>` is the CSA encryption keystore password in clear text.

This file is automatically deleted when the HPE Cloud Service Automation service is started.

2. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
3. If global search is enabled, do the following:
 - a. Right-click on the Elasticsearch 1.6.1 service and select **Restart**.
 - b. Wait for the Elasticsearch 1.6.1 service to restart, then right-click on HPE Search Service and select **Restart**.

Note: if global search is disabled, skip this step.

4. Right-click on the CSA service and select **Start**.
5. Right-click on the HPE Marketplace Portal service and select **Start**.
6. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Start**.

To start CSA on Linux, complete the following steps:

1. On the server that hosts CSA, type the following:

```
service csa start  
service mpp start
```

Refer to the `csa.provider.es.exists` property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 and type the following:

```
service elasticsearch start
```

By default, `elasticsearch` is enabled.

2. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPEOOInstallation>/central/bin/central start
```

For example, type `/usr/local/hpe/csa/00/central/bin/central start`

Restart CSA

To restart CSA on Windows, complete the following steps:

1. If you have configured CSA to be FIPS 140-2 compliant, create a CSA encryption keystore password file. The name and location of this file must match the value configured for the `keystorePasswordFile` property in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\WEB-INF\classes\csa.properties` file.

The password file must contain only the following content: `keystorePassword=<CSA encryption keystore password>`

where `<CSA encryption keystore password>` is the CSA encryption keystore password in clear text.

This file is automatically deleted when the CSA service is started.

2. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
3. If global search is enabled, do the following:
 - a. Right-click on the Elasticsearch 1.6.1 service and select **Restart**.
 - b. Wait for a minute for the Elasticsearch 1.6.1 service to restart, then right-click on HPE Search Service and select **Restart**.

Note: If global search is disabled, skip this step.

4. Right-click on the CSA service and select **Restart**.
5. Right-click on the HPE Marketplace Portal service and select **Restart**.
6. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Restart**.

To restart CSA on Linux, complete the following steps:

1. On the server that hosts CSA, type the following:

```
service csa restart
service mpp restart
```
2. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPE00installation>/central/bin/central stop
```

```
<embeddedHPE00installation>/central/bin/central start
```

For example, type:

```
/usr/local/hpe/csa/00/central/bin/central stop
```

```
/usr/local/hpe/csa/00/central/bin/central start
```

Stop CSA

To stop CSA on Windows, complete the following steps:

1. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
2. Right-click on the CSA service and select **Stop**.
3. Right-click on the HPE Marketplace Portal service and select **Stop**.
4. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Stop**.
5. If you enabled global search, do the following:
 - a. Right-click on the Elasticsearch 1.6.1 service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
 - b. Right-click on HPE Search Service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
6. If Elasticsearch is enabled (by default, Elasticsearch is enabled; refer to the [csa.provider.es.exists](#) property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 for more information), right-click on the Elasticsearch 1.6.1 service and select **Stop**.

To stop CSA on Linux, complete the following steps:

1. On the server that hosts CSA, type the following commands:

```
service csa stop
```

```
service mpp stop
```

2. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPE00installation>/central/bin/central stop
```

For example, type: `/usr/local/hpe/csa/00/central/bin/central stop`

Encrypt a password

To encrypt a password (for use with CSA configuration only; see ["Encrypt a Marketplace Portal Password" on page 180](#) for information about how to encrypt a Marketplace Portal password):

1. Open a command prompt and change to the `CSA_HOME/Tools/PasswordUtil` directory. For example:

```
/usr/local/hpe/csa/Tools/PasswordUtil
```

2. Run the following command:

```
CSA_JRE_HOME/bin/java -jar passwordUtil-standalone.jar encrypt  
<myPassword>,ROLE_REST,enabled
```

Clear the web browser cache

It may be necessary to clear your web browser cache on systems that previously accessed the Cloud Service Management Console after upgrading CSA.

To clear your Web browser cache:

- If you are using a Chrome Web browser:
 - a. Open the browser.
 - b. Select **<Ctrl>+<Shift>+<Delete>**.
 - c. For **Obliterate the following items from**, select **the beginning of time**.
 - d. Select only **Empty the cache**. Unselect all other items.
 - e. Click **Clear browsing data**.
- If you are using a Firefox Web browser:
 - a. Open the browser.
 - b. Select **<Ctrl>+<Shift>+<Delete>**.
 - c. For **Time range to clear**, select **Everything**.
 - d. Expand **Details**.

- e. Select only **Cache**. Unselect all other items.
- f. Click **Clear Now**.

Uninstall CSA

Uninstalling CSA removes the CSA_HOME directory and all of its contents (where CSA_HOME is the directory in which CSA is installed). If all the contents in CSA_HOME are not deleted, you must manually delete them and the CSA_HOME directory.

If you installed an embedded OO instance with CSA (you installed OO with CSA using the CSA installer), the embedded OO instance is removed. If you are using CSA with an external OO instance (you installed OO separately from CSA), the external OO instance is not removed.

Note: The CSA database is NOT updated or uninstalled.

Uninstall CSA on Windows

To uninstall CSA, complete the following steps:

1. Stop the CSA and HPE Marketplace Portal services.

To stop CSA on Windows, complete the following steps:

- a. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the CSA service and select **Stop**.
- c. Right-click on the HPE Marketplace Portal service and select **Stop**.
- d. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Stop**.
- e. If you enabled global search, do the following:
 - i. Right-click on the Elasticsearch 1.6.1 service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
 - ii. Right-click on HPE Search Service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).

- f. If Elasticsearch is enabled (by default, Elasticsearch is enabled; refer to the `csa.provider.es.exists` property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 for more information), right-click on the Elasticsearch 1.6.1 service and select **Stop**.

To stop CSA on Linux, complete the following steps:

- a. On the server that hosts CSA, type the following commands:

```
service csa stop  
service mpp stop
```

- b. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPE00installation>/central/bin/central stop
```

For example, type: `/usr/local/hpe/csa/00/central/bin/central stop`

2. Verify that the services were stopped.

If the CSA service is still running, open a command prompt, navigate to `CSA_HOME\jboss-as\bin`, and run the following command:

```
jboss-cli.bat --connect --command=:shutdown
```

3. Close all instances of Windows Explorer, close all command prompts, and exit all programs that are running on the system.
4. Navigate to **Control Panel > Uninstall a program**.
5. Right-click **HPE Cloud Service Automation** and select **Uninstall/Change**.
6. Click **Uninstall**.
7. Delete the `CSA_HOME` directory and any remaining contents, if they exist.
8. If they exist, delete all CSA entries from the following file:

```
C:\Program Files\Zero G Registry\.com.zerog.registry.xml
```

To uninstall CSA, complete the following steps:

1. Log in as the user who installed CSA (for example, `csauser`).
2. Stop all CSA services.

- a. On the server that hosts CSA, type the following:

```
service csa stop  
service mpp stop
```

- a. If Elasticsearch is enabled (by default, Elasticsearch is enabled; refer to the

`csa.provider.es.exists` property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 for more information), type the following:

```
service elasticsearch stop
```

- b. If you installed an embedded OO instance, type:

```
<embeddedHPEOOinstallation>/central/bin/central stop.
```

For example, type `/usr/local/hpe/csa/00/central/bin/central stop`

3. Verify that the services were stopped. For example, if CSA was installed in `/usr/local/hpe/csa`, enter the following:

```
ps -ef | grep /usr/local/hpe/csa
ps -ef | grep mpp
ps -ef | grep central
```

If there are CSA, Marketplace Portal, or OO services running, repeat step 2 or kill the CSA, Marketplace Portal, and OO services.

4. Go to the `CSA_HOME/_CSA_4_80_0_installation` directory. Enter the following:

```
cd CSA_HOME/_CSA_4_80_0_installation
```

5. Uninstall CSA. Enter the following:

```
./Change\ HPE\ Cloud\ Service\ Automation\ Installation
```

6. Confirm that you want to uninstall CSA.

7. When uninstallation completes, log in as root and do the following:

- a. If all the contents in `CSA_HOME` are not deleted, you must manually delete them and the `CSA_HOME` directory.

- b. Delete the CSA and HPE Marketplace Portal service scripts. Enter the following:

```
rm /etc/init.d/csa
rm /etc/init.d/mpp
```

- c. If they exist, delete all CSA entries from the following file:

```
/home/csouser/.com.zerog.registry.xml
```

- d. Optionally, remove the `csauser` user and `csagrps` group.

The Marketplace Portal

This chapter provides the following information on the Marketplace Portal:

- ["Configure Global Search" below](#)
- ["Configure the Showback Report Menu" on page 178](#)
- ["Encrypt a Marketplace Portal Password" on page 180](#)
- ["Configure Security Warning Messages for Marketplace Portal" on page 181](#)

For information about configurable attributes in the `mpp.json` file, see ["Appendix: Marketplace Portal Attributes" on page 349](#).

See the *Cloud Service Management Console Help* for information about customizing the Marketplace Portal.

Configure Global Search

Global search allows you to find a certain service offering, service instance, or subscription by a meaningful keyword. For service offerings, global search finds the keyword in the name, description, option sets, options, and properties. For service instances and subscriptions, global search finds the keyword in the name, description, and instance properties (name and value).

Note: The Search Results view displays the keyword found only in service offerings, service instances, and subscriptions within your organization. In the Search Results view, click on an object for more detailed information about a service offering or subscription.

Caution: You must disable global search in a FIPS 140-2 compliant environment.

By default, global search is enabled in CSA 4.70.

Note: When upgrading from CSA 4.5x to CSA 4.7x, global search is not enabled by default. Enable global search as described in the next section ["Enable Global Search" on the next page](#).

Enable Global Search

To enable global search, complete the following steps:

1. Configure the global search property:
 - a. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file in a text editor.
 - b. Set the `csa.provider.es.exists` property to **yes** or **true** (for example, `csa.provider.es.exists=yes`).
 - c. Save and exit the file.
2. Enable the global search icon in the top header of the Marketplace Portal:
 - a. Open the `CSA_HOME/portal/conf/dashboard.json` file in a text editor.
 - b. Set the `header: search: enable` attribute to **true** (for example,

```
"header": {  
  .  
  .  
  .  
  "search": {  
    "enable": true
```
 - c. Save and exit the file.
3. Start the global search services:
 - a. On the server that hosts CSA, navigate to **Control Panel > Administrative Tools > Services**.
 - b. Right-click on the Elasticsearch 1.6.1 service and select **Start**.
 - c. Wait for a minute for the Elasticsearch 1.6.1 service to start, then right-click on the HPE Search Service and select **Start**.
4. Restart CSA and HPE Marketplace Portal services. See ["Restart CSA" on page 170](#) for instructions.

Disable Global Search

To disable global search, complete the following steps:

1. Configure the global search property:
 - a. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file in a text editor.
 - b. Set the `csa.provider.es.exists` property to **no** or **false** (for example, `csa.provider.es.exists=no`).
 - c. Save and exit the file.
2. Disable the global search icon in the top header of the Marketplace Portal:
 - a. Open the `CSA_HOME/portal/conf/dashboard.json` file in a text editor.
 - b. Set the `header: search: enable` attribute to **false** (for example,

```
"header": {  
  .  
  .  
  .  
  "search": {  
    "enable": false
```
 - c. Save and exit the file.
3. Restart CSA and HPE Marketplace Portal services. See ["Restart CSA" on page 170](#) for instructions.

Configure the Showback Report Menu

The `[[[Undefined variable CSAVariables.tabMPP_ShowbackReport]]]` menu in the Marketplace Portal is a link to HPE IT Business Analytics, which automatically gathers metrics from CSA to build key performance indicators. HPE IT Business Analytics provides scorecards and dashboards so that a Consumer Organization Administrator has insight into how to measure and optimize the cost, risk, quality, and value of IT services and processes.

In the Marketplace Portal, the Consumer Organization Administrator role has access to the Showback Report tile. By default, the Showback Report menu is enabled in the Marketplace Portal. However, you must configure the hostname of the system on which HPE IT Business Analytics is installed to link to HPE IT Business Analytics from the Marketplace Portal. Additionally, to ensure seamless navigation between the Marketplace Portal and HPE IT Business Analytics, configure HP SSO between the Marketplace Portal and HPE IT Business Analytics.

Configure the Link to HPE IT Business Analytics

1. Navigate to the `CSA_HOME/portal/conf/` directory.
2. Make a backup copy of the `dashboard.json` file.
3. Open the `dashboard.json` file in a text editor.
4. Locate the following section:

```
"label": "common.items.SCORECARD",  
"icon": {  
  "className": "icon-status"  
},  
"link": {  
  "url": "https://<CONFIGURE_HOST_NAME>/  
fndwar/loadEmbeddedPage.jsp?com.hp.bsm.uim.pageUID=ef63ab7f-b86b-43c8-b8d8-  
bb81869b73dc",  
  "target": "_blank"  
}
```

5. Replace `<CONFIGURE_HOST_NAME>` with the host name of your HPE IT Business Analytics installation.
6. Save and exit the file.
7. If you are logged in to the Marketplace Portal, clear the browser cache (see ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache) and refresh the browser.

Note: The changes to the `dashboard.json` file do not require you to restart CSA.

Configure HP SSO

To ensure seamless navigation between the Marketplace Portal and HPE IT Business Analytics, HP SSO must be configured for CSA and HPE IT Business Analytics. Note the following:

- Verify that HP SSO for HPE IT Business Analytics is configured to enable logging on to the Marketplace Portal. See the *HPE IT Business Analytics Administrator Guide* for more information about configuring HP SSO for HPE IT Business Analytics.
- For SSO between CSA and HPE IT Business Analytics to work successfully, both products must

be installed on machines that are in the same Domain. The value of Domain and Protected Domain parameters specified for HP SSO configuration must be the same.

Note: As HP SSO is enabled by default, if you disable HP SSO manually, seamless navigation between CSA/Marketplace Portal and HPE IT Business Analytics will no longer work.

- You must configure users for both CSA and HPE IT Business Analytics for single sign-on (each user must have the same name and password). You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and HPE IT Business Analytics to use the same LDAP source or, if CSA and HPE IT Business Analytics use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the appropriate role to access the menu that launch HPE IT Business Analytics and the HPE IT Business Analytics user must be assigned a role that allows it to perform the expected functions in HPE IT Business Analytics.
- When configuring HP SSO, the `initString` setting for the Cloud Service Management Console/Marketplace Portal, and HPE IT Business Analytics must be configured to the same value. The value of the `initString` attribute is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file. Use this setting to configure the Cloud Service Management Console/Marketplace Portal, and HPE IT Business Analytics.

The `initString` value represents a secret key and should be treated as such in your environment.

Encrypt a Marketplace Portal Password

To encrypt a password used by the Marketplace Portal:

1. Open a command prompt and change to the `CSA_HOME/portal/bin` directory. For example:

Windows:

```
C:\Program Files\HPE\CSA\portal\bin
```

Linux:

```
/usr/local/hpe/csa/portal/bin
```

2. Run the following command:

Windows:

```
..\..\node.js\node passwordUtil --keyfilePath <keyfile> --password <myPassword>
```

Linux:

```
passwordUtil --keyfilePath <keyfile> --password <myPassword>
```

where <keyfile> is the path to (absolute or relative to the bin directory) and name of the file that contains the Marketplace Portal's encrypted symmetric key (if the file does not exist, it will create the file) and <myPassword> is the password to be encrypted.

Configure Security Warning Messages for Marketplace Portal

You can enable or disable security warning messages for the Marketplace Portal as described in the following sections.

Configure Uploaded or Downloaded File Security Warning Messages

You can enable/disable the security warning messages for files that are uploaded or downloaded in the Marketplace Portal.

The default upload message is:

Please make sure the files you upload are safe. Uploading malicious files will have legal consequences.

The default download message is:

Files you download may be potentially unsafe, it is advised to have a local antivirus software to prevent common threats.

To configure the security warning messages for Marketplace Portal:

1. Open the `CSA_HOME/portal/node_modules/mpp-consumption/dist/offerings/config.json` file in a text editor.
2. Find the `enableSecurityWarning` parameter value and set it to the desired value:

- `true` to enable the warning message.
- `false` to disable the warning message.

Configure Attach Documents Security Warning Message

You can enable/disable the Attach Documents security warning message on the Offerings page at checkout time in the Marketplace Portal.

The Attach Documents message is:

Security Warning: Please make sure the files you upload are free of viruses and other threats. Deliberately uploading malicious files may have legal consequences.

To configure the Attach Documents security warning message for Marketplace Portal:

1. Open the `CSA_HOME/portal/node_modules/mpp-server/conf/dashboard.json` file in a text editor.
2. Find the `"SecurityWarning": "enable"` parameter value and set it to the desired value:
 - `true` to enable the warning message.
 - `false` to disable the warning message.

The OO Designer

The OO Designer is a web-based environment for authoring flows, which is embedded in the CSA UI.

This chapter provides information about configuring the OO Designer to work in an HTML Inline Frame Element (iFrame).

Note: An iFrame embeds another HTML page into the current page.

Complete the following steps to configure CSA's OO Designer to appear in an iFrame instead of appearing in a new browser window.

1. After installing CSA, edit the following file:

```
CSA_HOME/jboss-as/standalone/deployments/csa.war/dashboard/config.json
```

2. Locate the following section:

```
{  
    "id": "oo_designer",  
    "name": "oo_designer",  
    "description": "oo_designer_description",  
    "enabled": true,  
    "style": "oo_designer",  
    "type": "link",  
    "target": "_blank",  
    "url": "/csa/oows_redirect.jsp",  
    "helptopic": "/csa/oowshelp_redirect.jsp",  
    "roles": ["CSA_ADMIN", "WORKFLOW_AUTHOR"]  
}
```

3. Remove the line containing `target` and change the value of the `type` field to `iframe` as shown in bold font:

```
{  
    "id": "oo_designer",  
    "name": "oo_designer",  
    "description": "oo_designer_description",  
    "enabled": true,
```

```
"style": "oo_designer",  
"type": "iframe",  
"url": "/csa/oows_redirect.jsp",  
"helptopic": "/csa/oowshelp_redirect.jsp",  
"roles": ["CSA_ADMIN", "WORKFLOW_AUTHOR"]  
}
```

4. Edit the following file:

CSA_HOME/jboss-as/standalone/deployments/csa.war/oows_redirect.jsp

5. Add "?embedded=true" to the URL as shown in bold font:

```
<%@page import="com.hp.csa.core.cfg.CSAProperties" %>  
<html>  
<head>  
<%  
String oowsURL = CSAProperties.get("OO_WS_URL");  
response.sendRedirect(oowsURL + "/oo-designer/?embedded=true");  
%>  
</head>  
</html>
```


6. Do the following to restart the HPE Cloud Automation service:

- o **Windows:** Restart the HPE Cloud Service Automation service.
- o **Linux:** Run the following commands:
 - i. Run `./csa stop`
 - ii. Run `./csa status` and check that the csa service is not running.
 - iii. Run `./csa start` and check that the csa service is running.

CSA includes a self-signed certificate. After you configure CSA to open the OO Designer in an iframe, the browser might not trust the certificate. So you might not be able to open the OO Designer.

Complete the following steps to remedy this certificate issue.

1. After you enter the CSA URL into a browser, the browser might mark the URL as being untrusted:

- o **Chrome:** 
- o **Internet Explorer:** Look for a certificate error in the URL.

2. Do the following to open a certificate dialog:

- **Chrome:** Click the red triangle, then click **Details**.
- **Internet Explorer:** Click the certificate error, then click **View certificates**.
- 3. After the certificate dialog appears, export the certificate to a file using DER encoded binary X.509 (.CER) format. Remember the file name and location to which you exported the file.
- 4. To Import the certificate to a browser's Trusted Root Certification Authorities do the following:
 - **Chrome**
 - i. Click **Settings >Show advanced settings...**
 - ii. Locate the **HTTPS/SSL** section and click **Manage certificates**.
 - iii. After the **Certificates** dialog appears, select **Trusted Root Certification Authorities**.
 - iv. Click **Import...**
 - v. Navigate to the certificate file that you exported previously.
 - vi. During the next step, select **Place all certificates in the following store**.
 - vii. Ensure that Certificate store is set to Trusted Root Certification Authorities, then click **Next**.
 - viii. Follow the remaining browser navigation instructions to install the certificate.
 - ix. Restart the browser and enter the CSA URL into the browser to test that you can now open CSA.
 - **Internet Explorer:**
 - i. Click **Internet Options >Content > Certificates**.
 - ii. After the **Certificates** dialog appears, select **Trusted Root Certification Authorities**.
 - iii. Click **Import...**
 - iv. Navigate to the certificate file that you exported previously.
 - v. During the next step, select **Place all certificates in the following store**.
 - vi. Ensure that Certificate store is set to Trusted Root Certification Authorities, then click **Next**.
 - vii. Follow the remaining browser navigation instructions to install the certificate.
 - viii. Restart the browser and enter the CSA URL into the browser to test that you can now open CSA.

Integrate CSA with HPE UCMDB

This chapter describes how to use the HPE CSA (CI based) palette and to configure and integrate CSA with a HPE UCMDB provider.

Using HPE CSA (CI Type based) Palette

HPE CSA (CI Type based) palette is available for Sequenced Designs and the components in this palette are similar to the components available in the HPE CSA palette. However, the component types in the HPE CSA (CI Types based) palette is aligned with HPE UCMDB's CI Type system, so that each new CSA component type is mapped to its corresponding CI Type in HPE UCMDB, and the attributes of the corresponding CI Types are mapped to the attributes of the CSA component type.

An HPE UCMDB provider should be configured if the new component types based on HPE UCMDB's CI Type system needs to be created, or if the service topology provisioned by CSA should be created in HPE UCMDB. If the HPE UCMDB provider is configured and a service offering is subscribed or a service design is deployed in CSA, then the equivalent topology will be created in HPE UCMDB.

Note: For the Cloud Subscription Service CI to be visible in HPE UCMDB tool, the HPE UCMDB Content Pack should be imported to UCMDB from HPLN.

To download the new package, go to **HPLN > Cloud Service Automation > Resource tab > Tools** tile and download the **CloudSubscriptionService UCMDB Package**. Deploy the package using the **UCMDB Package Manager** from the **Administration** tab.

This chapter describes the following:

- [Configuring CSA for HPE UCMDB Integration](#)
- [CSA Component Types aligned with HPE UCMDB's CI Types](#)
- [CI Type Component Types Hierarchy](#)
- [Creating a Custom CSA Component Type](#)

Configure CSA for HPE UCMDB Integration

Complete the following steps to configure CSA for HPE UCMDB integration:

1. Login to the server where HPE UCMDB is installed and extract the UCMDB server certificate using the following command:

Note: The UCMDB certificate must be imported into CSA's keystore if the UCMDB provider is configured to use secure HTTP for communication.

For Windows:

```
%UCMDB_JRE_HOME%\bin\jre\bin\keytool.exe -exportcert -keystore %UCMDB_JRE_HOME%\conf\security\server.keystore -alias hpcert -storepass hppass -file remoteServer.cert
```

Where, %UCMDB_JRE_HOME% is the directory where the Java Runtime Environment used by HPE UCMDB is installed.

For Linux:

```
$UCMDB_JRE_HOME/bin/jre/bin/keytool -exportcert -keystore $UCMDB_JRE_HOME/conf/security/server.keystore -alias hpcert -storepass hppass -file remoteServer.cert
```

Where, \$UCMDB_JRE_HOME is the directory where the Java Runtime Environment used by HPE UCMDB is installed.

2. Copy the file "remoteServer.cert" to the server where CSA is installed, to a temporary location.
3. Change to the following directory in the server where CSA is installed:

For Windows:

```
%CSA_HOME%\openjre\bin
```

For Linux:

```
$CSA_HOME/openjre/bin
```

4. Open a Command Shell at this location, and execute the following command:

For Windows:

```
keytool.exe -importcert -alias hpcert -file <FILE PATH>/remoteServer.cert -keystore "%CSA_HOME%\openjre\lib\security\cacerts"
```

For Linux:

```
keytool -importcert -alias hpcert -file <FILE PATH>/remoteServer.cert -keystore  
"$CSA_HOME/openjre/lib/security/cacerts"
```

Where, <FILE PATH> is the path where the "remoteServer.cert" file is copied to as in [step 2](#).

Note: If you are using an external JRE, then import the HPE UCMDB certificate into the keystore of the external JRE.

5. Restart CSA services after importing the certificate.

Once these steps are executed, create a new HPE UCMDB provider in CSA. If a HPE UCMDB provider is already configured, ensure the certificate is exported from the configured HPE UCMDB instance and imported into CSA.



Note: When designs having CI Type components are subscribed; during service provisioning if UCMDB is not reachable, the subscriptions will not fail. Subscriptions will be active however, the topology in UCMDB will not be created.

CSA Component Types aligned with HPE UCMDB's CI Types



The new component types are included in a palette called **HPE CSA (CI Types based)**. The component types in this palette are equivalent to similarly-named CI Types in the HPE UCMDB type system. The new component types cover most commonly used types in Compute, Network, Storage, Software and Load Balancer.

The table below describes the new CSA component types and how they are mapped to the equivalent CI Types in HPE UCMDB:






CI Types

Component Type	Description	Component Constraints	Resource Category Constraints
Configuration Item 	Root Component for all components derived from CI Type or equivalent.	None	None
Infrastructure Element 	Contains infrastructure service components such as, individual servers or server groups that are required in a service design.	None	None




CI Types, continued


Node 	Represents a general purpose machine and is a base component type from which other component types representing devices such as, Virtual Machines will be derived.	None	None
Server	Represents a general purpose machine which has an IP Address such as, Windows or Unix Servers.	Node Element Running Software	Application Compliance Compute Configuration Management IT Service Management Monitoring Service Usage Storage
CI Collection 	Generic logical collection of components that are of type - Infrastructure Element or Running Software.	None	None
Infrastructure Service	It contains a collection of infrastructure service components, such as Server Groups, that are needed in a service design.	Server Server Group	Compute Configuration Management Infrastructure Monitoring Network Storage
Server Group	Represents a server group which contains one or more servers.	Server Running Software	Compute Configuration Management Infrastructure Monitoring Network

CI Types, continued

			Storage
Running Software 	Represents the runtime aspects of a software that is running or is intended to be running on a server and is a base component type from which other component types representing various software will be derived.	None	Application Monitoring
Database 	Represents a database management system.	None	None
Oracle	Represents an Oracle database server.	None	None
MS SQL Server	Represents an MS SQL Server database server.	None	None
PostgreSQL	Represents a PostgreSQL database server.	None	None
Vertica	Represents a Verica Instance.	None	None
MySQL	Represents a MySQL database server.	None	None
Application Server 	Represents a server that hosts an API to expose business logic and business processes for use by third-party applications.	None	None
J2EE Server 	Represents a J2EE application server.	None	None
JBoss AS	Represents a JBoss application server.	None	None
Weblogic AS	Represents a Weblogic application server.	None	None
Load Balancer Software	Represents a component type for all Load Balancer software.	None	None
F5 BIG-IP LTM	Represents a F5 BIG-IP Local Traffic Manager software.	None	None
Web Server 	Represents a software that accepts http requests and delivers web pages as a	None	None

CI Types, continued

	response.		
Apache Web Server		None	None
Apache Tomcat	Represents an instance of Apache Tomcat web server.	None	None
IIS Web Server	Represents an instance of IIS Web Server.	None	None
Application Resource 	A base component type that is used to derive component types representing application resources.	None	None
WebService Resource 	A base component type used to derive component types that represent web service resources.	None	None
Web Service		None	None
Interface 	A component type that describes a logical interface.	None	None

 This indicates that the component type is not selected to be Designer Visible and therefore, the user cannot view these components in Service Designer while creating service designs. All the component types are system critical. The constraints defined for the new component types are same as the ones defined for older component types. Additional constraints can be defined as needed.

Designer Visible

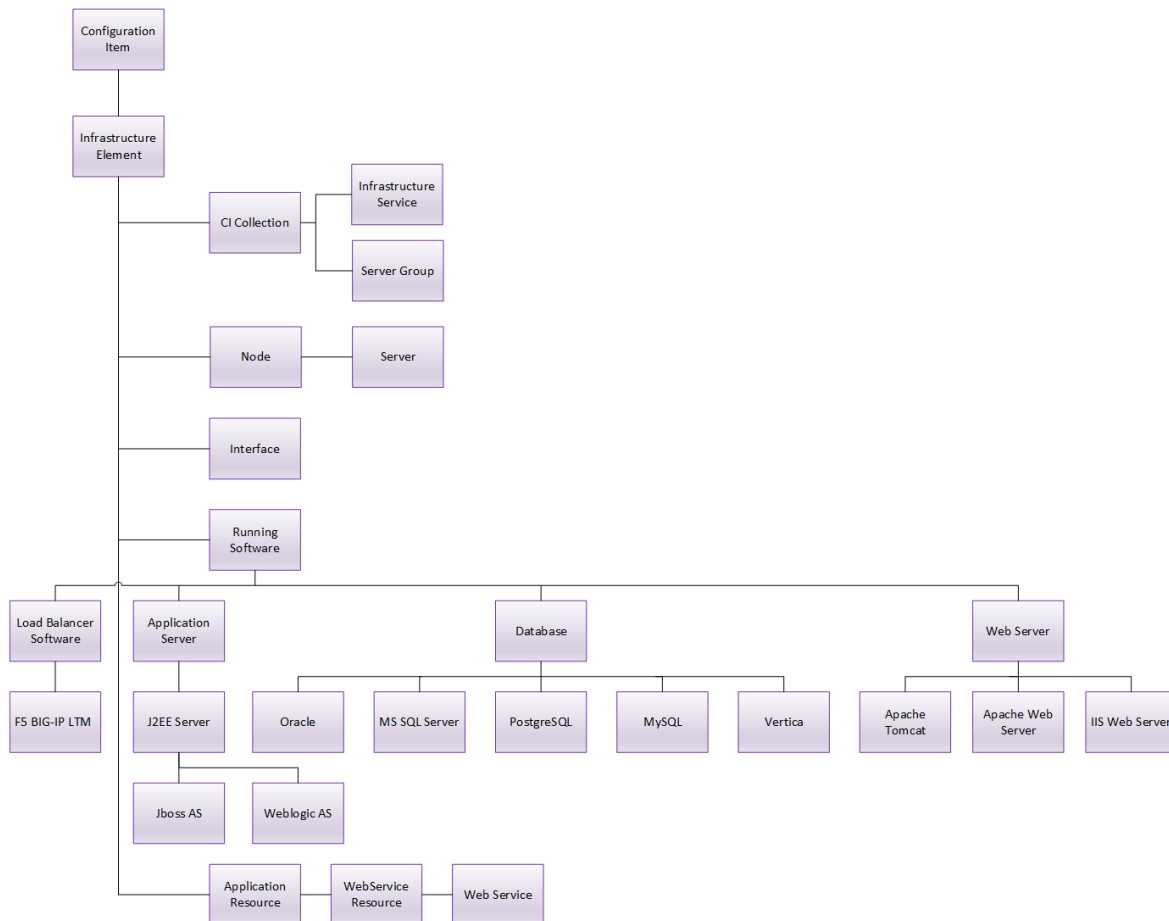
When the Designer Visible check box for a component type is selected, then that component type is concrete and can be used in service designs. Such component types are visible in the Service Designer during design creation. Component types that are not Designer Visible can be used as base component types and new component types that are concrete in nature can be created and used.

For example, the Server Group and Server component types are marked as Designer Visible. This means a designer will be able to see the components of this type in the service designer, and can use them as components in a sequence design.

Component types such as, Node, Node Element and Running Software are not visible in the service designer. Instead it is recommended to use more concrete sub-types of these component types such as, Server, Network Connection or Application Server. However, you can create custom component types using such component types as base component types only, and use the custom component types while creating sequence designs.

Note: Components created using abstract components as base components cannot be set as designer visible. By default, the **Designer Visible** check box will be disabled on selecting an abstract component as the base component.

CI Type Component Relationship



Creating a Custom CSA Component Type

This section describes the steps and prerequisites for creating a custom CSA component type.

Prerequisites:

The following are the prerequisites for defining the CI Type for a new component type:

- An HPE UCMDB provider must be configured. See, [Configure CSA for HPE UCMDB Integration](#) section for details.
- The CI Type must exist in the configured HPE UCMDB server.
- The CI Type of the base component type must be the parent of the CI Type set for the new component type, or it must be the same as the CI Type set for the new component type.

Note: The input for CI Type is not mandatory and can be left blank. When left blank, the CI Type of the base component type is used for the new component type.

- The UCMDB certificate must be imported into CSA's keystore if the UCMDB provider is configured to use secure HTTP for communication as given in ([step 1](#)).

Follow the below steps to create a custom CSA component type:

1. From the Cloud Service Automation Dashboard, select **Designs > Sequenced > Components**.

The **Sequenced Components** page is displayed.

2. Select **HPE CSA (CI Type) based** palette from the left panel > click the gear icon and select **Create Component Type**.

The **Create Component Type** dialog box is displayed.

3. Select a CSA component type as the Base Component Type.

Base Components that are mapped to CI Types in HPE UCMDB is indicated by the **CI Type** label displayed beside component type, as shown in the image below.






Component Types indicated with the Label CI Type

Select Component Type ? x

Select the component type to be used as the basis for the new component. Only base component types are shown which will not create a circular dependency with this palette.

HPE CSA (CI Type based) ▼

Search 🔍

-  **Apache Tomcat** CI Type
This component type represents an instance of Apache Tomcat web server.
-  **Apache Web Server** CI Type
This component type represents an instance of Apache Web Server.
-  **Application Resource** CI Type
A base component type that is used to derive component types representing application resources
-  **Application Server** CI Type
A component type that represents a server that hosts an API to expose business logic and business processes for use by third-party applications.
-  **CI Collection** CI Type
Generic logical collection of components that are of Infrastructure Element or Running Software type.

Done **Cancel**

Points to Note:

In HPE CSA (CI Type based) Palette

- You can only create CI Type based components from **HPE CSA (CI Type based)** palette. Though you can select other palettes from the **Select Component Type** dialog box, components for those palettes will not be displayed unless the component type is derived from a CI Type in HPE UCMDB.
- The **CI Type** field is visible in the **Create Component Type** dialog box by default.
- Selecting a Base Component Type is mandatory.
- When a Base Component Type is selected, the internal name of the selected Base Component Type is displayed in the CI Type field by default. However, you can change the CI Type to any other related CI Type component. But if the CI Type is not related, then the custom component type creation will fail. For example, if a custom component type for Windows-based Server is created, the CI Type should either be "host_node" or "nt". Instead if the CI Type is "disk_device" which is not related to "host_node" or "nt" the custom component type creation will fail.
- When the **CI Type** field is left blank, the selected Base Component is considered as the CI Type.

In palette other than HPE CSA (CI Type based)

- You can create components of all types including CI Type by selecting a CI Type component as the Base Component Type.
 - The **CI Type** field in the **Create Component Type** dialog box is not visible by default. Only on selecting a CI Type component as the Base Component, the **CI Type** field is visible.
 - Selecting Base Component Type is not mandatory for components other than CI Type.
4. Enter a name for the CI Type the new component type should be mapped to in HPE UCMDB. This CI Type must exist in HPE UCMDB else, the component type creation will fail.
 5. Enter a name and display name for the new component type.

Note: You must also have a valid HPE UCMDB provider defined in CSA as described in the section [Configure CSA for HPE UCMDB Integration](#).

6. Select the **Designer Visible** check box to set the component type to be visible and available for use in service designs.
7. Select the **Consumer Visible** check box to set the component type to be visible in the Marketplace Portal in Marketplace Portal.

8. Select the **Pattern** check box to set the component of this type to be accompanied by a Clone Pattern action in the parent service component.

The image below shows how a new CI Type component can be defined in CSA.

Create Component Type Dialog Box

Create Component Type ? x


Base Component Type *
Database ...

CI Type
database ?

Name *
ORACLE ?

Display Name *
Oracle

Description
Oracle Database

Image

[Change Image](#)

Consumer Visible ?

Create **Cancel**

User Administration

This chapter provides information for additional administration and configuration tasks.

Tasks include:

- "Allow Non-Administrator Users to Start and Stop the CSA, Marketplace Portal, or Global Search Service on Windows" below (optional)
- "Allow the CSA, Marketplace Portal, and Global Search Services to be Run as a Non-Administrator User on Windows" on page 201 (optional)
- "Change CSA Built-In User Accounts" on page 205 (optional)

Allow Non-Administrator Users to Start and Stop the CSA, Marketplace Portal, or Global Search Service on Windows

By default, only users with administrator privileges can start or stop the CSA, Marketplace Portal, and global search services. This procedure explains how to grant permissions to non-administrator users to start and stop these services. This process involves the following tasks:

- Create a non-administrator user account, if one does not exist.
- Update the security descriptor of the services.
- Change the permissions of the CSA installation directory for the non-administrator user.

To allow non-administrator users to start and stop the CSA, Marketplace Portal, or global search service, do the following:

1. Create a non-administrator user account:
 - a. Log in to the CSA system as administrator.
 - b. Navigate to **Start > Control Panel** on the CSA system and click **Add or remove user accounts** that is under **User Accounts**.
 - c. Click **Create a new account** in the Manage Accounts window.

- d. Enter a name for the user, select the **Standard user** radio button if it is not selected, and then click the **Create Account** button to create the user account.

2. Update the security descriptor of the services:

- a. Open a command prompt window and run the following command, as is applicable, to display the security descriptor for the CSA or HPE Marketplace Portal service:

For the CSA service: `sc sdshow csa`

For the HPE Marketplace Portal service: `sc sdshow hpemarketplaceportal.exe`

For the global search services:

- `sc sdshow hpesearchservice.exe`
- `sc sdshow elasticsearch-service-x64` or
`sc sdshow elasticsearch-service-x86`

The command returns a security descriptor in Security Descriptor Definition Language (SDDL), like the following example for the CSA service:

```
D: (A;;CCLCSWRPWPDTLOCRRC;;;SY) (A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)
(A;;CCLCSWLOCRRC;;;IU) (A;;CCLCSWLOCRRC;;;SU) S:
(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)
```

- b. Copy the security descriptor that was returned by the above command to a text editor such as Notepad.
- c. Run the following command to display the names and SIDs for all existing user accounts:

```
wmic useraccount get name,sid
```

- d. From the command output, copy the SID for the non-administrator user to the text editor.

The SID is usually in a format like S-1-5-21-3637136161-1358011849-3560387905-1014.

- e. Add (A;;RPWPCR;;;<SID of non-admin user>) before the S: (AU;... portion of the security descriptor that you copied to a text editor earlier in this procedure.

Using the security descriptor and SID from our example, the result would be as follows, with the added text highlighted in grey:

```
D: (A;;CCLCSWRPWPDTLOCRRC;;;SY) (A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)
(A;;CCLCSWLOCRRC;;;IU) (A;;CCLCSWLOCRRC;;;SU) (A;;RPWPCR;;;S-1-5-21-
3637136161-1358011849-3560387905-1014) S:
(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)
```

- f. Run the following command, as is applicable, to set the security descriptor for the CSA or

HPE Marketplace Portal service to the new value:

For the CSA service: `sc sdset csa "<new security descriptor>"`

For the HPE Marketplace Portal service:

`sc sdset hpemarketplaceportal.exe "<new security descriptor>"`

For the global search services:

- `sc sdset hpesearchservice.exe "<new security descriptor>"`
- `sc sdset elasticsearch-service-x64 "<new security descriptor>"` or
`sc sdset elasticsearch-service-x86 "<new security descriptor>"`

The message [SC] SetServiceObjectSecurity SUCCESS is returned if the command completes successfully.

Repeat this step for each user who will be allowed to start and stop the services.

3. Change the permissions of the CSA installation directory:

- In Windows Explorer, navigate to the CSA installation directory (for example, C:\Program Files\HPE\CSA), right-click on the folder, and select **Properties** in the menu that appears to open the CSA Properties dialog box.
- Click the **Security** tab in the CSA Properties dialog box.
- Check if the user is listed in the Group or user names list in the dialog box, and if it is not listed, continue with the next step. If it is listed, go to Step f to continue.
- Click the **Edit...** button, click the **Add...** button in the dialog box that appears, enter the non-administrator user name in the Enter the object names to select field, and then click the **Check Names** button.
- Select the name, and then click **OK** to add the user to the Group or user names list.
- Select the user name, select the **Allow** checkbox for the following permissions, and then click **OK**.
 - Read & execute
 - List folder contents
 - Read
 - Write

Log in to the CSA system using the non-administrator user account and start and stop the CSA, Marketplace Portal, and global search services.

Allow the CSA, Marketplace Portal, and Global Search Services to be Run as a Non-Administrator User on Windows

By default, the CSA, Marketplace Portal, and global search services are run as the service user. This section explains how to configure CSA so that these services can be run by non-administrator users. This process involves the following tasks:

- Create non-administrator users
- Configure the services
- Configure file system permissions

Caution: If the CSA, Marketplace Portal, and global search services are run as non-administrator users, you will not be able to do the following:

- Upgrade CSA
- Deploy hotfixes
- Install patches
- Use external tools such as the component tool, content archive tool, database purge tool, provider tool, schema installation tool, and support tool.
- Modify Autopass license data

Note: Certificates must be replaced and regenerated as the Administrator user.

Create Non-Administrator Users

The following tasks show how to create a non-administrator user account. You may choose to create a separate user for each service or one user to run all services. The examples in this section demonstrate how to run each service as a single and separate non-administrator user.

1. Log in as the Administrator.
2. Navigate to **Start > Control Panel** on the CSA system and click **Add or remove user accounts**

that is under **User Accounts**.

3. Click **Create a new account** in the Manage Accounts window that appears.
4. Enter a name for the user, select the **Standard user** radio button if it is not selected, and then click the **Create Account** button to create the user account.

Create three user accounts: CSAUser, MPPUser, and SearchUser.

Configure the Services

1. Log in as the Administrator.
2. Stop CSA. See "[Stop CSA](#)" on page 171 for instructions.
3. Back up and then delete the log files in the CSA_HOME\jboss-as\standalone\log\ directory.
4. Delete all files in the CSA_HOME\jboss-as\standalone\tmp\ directory.
5. Configure the CSA service to be run as CSAUser:
 - a. Navigate to **Start > Control Panel > Administrative Tools > Services**.
 - b. Right-click on the CSA service and select **Properties**.
 - c. Select the **Log On** tab.
 - d. Select **This account**.
 - e. In the first field, enter **CSAUser**.
 - f. Enter the password for CSAUser, confirm the password, and click **OK**.
6. Configure the HPE Marketplace Portal service to be run as MPPUser:
 - a. Navigate to **Start > Control Panel > Administrative Tools > Services**.
 - b. Right-click on the HPE Marketplace Portal service and select **Properties**.
 - c. Select the **Log On** tab.
 - d. Select **This account**.
 - e. In the first field, enter **MPPUser**.
 - f. Enter the password for MPPUser, confirm the password, and click **OK**.
7. Configure the Elasticsearch service to be run as SearchUser:

- a. Navigate to **Start > Control Panel > Administrative Tools > Services**.
- b. Right-click on the Elasticsearch service and select **Properties**.
- c. Select the **Log On** tab.
- d. Select **This account**.
- e. In the first field, enter **SearchUser**.
- f. Enter the password for SearchUser, confirm the password, and click **OK**.

Configure File System Permissions for the Non-Administrator Users

Assign permissions to each user for the specified directories in the CSA file system.

1. Log in as the Administrator.
2. Open the File Explorer.
3. For each of the directories listed in the following table, do the following (where C:\Program Files\HPE\CSA is the directory in which CSA has been installed):
 - a. Right-click on the directory and select **Properties**.
 - b. Click the **Security** tab.
 - c. Click **Edit**.
 - d. Select a user (CSAUser, MPPUser, or SearchUser) and select the permissions listed in the table.
 - e. Click **OK** to exit the Permissions dialog.
 - f. Click **OK** to exit the Properties dialog.

Directory	User(s)	Allowed Permission(s)
C:\	CSAUser MPPUser SearchUser	Full Control Modify Read & execute List folder contents Read Write

Directory	User(s)	Allowed Permission(s)
C:\Program Files\HPE	CSAUser MPPUser SearchUser	Full Control Modify Read & execute List folder contents Read Write
C:\Program Files\HPE\CSA	CSAUser MPPUser SearchUser	Full Control Modify Read & execute List folder contents Read Write
C:\Program Files\HPE\CSA\Autopass	CSAUser MPPUser	Full Control Read
C:\Program Files\HPE\CSA\CONTENT_IMPORT_LOGS	CSAUser	Write
C:\Program Files\HPE\CSA\csa-search-service	SearchUser	Read
C:\Program Files\HPE\CSA\csa-search-service\bin\daemon	SearchUser	Write
C:\Program Files\HPE\CSA\elasticsearch-1.6.1	SearchUser	Read
C:\Program Files\HPE\CSA\elasticsearch-1.6.1\logs	SearchUser	Write
C:\Program Files\HPE\CSA\jboss-as	CSAUser	Read
C:\Program Files\HPE\CSA\jboss-as\bin	CSAUser	Write
C:\Program Files\HPE\CSA\jboss-as\standalone	CSAUser	Write
C:\Program Files\HPE\CSA\jboss-as\standalone\deployments	CSAUser MPPUser SearchUser	Modify Read & execute List folder contents Read Write
C:\Program Files\HPE\CSA\jboss-as\standalone\configuration	CSAUser MPPUser SearchUser	Modify Read & execute List folder contents Read Write

Directory	User(s)	Allowed Permission(s)
C:\Program Files\HPE\CSA\node.js	MPPUser SearchUser	Read
C:\Program Files\HPE\CSA\openjre* *This is the JRE used by CSA. If you are using a different JRE, set the permissions to that JRE's directory.	CSAUser MPPUser SearchUser	Read & execute List folder contents Read Write
C:\Program Files\HPE\CSA\portal	MPPUser SearchUser	Read
C:\Program Files\HPE\CSA\portal\bin\daemon	MPPUser	Write
C:\Program Files\HPE\CSA\portal\logs	MPPUser	Write
C:\Program Files\HPE\CSA\scripts	CSAUser	Read
C:\Program Files\HPE\CSA\Tools	CSAUser	Read

4. Start CSA. See ["Start CSA" on page 168](#) for instructions.
5. Examine the `CSA_HOME\jboss-as\standalone\log\server.log` file and verify the changes deployed correctly.

Change CSA Built-In User Accounts

CSA ships with built-in user accounts. The user accounts are used to authenticate REST API calls and for initial setup and experimentation with the product. For security reasons, you may want to disable or change the passwords associated with these accounts (do not change the usernames).

Note: Do not create users in your LDAP directory that match the built-in users provided by CSA: `csaCatalogAggregationTransportUser`, `csaReportingUser`, `ooInboundUser`, and `codarintegrationUse`. Creating the same users in LDAP may allow the CSA built-in users unintended access to the Cloud Service Management Console or give the LDAP users unintended privileges.

Note: When you change or create passwords, you can use these special characters: `~ ` ! @ # $ % * () _ - + = { } [] \ / : ; [space] ?`

CSA does **not** support these characters: `^ & | " . > , <`

Cloud Service Management Console User Accounts

The following users are shipped with CSA and are used with the Cloud Service Management Console:

admin User: Cloud Service Management Console

Username	admin
Default Password	cloud
Default Role	ROLE_REST
Usage	This account is used to initially log in to the Cloud Service Management Console to configure the provider organization.
To Disable	<p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file. Update the <code>admin</code> property to disable this user account. For example, set <code>admin</code> to the following value (this value should be encrypted):</p> <pre>ccloud,DIAGNOSTICS_ADMIN,SUPPLIER_VIEWER,CONTENT_ADMIN,LICENSE_ADMIN,SUPER_IDM_ADMIN,ROLE_REST,disabled</pre> <p>Note: This property not only determines if the account is enabled, it also contains the password and the roles that control access to CSA.</p> <p>By default, the unencrypted value of this property is: <code>ccloud,ROLE_REST,enabled</code></p> <p>See "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p>
To Change Password	<p>If you change the password to this account, you must update the value of the password in the <code>provider-users.properties</code> file and the <code>securityAdminPassword</code> property in the <code>csa.properties</code> file (you must use the same password). You must also update and use the same password for every REST API call that uses the password.</p> <p>Updating the admin property in <code>provider-users.properties</code></p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file. Update the password portion of the <code>admin</code> value and encrypt the entire value, including the roles and account status (see "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p>

admin User: Cloud Service Management Console, continued

	<p>Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled. By default, the unencrypted value of this property is: <code>ccloud,ROLE_REST,enabled</code></p> <p>Updating the securityAdminPassword property in csa.properties</p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file (where <code>CSA_HOME</code> is the directory in which CSA is installed) and update the value of the <code>securityAdminPassword</code> property.</p> <p>Regenerate the newpassword for admin in the below format:</p> <p>For IDM Configuration (in file provider-users.properties) following value should be encrypted: <code>newpassword,DIAGNOSTICS_ADMIN,SUPPLIER_VIEWER,CONTENT_ADMIN,LICENSE_ADMIN,SUPER_IDM_ADMIN,ROLE_REST,enabled</code></p> <p>For csa.properties – Password should be encrypted as: “newpassword” Example: <code>encrypt "cloud"</code></p> <p>After modifying the <code>csa.properties</code> file, restart CSA. See "Restart CSA" on page 170 for instructions.</p>
--	---

consumerAdmin User: Marketplace Portal

Username	consumerAdmin
Default Password	cloud
Default Role	CONSUMER_ORGANIZATION_ADMINISTRATOR
Usage	This account is used to initially log in to the Cloud Service Management Console to configure and manage the sample CSA Consumer organization.
To Disable	<p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/consumer-users.properties</code> file. Update the <code>consumerAdmin</code> property to disable this user account. For example, set <code>consumerAdmin</code> to the following value (this value should be encrypted):</p> <p><code>ccloud,CONSUMER_ORGANIZATION_ADMINISTRATOR,disabled</code></p> <p>Note: This property not only determines if the account is enabled, it also contains the password and the roles that control access to CSA.</p> <p>By default, the unencrypted value of this property is: <code>ccloud,CONSUMER_ORGANIZATION_ADMINISTRATOR,enabled</code></p> <p>See "Encrypt a password" on page 172 for instructions about how to encrypt this</p>

consumerAdmin User: Marketplace Portal, continued

	<p>value). The encrypted value is preceded by ENC without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>For consumerAdmin (in file consumer-users.properties) following value should be encrypted:</p> <pre>newpassword,CONSUMER_ORGANIZATION_ADMINISTRATOR,enabled</pre>
To Change Password	<p>Edit the CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/consumer-users.properties file. Update the password portion of the consumerAdmin value and encrypt the entire value, including the roles and account status (see "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by ENC without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled.</p> <p>By default, the unencrypted value of this property is:</p> <pre>ccloud,CONSUMER_ORGANIZATION_ADMINISTRATOR,enabled</pre>

csaCatalogAggregationTransportUser User: Cloud Service Management Console

Username	csaCatalogAggregationTransportUser
Default Password	cloud
Usage	This account is used to authenticate REST API calls.
To Disable	Do not disable this account.
To Change Password	<p>If you change the password to this account, you must update the value of the securityCatalogAggregationTransportUserPassword property in csa.properties. You must also update the password using the catalog aggregation registration REST APIs.</p> <p>Edit the CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties file (where CSA_HOME is the directory in which CSA is installed) and update the value of the securityCatalogAggregationTransportUserPassword property. Determine a suitable new password (see "Encrypt a password" on page 172 for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>After modifying the csa.properties file, restart CSA. See "Restart CSA" on page 170 for instructions.</p>

csaReportingUser User: Cloud Service Management Console

Username	csaReportingUser
Default Password	cloud
Default Roles	ROLE_REST, ROLE_DYNAMIC
Usage	This account is used when a subscription is ordered or modified and a field for the subscription includes a dynamically generated list. The dynamically generated list is a subscriber option property configured to use a dynamic query. The dynamic query uses this account to access CSA to determine the values that will appear in the list. This account has read-only access to CSA .
To Disable	Do not disable this account.
To Change Password	<p>If you change the password to this account, you must update the value of the password in the <code>provider-users.properties</code> file and the <code>securityCsaReportingUserPassword</code> property in the <code>csa.properties</code> file (you must use the same password). You must also update and use the same password for every REST API call that uses the password.</p> <p>Updating the <code>csaReportingUser</code> property in <code>provider-users.properties</code></p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file. Update the password portion of the <code>csaReportingUser</code> value and encrypt the entire value, including the roles and account status (see "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled. By default, the unencrypted value of this property is: <code>cloud,ROLE_REST,ROLE_DYNAMIC,enabled</code></p> <p>Updating the <code>securityCsaReportingUserPassword</code> property in <code>csa.properties</code></p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file (where <code>CSA_HOME</code> is the directory in which CSA is installed) and update the value of the <code>securityCsaReportingUserPassword</code> property.</p> <p>Regenerate the newpassword for <code>csaReportingUser</code> in the below format:</p> <p>For IDM Configuration (in file provider-users.properties) following value should be encrypted: <code>newpassword,ROLE_REST,ROLE_DYNAMIC,enabled</code></p> <p>For csa.properties – Password should be encrypted as: “newpassword”</p>

csaReportingUser User: Cloud Service Management Console, continued

	<p>Example: encrypt "cloud"</p> <p>After modifying the <code>csa.properties</code> file, restart CSA. See "Restart CSA" on page 170 for instructions.</p>
--	---

csaTransportUser User: Cloud Service Management Console

Username	csaTransportUser
Default Password	csaTransportUser
Usage	This account is used to authenticate REST API calls.
To Disable	Do not disable this account.
To Change Password	<p>If you change the password to this account, you must update the value of the <code>securityTransportPassword</code> property in the <code>csa.properties</code> file and the <code>idm.csa.password</code> property in the <code>applicationContext.properties</code> file (you must use the same password). You must also update and use the same password for every REST API call that uses the password.</p> <p>Updating the <code>securityTransportPassword</code> property in <code>csa.properties</code></p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file (where <code>CSA_HOME</code> is the directory in which CSA is installed) and update the value of the <code>securityTransportPassword</code> property. Determine a suitable new password (see "Encrypt a password" on page 172 for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>Updating the <code>idm.csa.password</code> property in <code>applicationContext.properties</code></p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file and update the value of the <code>idm.csa.password</code> property.</p> <p>Regenerate the newpassword for <code>csaTransportUser</code> in the below format:</p> <p>For IDM Configuration (in file provider-users.properties) following value should be encrypted: <code>newpassword,ROLE_REST,enabled</code></p> <p>For csa.properties – Password should be encrypted as: "newpassword" Example: encrypt "cloud"</p> <p>After modifying and saving the changes to the files, restart CSA. See "Restart CSA" on page 170 for instructions.</p>

idmTransportUser User: Cloud Service Management Console

Username	idmTransportUser
Default Password	idmTransportUser
Default Roles	ROLE_ADMIN, PERM_IMPERSONATE
Usage	This account is used to authenticate REST API calls.
To Disable	Do not disable this account.
To Change Password	<p>If you change the password to this account, you must update the value of the <code>securityIdmTransportUserPassword</code> property in the <code>csa.properties</code> file, the <code>idmTransportUser</code> property in the <code>integrationusers.properties</code> file, and the <code>password</code> attribute in the <code>idmProvider</code> section of the <code>mpp.json</code> file (you must use the same password) and you must clear the JBoss server and web browser caches. You must also update and use the same password for every REST API call that uses the password.</p> <p>Updating the <code>securityIdmTransportUserPassword</code> property in <code>csa.properties</code></p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file (where <code>CSA_HOME</code> is the directory in which CSA is installed) and update the value of the <code>securityIdmTransportUserPassword</code> property. Determine a suitable new password (see "Encrypt a password" on page 172 for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>For <code>idmTransportUser</code> (in file <code>provider-users.properties</code>) following value should be encrypted:</p> <pre>newpassword,ROLE_ADMIN,PERM_IMPERSONATE,enabled</pre> <p>Updating the <code>idmTransportUser</code> property in <code>integrationusers.properties</code></p> <p>Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled.</p> <p>By default, the unencrypted value of this property is:</p> <pre>idmTransportUser,ROLE_ADMIN,PERM_IMPERSONATE,enabled</pre> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/integrationusers.properties</code> file and update the value of the <code>idmTransportUser</code> property. Use the same password that you used for the <code>securityIdmTransportUserPassword</code> property in the <code>csa.properties</code> file and encrypt the entire value of the <code>idmTransportUser</code> property, including the roles and account status (see "Encrypt a password" on page 172 for instructions about how to</p>

idmTransportUser User: Cloud Service Management Console, continued

encrypt this value). The encrypted value is preceded by ENC without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.

Updating the password attribute in mpp.json

Edit the `CSA_HOME/portal/conf/mpp.json` file (where `CSA_HOME` is the directory in which CSA is installed) and update the value of the `password` attribute in the `idmProvider` section and the `keyfile` attribute. Use the same password that you used for the `securityIdmTransportUserPassword` property in the `csa.properties` file and encrypt this password using the password utility that is provided by the Marketplace Portal:

1. Open a command prompt and navigate to the `CSA_HOME/portal/bin` directory. For example, on Windows the directory is `C:\Program Files\HPE\CSA\portal\bin` and on Linux the directory is `/usr/local/hpe/csa/portal/bin`.

2. Run the following command:

Windows:

```
..\..\node.js\node passwordUtil
```

Linux:

```
../..../node.js/node passwordUtil
```

When prompted, enter the name and location of the keyfile to generate (for example, `../conf/keyfile`) and the password to encrypt.

3. An encrypted password is displayed. Copy the encrypted password to the `password` attribute value in the `idmProvider` section. An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value. For example ENC (3oKr7eAo25bEn3Zn2t9wIA==)
4. Copy the keyfile name and location to the `keyfile` attribute.

Clearing the JBoss server and web browser caches

After modifying and saving the changes to the files, clear the JBoss server and web browser caches.

To clear the JBoss server cache, remove the contents from the `CSA_HOME/jboss-as/standalone/tmp` directory.

See ["Clear the web browser cache" on page 172](#) for information about how to clear the web browser cache.

Restarting CSA

After making these changes, restart CSA. See ["Restart CSA" on page 170](#) for instructions about how to restart CSA and the Marketplace Portal.

ooInboundUser User: Cloud Service Management Console

Username	ooInboundUser
Default Password	cloud
Default Role	ROLE_REST
Usage	This account is used by Operations Orchestration to authenticate REST API calls with CSA.
To Disable	Do not disable this account.
To Change Password	<p>If you change the password to this account, you must update the value of the password in the <code>provider-users.properties</code> file and the <code>securityOoInboundUserPassword</code> property in the <code>csa.properties</code> file (you must use the same password). You must also update and use the same password for every REST API call that uses the password.</p> <p>Updating the ooInboundUser property in provider-users.properties</p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file. Update the password portion of the <code>ooInboundUser</code> value and encrypt the entire value, including the roles and account status (see "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled.</p> <p>By default, the unencrypted value of this property is: <code>cloud,ROLE_REST,enabled</code></p> <p>You must also update and use the same password for the <code>CSA_REST_CREDENTIALS</code> system account in Operations Orchestration (located in the Configuration folder of the Public Repository).</p> <p>Updating the securityOoInboundUserPassword property in csa.properties</p> <p>If you change the password to this account, you must update the value of the <code>securityOoInboundUserPassword</code> property in <code>csa.properties</code>. You must also update and use the same password for the <code>CSA_REST_CREDENTIALS</code> system account in Operations Orchestration (located in the Configuration folder of the Public Repository).</p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file (where <code>CSA_HOME</code> is the directory in which CSA is installed) and update the value of the <code>securityOoInboundUserPassword</code> property. Use the same encrypted password that you entered for the <code>ooInboundUser</code> property in the <code>provider-users.properties</code> file.</p>

oolInboundUser User: Cloud Service Management Console, continued

	<p>Regenerate the newpassword for oolInboundUser in the below format:</p> <p>For IDM Configuration (in file provider-users.properties) following value should be encrypted: newpassword,ROLE_REST,enabled</p> <p>For csa.properties – Password should be encrypted as: "newpassword" Example: encrypt "cloud"</p> <p>After modifying the <code>csa.properties</code> file, restart CSA. See "Restart CSA" on page 170 for instructions.</p>
--	--

cdainboundUser User: Cloud Service Management Console

Username	cdainboundUser
Default Password	CDA2CSAIntegration!
Default Role	ROLE_REST
Usage	This account is used by Continuous Delivery Automation (Continuous Delivery Automation) to authenticate REST API calls with CSA.
To Disable	Do not disable this account.
To Change Password	<p>If you change the password to this account, you must update the value of the password in the <code>provider-users.properties</code> file and the <code>securityCdaInboundUserPassword</code> property in the <code>csa.properties</code> file (you must use the same password). You must also update and use the same password for every REST API call that uses the password.</p> <p>Updating the cdainboundUser property in provider-users.properties</p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file. Update the password portion of the <code>cdainboundUser</code> value and encrypt the entire value, including the roles and account status (see "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p> <p>Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled.</p> <p>By default, the unencrypted value of this property is: <code>CDA2CSAIntegration!,ROLE_REST,enabled</code></p> <p>Updating the securityCdaInboundUserPassword property in csa.properties</p> <p>If you change the password to this account, you must update the value of the <code>securityCdaInboundUserPassword</code> property in <code>csa.properties</code>. You must also</p>

cdaInboundUser User: Cloud Service Management Console, continued

	<p>update and use the same password in Continuous Delivery Automation.</p> <p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file (where <code>CSA_HOME</code> is the directory in which CSA is installed) and update the value of the <code>securityCdaInboundUserPassword</code> property. Use the same encrypted password that you entered for the <code>cdaInboundUser</code> property in the <code>provider-users.properties</code> file.</p> <p>After modifying the <code>csa.properties</code> file, restart CSA. See "Restart CSA" on page 170 for instructions.</p>
--	---

Marketplace Portal User Account

The following is a sample user that ships with CSA and is used to access the Marketplace Portal:

consumer User: Marketplace Portal

Username	consumer
Default Password	cloud
Default Roles	SERVICE_CONSUMER, ROLE_REST
Usage	This account is used to initially log in to and experiment with the Marketplace Portal (LDAP does not have to be configured). This user belongs to the "CSA consumer internal group" and is a member of the "CSA Consumer" organization (both the group and organization are provided as samples).
To Disable	<p>Edit the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/consumer-users.properties</code> file. Update the <code>consumer</code> property to disable this user account. For example, set <code>consumer</code> to the following value (this value should be encrypted):</p> <pre>ccloud,SERVICE_CONSUMER,ROLE_REST,disabled</pre> <p>Note: This property not only determines if the account is enabled, it also contains the password and the roles that control access to CSA.</p> <p>By default, the unencrypted value of this property is:</p> <pre>ccloud,SERVICE_CONSUMER,ROLE_REST,enabled</pre> <p>See "Encrypt a password" on page 172 for instructions about how to encrypt this value). The encrypted value is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.</p>

consumer User: Marketplace Portal, continued

To Change Password

Edit the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/consumer-users.properties` file. Update the password portion of the consumer value and encrypt the entire value, including the roles and account status (see ["Encrypt a password" on page 172](#) for instructions about how to encrypt this value). The encrypted value is preceded by ENC without any separating spaces and is enclosed in parentheses. Ensure there is no blank space at the end of the value.

For consumer (in file `consumer-users.properties`) following value should be encrypted:

```
newpassword,CONSUMER,SUPPORT,ROLE_REST,enabled
```

Note: This property not only contains the password, but also the roles that control access to CSA and if the account is enabled.

By default, the unencrypted value of this property is:
`cloud,SERVICE_CONSUMER,ROLE_REST,enabled`

LDAP Account Lockout Mechanism for the Cloud Service Management Console and Marketplace Portal

The account lockout mechanism is for LDAP users only. When an LDAP user attempts to log in to the Cloud Service Management Console or the Marketplace Portal, and enters the wrong password a specified number of times, the user account is locked out.

Note: There is no lockout error message. If you cannot log in, contact the administrator.

The lockout mechanism is configured on the LDAP server. See the LDAP vendor documentation for details about the lockout behavior and configuration.

Configure IPv6

This chapter explains how to configure CSA to support IPv6 (both dual-stack and IPv6-only). Make sure that IPv6 has been implemented on the system on which CSA is running (including configuring the network and DNS) and that your Web browser, such as Firefox or Chrome, have been enabled for IPv6 support.

To configure CSA to support IPv6, open `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` in a text editor and make the following changes:

1. Locate the following line:

```
<wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
```

and replace `127.0.0.1` with `:::1`. For example,

```
<wsdl-host>${jboss.bind.address:::1}</wsdl-host>
```

2. Locate the following lines:

```
<interface name="management">  
  <inet-address value="127.0.0.1" />  
</interface>
```

and replace `127.0.0.1` with `:::1`. For example,

```
<interface name="management">  
  <inet-address value=":::1" />  
</interface>
```

3. Locate the following lines:

```
<interface name="public">  
  <inet-address value="0.0.0.0" />  
</interface>
```

and replace `0.0.0.0` with `:::1`. For example,

```
<interface name="public">  
  <inet-address value=":::1" />  
</interface>
```

4. Locate the following lines:

```
<interface name="unsecure">  
  <inet-address value="{jboss.bind.address.unsecure:127.0.0.1}" />  
</interface>
```

and replace 127.0.0.1 with [::1]. For example,

```
<interface name="public">  
  <inet-address value="{jboss.bind.address.unsecure:[::1]}" />  
</interface>
```

To configure the Marketplace Portal to support IPv6, do the following:

- Open the `CSA_HOME\portal\conf\mpp.json` file in a text editor.
- In the general attribute section (for example, after the `uid` attribute), add a `bindIP` attribute and set the value to the IPv6 address to which the Marketplace Portal binds.
- Save and close the file.

To configure CSA tools (such as the purge tool, schema installation tool, provider tool, or content archive tool) to support IPv6, when configuring the `db.url`, `dbUrl`, or `jdbc.databaseUrl` attribute in the database file used by the tool (for example, `config.properties`, `jdbc.properties`, or `db.properties`), enclose the IPv6 address in square brackets (for example, `[f000:253c::9c10:b4b4]` or `[::1]`).

Launch the Cloud Service Management Console

Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`

Common Access Card

This chapter provides information about the integration between a Common Access Card (CAC) and CSA, where CAC is used as the user authentication mechanism. By configuring CAC, you are able to log into CSA using a Personal Identity Verification (PIV) card.

Caution: If you are configuring CSA on Windows to be compliant with FIPS 140-2, do NOT configure CAC before configuring CSA to be compliant with FIPS 140-2. If you have configured any feature before configuring CSA to be compliant with FIPS 140-2, you must re-install CSA.

After integrating CSA with CAC, the following log in rules apply:

- You can log in to the Cloud Service Management Console and the Marketplace Portal using a PIV card with a valid certificate.
- Log in to the Cloud Service Management Console and the Marketplace Portal using a CSA built-in user account without a PIV card.
- You can only log in to the Cloud Service Management Console and the Marketplace Portal as a valid LDAP user **with** a PIV card.

Caution: For the Cloud Service Management Console on Windows, in a standard environment (not a FIPS 140-2 compliant environment), only the JKS keystore type is supported for CAC. In a FIPS 140-2 compliant environment, only the PKCS #12 keystore type is supported for CAC.

Complete the following tasks to integrate CSA with CAC:

- ["Stop CSA" on the next page](#)
- ["Update JBoss Configuration to Set Up Client Authentication" on the next page](#)
- ["Configure the Identity Management component " on page 222](#)
- ["Configure Certificate Revocation" on page 227](#)
- ["Restart CSA" on page 229](#)

Stop CSA

If CSA is running, stop CSA. See ["Stop CSA" on page 171](#) for instructions.

Update JBoss Configuration to Set Up Client Authentication

1. Download the CA certificate for the digital certificate from the PIV card.
2. Import the CA certificate into a new truststore.

The truststore type is determined by the CSA environment. That is, if CSA is running in a standard environment, the truststore type must be JKS. If CSA is running in a FIPS 140-2 compliant environment, the truststore type must be PKCS #12.

For example, if you named the CA certificate from step 1 CACcert.cer, saved it in /tmp/ for Linux or \Temp\ for Windows, and want to create a truststore named CSA_HOME/jboss-as/standalone/configuration/.piv_keystore, run the following command:

Windows:

```
CSA_JRE_HOME\bin\keytool" -importcert -file C:\Temp\CACcert.cer -alias caccert  
-keystore CSA_HOME\jboss-as\standalone\configuration\.piv_keystore -storepass  
changeit
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -file /tmp/CACcert.cer -alias caccert  
-keystore CSA_HOME/jboss-as/standalone/configuration/.piv_keystore -storepass  
changeit
```

3. Edit the CSA_HOME/jboss-as/standalone/configuration/standalone.xml file:
 - a. Locate the <security-realm name="CsaRealm"> element. Within this element and after </server-identities>, add the following:

```
<authentication>  
  <truststore path="<Location of truststore>" keystore-password="<truststore  
password"/>  
</authentication>
```

For example,

Windows:

```
<security-realm name="CsaRealm">
  <server-identities>
    <ssl>
      <keystore keystore-password="changeit" path="C:\Program Files\HPE\CSA\jboss-
as/standalone/configuration/.keystore"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="C:\Program Files\HPE\CSA\jboss-as\
standalone\configuration\.piv_keystore" keystore-password="TruststorePassword"/>
  </authentication>
</security-realm>
```

Linux:

```
<security-realm name="CsaRealm">
  <server-identities>
    <ssl>
      <keystore keystore-password="changeit" path="/usr/local/hpe/jboss-
as/standalone/configuration/.keystore"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/usr/local/hpe/jboss-as/standalone/configuration/.piv_keystore"
keystore-password="TruststorePassword"/>
  </authentication>
</security-realm>
```

Note: This example stores the password in clear text. If you want to use an encrypted password, see ["Masking Passwords in standalone.xml Using the JBoss vault Script" on page 59](#) for information about creating a password vault for JBoss.

- b. Locate the `https-listener` element that contains the `name="https` and `security-realm="CsaRealm"` attributes. Add the `verify-client="REQUESTED"` attribute to this element.

For example,

```
<https-listener enabled-cipher-suites="TLS_ECDHE_ECDSA_WITH_AES_256_CBC_
SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384, ... " name="https" security-
realm="CsaRealm" socket-binding="https" verify-client="REQUESTED"/>
```

Configure the Identity Management component

To configure the Identity Management component, complete the following steps:

Note: If you wish to configure CAC without HP SSO, do the following in this order:

1. Follow the instructions to manually disable HP SSO, see "Disable HP Single Sign-On (HPSSO)" in ["Integrate with HP Single Sign-On" on page 230](#).
2. Continue to follow the steps below, but you should skip steps 4a and 4f. These steps are only relevant when HP SSO is used in CSA (HP SSO is enabled by default).

1. Extract the user name from the certificate using the username extraction mechanism.

The username extraction mechanism depends on the format of your certificate. The user name extracted from the certificate should match the user names configured in the LDAP configuration configured in CSA. CSA enables you to extract the user name using the **SubjectDN** and **Subject Alternative Name (SAN)** mechanisms. To configure the username extraction mechanism you must make the changes to the following properties in the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties` file:

Property	Description
<code>idm.cac.x509Attribute</code>	<p>The name of the X.509 certificate attribute from which the user name will be extracted.</p> <p>Set this property to <code>subjectDN</code> or <code>san</code> or <code>subjectDN,san</code>. If this property is set to contain both attributes such as <code>subjectDN,san</code> or <code>san,subjectDN</code>, then username will be extracted from the <code>subjectDN</code> attribute only if the SAN attribute is not present in the certificate. If this property is not set, then the default value for the property is "subjectDN".</p> <p>Uncomment this property <code>#idm.cac.x509Attribute=subjectDN</code></p>
<code>idm.cac.regex</code>	<p>The regular expression used to extract a user name from the <code>subjectDN</code> X.509 attribute. If this property is not set, then the default for <code>regex</code> is <code>CN=(.*)</code>. This property need not be set if the property <code>idm.cac.x509Attribute</code> is set to "san".</p> <p>Note: To retrieve the data between the parentheses from</p>

Property	Description
	the subjectDN X.509 attribute, use the filter <code>csa.cac.regex=\\((.*?)\\)</code> .
<code>idm.cac.san.type</code>	The type of the subject alternative name. The allowed types are <code>othername</code> and <code>rfc822name</code> . If this property is not set, then the default value for the property is <code>otherName</code> . This property need not be set if <code>idm.cac.x509Attribute</code> is set to <code>"subjectDN"</code> .
<code>idm.cac.default_tenant_org</code>	Name of the default organization to use with CAC if no organization is defined in the request. Uncomment this property <code>#idm.cac.default_tenant_org=CONSUMER</code>

2. Navigate to the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring` directory.
3. Make a backup copy of the `applicationContext-security.xml` file.
4. Edit the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-security.xml` file:
 - a. (Skip this step if HP SSO has been disabled manually.) If you are not using HP SSO, locate and uncomment the content below the line `START Certificate Authentication with subjectAlternativeName (with HP SSO)` so that it appears as follows:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-config="false">
  <security:http-basic />
  <security:csrf disabled="true" />
  <security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER" />
  <security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER" />
  <security:custom-filter ref="requestTokenCompositeFilter" position="FIRST"/>
  <security:custom-filter position="X509_FILTER" ref="cacX509AuthenticationFilter" />
  <security:custom-filter ref="cacFilter" before="LAST" />
  <security:custom-filter ref="noPromptFilter" position="LAST" />
</security:http>

<bean id="cacFilter" class="com.hp.ccue.identity.filter.certificate.CertificateFilter">
  <property name="generateTokenUtil" ref="generateTokenUtil" />
  <property name="tokenFactory" ref="tokenFactory" />
  <property name="tokenWriter" ref="hpssoTokenWriter" />
  <property name="loginRedirectionHandler" ref="loginRedirectionHandler" />
  <property name="authenticationFactory" ref="authnFactory" />
  <property name="persistenceService" ref="persistenceService"/>
  <property name="rolesPopulator" ref="csaRolesPopulator"/>
  <property name="userAndRepFactory" ref="ldapUserAndRepFactory"/>
  <property name="tenantFactory" ref="tenantFactory"/>
  <property name="defaultTenantOrganization" value="{idm.cac.default_tenant_org}" />
</bean>
```

- b. Locate the line `START Certificate Authentication (beans)` and uncomment the bean definitions below this comment so that it appears as follows:

Note: To retrieve the data between the parentheses for `<property name="regex" value="${idm.cac.regex:CN=(.*?),}" />`, use the filter `<property name="regex" value="${idm.cac.regex:\\((.*?)\\)" />`.

```
<!--START Certificate Authentication (beans) -->
    <bean id="cacX509AuthenticationFilter"

class="org.springframework.security.web.authentication.preauth.x509.X509Auth
enticationFilter">

        <property name="authenticationManager" ref="authManager" />
        <property name="principalExtractor" ref="customX509Extractor" />
    </bean>

    <bean id="customX509AttrPreAuthAuthProvider"

class="org.springframework.security.web.authentication.preauth.PreAuthentic
atedAuthenticationProvider">
        <property name="preAuthenticatedUserDetailsService"
ref="customAuthenticationUserDetailsService" />
    </bean>

    <bean id="customAuthenticationUserDetailsService"

class="org.springframework.security.core.userdetails.UserDetailsByNameServic
eWrapper">
        <property name="userService" ref="cacUserService" />
    </bean>

    <bean id="customX509Extractor"

class="com.hp.ccue.identity.filter.certificate.CustomX509PrincipalExtracto
r">
        <property name="x509Attribute"
value="${idm.cac.x509Attribute:subjectDN}" />
        <property name="regex" value="${idm.cac.regex:CN=(.*?),}" />
        <property name="sanType" value="${idm.cac.san.type:OtherName}" />
        <property name="UPNResolver" ref="userPrincipalNameResolver" />
    </bean>

<!-- Uncomment a userPrincipalNameResolver implementation for extracting the
user principal name -->
<!--
    <bean id="userPrincipalNameResolver"
```



```
class="com.hp.ccue.identity.filter.certificate.DefaultUserPrincipalNameExtra  
ctor" />  
-->  
  
    <bean id="userPrincipalNameResolver"  
class="com.hp.ccue.identity.filter.certificate.CsaBouncyCastleUpnExtractor"  
/>
```

- c. Locate the line `<security:authentication-provider ref="customX509AttrPreAuthAuthProvider"/>` and uncomment this line so that it appears as below:

```
<!-- START Certificate Authentication with subjectAlternativeName -->  
  
    <security:authentication-provider  
ref="customX509AttrPreAuthAuthProvider"/>  
  
<!-- END Certificate Authentication with subjectAlternativeName -->
```

- d. Locate the line `<!-- START Simplified Logout Configuration -->` and uncomment the section below the line so that it appears as follows:

```
<!-- START Simplified Logout Configuration -->  
  
    <security:http pattern="/idm/v0/logout" use-expressions="true" auto-  
config="false">  
        <security:csrf disabled="true" />  
        <security:custom-filter ref="simpleLogoutRedirect" position="FIRST"/>  
        <security:http-basic />  
    </security:http>  
  
<!--<security:http pattern="/idm/v0/logout/close" use-expressions="true"  
auto-config="false">  
    <security:csrf disabled="true" />  
    <security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_  
FILTER"/>  
    <security:http-basic />  
    </security:http-->  
  
<bean id="simpleLogoutRedirect"  
class="com.hp.ccue.identity.filter.RedirectFilter">
```

```
        <property name="url" value="/idm/v0/logout/close"/>
    </bean>
```

```
<!-- END Simplified Logout Configuration -->
```

- e. Locate the line `<!--START Certificate Authentication / SiteMinder SSO / HP SSO Configuration-->` and uncomment the section below this line so that it appears as follows:

```
<!-- START Certificate Authentication / SiteMinder SSO / HP SSO
Configuration -->
```

```
    <bean class="com.hp.ccue.identity.filter.LoginRedirectionHandler"
id="loginRedirectionHandler">
        <property name="tokenService" ref="tokenService"/>
    </bean>
```

```
    <bean class="com.hp.ccue.identity.utilities.GenerateResponseTokenUtil"
name="generateTokenUtil">
        <property name="tenantFactory" ref="tenantFactory"/>
        <property name="userFactory" ref="userFactory"/>
        <property name="authenticationResponseFactory"
ref="authenticationResponseFactory"/>
        <property name="roles">
            <list>
                <value>ROLE_REST</value>
            </list>
        </property>
    </bean>
```

```
<!-- END Certificate Authentication / SiteMinder SSO / HP SSO Configuration
-->
```

- f. (Skip this step if SSO has been disabled manually). Search for `START HP SSO ONLY Configuration` and comment out the section below:

```
<security:http auto-config="false" pattern="/idm/v0/login" use-
expressions="true">
    <security:csrf disabled="true"/>
    <security:custom-filter position="FIRST"
ref="requestTokenCompositeFilter"/>
    <security:custom-filter before="PRE_AUTH_FILTER"
ref="hpssoProvidedFilter"/>
    <security:custom-filter after="PRE_AUTH_FILTER"
ref="hpssoIntegrationFilter"/>
    <security:custom-filter before="FORM_LOGIN_FILTER"
ref="noPromptFilter"/>
```

```
    <security:http-basic/>
</security:http>

<security:http auto-config="false" pattern="/idm/v0/logout" use-
expressions="true">
    <security:csrf disabled="true"/>
    <security:custom-filter position="FIRST"
ref="requestTokenCompositeFilter"/>
    <security:custom-filter before="PRE_AUTH_FILTER"
ref="hpssoProvidedFilter"/>
    <security:custom-filter after="PRE_AUTH_FILTER"
ref="hpssoIntegrationFilter"/>
    <security:http-basic/>
</security:http>
```

5. Edit the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.xml` file:
 - a. Comment out `activeDirectoryAuthProvider` and `ldapAuthProvider` so that they appear as follows:

Note: Ignore this step if it is already done.

```
<bean id="multiTenantAuthProvider"
class="com.hp.ccue.identity.authn.MultiTenantAuthenticationProvider">
    <property name="providers">
        <list>
            <!-- <ref bean="activeDirectoryAuthProvider"/> -->
            <!-- <ref bean="ldapAuthProvider"/> -->
            <ref bean="seededAuthProvider"/>
        </list>
    </property>
    .....
</bean>
```

Configure Certificate Revocation

You will need to revoke a certificate if it has been compromised in any way or if an employee leaves your organization.

The following are the methods to revoke a certificate:

- Configure CSA to use a Certificate Revocation List (CRL)
- Configure CSA to Use a Certificate Revocation List Distribution Point (CRL DP)
- Configure CSA to Use the Online Certificate Status Protocol (OCSP)

Configure CSA to Use a Certificate Revocation List

The following is an example of how to revoke a certificate that was generated by the certificate authority and publish a Certificate Revocation List (CRL) that contains this certificate ID in the list. The CRL must already exist. You will download and save it in a folder on the system where CSA is installed and point to its location using the `ca-revocation-url` parameters.

1. Copy the CRL file to the system where CSA is installed (for example, copy it to the `<crl_file_directory>` directory).
2. In the `CSA_HOME\jboss-as\standalone\configuration\standalone.xml` file, add the `ca-revocation-url="<crl_file>"` attribute to the `<truststore path="<Location of truststore>" keystore-password="<truststore password>"/>` element.

For example, change the following from:

```
<authentication>
  <truststore path="<Location of truststore>" keystore-password="<truststore password>"/>
</authentication>
```

to:

```
<authentication>
  <truststore path="<Location of truststore>" keystore-password="<truststore password>" ca-
  revocation-url="<crl_file>"/>
</authentication>
```

3. Log in to the Cloud Service Management Console or the Marketplace Portal using a revoked certificate. The `Secure Connection Failed` message should display in the browser.

After restarting CSA (described below), you should log in to the Cloud Service Management Console or the Marketplace Portal using a revoked certificate. The `Secure Connection Failed` message should display in the browser.

Configure CSA to Use a Certificate Revocation List Distribution Point

To enable a Certificate Revocation List Distribution Point (CRL DP), edit the `CSA_HOME\jboss-as\standalone\configuration\standalone.xml` file and enable revocation and CRL DP by adding the following lines under `<system-properties>`:

```
<property name="com.sun.net.ssl.checkRevocation" value="true"/>
<property name="com.sun.security.enableCRLDP" value="true"/>
```

Configure CSA to Use the Online Certificate Status Protocol

To enable the Online Certificate Status Protocol (OCSP), do the following:

1. Edit the `CSA_HOME\jboss-as\standalone\configuration\standalone.xml` file and enable revocation by adding the following line under `<system-properties>`:
2. Edit the `CSA_JRE_HOME\lib\security\java.security` file and uncomment the following line (where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed):

```
ocsp.enable=true
```

Restart CSA

See ["Restart CSA" on page 170](#) for instructions.

Single Sign-On

This chapter provides information about integrating CSA with a single sign-on solution.

Tasks include:

- ["Integrate with HP Single Sign-On" below](#)
- ["Integrate CSA with a Single Sign-On Solution" on page 239](#)
- ["Integrate CSA with CA SiteMinder" on page 243](#)

Integrate with HP Single Sign-On

HP Single Sign-On (HP SSO) is included with CSA and can be used from the Cloud Service Management Console or Marketplace Portal when launching an application from the Cloud Service Management Console or Marketplace Portal. HP SSO must be installed and configured on the application before single sign-on can be integrated between it and CSA.

This guide provides details about how to integrate HP SSO between the Cloud Service Management Console and the following:

- OO
- HPE IT Business Analytics, see ["Enable the Cloud Analytics Side-Bar Menu" on page 109](#)
- HPE Enterprise Maps, see ["Enable the `\[\[\[Undefined variable CSAVariables.tabMigrateToCloud\]\]\] Secondary Tiles" on page 111`](#)
- Cloud Optimizer, see ["Configure the Cloud Optimizer Side-Bar Menu" on page 113](#)
- Workflow Designer, see [Workflow_Designer_SSO](#)

This guide also provides details about how to integrate HP SSO between the Marketplace Portal and HPE IT Business Analytics, see ["Configure the Showback Report Menu" on page 178](#).

You must configure a user (with the same name) for both CSA and the other application for single sign-on. You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and the application to use the same LDAP source or, if CSA and the application use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the appropriate role to access the tiles that launch the application and

the application user must be assigned a role that allows it to perform the expected functions in the application (for example, viewing flows or viewing reports).

Important: HP SSO is enabled by default for the Cloud Service Management Console and the Marketplace Portal.

HP Single Sign-On can be disabled. However, you cannot disable HP SSO only for CSA or only for Marketplace Portal. HP SSO can only be disabled for all applications, or not at all. See the next section "[Disable HP Single Sign-On \(HPSSO\)](#)" below for details.

Disable HP Single Sign-On (HPSSO)

If you intend to disable HP Single Sign-On for all applications, you must complete the following two tasks:

- ["Configure the Cloud Service Management Console" below](#)
- ["Configure the Identity Management component" on the next page](#)

Caution: If you do disable HP SSO, seamless login without prompting for a password between Marketplace Portal, CSA, OO, HPE IT Business Analytics, and any other application with LWSSO/HPSSO support will no longer work.

Configure the Cloud Service Management Console

To disable validation of the HP SSO token in the Cloud Service Management Console, complete the following steps:

1. Navigate to the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF` directory.
2. Make a backup copy of the `applicationContext-security.xml` file.
3. Open the `applicationContext-security.xml` file in a text editor.
4. Search for the `checkSSOCookie` bean and change the value of attribute `checkSSOCookie` from `true` to `false`.

```
<beans:bean id="tokenValidityFilter"  
class="com.hp.csa.security.TokenValidityFilter">  
<beans:property name="checkSSOCookie" value="false"/>  
</beans:bean>
```

5. Save and exit the file.

Configure the Identity Management component

To disable HP SSO for the Cloud Service Management Console and the Marketplace Portal, complete the following steps:

1. Navigate to the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF` directory.
2. Make a backup copy of the `web.xml` file.
3. Locate the comment `START HP SSO Configuration` and comment out the following content:

```
<listener>
<listener-class>com.hp.ccue.identity.hpssoImpl.HpSsoContextListener</listener-
class>
</listener>

<context-param>
<param-name>com.hp.sw.bto.ast.security.lwssso.conf.fileLocation</param-name>
  <param-value><CSA_HOME>/jboss-as/standalone/deployments/idm-
service.war/WEB-INF/hpssoConfig.xml
  </param-value>
</context-param>
```

4. Save and exit the file.
 5. Navigate to the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring` directory.
 6. Make a backup copy of the `applicationContext-v0.xml` file.
 7. Open the `applicationContext-v0.xml` file in a text editor.
 8. Locate the comment `START HP SSO Configuration` and comment out the following line:
- ```
<property name="tokenWriter" ref="hpssoTokenWriter"/>
```
9. Save and exit the file.
  10. Make a backup copy of the `applicationContext-security.xml` file in the same directory.
  11. Open the `applicationContext-security.xml` file in a text editor.
  12. Locate the comment `START HP SSO ONLY Configuration` and comment out the following content:

**Note:** If CAC, SAML or SiteMinder are configured, the security tags listed below should already be commented out).



```
<security:http auto-config="false" pattern="/idm/v0/login" use-
expressions="true">
<security:csrf disabled="true"/>
<security:custom-filter position="FIRST" ref="requestTokenCompositeFilter"/>
<security:custom-filter before="PRE_AUTH_FILTER" ref="hpssoProvidedFilter"/>
<security:custom-filter after="PRE_AUTH_FILTER" ref="hpssoIntegrationFilter"/>
<security:custom-filter before="FORM_LOGIN_FILTER" ref="noPromptFilter"/>
<security:http-basic/>
</security:http>
```

```
<security:http auto-config="false" pattern="/idm/v0/logout" use-
expressions="true">
<security:csrf disabled="true"/>
<security:custom-filter position="FIRST" ref="requestTokenCompositeFilter"/>
<security:custom-filter before="PRE_AUTH_FILTER" ref="hpssoProvidedFilter"/>
<security:custom-filter after="PRE_AUTH_FILTER" ref="hpssoIntegrationFilter"/>
<security:http-basic/>
</security:http>
```

```
<bean class="com.hp.ccue.identity.filter.hpsso.HpSsoFilter"
id="hpssoIntegrationFilter">
<property name="generateTokenUtil" ref="generateTokenUtil"/>
<property name="tokenFactory" ref="tokenFactory"/>
<property name="tenantFactory" ref="tenantFactory"/>
<property name="loginRedirectionHandler" ref="loginRedirectionHandler"/>
<property name="securityContextConverter" ref="securityContextConverter"/>
<property name="tokenWriter" ref="hpssoTokenWriter"/>
<property name="tokenService" ref="tokenService"/>
</bean>
```

```
<bean class="com.hp.ccue.identity.filter.hpsso.HpSsoFilter"
id="hpssoVerifyWithoutRedirectFilter">
<property name="generateTokenUtil" ref="generateTokenUtil"/>
<property name="tokenFactory" ref="tokenFactory"/>
<property name="tenantFactory" ref="tenantFactory"/>
<property name="redirectOnSuccess" value="false"/>
<property name="securityContextConverter" ref="securityContextConverter"/>
<property name="tokenWriter" ref="hpssoTokenWriter"/>
<property name="tokenService" ref="tokenService"/>
</bean>
```

```
<bean class="com.hp.ccue.identity.hpsso.SecurityContextConverter"
id="securityContextConverter">
<property name="tokenFactory" ref="tokenFactory"/>
<property name="tenantFactory" ref="tenantFactory"/>
<property name="provider" ref="multiTenantAuthProvider"/>
<property name="userFactory" ref="userFactory"/>
```

```
<property name="userService" ref="userService"/>
<property name="authenticationFactory" ref="authnFactory"/>
<property name="orgService" ref="organizationService"/>
</bean>

<bean class="com.hp.ccue.identity.hpsso.HpSsoCookieTokenWriter"
id="hpssoTokenWriter">
<property name="tokenStore" ref="tokenStore"/>
<property name="tokenService" ref="tokenService"/>
<property name="tokenFactory" ref="tokenFactory"/>
<property name="userService" ref="userService"/>
</bean>
```

13. If SAML is configured or you are about to configure it without HP SSO, do the following:

- a. Locate the comment `START SAML Web SSO with HP SSO` and verify that the following content is commented out. If not, comment it out:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
<security:csrf disabled="true" /><security:custom-filter
ref="requestTokenCompositeFilter" position="FIRST" />
<security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER"
/>
/>
<security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER"
/>
/>
<security:custom-filter ref="samlSsoFilter" before="CAS_FILTER" />
<security:custom-filter ref="noPromptFilter" before="FORM_LOGIN_FILTER" />
<security:http-basic />
</security:http>
```

```
<security:http pattern="/idm/v0/logout" use-expressions="true" auto-
config="false">
<security:csrf disabled="true" />
<security:custom-filter ref="requestTokenCompositeFilter" position="FIRST"
/>
/>
<security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER"
/>
/>
<security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER"
/>
/>
<security:custom-filter ref="samlSsoFilter" before="CAS_FILTER" />
<security:custom-filter ref="noPromptFilter" before="FORM_LOGIN_FILTER" />
<security:http-basic />
</security:http>
```

- b. Locate the comment `START SAML Web SSO without HP SSO` and uncomment the following content:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
<security:http-basic />
<security:csrf disabled="true" />
<security:custom-filter ref="requestTokenCompositeFilter" position="FIRST"/>
<security:custom-filter position="PRE_AUTH_FILTER" ref="samlSsoFilter" />
 <security:custom-filter ref="noPromptFilter" before="FORM_LOGIN_
FILTER" />
</security:http>
<security:http pattern="/idm/v0/logout" use-expressions="true" auto-
config="false">
<security:csrf disabled="true" />
<security:custom-filter ref="requestTokenCompositeFilter" position="FIRST"
/>
<security:custom-filter ref="samlSsoFilter" before="CAS_FILTER" />
<security:custom-filter ref="noPromptFilter" before="FORM_LOGIN_FILTER" />
<security:http-basic />
</security:http>
```

14. If CAC is configured or you are about to configure it without HP SSO, do the following:

- a. Locate the comment START Certificate Authentication with subjectAlternativeName (with HP SSO) and verify that the following content is commented out. If not, comment it out:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
 <security:http-basic />
 <security:csrf disabled="true" />
 <security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_
FILTER" />
 <security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_
FILTER" />
 <security:custom-filter ref="requestTokenCompositeFilter"
position="FIRST"/>
 <security:custom-filter position="X509_FILTER"
ref="cacX509AuthenticationFilter" />
 <security:custom-filter ref="cacFilter" before="LAST" />
 <security:custom-filter ref="noPromptFilter" position="LAST" />
</security:http>

<bean id="cacFilter"
class="com.hp.ccue.identity.filter.certificate.CertificateFilter">
 <property name="generateTokenUtil" ref="generateTokenUtil" />
 <property name="tokenFactory" ref="tokenFactory" />
 <property name="tokenWriter" ref="hpssoTokenWriter" />
 <property name="loginRedirectionHandler" ref="loginRedirectionHandler"
/>
```

```
<property name="authenticationFactory" ref="authnFactory" />
<property name="persistenceService" ref="persistenceService"/>
<property name="rolesPopulator" ref="csaRolesPopulator"/>
<property name="userAndRepFactory" ref="ldapUserAndRepFactory"/>
<property name="tenantFactory" ref="tenantFactory"/>
<property name="defaultTenantOrganization" value="{idm.cac.default_
tenant_org}" />
</bean>
```

- b. Locate the comment `START Certificate Authentication` with `subjectAlternativeName` (without HP SSO) and uncomment the following content:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
 <security:http-basic />
 <security:custom-filter ref="requestTokenCompositeFilter"
position="FIRST"/>
 <security:custom-filter position="LAST" ref="cacFilter" />
 <security:custom-filter position="X509_FILTER"
ref="cacX509AuthenticationFilter" />
</security:http>

<bean id="cacFilter"
class="com.hp.ccue.identity.filter.certificate.CertificateFilter">
 <property name="generateTokenUtil" ref="generateTokenUtil" />
 <property name="tokenFactory" ref="tokenFactory" />
 <property name="loginRedirectionHandler" ref="loginRedirectionHandler"
/>
 <property name="authenticationFactory" ref="authnFactory" />
 <property name="persistenceService" ref="persistenceService"/>
 <property name="rolesPopulator" ref="csaRolesPopulator"/>
 <property name="userAndRepFactory" ref="ldapUserAndRepFactory"/>
 <property name="tenantFactory" ref="tenantFactory"/>
 <property name="defaultTenantOrganization" value="{idm.cac.default_
tenant_org}" />
</bean>
```

15. If SiteMinder is configured or you are about to configure it without HP SSO, do the following:

- a. Locate the comment `START SiteMinder SSO` (with HP SSO) and verify that following content is commented out. If not, comment it out:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
<security:http-basic />
<security:csrf disabled="true" />
<security:custom-filter ref="requestTokenCompositeFilter" position="FIRST"
/>
<security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER"
```

```
 />
 <security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER"
 />
 <security:custom-filter ref="ssoHeaderFilter" before="CAS_FILTER" />
 <security:custom-filter ref="ssoFilter" before="LAST" />
 <security:custom-filter ref="noPromptFilter" position="LAST" />
 </security:http>

 <bean id="ssoFilter" class="com.hp.ccue.identity.filter.sso.SSOFilter">
 <property name="generateTokenUtil" ref="generateTokenUtil" />
 <property name="tokenFactory" ref="tokenFactory" />
 <property name="loginRedirectionHandler" ref="loginRedirectionHandler" />
 <property name="tokenWriter" ref="hpssoTokenWriter" />
 <property name="userAndRepFactory" ref="siteMinderUserAndRepFactory" />
 <property name="groupMembershipHeader" value="{idm.sso.group_membership_
header}"/>
 <property name="groupMembershipDelimiter" value="{idm.sso.group_membership_
delimiter}"/>
 <property name="defaultTenantOrganization" value="{idm.sso.default_tenant_
org}" />
 <property name="enableDefaultOrg" value="{idm.sso.enable_default_org}" />
 <property name="headerMetadataMap" ref="customHeaderMapping"/>
 </bean>

 <util:map id="customHeaderMapping" map-class="java.util.HashMap">
 <entry key="header1" value="metadataName1" />
 <entry key="header2" value="metadataName2" />
 </util:map>
```

- b. Locate the comment `START SiteMinder SSO (without HP SSO)` and uncomment the following content:

```
 <security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
 <security:http-basic />
 <security:csrf disabled="true" />
 <security:custom-filter ref="requestTokenCompositeFilter" position="FIRST"
 />
 <security:custom-filter ref="ssoHeaderFilter" position="PRE_AUTH_FILTER" />
 <security:custom-filter ref="ssoFilter" before="LAST" />
 <security:custom-filter ref="noPromptFilter" position="LAST" />
 </security:http>

 <bean id="ssoFilter" class="com.hp.ccue.identity.filter.sso.SSOFilter">
 <property name="generateTokenUtil" ref="generateTokenUtil" />
 <property name="tokenFactory" ref="tokenFactory" />
 <property name="loginRedirectionHandler" ref="loginRedirectionHandler" />
```

```
<property name="userAndRepFactory" ref="siteMinderUserAndRepFactory" />
</bean>
```

16. Save and exit the file.
17. Restart CSA. See ["Restart CSA" on page 170](#) for instructions.

## Workflow Designer Configuration - SSO

- a. Open the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file on first CSA node in text editor and locate the `initString` value in `crypto` element.

Example:

```
<crypto initString="2kDcHB0e0HrHcAGeArIPr7TNfuiVOpKqjj29SwkOQIoI"
cipherType="symmetricBlockCipher" engineName="AES"

paddingMode="CBC" keySize="256" encodingMode="Base64Url"
algorithmPaddingName="PKCS7Padding" checkIntegrity="disabled"

cryptoSource="lw" directKeyEncoded="false" directKeyEncoding="Hex"
jcePbeAlgorithmName="PBEWithHmacSHA1"

jcePbeMacAlgorithmName="PBEWithHmacSHA1" macAlgorithmName="SHA1"
macKeySize="256" macPbeCount="20" macType="hmac"

pbeCount="20" pbeDigestAlgorithm="SHA1"/>
```

If not already done, copy this `initString` to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` of all other CSA nodes.

Create encrypted version of `initString` with `encrypt-password` script. When prompted for password, provide the `initString` to the script:

```
sh# cd $CSA_HOME/workflow-designer/designer/bin
sh# ./encrypt_password

Password (typing will be hidden):
Confirm password (typing will be hidden):
{ENCRYPTED}xxts33/07Dtyz0iZ3e0QhzFVuqXvZ7KK6wDNm1A4E5+byAx1DZ+1HzwNRPvLgqXf
sh#
```

- b. Edit the file `CSA_HOME/workflow-designer/designer/var/securitysecured.properties` on every CSA node.

- c. Add/Edit the `lwssso.initString` property with the encrypted `initString` from `encrypt-password sript`:

```
#This is for limit the size of single CP upload, default 200MB
upload.max.fileSize.limit = 209715200

#This is for limit the number of parallel CP creation
max.parallel.cp.creation = 30

#This is for limit the number of parallel CP upload
max.parallel.cp.upload = 50

is.secured.cookie = true

lwssso.initString = {ENCRYPTED}
xxts33/07Dtyz0iZ3eOQhzFVuqXvZ7KK6wDNm1A4E5+byAx1DZ+1HzwNRPvLgqXf
```

- d. Edit file `CSA_HOME/workflow-designer/designer/tomcat/conf/server.xml` on every CSA node. Locate `Engine` element and add `jvmRoute` property with unique node name for every `workflow-designer` node:

Engine element in `server.xml` file:

```
<Engine defaultHost="localhost" name="Catalina">
```

On first node change to:

```
<Engine defaultHost="localhost" name="Catalina" jvmRoute="ood1">
```

On second node change to:

```
<Engine defaultHost="localhost" name="Catalina" jvmRoute="ood2">
```

...

- e. Restart `workflow-designer` on every node using command: `CSA_HOME/workflow-designer/designer/bin/designer restart`

## Integrate CSA with a Single Sign-On Solution

While CSA provides a Single Sign-On solution using CA SiteMinder, there are a variety of scenarios where you may need to perform the integration with CSA using another Single Sign-On solution. For example, you may be using:

- An implementation where you need to authenticate with an Single Sign-On vendor other than CA SiteMinder.
- A different deployment architecture than what is provided by CSA.
- A different version of CA SiteMinder than what is supported by CSA.
- An entirely different architecture than that which is supported.

In such cases it makes sense to create a custom Single Sign-On solution so that you can extend the provided implementation to your own.

For the Cloud Service Management Console and for the Marketplace Portal, Single Sign-On cannot be enabled at the same time as CAC.

## Verify the CSA Provider Organization's LDAP Server Configuration

You should verify that an LDAP user can log into the Cloud Service Management Console and the Marketplace Portal, which should already be configured. By performing this verification, you can be confident that any login issues that occur after integration have nothing to do with this particular configuration.

If there are any login issues, then update or configure the LDAP server for both the provider organization and the consumer organization from the Cloud Service Management Console, which is the interface from which you perform all administration tasks for *both* the Cloud Service Management Console and the Marketplace Portal.

**Note:** You must configure the CSA Provider organization to use the same LDAP server used by the custom SSO Server. If you do not configure this access point, no one will be able to access the Cloud Service Management Console.

To configure or update the provider organization's LDAP server:

1. Launch the Cloud Service Management Console by typing the following URL in a supported web browser: `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which the Cloud Service Management Console resides.

Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`

2. Log in to the Cloud Service Management Console as a CSA Administrator.



3. Click the **Administration** tile.
4. In the left-navigation frame, select the provider organization.
5. From the provider organization's navigation frame, select **LDAP**.
6. Update the LDAP server information.
7. Click **Save**.

## Verify the CSA Consumer Organization's LDAP Server Configuration

**Note:** The same LDAP server must be used by the CSA Provider organization, CSA consumer organization and custom SSO Server.

To configure or update the consumer organization's LDAP server:

1. Launch the Cloud Service Management Console by typing the following URL in a supported web browser: `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which the Cloud Service Management Console resides.  
  
Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`
2. Log in to the Cloud Service Management Console as the CSA Administrator.
3. Click the **Administration** tile.
4. In the left-navigation frame, select a consumer organization.
5. From the consumer organization's navigation frame, select **LDAP**.
6. Update the LDAP server information.
7. Click **Save**.
8. Repeat these steps for every consumer organization configured in CSA.

Only the `/csa` and `/mpp` contexts are supported (this is required by the SSO proxy setup).

## Configure the Custom SSO Server to Work with CSA

To configure your custom SSO server to work with CSA, follow the instructions provided with your SSO application.

### Stop CSA

See ["Stop CSA" on page 171](#) for instructions.

## Configure the Cloud Service Management Console

To configure the Cloud Service Management Console:

1. Update the `applicationContext-security.xml` file as appropriate for your custom SSO solution (based on the Spring Security Framework documentation).
2. Update the `csa.properties` file by uncommenting the string `enableSSO=true` and setting the value of `csa.subscriber.portal.url` to `{<protocol>}://{<host>}/mpp/org/{<orgName>}`.

## Configure the Marketplace Portal

To configure the Marketplace Portal:

1. Change `proxy` in the `mpp.json` file to the IP address of the proxy to be used by SSO. See the *Configure Proxy Mapping* section for details.
2. Update the `applicationContext-security.xml` file as appropriate for your custom SSO solution (based on the Spring Security Framework documentation).
3. Update the `applicationContext.xml` file as appropriate for your custom SSO solution (based on the Spring Security Framework documentation).

## Configure Proxy Mapping

To configure proxy mapping:

1. Map the `/csa` proxy to the CSA deployment.

**Caution:** Use only `/csa` as the alias. Using another alias may cause CSA to fail.

For example, when configuring the alias in an Apache proxy server, set the following:

```
ProxyPass /csa/ https://<csahostname>:8444/csa/
```

```
ProxyPassReverse /csa/ https://<csahostname>:8444/csa/
```

2. Map the `/idm-service` proxy to the Identity Management component deployment.
3. Map the `/mpp` proxy to the Marketplace Portal deployment.

## Start CSA

See ["Start CSA" on page 168](#) for instructions.

## Verify the Single Sign-On Integration

You should verify that the Single Sign-On integration works by logging into both the Cloud Service Management Console and the Marketplace Portal using the newly-integrated Single Sign-On solution.

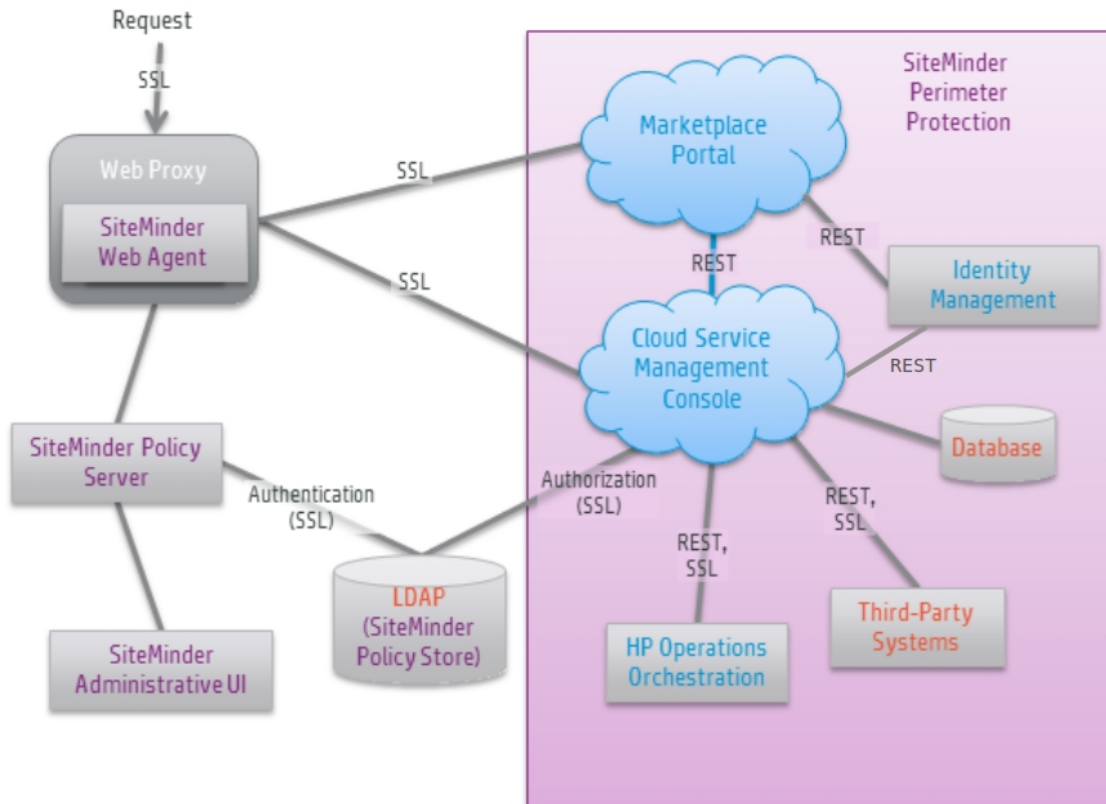
## Integrate CSA with CA SiteMinder

CSA, as well as SiteMinder (also called CA Single Sign-On) with a reverse proxy solution, must already be installed and configured before you can integrate them. The LDAP server shared by CSA and SiteMinder must be configured for the CSA provider and consumer organization (from the Cloud Service Management Console) before integration between CSA and SiteMinder is started.

SiteMinder is made up of several components that work with CSA and your LDAP server to provide secure access. The information provided in this section configures CSA to work with a reverse proxy solution, as shown in the following diagram.

**Note:** When CSA is integrated with SiteMinder, and CAC or SiteMinder is enabled, secondary authentication is not supported against keystone. In this case, Openstack providers cannot be used with SiteMinder. You must set `idm.keystone.enabled` to `false`.

### Supported CA SiteMinder Deployment Architecture Diagram



For more information about how to install and configure CA SiteMinder for a reverse proxy solution, refer to the *Configure Reverse Proxy Servers* section in the *Web Agent Configuration Guide* (a Web Agent guide). Documentation for SiteMinder can be found using the following URL:

<https://support.ca.com/irj/portal/anonymous/DocumentationSearch>

Complete the following steps to integrate CSA and SiteMinder:

- "Configure the CSA Provider Organization's LDAP Server" on the next page
- "Configure the CSA Consumer Organization's LDAP Server" on the next page
- "Configure the SiteMinder Policy Server for CSA Integration" on page 246
- "Configure the SiteMinder Web Agent for CSA Integration" on page 247
- "Configure CSA for SiteMinder Integration" on page 248

## Configure the CSA Provider Organization's LDAP Server

You must configure the CSA provider organization to use the same LDAP server used by the SiteMinder Policy Server. If you do not configure this access point before integrating CSA and SiteMinder, you will not be able to access CSA after integration.

**Caution:** LDAP must be configured for the CSA provider organization before you begin the integration between CSA and SiteMinder. After integrating CSA and SiteMinder, you can only log in to the Cloud Service Management Console via SiteMinder using a valid user from this LDAP directory. The CSA built-in users can no longer be used to log in to CSA.

When using the REST API, the built-in CSA users are still valid after integration.

To configure the provider organization's LDAP server, do the following:

1. Launch the Cloud Service Management Console by typing the following URL in a supported web browser: `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which the Cloud Service Management Console resides.

Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`

2. Log in to the Cloud Service Management Console as a CSA Administrator.
3. Click the **Administration** tile.
4. In the left-navigation frame, select the provider organization.
5. From the provider organization's navigation frame, select **LDAP**.
6. Update the LDAP server information.
7. Click **Save**.

## Configure the CSA Consumer Organization's LDAP Server

You must configure each CSA consumer organization to use the same LDAP server used by the SiteMinder Policy Server. If you do not configure this access point, no one will be able to access the

## Marketplace Portal.

To configure a consumer organization's LDAP server, do the following:

1. Launch the Cloud Service Management Console by typing the following URL in a supported web browser: `https://<csahostname>:8444/csa` where `<csahostname>` is the fully-qualified domain name of the system on which the Cloud Service Management Console resides.

Launch the Cloud Service Management Console using an IPv6 address by typing the following URL in a supported web browser: `https://<ipv6_address>:8444/csa/login`

2. Log in to the Cloud Service Management Console as the CSA Administrator.
3. Click the **Administration** tile.
4. In the left-navigation frame, select a consumer organization.
5. From the consumer organization's navigation frame, select **LDAP**.
6. Update the LDAP server information.
7. Click **Save**.
8. Repeat these steps for every consumer organization configured in CSA.

## Configure the SiteMinder Policy Server for CSA Integration

Complete the following steps to configure the SiteMinder Policy Server for CSA integration.

1. Navigate to **Control Panel > Administrative Tools > Services**.
2. Right-click on the HPE Marketplace Portal service and select **Stop**.
3. Configure the SiteMinder Policy Server to use the LDAP server that will be shared between CSA and SiteMinder.
4. Configure the SiteMinder Policy Server idle timeout, the Cloud Service Management Console session timeout, and the Marketplace Portal session timeout to be the same amount of time, regardless of the units (minutes or seconds) used by the parameters in the respective configuration files. By default, the session timeout value for the Cloud Service Management Console is 60 minutes, and for the Marketplace Portal, it is 1800 seconds.

The session timeout for the Cloud Service Management Console is configured using the `session-timeout` parameter in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/web.xml` file:

```
...
<session-config>
...
 <session-timeout>60</session-timeout>
...
```

5. To process image file names that contain spaces, from the SiteMinder Policy Server, either comment out the `BadUrlChars` parameter or modify the SiteMinder Policy Server to allow image file names that contain spaces.
6. Navigate to **Control Panel > Administrative Tools > Services**.
7. Right-click on the HPE Marketplace Portal service and select **Start**.

## Configure the SiteMinder Web Agent for CSA Integration

Configure proxy mapping for the SiteMinder Web Agent. To configure proxy mapping:

1. Map the `/csa` proxy to the CSA deployment.

**Caution:** Use only `/csa` as the alias. Using another alias may cause CSA to fail.

For example:

```
ProxyPass /csa/ https://<csahostname>:8444/csa/
ProxyPassReverse /csa/ https://<csahostname>:8444/csa/
```

2. Map the `/idm-service` proxy to the Identity Management component deployment. For example:

```
ProxyPass /idm-service/ https://<csahostname>:8444/idm-service/
ProxyPassReverse /idm-service/ https://<csahostname>:8444/idm-service/
```

3. Map the `/mpp` proxy to the Marketplace Portal deployment. For example:

```
ProxyPass /mpp/ https://<csahostname>:8090/mpp/
ProxyPassReverse /mpp/ https://<csahostname>:8090/mpp/
```

**Note:** The port number must match the value configured for the `port` attribute of the proxy

element in the `CSA_HOME/portal/conf/mpp.json` file. By default, this port is 8090.

If you are configuring a remote instance of the Marketplace Portal, use the hostname of the system on which the remote instance of the Marketplace Portal is installed.

## Configure CSA for SiteMinder Integration

To configure CSA for SiteMinder integration, you must:

- ["Stop CSA" below](#)
- ["Configure the Cloud Service Management Console" below](#)
- ["Configure the Marketplace Portal" on the next page](#)
- ["Configure the Identity Management component" on page 250](#)
- ["Start CSA" on page 253](#)

### Stop CSA

See ["Stop CSA" on page 171](#) for instructions.

## Configure the Cloud Service Management Console

Complete the following steps to configure the Cloud Service Management Console for a SiteMinder reverse proxy solution. Update the `applicationContext-security.xml` file:

1. Navigate to the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF` directory where  
`CSA_HOME` is the directory in which CSA is installed  
. For example:

**Windows:**

```
C:\Program Files\HPE\CSA\jboss-as\standalone\deployments\csa.war\WEB-INF
```

**Linux:**

```
/usr/local/hpe/csa/jboss-as/standalone/deployments/csa.war/WEB-INF
```

2. Make a backup copy of the `applicationContext-security.xml` file.



3. Open the `applicationContext-security.xml` file in a text editor.
4. Locate the comment `SiteMinder Configuration` and uncomment the following content:

```
<http pattern="/dashboard/index.jsp" use-expressions="true" entry-point-ref="idmEntryPoint">
 <intercept-url pattern="/dashboard/index.jsp" access="isAuthenticated()"/>
 <custom-filter ref="tokenValidityFilter" before="PRE_AUTH_FILTER" />
</http>
```

5. Locate the `<beans:constructor-arg value="/ssologout.jsp"/>` and uncomment.
6. In the same section of the file, comment out the following content (if it is not already commented out):

```
<beans:constructor-arg value="/logout.jsp"/>
```

7. Save and exit the file.
8. Navigate to the `classes` subdirectory (`CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes`).
9. Open the `csa.properties` file in a text editor.
10. Edit the following line to configure the URL to display for the organization in the Cloud Service Management Console:

```
csa.subscriber.portal.url={protocol}://{host}:8089/org/{orgName}
```

You can define a hard-coded URL or a URL that is replaced by information as known by the client-side browser. The following tokens are supported: `protocol` (`http` or `https`), `host` (the host in the browser URL used to access the Cloud Service Management Console), and `orgName` (the organization name of the selected organization in the browser). For example, if the client URL is `https://csa-server.company.com:8444/csa`, for a selected organization named `devteam`, then after the token replacement, the client displays a URL of `https://csa-server.company.com:8089/#/login/devteam`. No port is defined, and the `mpp` context is added to the URL. The context should be the same as is defined for the Marketplace Portal in the `mpp.json` file.

11. Locate the comment `Needed for SSO` and uncomment the following content:

```
enableSSO=true
```
12. Save and exit the file.

## Configure the Marketplace Portal

Complete the following steps to configure the Marketplace Portal for a SiteMinder reverse proxy solution.

1. Open the `CSA_HOME/portal/conf/mpp.json` file in a text editor.
2. In the `idmProvider` section, for `returnUrl`, change `proxy` to the IP address of the SiteMinder Web Agent proxy and add `redirectUrl` with its value set to the IP address of the SiteMinder Web Agent proxy:

```
"idmProvider": {

 "returnUrl": "https://{proxy}/mpp",
 "redirectUrl": "https://{proxy}",

}
```

For example:

```
"idmProvider": {

 "returnUrl": "https://101.32.24.101/mpp",
 "redirectUrl": "https://101.32.24.101",

}
```

3. Enable the proxy element to be used by the SiteMinder Web Agent by setting `enabled` to `true` as follows:

```
"proxy": {
 "enabled": true,
 "port": 8090,
 "contextPath": "/mpp"
}
```

To enable single sign-on for the Marketplace Portal, you must also set up proxy mapping on the SiteMinder Web Agent for the Marketplace Portal and for the Identity Management component service. The proxy mapping for the Marketplace Portal must use the same context name (`/mpp`) and port (`8090`) as defined here.

## Configure the Identity Management component

Complete the following steps to configure the Identity Management component for a SiteMinder reverse proxy solution.

**Note:** If you wish to configure SiteMinder without HP SSO, do the following in this order:

1. Follow the instructions to manually disable HP SSO, see "Disable Single Sign-On" in ["Integrate with HP Single Sign-On" on page 230](#).
2. Continue to follow the steps below, but you should skip steps 4, 6, and 9. These steps are only relevant when HP SSO is used in CSA (HP SSO is enabled by default).

1. Navigate to the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring directory`.
2. Make a backup copy of the `applicationContext-security.xml`, `applicationContext.xml` and `applicationContext.properties` files.
3. Open the `applicationContext-security.xml` file in a text editor.
4. (Skip this step if HP SSO has been disabled manually.) Locate the `START SiteMinder SSO (with HP SSO)` section and uncomment the following content:

```
<security:http pattern="/idm/v0/login" use-expressions="true" auto-config="false">
 <security:http-basic />
 <security:csrf disabled="true" />
 <security:custom-filter ref="requestTokenCompositeFilter" position="FIRST" />
 <security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER" />
 <security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER" />
 <security:custom-filter ref="ssoHeaderFilter" before="CAS_FILTER" />
 <security:custom-filter ref="ssoFilter" before="LAST" />
 <security:custom-filter ref="noPromptFilter" position="LAST" />
</security:http>

<bean id="ssoFilter" class="com.hp.ccue.identity.filter.sso.SSOFilter">
 <property name="generateTokenUtil" ref="generateTokenUtil" />
 <property name="tokenFactory" ref="tokenFactory" />
 <property name="loginRedirectionHandler" ref="loginRedirectionHandler" />
 <property name="tokenWriter" ref="hpssoTokenWriter" />
 <property name="userAndRepFactory" ref="siteMinderUserAndRepFactory" />
 <property name="groupMembershipHeader" value="${idm.sso.group_membership_header}"/>
 <property name="groupMembershipDelimiter" value="${idm.sso.group_membership_delimiter}"/>
 <property name="defaultTenantOrganization" value="${idm.sso.default_tenant_org}" />
 <property name="enableDefaultOrg" value="${idm.sso.enable_default_org}" />
 <property name="headerMetadataMap" ref="customHeaderMapping"/>
</bean>

<util:map id="customHeaderMapping" map-class="java.util.HashMap">
 <entry key="header1" value="metadataName1" />
 <entry key="header2" value="metadataName2" />
</util:map>
```

5. Locate the `START SiteMinder SSO (all)` section and uncomment the following content:

```
<security:authentication-manager id="ssoAuthManager">
 <security:authentication-provider ref="ssoAuthenticationProvider"/>
</security:authentication-manager>

<bean id="ssoHeaderFilter"
class="org.springframework.security.web.authentication.preauth.RequestHeaderAuthenticationFilter">
```

```
<property name="principalRequestHeader" value="${idm.sso.username_header}"/>
<property name="authenticationManager" ref="ssoAuthManager" />
<property name="exceptionIfHeaderMissing" value="true" />
</bean>
```

6. (Skip this step if HP SSO has been disabled manually.) Locate the START HP SSO ONLY Configuration section and comment out the following content:

```
<security:http auto-config="false" pattern="/idm/v0/login" use-expressions="true">
 <security:csrf disabled="true"/>
 <security:custom-filter position="FIRST" ref="requestTokenCompositeFilter"/>
 <security:custom-filter before="PRE_AUTH_FILTER" ref="hpssoProvidedFilter"/>
 <security:custom-filter after="PRE_AUTH_FILTER" ref="hpssoIntegrationFilter"/>
 <security:custom-filter before="FORM_LOGIN_FILTER" ref="noPromptFilter"/>
 <security:http-basic/>
</security:http>

<security:http auto-config="false" pattern="/idm/v0/logout" use-expressions="true">
 <security:csrf disabled="true"/>
 <security:custom-filter position="FIRST" ref="requestTokenCompositeFilter"/>
 <security:custom-filter before="PRE_AUTH_FILTER" ref="hpssoProvidedFilter"/>
 <security:custom-filter after="PRE_AUTH_FILTER" ref="hpssoIntegrationFilter"/>
 <security:http-basic/>
</security:http>
```

7. Locate the START Simplified Logout Configuration section and uncomment the following content:

```
<!-- START Simplified Logout Configuration -->
<!--
<security:http pattern="/idm/v0/logout" use-expressions="true" auto-
config="false">
 <security:csrf disabled="true" />
 <security:custom-filter ref="simpleLogoutRedirect" position="FIRST"/>
 <security:http-basic />
</security:http>

<security:http pattern="/idm/v0/logout/close" use-expressions="true" auto-
config="false">
 <security:csrf disabled="true" />
 <security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_
FILTER"/>
 <security:http-basic />
</security:http>

<bean id="simpleLogoutRedirect"
class="com.hp.ccue.identity.filter.RedirectFilter">
 <property name="url" value="/idm/v0/logout/close"/>
</bean>
<!-- END Simplified Logout Configuration -->
```

8. Open the applicationContext-v0.xml file in a text editor.
9. Open the applicationContext.xml file in a text editor.

10. Locate the START SiteMinder SSO Configuration section and uncomment the following content:

```
<bean id="ssoAuthenticationProvider"

class="org.springframework.security.web.authentication.preauth.PreAuthenticatedAuthenticatio
nProvider">
 <property name="preAuthenticatedUserDetailsService">
 <bean id="userDetailsServiceWrapper"
class="org.springframework.security.core.userdetails.UserDetailsServiceWrapper">
 <property name="userDetailsService" ref="ssoPreAuthenticatedUserDetailsService" />
 </bean>
 </property>
</bean>

<bean id="ssoPreAuthenticatedUserDetailsService"
class="com.hp.ccue.identity.filter.sso.SSOUserDetailsServiceImpl">
 <property name="restRole" value="ROLE_REST" />
</bean>
```

11. Open the `applicationContext.properties` file in a text editor.
12. Locate the property `idm.sso.username_header` and change its value to `SM_USER` :

```
idm.sso.username_header=SM_USER
```

## Start CSA

See ["Start CSA" on page 168](#) for instructions.

## Launch the Marketplace Portal

After completing the Marketplace Portal changes and restarting CSA, launch the Marketplace Portal using the URL: `https://<proxy_server_ip>/mpp/`. Depending on the Web agent configuration being used, a proxy server port *may* be required.

**Note:** If the single sign-on prompt appears multiple times when accessing the Marketplace Portal, you may need to [configure the Marketplace Portal to use the fully-qualified domain name of the SiteMinder Web Agent](#).

## Customize the Marketplace Portal Landing Page (Optional)

When accessing the Marketplace Portal during a single sign-on session, the user lands on the landing page displaying a button to be clicked to get to the Marketplace Portal dashboard. By default, the button is labeled "Log In." This might cause confusion as the authentication has already been completed using a single sign-on login prompt. To avoid this confusion, the label of the button can be modified:

1. Edit the `CSA_HOME/portal/node_modules/mpp-ui/dist/locales/<Locale>/rb.json` file. The location of the file depends on the locale being used. For example, for English, the file is `CSA_HOME/portal/node_modules/mpp-ui/dist/locales/en/rb.json`:

Modify the label of the login button. For example, to change the label to "Click to continue," make the following modification:

```
"login": {

 "login": "Click to continue",

}
```

2. Restart the HPE Marketplace Portal service:

### **Windows:**

- a. Navigate to **Control Panel > Administrative Tools > Services**.
- b. Right-click on the HPE Marketplace Portal service.
- c. Select **Restart**.

### **Linux:**

From a command prompt, type `service mpp restart`.

## Customize the Logout Page (Optional)

After clicking the Log out link from the Cloud Service Management Console or the Marketplace Portal, the user is directed to a logout page. This page is customizable.

The following is the name and location of the logout file. There is one file for the Cloud Service Management Console and another file for the Marketplace Portal.

- Cloud Service Management Console:

`CSA_HOME/jboss-as/standalone/deployments/csa.war/ssologout.jsp`

where

`CSA_HOME` is the directory in which CSA is installed

. For example:

**Windows:**

`C:\Program Files\HPE\CSA\jboss-as\standalone\deployments\csa.war\ssologout.jsp`

**Linux:**

`/usr/local/hpe/csa/jboss-as/standalone/deployments/csa.war/ssologout.jsp`

- Marketplace Portal:

`CSA_HOME/portal/node_modules/mpp-ui/dist/locales/en/rb.json`

where

`CSA_HOME` is the directory in which CSA is installed

. For example:

**Windows:**

`C:\Program Files\HPE\CSA\portal\node_modules\mpp-ui\dist\locales\en\rb.json`

**Linux:**

`/usr/local/hpe/csa/portal/node_modules/mpp-ui/dist/locales/en/rb.json`

In the above example, the `rb.json` file is for the English locale (language) and is therefore in the `en` folder.

You customize the logout message for your locale by modifying the `youAreOut` text. For example, for English locales, you can modify the text as follows:

```
"logout":{
 ...
 "youAreOut": "Please close your browser window. This prevents the
possibility of someone pressing the ''Back'' button on your browser and possibly
viewing confidential information.",
 ...
},
```

For other locales, modify the corresponding `rb.json` files.

**Note:** By default, after logging out, the user must close the Web browser to completely clear the

SiteMinder session.

The logout page can be customized to point to a SiteMinder logout page if one is available.

## Configure the Marketplace Portal to Use the Fully-Qualified Domain Name of the SiteMinder Web Agent (Optional)

The single sign-on prompt might appear multiple times when trying to access the Marketplace Portal when the domain name generated in the SiteMinder cookie (SMSESSION) does not match the address that is used to access the Marketplace Portal. If this problem occurs, do the following:

1. If the system (from which the browser that accesses the Marketplace Portal is launched) is unable to recognize the fully-qualified domain name of the SiteMinder Web Agent, update the system configuration to define an alias for the fully-qualified domain name to the IP address of the SiteMinder Web Agent. For example, define an alias in the host file.
2. On the system on which the Marketplace Portal is installed, do the following:
  - a. Update the following properties in the `CSA_HOME/portal/conf/mpp.json` file:

```
"idmProvider": {

 "returnUrl": "https://<FQDN_OF_SITE_MINDER_WEB_AGENT>/mpp",
 "redirectUrl": "https://<FQDN_OF_SITE_MINDER_WEB_AGENT>",

}
```
  - b. Update the system configuration to define an alias for the fully-qualified domain name to the IP address of the SiteMinder Web Agent. For example, define an alias in the host file.
  - c. Restart the system. Verify that the HPE Marketplace Portal service has restarted.
3. On the system on which CSA is installed, do the following:
  - a. Verify that the Organization URL (the URL used to access the Marketplace Portal) displayed in the Cloud Service Management Console uses the fully-qualified domain name of the SiteMinder Web Agent. To view the Organization URL, from the Cloud Service Management Console dashboard, select the Administration tile. In the left navigation frame, select the organization. In the organization's navigation frame, select **General Information**.
  - b. If the Organization URL does not use the fully-qualified domain name of the SiteMinder Web



- Agent, update the `csa.subscriber.portal.url` property in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file.
- c. If you updated the `csa.subscriber.portal.url` property, restart the CSA service:

**Windows:**

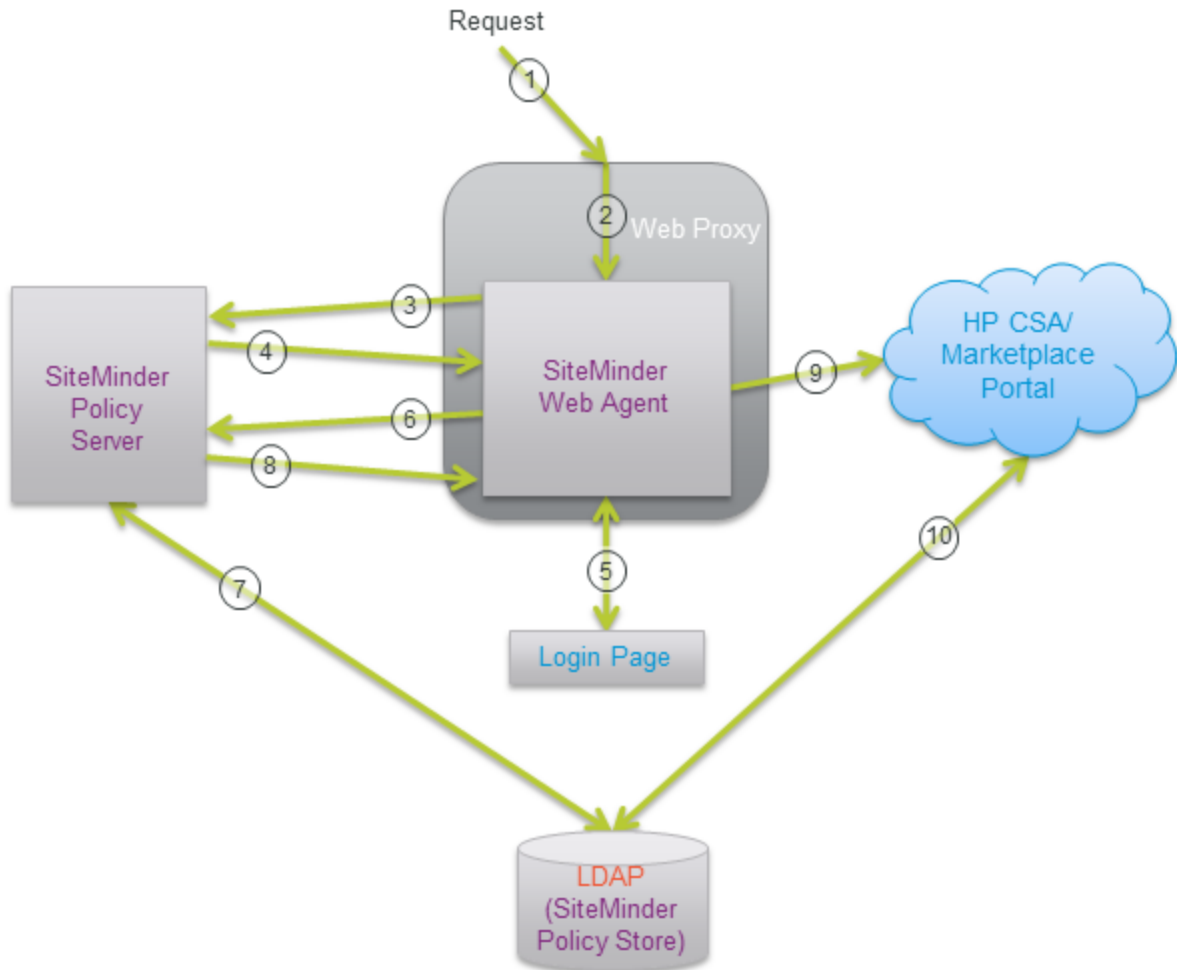
- i. Navigate to **Control Panel > Administrative Tools > Services**.
- ii. Right-click on the CSA service, and select **Restart**.

**Linux:**

From a command prompt, type `service csa restart`).

## Request Flow

The following diagram shows how a request is processed when CSA and SiteMinder are integrated.



1. A user sends a request to launch the Marketplace Portal.
2. The request is intercepted by the SiteMinder Web Agent.
3. The SiteMinder Web Agent queries the SiteMinder Policy Server to determine if it is a protected URL.
4. The SiteMinder Policy Server verifies that the URL is protected.
5. The user is redirected by the SiteMinder Web Agent to a login page where the user's credentials are collected.
6. The SiteMinder Web Agent sends the user's credentials to the SiteMinder Policy Server for authentication.
7. The SiteMinder Policy Server authenticates the user's credentials using the LDAP server (SiteMinder Policy Store).
8. The verification of the authenticated user is returned to the SiteMinder Web Agent.
9. The SiteMinder Web Agent redirects the user's request to launch the Marketplace Portal, which uses the Identity Management component to generate the necessary token.
10. CSA uses the token (included in the X-Auth-Token HTTP header) to perform the authorization. The name of the HTTP header may be different if you customized the `xAuthToken` configuration property in the `csa.properties` configuration file.

Additional requests from the user using the same SiteMinder session are automatically directed by the SiteMinder Web Agent to CSA.

## Configure SAML

CSA as a service provider can be configured to support SAML (Security Assertion Markup Language) by configuring the Identity Provider endpoint.

SAML is configured in the Cloud Service Management Console.

SAML is used for Federated Identity Management to implement scalable, secure, Single Sign-On across organizations.

**Note:** For supported SAML versions, see the *Cloud Service Automation System and Software Support Matrix*.

**Note:** In a SAML federated logout, CSA will clear the federated Identity Provider (IDP) sessions of the user.

## SAML Configuration on a CSA Fresh Install

Follow the below steps for SAML configuration on a CSA 4.7 Install:

**Note:** If you wish to configure SAML without HP SSO, do the following in this order:

1. Follow the instructions to manually disable HP SSO, see "Disable HP Single Sign-On (HPSSO)" in ["Integrate with HP Single Sign-On" on page 230](#).
2. Continue to follow the steps below, but you should skip steps 5 and 6. These steps are only relevant when HP SSO is used in CSA (HP SSO is enabled by default).

1. Navigate to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/web.xml`

`CSA_HOME` is the directory in which CSA is installed.

By default `CSA_HOME` is:

**Windows:** `C:\Program Files\HPE\CSA\`

**Linux:** `/usr/local/hpe/csa/`

2. Open **web.xml** file and uncomment the line:

```
/WEB-INF/spring/applicationContext-saml.xml
```

3. Perform the following step (step 3) only in a HA Cluster environment on two or more CSA nodes:

Open the file `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-saml.xml`

and uncomment the line

```
<property name="entityBaseURL" value="http://localhost/idm-service"/>
```

and replace with

```
<property name="entityBaseURL" value="https://<fqdn>:<port>/idm-service"/>
```

4. Navigate to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-security.xml`

5. (Skip this step if HP SSO has been disabled manually.) Uncomment the segment between the lines:

```
<!-- START SAML Web SSO with HP SSO --> and <!-- END SAML Web SSO with HP SSO -->
>:

<!-- START SAML Web SSO with HP SSO -->

<security:http pattern="/idm/v0/login" use-expressions="true" auto-
config="false">
<security:csrf disabled="true" />
<security:custom-filter ref="requestTokenCompositeFilter" position="FIRST" />
<security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER" />
<security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER" />
<security:custom-filter ref="samlSsoFilter" before="CAS_FILTER" />
<security:custom-filter ref="noPromptFilter" before="FORM_LOGIN_FILTER" />
<security:http-basic />
</security:http>
<security:http pattern="/idm/v0/logout" use-expressions="true" auto-
config="false">
<security:csrf disabled="true" />
<security:custom-filter ref="requestTokenCompositeFilter" position="FIRST" />
<security:custom-filter ref="hpssoProvidedFilter" before="PRE_AUTH_FILTER" />
<security:custom-filter ref="hpssoIntegrationFilter" after="PRE_AUTH_FILTER" />
<security:custom-filter ref="samlSsoFilter" before="CAS_FILTER" />
<security:custom-filter ref="noPromptFilter" before="FORM_LOGIN_FILTER" />
<security:http-basic />
</security:http>

<!-- END SAML Web SSO with HP SSO -->
```

6. (Skip this step if HP SSO has been disabled manually.) Comment out the segment below the line:

```
<!-- START HP SSO ONLY Configuration -->:

<!-- START HP SSO ONLY Configuration -->

<security:http auto-config="false" pattern="/idm/v0/login" use-
expressions="true">
<security:csrf disabled="true"/>
<security:custom-filter position="FIRST" ref="requestTokenCompositeFilter"/>
<security:custom-filter before="PRE_AUTH_FILTER" ref="hpssoProvidedFilter"/>
<security:custom-filter after="PRE_AUTH_FILTER" ref="hpssoIntegrationFilter"/>
<security:custom-filter before="FORM_LOGIN_FILTER" ref="noPromptFilter"/>
<security:http-basic/>
</security:http>
<security:http auto-config="false" pattern="/idm/v0/logout" use-
expressions="true">
<security:csrf disabled="true"/>
<security:custom-filter position="FIRST" ref="requestTokenCompositeFilter"/>
<security:custom-filter before="PRE_AUTH_FILTER" ref="hpssoProvidedFilter"/>
<security:custom-filter after="PRE_AUTH_FILTER" ref="hpssoIntegrationFilter"/>
```

```
<security:http-basic/>
</security:http>
```

7. Restart the CSA service. See "[Restart CSA](#)" on page 170 for instructions.

## ADFS Group Claim Configuration

### Pre-requisites:

Download CSA metadata and import to the ADFS machine.

**Note:** Identity Management component metadata can be downloaded from the following URL:  
<https://<CSA-FQDN>:8444/idm-service/saml/metadata>

This section contains two alternate ways to create a rule in Microsoft Server Manager to send group membership as a claim in Active Directory Federation Services (ADFS).

**Note:** If you change your **Group Name** in the Active Directory make sure you do not modify your **SAMAccount** name in the back end for the Group.

**Note:** If **SAMAccount** name is changed, the log in will be denied as the ADFS will return **SAMAccountname** for the Group configured.

### Rule creation method #1:

1. Open **ADFS Management** in the Server Manager.
2. In the console tree, navigate to **ADFS >Trust Relationships**.
3. Click **Relying Party Trusts** and select the trust where you will create the rule.
4. Right-click the trust and choose **Edit Claim Rules**.
5. In the **Edit Claim Rules** dialog, click **Add Rule**. The Add Transform Claim Rule Wizard is displayed.
6. In the **Select Rule Template** dialog, in the **Claim rule template** section, choose **Send Claims Using a Custom Rule**.
7. Enter a name for the rule.
8. In the **Custom Rule** box, enter the following rule syntax:

```
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer == "AD AUTHORITY"]=> issue(store = "Active Directory", types =
("Group"), query = ";tokenGroups;{0}", param = c.Value
```

9. Click **Finish** to apply the changes.

## Rule creation method #2:

Alternatively, you can add the rule using the following steps:

1. Open **ADFS Management** in the Server Manager.
2. In the console tree, navigate to **ADFS >Trust Relationships**.
3. Click **Relying Party Trusts** and select the trust where you will create the rule.
4. Right-click the trust party and choose **Edit Claim Rules**.
5. Click **Add Rule**. The Add Transform Claim Rule Wizard is displayed.
6. In the **Select Rule Template** dialog, in the **Claim rule template** section, select **Send LDAP Attributes as Claims**.
7. Enter a name for the rule.
8. Enter `active directory` in the **attribute stores** field.
9. For the LDAP attribute, select **Token-Groups – Unqualified Names**.
10. Create two rules.  
First rule, map **samAccountName** in **LDAP Attribute** (from the drop-down) to **NameID** in the **Outgoing Claim Type**.  
For second rule, for the outgoing claim type, specify any name but the same **Outgoing Claim Type** name should be used in the CSA SAML Configuration user interface under **Attribute** property.
11. Click **Finish**.

## Exporting the ADFS Certificate and Importing the Certificate in Identity Management component

1. Go to certificates in the **ADFS** wizard. Select the option to **Set Service Communications Certificate**.
2. Select the certificate with 5 years validity (5 years is the default validity period; this comes up when you are configuring Certificate Services).
3. Select copy to file. Certificate import wizard will open. Leave the default settings and save it in `CSA_HOME\jboss-as\standalone\deployments\idm-service.war\WEB-INF\classes\security`.
4. Save the backup of `samlKeystore.jks` from `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/security`
5. Import the certificate of Identity Provider (IDP) to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/security`

`CSA_HOME` is the directory in which CSA is installed.

By default `CSA_HOME` is:

**Windows:** `C:\Program Files\HPE\CSA\`

**Linux:** `/usr/local/hpe/csa/`

6. Run the following command from the above directory path:

```
keytool -import -alias adfs_alias -file your_certificate_name.cer -keystore
samlKeystore.jks
```

7. Provide the Destination store Identity Management component password.

**Note:** `nalle123` is the default password.

**Note:** Change the default password of `samlKeystore.jks`. After changing the password, update `idm.saml.keystore.password` in `applicationContext.properties`

## Adding SAML Configuration for the Organization

1. In the Cloud Service Management Console, navigate to **Organizations > Selected Organization > SAML**.
2. In the **SAML** tab of the organization, provide the Identity Provider metadata URL in the SAML URL field. The Identity Provider metadata URL can be a **web URL** OR a **relative directory path**.

- o For example:

**Web URL** for ADFS will be as shown in the format:

```
https://<hostname>/FederationMetadata/2007-06/FederationMetadata.xml
```

where <hostname> refers to the Identity Provider hostname.

- o For **relative directory path**, first create a **metadata** folder and then download metadata XML using the Identity Provider metadata URL and place the file to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/metadata`

`CSA_HOME` is the directory in which CSA is installed.

By default `CSA_HOME` is:

**Windows:** `C:\Program Files\HPE\CSA\`

**Linux:** `/usr/local/hpe/csa/`

Restart the CSA service and set the **SAML URL** as: `/metadata/FederationMetadata.xml`,

where `FederationMetadata` corresponds to the name of the metadata file.

3. The **attribute** name must be set to the name of Identity Provider attribute resolver.

**For Example:** For ADFS, the **attribute** will be Outgoing Claim type of a Rule in the **Relying Party Trust**. For instance, **Group**.

4. Configure LDAP for your organization (select **Organizations ->Selected Organization ->LDAP**).
5. In the access control add the group.

**NOTE:** Updating groups in access control is mandatory when groups are added before configuring SAML for an organization, irrespective whether it is fresh/upgrade setup.

**Note:** If CSA is configured for FIPS, see "*Configure CSA to Use SAML*" section in *Cloud Service Automation FIPS 140-2 Compliance Configuration Guide* for more information.



6. Download JCE Unlimited Strength Jurisdiction Policy Files from:  
<http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html>

7. Copy the `local_policy.jar` and `US_export_policy.jar` to `jre/lib/security` location.

For Example: If CSA is using OpenJre, then the path must be `CSA_HOME/openjre/lib/security/`

**Windows:** `C:\Program Files\HPE\CSA\openjre\lib\security\`

**Linux:** `/usr/local/hpe/csa/openjre/lib/security/`.

## SAML Configuration on a CSA Upgrade

**NOTE:** If SAML is configured in CSA 4.7 and you have upgraded to CSA 4.8, you need to configure SAML again.

To configure SAML when CSA is upgraded, complete the following steps:

1. Navigate to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/web.xml`

`CSA_HOME` is the directory in which CSA is installed.

For example, `CSA_HOME` on Windows is `C:\Program Files\HPE\CSA` and on Linux is `/usr/local/hpe/csa/`

2. Open **web.xml** file and add new entry `/WEB-INF/spring/applicationContext-saml.xml` below the line

`/WEB-INF/spring/applicationContext-common.xml`

3. Perform the following step (step 3) only in a HA Cluster environment on two or more CSA nodes:

Open the file `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-saml.xml`

and add the line `<property name="entityBaseURL" value="https://<fqdn>:<port>/idm-service"/>` next to

```
<!-- Filter automatically generates default SP metadata -->
<bean id="metadataGeneratorFilter"
class="org.springframework.security.saml.metadata.MetadataGeneratorFilter">
<constructor-arg>
<bean class="org.springframework.security.saml.metadata.MetadataGenerator">
```

# Database Administration

This chapter provides miscellaneous information about maintaining the database.

Tasks include:

- ["Restart the Database" below](#)
- ["Configure the CSA Reporting Database User" below](#)
- ["Update the CSA Database User or Password" on page 271](#) (required if you change the database user or password)
- ["Import Large Archives" on page 273](#)
- ["Purge Service Subscriptions and Audit Data" on page 277](#)
- ["Upgrade or Install a Fresh CSA Database Schema" on page 290](#)
- ["Configure CSA to Mitigate Frequently Dropped Database Connections" on page 297](#)

## Restart the Database

If you restart the database, you must restart the CSA service. If you do not restart the service, you may not be able to log in to the Cloud Service Management Console or Marketplace Portal.

**Note:** You only need to restart the CSA service. You do not need to restart the HPE Marketplace Portal service.

**To restart the service on Linux, on the server that hosts CSA, type the following:**

```
service csa start
```

## Configure the CSA Reporting Database User

This section explains how to configure the CSA reporting database user and role and run the schema installation script to define a read-only user required to use the reporting capabilities of CSA.

If you already configured the CSA reporting database user and role and defined the CSA reporting database user when running the installer or upgrade installer, you do not need to repeat these steps (the CSA reporting database user is already configured).

If you installed or upgraded CSA but did not configure the CSA reporting database user during the installation or upgrade and want to use the reporting capabilities of CSA, complete the tasks in this section.

## Create the CSAReportingDBUser

To configure the CSA reporting database user, do the following:

1. Create a read-only user.

**Caution:** The username cannot contain more than one dollar sign symbol (\$). For example, c\$adb is a valid name but c\$\$adb and c\$ad\$b are not valid names.

For example, do the following, based on the database you are using with CSA:

### Oracle

Run the following commands to create the CSAReportingDBRole role and CSAReportingDBUser user:

```
Create user CSAReportingDBUser identified by CSAReportingDBUser;
Create role CSAReportingDBRole;
Grant CREATE SESSION to CSAReportingDBUser;
Grant CSAReportingDBRole to CSAReportingDBUser;
Alter user CSAReportingDBUser default role CSAReportingDBRole;
```

You will also need to add the CREATE ANY SYNONYM privilege to the CSA database user. This allows the CSA database user to create synonyms for the CSA reporting (read-only) database user.

For example, if the CSA database user is named CSADBUser, run the following command:

```
Grant CREATE ANY SYNONYM to CSADBUser
```

### Microsoft SQL

Add a reporting database user (CSAReportingDBUser) to the CSA database with no roles:

```
CREATE LOGIN CSAReportingDBUser WITH PASSWORD = '<csareportingdbuser_
password>';
```

```
CREATE USER CSAReportingDBUser FOR LOGIN CSAReportingDBUser WITH DEFAULT_SCHEMA
= csa;
```

### PostgreSQL

From the psql prompt, enter the following:

```
CREATE ROLE CSAReportingDBUser LOGIN PASSWORD '<csareportingdbuser_password>'
NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT;
GRANT CONNECT ON DATABASE csadb to CSAReportingDBUser;
```

2. Run the following script:

### Oracle

```
CSA_HOME/scripts/reporting/oracle/grant-reporting-user.sql
```

### Microsoft SQL

```
CSA_HOME\scripts\reporting\mssql\grant-reporting-user.sql
```

### PostgreSQL

```
CSA_HOME/scripts/reporting/postgresql/grant-reporting-user.sql
```

3. Restart CSA. See ["Restart CSA" on page 170](#) for instructions.
4. The CSA reporting database user can access the data using the following view:

```
RPT_RSC_CAPACITY_V
```

## Edit the CSAReportingDBUser Password

To modify the password for the CSA reporting database user, do the following:

1. Run the database command to modify the CSAReportingDBUser password:

**Note:** See the database vendor documentation for the database user password requirements.

For example, do one of the following, based on the database you are using with CSA:

### Oracle:

```
$ALTER USER CSAReportingDBUser IDENTIFIED BY 'newpassword'
```

### Microsoft SQL:

```
$ALTER login CSAReportingDBUser WITH PASSWORD = 'newpassword' OLD_PASSWORD =
'oldpassword'
```

**PostgreSQL:**

```
$ALTER USER CSAReportingDBUser PASSWORD 'newpassword'
```

2. Restart CSA. See ["Restart CSA" on page 170](#) for instructions.

## Update the CSA Database System

If you changed the hostname, domain, IP address, or port of the system on which the database used by CSA is installed, you must update the CSA configuration files that store this information.

1. Stop the CSA service.

**To stop CSA on Windows, complete the following steps:**

- a. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the CSA service and select **Stop**.
- c. Right-click on the HPE Marketplace Portal service and select **Stop**.
- d. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Stop**.
- e. If you enabled global search, do the following:
  - i. Right-click on the Elasticsearch 1.6.1 service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
  - ii. Right-click on HPE Search Service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
- f. If Elasticsearch is enabled (by default, Elasticsearch is enabled; refer to the `csa.provider.es.exists` property in ["Appendix: Cloud Service Management Console Properties" on page 300](#) for more information), right-click on the Elasticsearch 1.6.1 service and select **Stop**.

**To stop CSA on Linux, complete the following steps:**

- a. On the server that hosts CSA, type the following commands:

```
service csa stop
service mpp stop
```

- b. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPE00installation>/central/bin/central stop
```

For example, type: `/usr/local/hpe/csa/00/central/bin/central stop`

2. On the system running CSA, open a command prompt and change to the `CSA_HOME/jboss-as/standalone/configuration` directory where `CSA_HOME` is the directory in which CSA is installed  
.
3. In a text editor, open the `standalone.xml` file.
4. In the file, locate the `<datasource>` element of the CSA database and the system information to be updated. For example:

### PostgreSQL

```
<datasource enabled="true" jndi-name="java:jboss/datasources/csaDS" jta="true"
pool-name="csaPostgresDS" use-ccm="true" user-java-context="true">
 <connection-url>jdbc:postgresql://127.0.0.1:5432/csadb</connection-url>
 <driver>pgsqlDriver</driver>
.
.
.
</datasource>
```

5. The highlighted text should contain the old fully-qualified domain name, IP address, and/or port that must be updated. Replace this highlighted text with the new fully-qualified domain name, IP address, and/or port.
6. Save the `standalone.xml` file.
7. Restart the CSA service.

See ["Restart CSA" on page 170](#) for instructions.

If you are using a tool (such as the content archive tool, provider tool, purge tool, or schema installation tool) that uses a database or configuration properties file (for example, `db.properties` or `config.properties`), update the appropriate property or properties in the file. By default, the file is located in the `CSA_HOME/Tools/<Tool_Name>` directory.

## Update the CSA Database User or Password

If you changed the user or password of the database used by HPE Cloud Service Automation, you must update the JBoss DataSource and other files that store this information.

1. On the system running CSA, open a command prompt and change to the directory `CSA_HOME/jboss-as` where `CSA_HOME` is the directory in which CSA is installed.  
.
2. Run the following command to generate an encoded version of the new database password:

### Windows:

```
"CSA_JRE_HOME\bin\java" -cp
modules\system\layers\base\org\picketbox\main\picketbox-4.0.21.Final.jar
org.picketbox.datasource.security.SecureIdentityLoginModule <password>
```

### Linux:

```
CSA_JRE_HOME/bin/java -cp
modules/system/layers/base/org/picketbox/main/picketbox-4.0.21.Final.jar
org.picketbox.datasource.security.SecureIdentityLoginModule <password>
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

Copy the encoded password value that is returned (do not include spaces).

3. Stop the CSA service. See ["Stop CSA" on page 171](#) for instructions.
4. In a text editor, open the `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` file.
5. In the file, locate the following content:

### Microsoft SQL Server

```
<security-domain name="csa-encryption-sec" cache-type="default">
 <authentication>
 <login-module
 code="org.picketbox.datasource.security.SecureIdentityLoginModule"
 flag="required">
 <module-option name="username" value="<old_user_name>"/>
 <module-option name="password" value="<old_encoded_password>"/>
 <module-option name="managedConnectionFactoryName"
```

```
value="jboss.jca:service=LocalTxCM,name=mssqlDS"/>
 </login-module>
</authentication>
</security-domain>
```

### Oracle

```
<security-domain name="csa-encryption-sec" cache-type="default">
 <authentication>
 <login-module
code="org.picketbox.datasource.security.SecureIdentityLoginModule"
flag="required">
 <module-option name="username" value="<old_user_name>"/>
 <module-option name="password" value="<old_encoded_password>"/>
 <module-option name="managedConnectionFactoryName"
value="jboss.jca:service=LocalTxCM,name=OracleDS"/>
 </login-module>
 </authentication>
</security-domain>
```

### PostgreSQL

```
<security-domain name="csa-encryption-sec" cache-type="default">
 <authentication>
 <login-module
code="org.picketbox.datasource.security.SecureIdentityLoginModule"
flag="required">
 <module-option name="username" value="<old_user_name>"/>
 <module-option name="password" value="<old_encoded_password>"/>
 <module-option name="managedConnectionFactoryName"
value="jboss.jca:service=LocalTxCM,name=PostgresDS"/>
 </login-module>
 </authentication>
</security-domain>
```

6. Replace `<old_encoded_password>` with the new encoded password you copied in step 2 and `<old_user_name>` with the new user name.
7. Save the `standalone.xml` file.
8. Restart CSA service. See ["Restart CSA" on page 170](#) for instructions.
9. If you are using a tool (such as the content archive tool, provider tool, purge tool, or schema



installation tool) that uses a database or configuration properties file (for example, `db.properties` or `config.properties`), update the appropriate property or properties in the file. By default, the file is located in the `CSA_HOME/Tools/<Tool_Name>` directory.

The password property value should be *encrypted* (see ["Encrypt a password" on page 172](#) for instructions). An encrypted password is preceded by `ENC` without any separating spaces and is enclosed in parentheses.

## Import Large Archives

Archives exported from CSA can be imported to install artifacts or update existing artifacts in CSA. Archives can be imported using the CSA Content Archive Tool, the Cloud Service Management Console, or the REST API.

The default configuration for importing archives supports an archive up to 2 MB in size. When an archive larger than 2 MB is imported (typically, a catalog), the import operation may hang or take a very long time to complete. If an archive is larger than 2 MB, it is recommended that you use the Content Archive Tool and increasing the JVM heap size.

## Import Large Archives Using the CSA Content Archive Tool

If you want to import an archive larger than 2 MB, it is recommended that you use the Content Archive Tool because the tool uses its own JVM heap (it does not share the JVM heap used by CSA). When you reconfigure the JVM heap size for the tool, you do not need to restart CSA and CSA performance is not affected by the import.

To increase the JVM heap size when running the Content Archive Tool, add the `-Xms<heap_size>M -Xmx<heap_size>M` options to the command line. For example, to increase the JVM heap size to 3 GB, type:

### Windows:

```
"CSA_JRE_HOME\bin\java -Xms3072M -Xmx3072M -jar content-archive-tool.jar -i -z catalog_archive.zip
```

### Linux:

```
CSA_JRE_HOME/bin/java -Xms3072M -Xmx3072M -jar content-archive-tool.jar -i -z
catalog_archive.zip
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

**Note:** By default, the JVM heap size used by the Content Archive Tool is 2 GB. If you want to use a larger JVM heap size, you must always specify the two options listed above when running the Content Archive Tool.

For more information about the Content Archive Tool, refer to the *CSA Content Archive Tool* guide.

## Import Large Archives from the Cloud Service Management Console or through the REST API

If you want to import an archive larger than 2 MB, it is recommended that you use the Content Archive Tool. If you must use the Cloud Service Management Console or REST API to import a large archive, you must update the JVM heap size for CSA which requires CSA to be restarted. Also, importing a large archive from the Cloud Service Management Console or through the REST API may slow the performance of CSA.

To increase the JVM heap size before importing a large archive from the Cloud Service Management Console or through the REST API, do the following:

1. Stop CSA.

**To stop CSA on Windows, complete the following steps:**

- a. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the CSA service and select **Stop**.
- c. Right-click on the HPE Marketplace Portal service and select **Stop**.
- d. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Stop**.
- e. If you enabled global search, do the following:
  - i. Right-click on the Elasticsearch 1.6.1 service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
  - ii. Right-click on HPE Search Service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).

- f. If Elasticsearch is enabled (by default, Elasticsearch is enabled; refer to the `csa.provider.es.exists` property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 for more information), right-click on the Elasticsearch 1.6.1 service and select **Stop**.

**To stop CSA on Linux, complete the following steps:**

- a. On the server that hosts CSA, type the following commands:

```
service csa stop
service mpp stop
```

- b. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPE00installation>/central/bin/central stop
```

For example, type: `/usr/local/hpe/csa/00/central/bin/central stop`

2. Increase the JVM heap size for CSA.

- a. Open the `CSA_HOME/jboss-as/bin/standalone.conf.bat` file in a text editor.

- b. Locate the following line:

**Windows:**

```
set "JAVA_OPTS=%JAVA_OPTS% -Xms2048M -Xmx2048M -
XX:ReservedCodeCacheSize=256M"
```

**Linux:**

```
$JAVA_OPT -Xms2048M -Xmx2048M -XX:ReservedCodeCacheSize=256M"
```

- c. Increase the JVM heap size (by default, the JVM heap size is 1 GB). For example, to change the JVM heap size to 3 GB, change the line to:

**Windows:**

```
set "JAVA_OPTS=%JAVA_OPTS% -Xms3072M -Xmx3072M -
XX:ReservedCodeCacheSize=256M"
```

**Linux:**

```
$JAVA_OPT -Xms3072M -Xmx3072M -XX:ReservedCodeCacheSize=256M"
```

- d. Save and close the file.

3. Start CSA. See "[Start CSA](#)" on page 168 for instructions.

**To start CSA on Windows, complete the following steps:**

- a. If you have configured CSA to be FIPS 140-2 compliant, create a CSA encryption keystore password file. The name and location of this file must match the value configured for the `keystorePasswordFile` property in the `CSA_HOME\jboss-as\standalone\deployments\csa.war\WEB-INF\classes\csa.properties` file.

The password file must contain only the following content: `keystorePassword=<CSA encryption keystore password>`

where `<CSA encryption keystore password>` is the CSA encryption keystore password in clear text.

This file is automatically deleted when the HPE Cloud Service Automation service is started.

- b. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
- c. If global search is enabled, do the following:
  - i. Right-click on the Elasticsearch 1.6.1 service and select **Restart**.
  - ii. Wait for the Elasticsearch 1.6.1 service to restart, then right-click on HPE Search Service and select **Restart**.

**Note:** if global search is disabled, skip this step.

- d. Right-click on the CSA service and select **Start**.
- e. Right-click on the HPE Marketplace Portal service and select **Start**.
- f. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Start**.

**To start CSA on Linux, complete the following steps:**

- a. On the server that hosts CSA, type the following:

```
service csa start
service mpp start
```

Refer to the `csa.provider.es.exists` property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 and type the following:

```
service elasticsearch start
```

By default, `elasticsearch` is enabled.

- b. If you installed an embedded Operations Orchestration instance, type:

```
<embeddedHPE00installtion>/central/bin/central start
```

For example, type `/usr/local/hpe/csa/00/central/bin/central start`

For more information about importing archives from the Cloud Service Management Console, refer to the *Cloud Service Management Console Help*. For more information about importing archives through the REST API, refer to the *CSA API Reference* guide.

## Purge Service Subscriptions and Audit Data

The purge tool can be used to delete service subscriptions and audit data.

### About Service Subscriptions

Canceled, expired, failed, and retired service subscriptions store information in the database that, over time, is no longer needed. The purge tool can be used to delete canceled, expired, failed, and retired subscriptions along with specific associated or referenced artifacts and entities. Canceled, expired, and failed subscriptions must have a service instance status of failed, canceled, cancellation failed, or expiration failed to be deleted. Canceled, expired, and failed subscriptions that are not in one of these states will not be deleted. All retired subscriptions are deleted.

By default, when the purge tool is run, canceled, expired, failed, and retired subscriptions that are older than 400 days (subscriptions that have been in a canceled, expired, failed, or retired state longer than 400 days) and certain referenced artifacts and entities are deleted from the database. The age of deleted subscriptions can be increased or decreased by modifying the `age.in.days.to.purge.subscription` property in the configuration properties file used by the purge tool.

When a subscription is deleted, the following artifacts and entities are deleted from the database:

Deleted Artifact	Referenced by (Reference Fields)	Referenced Artifacts and Entities that are Deleted
ServiceSubscription		action associatedRequest basePrice catalogItem initiatingServiceRequest pricingModel property

Deleted Artifact	Referenced by (Reference Fields)	Referenced Artifacts and Entities that are Deleted
		serviceInstance totalPrice Notifications
ServiceRequest	ServiceSubscription (associatedRequest or initiatingServiceRequest)	action basePrice pricingModel property totalPrice Notifications
ServiceInstance	ServiceSubscription (serviceInstance)	componentRoot Notifications
ServiceComponent	ServiceInstance (componentRoot)	action property resourceBinding
ResourceBinding	ServiceComponent (resourceBinding)	action catalogItem lifecycleProperties property resourceInstance
ResourceSubscription	ResourceBinding (resourceInstance)	action catalogItem lifecycleProperties property
ProcessInstance		

## About Audit Data

CSA creates audit event records in the database for events that occur during the lifetime of a running instance of CSA.

By default, when the purge tool is run, audit data that is older than 400 days is deleted from the database. The age of deleted audit data can be increased or decreased by modifying the `age.in.days.to.purge.audit` property in the configuration properties file used by the purge tool.

For more information about auditing data, refer to the *Reporting and Auditing* whitepaper.

## Deleting Service Subscriptions and Audit Data

To delete canceled, expired, failed, and retired subscriptions or audit data from the database, do the following:

**Caution:** Deleted subscriptions and audit data cannot be restored unless you have backed up the database.

1. Change to the `CSA_HOME/Tools/DBPurgeTool/` directory where `CSA_HOME` is the directory in which CSA is installed.
2. Generate the sample configuration files by running the following command (a sample configuration file is generated for each type of database supported by CSA):

### Oracle

- o **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar -g -j ojdbc6.jar
```

- o **Linux:**

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -g -j ojdbc6.jar
```

where `ojdbc6.jar` is the name of the Oracle JDBC driver installed in `CSA_HOME/Tools/DBPurgeTool/`.

**Note:** Additional command line options are required if a secure connection is enabled between the Oracle database and CSA. See step 4 below for more information.

### MS SQL and PostgreSQL

- o **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar -g
```

- o **Linux:**

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -g
```

where `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

3. In the current directory, copy the sample configuration file that corresponds to the type of database you are using to a file named `config.properties`. For example, if you are using an Oracle

database, make a copy of the `config.properties.oracle` file and rename it to `config.properties`. Update the content of `config.properties` as needed, as described in the table:

Property Name	Description
<code>jdbc.driverClassName</code>	<p>The JDBC driver class.</p> <p><b>Example</b></p> <p>Oracle:  <code>jdbc.driverClassName=oracle.jdbc.driver.OracleDriver</code>            MS SQL:  <code>jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver</code>            PostgreSQL: <code>jdbc.driverClassName=org.postgresql.Driver</code></p>
<code>jdbc.dialect</code>	<p>The classname that allows JDBC to generate optimized SQL for a particular database.</p> <p><b>Example</b></p> <p>Oracle: <code>jdbc.dialect=org.hibernate.dialect.OracleDialect</code>            MS SQL:  <code>jdbc.dialect=org.hibernate.dialect.SQLServerDialect</code>            PostgreSQL:  <code>jdbc.dialect=org.hibernate.dialect.PostgreSQLDialect</code></p>
<code>jdbc.databaseUrl</code>	<p>The JDBC URL. When specifying an IPv6 address, it must be enclosed in square brackets (see examples below).</p> <p><b>Example</b></p> <p>Oracle (TLS not enabled):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@//127.0.0.1:1521/XE</code></p> <p>Oracle (TLS not enabled, using an IPv6 address):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@//[f000:253c::9c10:b4b4]:1521/XE</code></p> <p>Oracle (TLS enabled, CSA does not check the database DN):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL = TCPS)(HOST = &lt;host&gt;)(PORT = 1521))) (CONNECT_DATA =(SERVICE_NAME = ORCL)))</code> where <code>&lt;host&gt;</code> is the name of the system on which the Oracle database server is installed.</p> <p>Oracle (TLS enabled, CSA checks the database DN):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@(DESCRIPTION =</code></p>



Property Name	Description
	<p>is the name of the system on which the Oracle database server is installed and the values for SSL_SERVER_CERT_DN are for the DN of the Oracle database server.</p> <p>MS SQL (TLS not enabled):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=request</code></p> <p>MS SQL (TLS not enabled, using an IPv6 address):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://[::1]:1433/example;ssl=request</code></p> <p>MS SQL (TLS enabled):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=authenticate</code></p> <p>MS SQL (FIPS 140-2 compliant):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=authenticate</code></p> <p>PostgreSQL:  <code>jdbc.databaseUrl=jdbc:postgresql://127.0.0.1:5432/csadb</code></p>
securityAdminPassword	<p>The password for the CSA admin user. Required for the purge tool. The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p><b>Example</b></p> <p><code>securityAdminPassword=ENC(1v72sEeKj0cDCRxSqZV23w==)</code></p>
jdbc. username	<p>The user name of the database user you configured for CSA after installing the database.</p>
jdbc. password	<p>The password for the database user. The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>On Windows, if you have configured CSA to be FIPS 140-2 compliant, encrypt this password after you have configured CSA to be FIPS 140-2 compliant (that is, you should use the updated encryption tools to encrypt the password).</p> <p><b>Example</b></p>

Property Name	Description
	<code>jdbc.password=ENC(fc5e38d38a5703285441e7fe7010b0)</code>
<code>idmConfig.Url</code>	The system on which CSA is installed. <b>Default:</b> <code>https://127.0.0.1:8444</code>
<code>securityTransport.UserName</code>	The user used to authenticate legacy REST API calls. <b>Default:</b> <code>csaTransportUser</code>
<code>securityTransport.password</code>	The password for the user used to authenticate legacy REST API calls. The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.  <b>Note:</b> Passwords generated in the <code>config.properties</code> file are sample passwords. You must sync the passwords with the passwords in the <code>csa.properties</code> file.  On Windows, if you have configured CSA to be FIPS 140-2 compliant, encrypt this password after you have configured CSA to be FIPS 140-2 compliant (that is, you should use the updated encryption tools to encrypt the password).  <b>Example</b>  <code>securityTransport.password=ENC(r1bE8430uFSD1jert85441e7fe701jkY)</code>
<code>securityIdmTransport.UserName</code>	The user name to authenticate with Identity Management component. <b>Default:</b> <code>idmTransportUser</code>
<code>securityIdmTransport.password</code>	The password to authenticate with Identity Management component. The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.  <b>Note:</b> Passwords generated in the <code>config.properties</code> file are sample passwords. You must sync the passwords with the passwords in the <code>csa.properties</code> file.  On Windows, if you have configured CSA to be FIPS 140-2 compliant, encrypt this password after you have configured CSA to be FIPS 140-2 compliant (that is, you should use the updated

Property Name	Description
	<p>encryption tools to encrypt the password).</p> <p><b>Example</b></p> <p>securityIdmTransport.password=ENC (1Ddh98Kfe76op8lhjE0E1897k1RCB5321sb)</p>
age.in.days. to.purge. audit	<p>The age of audit data, in days, that the audit data must be equal to or older than to be deleted by this tool.</p> <p><b>Default:</b> 400</p>
age.in.days. to.purge. subscription	<p>The amount of time, in days, a subscription has been in a canceled, expired, failed, or retired state before it is deleted by this tool.</p> <p><b>Default:</b> 400</p>

#### Example config.properties content

##### Oracle (TLS not enabled)

```

jdbc.driverClassName=oracle.jdbc.driver.OracleDriver
jdbc.databaseUrl=jdbc:oracle:thin:@//127.0.0.1:1521/XE
jdbc.username=csa
jdbc.password=ENC(fc5e38d38a5703285441e7fe7010b0)
jdbc.dialect=org.hibernate.dialect.OracleDialect
idmConfig.Url=https://127.0.0.1:8444
securityTransportUserName=csaTransportUser
securityTransport.password=ENC(rlbE8430uFSDljert85441e7fe701jkY)
securityIdmTransportUserName=idmTransportUser
securityIdmTransport.password=ENC(1Ddh98Kfe76op8lhjE0E1897k1RCB5321sb)
age.in.days.to.purge.audit=400
age.in.days.to.purge.subscription=400

```

##### MS SQL (TLS not enabled)

```

jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver
jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=request
jdbc.username=csa
jdbc.password=ENC(fc5e38d38a5703285441e7fe7010b0)
jdbc.dialect=org.hibernate.dialect.SQLServerDialect
idmConfig.Url=https://127.0.0.1:8444
securityTransportUserName=csaTransportUser
securityTransport.password=ENC(rlbE8430uFSDljert85441e7fe701jkY)
securityIdmTransportUserName=idmTransportUser
securityIdmTransport.password=ENC(1Ddh98Kfe76op8lhjE0E1897k1RCB5321sb)

```

```
age.in.days.to.purge.audit=400
age.in.days.to.purge.subscription=400
```

### **MS SQL (TLS enabled)**

```
jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver
jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/
example;ssl=authenticate
jdbc.username=csa
jdbc.password=ENC(fc5e38d38a5703285441e7fe7010b0)
jdbc.dialect=org.hibernate.dialect.SQLServerDialect
```

### **MS SQL (FIPS 140-2 compliant on Windows)**

```
jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver
jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/
example;ssl=authenticate
jdbc.username=csa
jdbc.password=ENC(fc5e38d38a5703285441e7fe7010b0)
jdbc.dialect=org.hibernate.dialect.SQLServerDialect
idmConfig.Url=https://127.0.0.1:8444
securityTransportUserName=csaTransportUser
securityTransport.password=ENC(rlbE8430uFSDljert85441e7fe701jkY)
securityIdmTransportUserName=idmTransportUser
securityIdmTransport.password=ENC(1Ddh98Kfe76op8lhjE0E1897k1RCB5321sb)
age.in.days.to.purge.audit=400
age.in.days.to.purge.subscription=400
```

### **PostgreSQL**

```
jdbc.driverClassName=org.postgresql.Driver
jdbc.databaseUrl=jdbc:postgresql://127.0.0.1:5432/csadb
jdbc.username=csadbuser
jdbc.password=ENC(fc5e38d38a5703285441e7fe7010b0)
jdbc.dialect=org.hibernate.dialect.PostgreSQLDialect
idmConfig.Url=https://127.0.0.1:8444
securityTransportUserName=csaTransportUser
securityTransport.password=ENC(rlbE8430uFSDljert85441e7fe701jkY)
securityIdmTransportUserName=idmTransportUser
securityIdmTransport.password=ENC(1Ddh98Kfe76op8lhjE0E1897k1RCB5321sb)
age.in.days.to.purge.audit=400
age.in.days.to.purge.subscription=400
```

4. Run the following command to delete subscriptions and audit data (you can specify options to

delete only subscriptions or only audit data):

**Caution:** THE PURGE TOOL RUNS WITHOUT PROMPTING FOR A CONFIRMATION.

Deleted subscriptions and audit data cannot be restored unless you have backed up the database.

Verify that you have entered the correct information into the `config.properties` file before running this tool.

**Note:** When running the tool to delete subscriptions or audit data, you will be prompted for a username and password. This user **MUST** be assigned to the CSA Administrator role. Users who are not assigned to this role cannot delete subscriptions nor audit data.

#### Oracle (TLS not enabled)

- **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar -j ojdbc6.jar
```

- **Linux:**

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -j ojdbc6.jar
```

where `ojdbc6.jar` is the name of the Oracle JDBC driver installed in `CSA_HOME\Tools\DBPurgeTool` and `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

#### Oracle (TLS enabled, CSA does not check the database DN, client authentication is enabled on the Oracle database server)

- **Windows:**

```
"CSA_JRE_HOME\bin\java"
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar db-purge-tool.jar -j ojdbc6.jar
```

- **Linux:**

```
CSA_JRE_HOME/bin/java
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar db-purge-tool.jar -j ojdbc6.jar
```

where `ojdbc6.jar` is the name of the Oracle JDBC driver installed in `CSA_HOME\Tools\DBPurgeTool`, `certificate_key_file` is the same keystore file defined by the `certificate-key-file` attribute in the `ssl` element of the `CSA_HOME\jboss-as/standalone/configuration/standalone.xml` file (for example, `CSA_HOME/jboss-as/standalone/configuration/.keystore`), `certificate_key_file_password` is the password to the keystore file (for example, `changeit`), `certificate_key_file_type` is the keystore type (for example, `JKS` or `PKCS12`), and `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

**Oracle (TLS enabled, CSA does not check the database DN, client authentication is NOT enabled on the Oracle database server)**

o **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar -j ojdbc6.jar
```

o **Linux:**

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -j ojdbc6.jar
```

where `ojdbc6.jar` is the name of the Oracle JDBC driver installed in `CSA_HOME\Tools\DBPurgeTool` and `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed. .

**Oracle (TLS enabled, CSA checks the database DN, client authentication is enabled on the Oracle database server)**

o **Windows:**

```
"CSA_JRE_HOME\bin\java"
-Doracle.net.ssl_server_dn_match=true
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar db-purge-tool.jar -j ojdbc6.jar
```

o **Linux:**

```
CSA_JRE_HOME/bin/java
-Doracle.net.ssl_server_dn_match=true
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar db-purge-tool.jar -j ojdbc6.jar
```

where **ojdbc6.jar** is the name of the Oracle JDBC driver installed in `CSA_HOME\Tools\DBPurgeTool`, `certificate_key_file` is the same keystore file defined by the

certificate-key-file attribute in the ssl element of the CSA\_HOME\jboss-as\standalone\configuration\standalone.xml file (for example, CSA\_HOME\jboss-as\standalone\configuration\.keystore, *certificate\_key\_file\_password* is the password to the keystore file (for example, changeit), *certificate\_key\_file\_type* is the keystore type (for example, JKS or PKCS12), and CSA\_JRE\_HOME is the directory in which the JRE that is used by CSA is installed.

**Oracle (TLS enabled, CSA checks the database DN, client authentication is NOT enabled on the Oracle database server)**

- o **Windows:**  
 "CSA\_JRE\_HOME\bin\java"  
 -Doracle.net.ssl\_server\_dn\_match=true -jar db-purge-tool.jar -j ojdbc6.jar
- o **Linux:**  
 CSA\_JRE\_HOME/bin/java  
 -Doracle.net.ssl\_server\_dn\_match=true -jar db-purge-tool.jar -j ojdbc6.jar

where ojdbc6.jar is the name of the Oracle JDBC driver installed in CSA\_HOME/Tools/DBPurgeTool and CSA\_JRE\_HOME is the directory in which the JRE that is used by CSA is installed. .

**MS SQL and PostgreSQL**

- o **Windows:**  
 "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar
- o **Linux:**  
 CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar

where CSA\_JRE\_HOME is the directory in which the JRE that is used by CSA is installed.

The following options are available in the purge tool

Option	Description
-jar db-purge-tool.jar	Required. The name of the tool to run.
-a, --audit	Optional. Purge audit data. If neither -a nor -s are specified, the tool purges both audit data and subscriptions.  <b>Note:</b> When running the tool to delete subscriptions or audit data, you will be prompted for a username and password. This user

Option	Description
	<p>MUST be assigned to the CSA Administrator role. Users who are not assigned to this role cannot delete subscriptions nor audit data.</p>
-c <config_properties>, --config <config_properties>	<p>Optional. The name and location of the configuration properties file. By default, the tool looks for the configuration properties file in the working directory (the directory from which the tool is run). If this option is not specified, the tool looks for the <code>config.properties</code> in the working directory. The examples in this document assume the file is located in the working directory and is named <code>config.properties</code>.</p>
-g, --generate	<p>Optional. Generate example configuration properties files for supported databases.</p>
-h, --help	<p>Optional. List the options available in this tool.</p>
-j <jdbc_drivers>, --jars <jdbc_drivers>	<p>Optional. The name and location of the JDBC driver(s) to be used by this tool. If more than one driver needs to be specified, separate each driver by a space. By default, the tool looks for the JDBC driver(s) in the working directory (the directory from which the tool is run). If you are not running the tool from <code>CSA_HOME/Tools/DBPurgeTool</code>, specify the name and location of the JDBC driver(s) to be used. On Windows, if the path name contains a space, the path and file name should be enclosed in quotation marks. For example:</p> <pre>-j "C:\Program Files\jdbc\ojdbc6.jar"</pre> <p>For a list of supported JDBC driver versions, see the <i>Cloud Service Automation System and Software Support Matrix</i>.</p>
-s, --subscription	<p>Optional. Purge subscription data. If neither -s nor -a are specified, the tool purges both subscriptions and audit data.</p> <p><b>Note:</b> When running the tool to delete subscriptions or audit data, you will be prompted for a username and password. This user MUST be assigned to the CSA Administrator role. Users who are not assigned to this role cannot delete subscriptions nor audit data.</p>

### Examples for Oracle (TLS is not Enabled)

Display the purge tool help:

- Windows:
 

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar -h -j ojdbc6.jar
```
- Linux:
 

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -h -j ojdbc6.jar
```



Generate sample configuration properties files: \

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -g -j ojdbc6.jar

- **Linux:**

- CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar -g -j ojdbc6.jar

Purge subscriptions and associated entities:

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -s -j ojdbc6.jar

- **Linux:**

- CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar -s -j ojdbc6.jar

Purge audit data:

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -a -j ojdbc6.jar

- **Linux:**

- CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar -a -j ojdbc6.jar

Purge subscriptions and associated entities and audit data:

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -j ojdbc6.jar

- **Linux:**

- CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar -j ojdbc6.jar

### Examples for PostgreSQL

Display the purge tool help:

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -h

- CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar -h

Generate sample configuration properties files:

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -g

- CSA\_JRE\_HOME/bin/java -jar db-purge-tool.jar -g

Purge subscriptions and associated entities:

- **Windows:**

- "CSA\_JRE\_HOME\bin\java" -jar db-purge-tool.jar -s

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -s
```

Purge audit data:

- **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar -a
```

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar -a
```

Purge subscriptions and associated entities and audit data:

- **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar db-purge-tool.jar
```

```
CSA_JRE_HOME/bin/java -jar db-purge-tool.jar
```

## Upgrade or Install a Fresh CSA Database Schema

The schema installation tool is used to upgrade the existing CSA database schema or to install a fresh database schema without re-installing CSA. Use this tool if you did not install CSA database components onto the database during installation, did not upgrade the database schema during an upgrade, or if you want to drop the existing schema and install a fresh CSA database schema.

**Note:** If you do not install CSA database components during an install or upgrade, and you are going to run the schema installation tool, you need to run the org migration tool manually after that. For details see the *Cloud Service Automation Upgrade Guide*.

You can also use this tool to complete an upgrade if the upgrade failed, the database schema was not updated, the failure was not due to a database problem, and the problem can be fixed without rerunning the upgrade installer. For example, if the upgrade failed but can be completed successfully by manual configuration but the database schema was not updated, you can simply make the manual changes to complete the upgrade and run the schema installation tool instead of reverting CSA back to the previous version and running the upgrade installer again.

**Note:** Do not run the schema installation tool if you installed the database components during the installation of CSA or if you upgraded the database schema when you upgraded CSA.

If you run the schema installation tool on an existing schema (where CSA has been upgraded but the database schema was not upgraded), the schema is upgraded and no data in the database is lost. However, if you drop the existing schema and run this tool, all data in the database associated with the

dropped schema is lost. Once you run the tool, a fresh schema is installed and you cannot revert back to the dropped schema.

**Caution:** Once you drop an existing schema and run the database schema installation tool, you cannot revert back to the dropped schema.

## Upgrading or Installing the Database Schema

To upgrade or install a fresh CSA database schema, do the following:

1. If CSA is running, stop CSA.

**To stop CSA on Windows, complete the following steps:**

- a. On the server that hosts CSA, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the CSA service and select **Stop**.
- c. Right-click on the HPE Marketplace Portal service and select **Stop**.
- d. If you installed an embedded Operations Orchestration instance, right-click on the HPE Operations Orchestration Central service and select **Stop**.
- e. If you enabled global search, do the following:
  - i. Right-click on the Elasticsearch 1.6.1 service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
  - ii. Right-click on HPE Search Service and select **Stop**. You do not need to stop this service if global search is disabled (by default, global search is disabled).
- f. If Elasticsearch is enabled (by default, Elasticsearch is enabled; refer to the `csa.provider.es.exists` property in "[Appendix: Cloud Service Management Console Properties](#)" on page 300 for more information), right-click on the Elasticsearch 1.6.1 service and select **Stop**.

**To stop CSA on Linux, complete the following steps:**

- a. On the server that hosts CSA, type the following commands:

```
service csa stop
service mpp stop
```
- b. If you installed an embedded Operations Orchestration instance, type:

`<embeddedHPE00installation>/central/bin/central stop`

For example, type: `/usr/local/hpe/csa/00/central/bin/central stop`

2. Change to the `CSA_HOME/Tools/SchemaInstallationTool/` directory where `CSA_HOME` is the directory in which CSA is installed.
3. During upgrade or installation of CSA, a file named `db.properties` was generated in `CSA_HOME/Tools/SchemaInstallationTool/`. Verify the property values in this file. If you changed any database property values in the `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` file after installation, the values in `db.properties` may not be up-to-date.

If you have dropped the existing database schema and are installing a fresh database schema after upgrading to CSA 4.80, you must update the `driverFiles` property value. The properties defined in `db.properties` are described in the table.

Property Name	Description
<code>dbUrl</code>	<p>The JDBC URL. When specifying an IPv6 address, it must be enclosed in square brackets (see examples below).</p> <p><b>Examples</b></p> <p>Oracle (TLS not enabled):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@//127.0.0.1:1521/XE</code></p> <p>Oracle (TLS not enabled, using an IPv6 address):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@//[f000:253c::9c10:b4b4]:1521/XE</code></p> <p>Oracle (TLS enabled, CSA does not check the database DN):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL = TCPS)(HOST = &lt;host&gt;)(PORT = 1521))) (CONNECT_DATA =(SERVICE_NAME = ORCL)))</code> where <code>&lt;host&gt;</code> is the name of the system on which the Oracle database server is installed.</p> <p>Oracle (TLS enabled, CSA checks the database DN):  <code>jdbc.databaseUrl=jdbc:oracle:thin:@(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCPS)(HOST = &lt;host&gt;)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = ORCL))(SECURITY=(SSL_SERVER_CERT_DN="CN=abc,OU=dbserver,O=xyz,L=Sunnyvale,ST=CA,C=US")))</code> where <code>&lt;host&gt;</code> is the name of the system on which the Oracle database</p>

Property Name	Description
	<p>MS SQL (TLS not enabled):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=request</code></p> <p>MS SQL (TLS not enabled, using an IPv6 address):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://[::1]:1433/example;ssl=request</code></p> <p>MS SQL (TLS enabled):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=authenticate</code></p> <p>MS SQL (FIPS 140-2 compliant):  <code>jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=authenticate</code></p> <p>PostgreSQL:  <code>jdbc.databaseUrl=jdbc:postgresql://127.0.0.1:5432/csadb</code></p>
dbUserName	<p>The user name of the database user you configured for CSA after installing the database.</p>
dbPassword	<p>The password for the database user. The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>While you may enter a password in clear text, after you run the tool, the clear text password is automatically replaced by an encrypted password.</p> <p>If you have configured CSA to be FIPS 140-2 compliant on Windows, encrypt this password after you have configured CSA to be FIPS 140-2 compliant (that is, you should use the updated encryption tools to encrypt the password).</p> <p><b>Example</b></p> <pre>dbPassword=ENC(fc5e38d38a5703285441e7fe7010b0)</pre>
driverFiles	<p>The database driver files used by this tool. If you are running a fresh installation of CSA 4.80 (you did not upgrade to CSA 4.80), you do not need to change these values.</p> <p>If you have upgraded to CSA 4.80 and want to upgrade the existing schema, you do not need to change these values.</p> <p>If you have upgraded to CSA 4.80, have dropped the existing database schema, and are installing a fresh database schema, you must update this value to the following:</p> <p><b>Oracle</b> (upgrade and dropped schema only)</p> <pre>driverFiles=CSA_HOME\scripts\schemainstallforupg\ (4.80) create-oracle-schema.sql, CSA_HOME\scripts\schemainstallforupg\</pre>

Property Name	Description
	<p><b>Microsoft SQL</b> (upgrade and dropped schema only)  driverFiles=CSA_HOME/scripts/schemainstallforupg/  alterdb.sql,  CSA_HOME/scripts/schemainstallforupg\  create-mssql-schema.sql,  CSA_HOME/scripts/schemainstallforupg\  create-mssql-topology-schema.sql,  CSA_HOME/scripts/schemainstallforupg\  mssql\seed_data_driver.sql,  CSA_HOME/scripts/reporting/mssql\  install_views_driver.sql,  CSA_HOME/scripts/reporting/mssql\  grant-reporting-user.sql</p> <p><b>Note:</b> Add the grant-reporting-user.sql file only if you have created the reporting database user for CSA.</p>
jdbcDriverClassName	<p>The JDBC driver class. Do not change this value.</p> <p><b>Examples</b></p> <p>Oracle:  jdbc.driverClassName=oracle.jdbc.driver.OracleDriver  MS SQL:  jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver  PostgreSQL: jdbc.driverClassName=org.postgresql.Driver</p>
jdbcDriverDir	<p>The location of the JDBC driver(s) used by this tool. Do not change this value.</p>

- Run the following command:

**Oracle (TLS not enabled), MS SQL, and PostgreSQL**

- Windows:**  
"CSA\_JRE\_HOME\bin\java" -jar schema-installation-tool.jar
- Linux:**  
CSA\_JRE\_HOME/bin/java -jar schema-installation-tool.jar

where CSA\_JRE\_HOME is the directory in which the JRE that is used by CSA is installed. .

**Oracle (TLS enabled, CSA does not check the database DN, client authentication is enabled on the Oracle database server)**

- Windows:**  
"CSA\_JRE\_HOME\bin\java"

```
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar schema-installation-tool.jar
```

- o **Linux:**

```
CSA_JRE_HOME/bin/java
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar schema-installation-tool.jar
```

where `certificate_key_file` is the same keystore file defined by the `certificate-key-file` attribute in the `ssl` element of the `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` file (for example, `CSA_HOME/jboss-as/standalone/configuration/.keystore`), `certificate_key_file_password` is the password to the keystore file (for example, `changeit`), `certificate_key_file_type` is the keystore type (for example, `JKS` or `PKCS12`) and `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed. .

**Oracle (TLS enabled, CSA does not check the database DN, client authentication is NOT enabled on the Oracle database server)**

- o **Windows:**

```
"CSA_JRE_HOME\bin\java" -jar schema-installation-tool.jar
```

- o **Linux:**

```
CSA_JRE_HOME/bin/java -jar schema-installation-tool.jar
```

**Oracle (TLS enabled, CSA checks the database DN, client authentication is enabled on the Oracle database server)**

- o **Windows:**

```
"CSA_JRE_HOME\bin\java"
-Doracle.net.ssl_server_dn_match=true
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar schema-installation-tool.jar
```

- o **Linux:**

```
CSA_JRE_HOME/bin/java
-Doracle.net.ssl_server_dn_match=true
-Djavax.net.ssl.keyStore="<certificate_key_file>"
-Djavax.net.ssl.keyStorePassword=<certificate_key_file_password>
-Djavax.net.ssl.keyStoreType=<certificate_key_file_type>
-jar schema-installation-tool.jar
```

where `certificate_key_file` is the same keystore file defined by the `certificate-key-file` attribute in the `ssl` element of the `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` file (for example, `CSA_HOME/jboss-as/standalone/configuration/.keystore`), `certificate_key_file_password` is the password to the keystore file (for example, `changeit`), `certificate_key_file_type` is the keystore type (for example, `JKS` or `PKCS12`), and `CSA_JRE_HOME` is the directory in which the JRE that is used by CSA is installed.

**Oracle (TLS enabled, CSA checks the database DN, client authentication is NOT enabled on the Oracle database server)**

- o **Windows:**

```
"CSA_JRE_HOME\bin\java" -Doracle.net.ssl_server_dn_match=true -jar schema-
installation-tool.jar
```

- o **Linux:**

```
CSA_JRE_HOME/bin/java -Doracle.net.ssl_server_dn_match=true -jar schema-
installation-tool.jar
```

5. Prepare icons and java scripts to be loaded to the database:

a. Change to the `CSA_HOME/images` directory.

If there is a single file named `icons-backup<time_stamp><ip_address>.zip` then unzip this file into the current directory. When CSA is started the images will be migrated to the database.

If the `icons-backup<time_stamp><ip_address>.zip` file is missing or the directory contains these directories: `catalog`, `categories` and `library`, then skip this step 5 completely.

b. Change to the `CSA_HOME/propertysources` directory.

If there is a file named `js-backup<time_stamp><ip_address>.zip` then unzip this file into the current directory. When CSA is started the java scripts will be migrated to the database.



# Configure CSA to Mitigate Frequently Dropped Database Connections

If you are experiencing frequently dropped database connections, configure the JBoss data source connections to mitigate the problem.

In a standalone environment, do the following:

1. Stop the CSA service:

**Windows:**

- a. Navigate to **Control Panel > Administrative Tools > Services**.
- b. Right-click on the CSA service.
- c. Select **Stop**.

**Linux:**

From a command prompt, type `service csa stop`.

2. Edit the `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` file:
  - a. Find the `dataSource` tag which is used for CSA database configuration.
  - b. Add the following after the line that ends with `</security>`:

**Oracle**

```
<validation>
<check-valid-connection-sql>select 1 from DUAL</check-valid-connection-sql>
<validate-on-match>>false</validate-on-match>
</validation>
```

**MS SQL or PostgreSQL**

```
<validation>
<check-valid-connection-sql>select 1</check-valid-connection-sql>
<validate-on-match>>false</validate-on-match>
</validation>
```

3. Start the CSA service:

**Windows:**

- a. Navigate to **Control Panel > Administrative Tools > Services**.
- b. Right-click on the CSA service.
- c. Select **Start**.

**Linux:**

From a command prompt, type `service csa start`.

In a clustered environment, do the following:

1. Stop the CSA service:

**Windows:**

- a. Navigate to **Control Panel > Administrative Tools > Services**.
- b. Right-click on the CSA service.
- c. Select **Stop**.

**Linux:**

From a command prompt, type `service csa stop`.

2. Edit the `CSA_HOME/jboss-as/standalone/configuration/standalone-full-ha.xml` file:

- a. Find the `dataSource` tag which is used for CSA database configuration.
- b. Add the following after the line that ends with `</security>`:

**Oracle:**

```
<validation>
<check-valid-connection-sql>select 1 from DUAL</check-valid-connection-sql>
<validate-on-match>>false</validate-on-match>
</validation>
```

**MS SQL or PostgreSQL:**

```
<validation>
<check-valid-connection-sql>select 1</check-valid-connection-sql>
<validate-on-match>>false</validate-on-match>
</validation>
```

3. Start the CSA service:

**Windows:**

- a. Navigate to **Control Panel > Administrative Tools > Services**.

- b. Right-click on the CSA service.
- c. Select **Start**.

**Linux:**

From a command prompt, type `service csa start`.

# Appendix: Cloud Service Management Console Properties

This section lists and describes the properties that can be configured for the Cloud Service Management Console, which are located in one of the following files:

- `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties`
- `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/web.xml`
- `CSA_HOME/jboss-as/standalone/deployments/csa.war/offerings/config.json`

where

`CSA_HOME` is the directory in which CSA is installed

The following areas contain properties that can be configured (for many properties, default values are provided):

- [Authentication](#)
- [Action Selection Wizard](#)
- [Security banner](#)
- [Email notifications](#)
- [Marketplace Portal URL](#)
- [Dashboard](#)
- [Security](#)
- [CSA keystore](#)
- [Service request processor scheduler](#)
- [Auditing](#)
- [Process execution manager](#)
- [Lifecycle engine](#)
- [Approval engine scheduler](#)
- [LDAP cache scheduler](#)

- [Clustering](#)
- [Dynamic property](#)
- [Group Approval](#)
- [Marketplace Portal](#)
- [FIPS 140-2 on Windows](#)
- [Common Access Card](#)
- [Single Sign-On](#)
- [Process executor delegate](#)
- [Miscellaneous](#)
- [Operations Orchestration](#)
- [CSA 3.x API authentication](#)
- [Topology designer](#)
- [Elasticsearch](#)
- [Microservices](#)
- [Secure connections](#)
- [LDAP access point](#)
- [Service design, service offering, and catalog content archive verification](#)
- [HPE ITOC Integration](#)
- [Session timeout](#)
- [REST](#)

For information about Codar properties, see the Codar documentation.

After modifying the `csa.properties` file, restart CSA. See ["Restart CSA" on page 170](#) for instructions.

## Authentication

These properties are used for authentication.

These properties are configured in `csa.properties`.

Property	Description
<code>csa.provider.hostname</code>	<p>Required. The fully-qualified domain name of the system on which CSA is running.</p> <p>If you change this hostname, you must update the value of the <code>idm.csa.hostname</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file.</p>
<code>csa.provider.port</code>	<p>Required. The port used to connect to the system on which CSA is running.</p> <p>If you change this port, you must update the value of the <code>idm.csa.port</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file.</p>
<code>csa.provider.rest.protocol</code>	<p>Required. The protocol used by the REST API to connect to the system on which CSA is running.</p> <p>This attribute must be set to <b>https</b>.</p> <p>If you change this protocol, you must update the value of the <code>idm.csa.protocol</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file.</p>
<code>csa.orgName.identifier</code>	<p>Required. The provider organization identifier assigned to the organization who is providing this instance of the Cloud Service Management Console.</p> <p>This attribute must be set to <b>CSA-Provider</b>.</p>

### Action Selection Wizard

These properties are used for the Action Selection Wizard.





These properties are configured in `csa.properties`.

Property	Description
<code>csa.cache.default.timeout.seconds</code>	The <code>csa.properties</code> key that controls the cache timeout. <b>Default:</b> 300 seconds
<code>csa.oo.content.root.lifecycle.action</code>	Comma-separated root folder names from Operations Orchestration for the Action Selection Wizard when used in the Lifecycle Action and User Operations areas. <b>Default:</b> <code>=/Library/CSA Content Pack/CSA3.2/Providers,/Library/CSA Content Pack/CSA3.2/CSA Import and Migration Pack</code>
<code>csa.oo.content.root.external.approval</code>	Comma-separated root folder names from Operations Orchestration for the Action Selection Wizard when used in the Approvals area. <b>Default:</b> <code>=/Library/CSA Content Pack/CSA3.2/External Approval System/Service Manager/Actions</code>
<code>csa.oo.content.root.resource.pool.sync</code>	Comma-separated root folder names from Operations Orchestration for the Action Selection Wizard when used in the Resource Pool area. <b>Default:</b> <code>=/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/vCenter/Resource Pool Sync/Actions</code>





### Security banner attributes

The attributes in the following table are used by the Cloud Service Management Console to enable or disable the display of a disclaimer upon logging in to the Cloud Service Management Console and a color-coded banner that appears at the top and bottom of the Cloud Service Management Console.

These properties are configured in `csa.properties`.

Attribute	Description
<p><code>csa.provider.agency</code></p>	<p>By default, this attribute is commented out. When this attribute is commented out or does not contain a valid value, the login disclaimer and color-coded banners are not displayed for the Cloud Service Management Console.</p> <p>If you want to enable the login disclaimer and color-coded banners, uncomment this attribute and set the value to <b>GOVERNMENT</b>. If set to any other value, the login disclaimer and color-coded banners are not displayed.</p> <p>To edit the disclaimer page, edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/static/template/disclaimerNote.jsp</code> file.</p> <p>To edit the disclaimer content, edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/msgs/messages_en.properties</code> file. To locate the disclaimer content in this file, search for message property entries beginning with <code>csa.security.warning</code>.</p>
<p><code>csa.provider.contentType</code></p>	<p>By default, this attribute is commented out. This attribute defines the color and content that displays in the security banner. The security banners appear at the top and bottom of the Cloud Service Management Console.</p> <p>The following values are shipped with CSA:</p> <ul style="list-style-type: none"> <li> <p>UNCLASSIFIED. The banner is light green and contains no content. An example is shown below.</p>  </li> <li> <p>UNCLASSIFIED_FOUO. For official use only. The banner is light green and displays the text "FOUO." An example is shown below.</p>  </li> <li> <p>UNCLASSIFIED_NOFORN. Not releasable to foreign nationals. The banner is light green and displays the text "NOFORN." An example is shown below.</p>  </li> <li> <p>CONFIDENTIAL. The banner is light blue and displays the text</p>  </li> </ul>



Attribute	Description
	<ul style="list-style-type: none"><li data-bbox="505 306 1341 369">• CONFIDENTIAL_FOUO. The banner is light blue and displays the text "CONFIDENTIAL-FOUO." An example is shown below. </li><li data-bbox="505 453 1382 516">• CONFIDENTIAL_NOFORN. The banner is light blue and displays the text "CONFIDENTIAL-NOFORN." An example is shown below. </li><li data-bbox="505 600 1279 663">• SECRET. The banner is red and displays the text "SECRET." An example is shown below. </li><li data-bbox="505 747 1195 810">• TOPSECRET. The banner is orange and displays the text "TOPSECRET." An example is shown below. </li></ul> <p data-bbox="495 905 1328 1066">To edit the banner content, edit the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/mgs/messages_en.properties</code> file. To locate the banner content in this file, search for message property entries beginning with <code>csa.security.label</code>.</p>

## Email notifications

These properties are used to define email notifications.

This property is configured in `csa.properties`.

Property	Description
<code>csa.notification.type</code>	<p>Defines the type of email notification: html/text.</p> <ul style="list-style-type: none"><li>• <code>html</code> enables custom HTML notifications.</li><li>• <code>text</code> enables the legacy text-based notification.</li></ul> <p><b>Default:</b> <code>html</code></p>
<code>csa.notification.cacheTemplates</code>	<p>The notification templates will be cached so that I/O performance is improved while sending notifications. If any notification template used by CSA is changed, then the changes will not be seen in later notifications unless the CSA service is restarted.</p> <p>The value of <code>csa.notification.cacheTemplates</code> may be set to <code>false</code> during development of custom notifications so that a service restart is not required every time a notification template is changed.</p> <p><b>Default:</b> <code>true</code></p>

## Marketplace Portal URL

This property is used to define the URL of the Marketplace Portal for an organization and is displayed in the Cloud Service Management Console.

This property is configured in `csa.properties`.

Property	Description
<code>csa.subscriber.portal.url</code>	<p>The URL used to access the Marketplace Portal of an organization and is displayed in the Organization URL field in the General Information section of an organization's page in the Cloud Service Management Console.</p> <p>You can use specific values or one or more of the following variables:</p> <ul style="list-style-type: none"> <li>• <code>{protocol}</code> - The protocol used to connect to the Marketplace Portal. This is either <code>http</code> or <code>https</code>. The variable value is the same protocol used to access the Cloud Service Management Console.</li> <li>• <code>{host}</code> - The fully-qualified domain name or IP address of the system on which the Marketplace Portal is installed. The variable value is the same host on which the Cloud Service Management Console is installed.</li> <li>• <code>{orgName}</code> - The organization's name. The variable value is the Organization Identifier displayed in the General Information section of an organization's page. The Organization Identifier is based on the value entered in the Organization Display Name field.</li> </ul> <p>The port configured for the Marketplace Portal in this property should match the <code>port</code> attribute value configured in the <code>CSA_HOME/portal/conf/mpp.json</code> file.</p> <p>If a variable's value is incorrect, you can enter a specific value in place of the variable. For example,  <code>https://{host}:8089/org/{orgName}</code> or  <code>{protocol}://csa_system.xyz.com:8089/#/login/marketing</code></p> <p><b>Default:</b> <code>{protocol}://{host}:8089/org/{orgName}</code></p>

## Dashboard

This property is used to control whether the Dashboard Mashup Widgets can be edited.

This property is configured in `csa.properties`.

Property	Description
<code>csa.ui.organizations.dashboardwidgets.enableEditingMashup</code>	<p>This property is disabled by default in a fresh install, which prevents the administrator from modifying organization widgets. This property controls whether the administrator only sees the widgets, or has the ability to edit the widgets.</p> <ul style="list-style-type: none"><li>• <code>false</code> disables editing the Mashup Widgets, they can only be seen. If the administrator tries to edit the Mashup Widget, a pop-up message appears stating that support for adding and editing Mashup Widgets is currently disabled.</li><li>• <code>true</code> enables editing the Mashup Widgets.</li></ul> <p><b>Default:</b> <code>false</code></p>

## Security

These properties are used to configure security settings for the Cloud Service Management Console.

Most of these properties are configured in `csa.properties`, and also in `offerings/config.json` for `enableSecurityWarning`.

Property	Description
<p><code>securityAdminPassword</code></p>	<p>Required. The encrypted password used by the CSA built-in admin user (defined in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file). The admin user account is used for initial login to the Cloud Service Management Console and can also be used to authenticate REST API calls.</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must also update the password of any REST API calls that use this password. For more information about the REST APIs, refer to the <i>Cloud Service Automation API Quick Start Guide</i> and <i>Cloud Service Automation API Guide</i>.</p>
<p><code>securityCsaReportingUserPassword</code></p>	<p>Required. The encrypted password used by the CSA built-in <code>csaReportingUser</code> user (defined in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties</code> file). The <code>csaReportingUser</code> user account is used when a subscription is ordered or modified and a field for the subscription includes a dynamically generated list. The dynamically generated list is a subscriber option property configured to use a dynamic query. The dynamic query uses this account to access CSA to determine the values that will appear in the list. This account has read-only access to Cloud Service Automation.</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must also update the password of any REST API calls that use this</p>

Property	Description
	password. For more information about the REST APIs, see the <i>Cloud Service Automation API Guide</i> .
securityTransport UserName	<p>Required. The CSA built-in user used to authenticate REST API calls between the Marketplace Portal and Cloud Service Management Console (it should not be used to log in to the Cloud Service Management Console).</p> <p>If you change this username, you must update the value of the <code>idm.csa.username</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file.</p> <p>For more information about the integration user account, see <a href="#">"Change CSA Built-In User Accounts" on page 205</a>. For more information about the REST APIs, see the <i>Cloud Service Automation API Guide</i>.</p>
securityTransportPassword	<p>Required. The encrypted password used by the CSA built-in <code>csaTransportUser</code> user (defined in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/applicationContext-security.xml</code> file). The <code>csaTransportUser</code> user account is used to authenticate REST API calls between the Marketplace Portal and Cloud Service Management Console (it should not be used to log in to the Cloud Service Management Console).</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must update the value of the <code>idm.csa.password</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file.</p> <p>For more information about the integration user account, see <a href="#">"Change CSA Built-In User Accounts" on page 205</a>. For more information about the REST APIs, see the <i>Cloud Service Automation API Guide</i>.</p>
securityOoInbound UserPassword	<p>Required. The encrypted password used by the CSA built-in <code>ooInboundUser</code> user (defined in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-</code></p>

Property	Description
	<p>service.war/WEB-INF/classes/provider-users.properties file). The oolInboundUser user account is used by Operations Orchestration to authenticate REST API calls with CSA (it should not be used to log in to the Cloud Service Management Console).</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must also update and use the same password for the CSA_REST_CREDENTIALS system account in Operations Orchestration (located in the Configuration folder of the Public Repository).</p>
securityCdaInboundUserPassword	<p>Required. The encrypted password used by the CSA built-in cdaInboundUser user (defined in the CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/provider-users.properties file). The cdaInboundUser user account is used by Continuous Delivery Automation to authenticate REST API calls with CSA (it should not be used to log in to the Cloud Service Management Console).</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must also update and use the same password in Continuous Delivery Automation. For more information about this user account, see <a href="#">"Change CSA Built-In User Accounts" on page 205</a>.</p>
securityIdmTransportUserPassword	<p>Required. The encrypted password used by the CSA built-in idmTransportUser user (defined in the CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/applicationContext-security.xml file). The idmTransportUser user account is used to authenticate REST API calls (it should not be used to log in to the Cloud Service Management Console).</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted</p>

Property	Description
	<p>password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must also update the following passwords (you must use the same password):</p> <ul style="list-style-type: none"> <li>the <code>idmTransportUser</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/classes/integrationusers.properties</code> file.</li> <li>the <code>password</code> attribute in the <code>idmProvider</code> section of the <code>CSA_HOME/portal/conf/mpp.json</code> file (this password uses a different password encryption utility; see <a href="#">"Encrypt a Marketplace Portal Password" on page 180</a> for more information about encrypting the password attribute).</li> <li>the password of any REST API calls that use this password.</li> </ul> <p>For more information about this user account, see <a href="#">"Change CSA Built-In User Accounts" on page 205</a>.</p>
<p><code>securityCatalogAggregationTransportUserPassword</code></p>	<p>Required. The encrypted password used by the CSA built-in <code>csaCatalogAggregationTransportUser</code> user (defined in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/applicationContext-security.xml</code> file). The <code>csaCatalogAggregationTransportUser</code> user account is used to authenticate catalog aggregation REST API calls with CSA (it should not be used to log in to the Cloud Service Management Console).</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>If you change this password, you must also update the password using the catalog aggregation registration REST APIs. For more information about this user account, see <a href="#">"Change CSA Built-In User Accounts" on page 205</a>.</p>
<p><code>securityEncryptedSigningKey</code></p>	<p>CSA's encrypted signing key used to encrypt and decrypt authentication data passed between CSA and the Identity Management component.</p>



Property	Description
	<p>If you change this key, you must also update the <code>idm.encryptedSigningKey</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties</code> file.</p> <p>The key should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions about how to encrypt this key). The encrypted key is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses.</p>
<p><code>com.hp.csa.service.ssl.certificate.validation</code></p>	<p>Required. Determines if certificate validation, hostname verification, and certificate authentication are performed by CSA when making a secure connection (only using HTTPS) with an application or a component of CSA. Examples of an application include Operations Orchestration or a resource provider. Examples of a component of CSA include the Marketplace Portal and the Identity Management component. Other non-HTTP connections that have been configured to be secure are not affected by this property. For example, secure connections to the database, LDAP server, or SMTP server are not affected.</p> <p><b>Note:</b> If CSA is running in a FIPS-compliant environment, this property is not used. In a FIPS-compliant environment, certificate validation, hostname verification, and certificate authentication will always be performed when making a secure connection with CSA.</p> <p>By default, this property is set to <code>false</code>. That is, when CSA establishes a secure connection with another application or component, the connection will only be encrypted. No validation, verification, or authentication is performed. This mode should only be used during post-installation configuration or when troubleshooting problems with certificates. This mode should NOT be used in a production environment.</p> <p>When set to <code>true</code>, when CSA establishes a secure connection with another application or component, the following occurs:</p> <ul style="list-style-type: none"> <li>• The connection will be encrypted</li> <li>• Certificate validation - Checks that the certificate used by the application/component has not expired</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>• Hostname verification - Checks that the certificate hostname matches the URL hostname of the application/component to which CSA is connecting</li> <li>• Certificate authentication - Checks that the certificate or the root certificate used to sign the certificate has been imported into CSA's JRE truststore (for example, CSA_JRE_HOME/lib/security/cacerts)</li> </ul> <p><b>Default:</b> false</p>
com.hp.ccue.consumption disallowedExtensions	<p>A comma-delimited list of the file extensions that designate the types of documents or files that cannot be uploaded to the Cloud Service Management Console.</p> <p><b>Default:</b> exe,bat,com,cmd</p>
csa.additionalSupported ExtensionsForImport	<p>A comma-delimited list of the file extensions that designate the types of documents or files that can be uploaded to the Cloud Service Management Console. The file extensions listed can be the sole extension of the file or the start of the file extension followed by one or more characters. For example, listing txt as a file extension will match both mydocument.txt and mydocument.txt_3491767613.</p> <p>Files can be uploaded using the Cloud Service Management Console, the content archive tool, or the import API. Refer to the <i>Cloud Service Management Console Help</i>, <i>Cloud Service Automation API Guide</i>, or <i>Cloud Service Automation Content Archive Tool</i> for more information about using these features.</p> <p>The following extensions are automatically supported (and do not need to be defined by this property): jpg, jpeg, jpe, jfif, svg, tif, tiff, ras, cmx, ico, pnm, pbm, pgm, ppm, rgb, xbm, xpm, xwd, png, gif, bmp, cod, ief, json, xml, jsp, jspf.</p> <p><b>Default:</b> (no default defined)</p> <p>Example: txt,log</p>
csa.maxFileUploadSize	<p>The maximum size of a file, in megabytes (MB), that can be uploaded to the CSA system using the Cloud Service Management Console. If this property is not listed or is not set in the csa.properties file, the default maximum size of 50 MB is used.</p>

Property	Description
	<b>Default:</b> 50 (MB)
csa.war.images.directory.byteLimit	<p>A total size limit for all images or icons that are uploaded into <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/images</code>. The limit is used to prevent exhausting of server disk space through image upload in UI.</p> <p>Unit: bytes.</p> <p><b>Default:</b> 500000000 bytes (500 MB)</p>
csa.war.images.directory.smallFileByte Overhead	<p>Used when computing space occupied by existing image/icon files (see above <code>csa.war.images.directory.byteLimit</code>). For each file in the images directory, a value of this property is added to its size to account for the overhead of small files on the file system.</p> <p>Unit: bytes.</p> <p><b>Default:</b> 4096 bytes</p>
enableSecurityWarning	<p>Enables/disables the security warning messages for files that are uploaded or downloaded in the Cloud Service Management Console. Value is true or false.</p> <p><code>enableSecurityWarning</code> is in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/offerings/onfig.json</code> file.</p> <p><b>Default:</b> true</p>

### CSA keystore

These properties are used to configure information about Cloud Service Automation's keystore.

These properties are configured in `csa.properties`.

Property	Description
<p><code>csaTruststore</code></p>	<p>Required. The CSA keystore that stores trusted Certificate Authority certificates.</p> <p>Default: No default specified</p> <p><b>Example</b></p> <p><b>Windows:</b> C:\Program Files\HPE\CSA\openjre\lib\security\cacerts</p> <p><b>Linux:</b> /usr/local/hpe/csa/openjre/lib/security/cacerts</p> <p><b>Note:</b> On Windows, use only forward slashes (/) as your path separators.</p>
<p><code>csaTruststorePassword</code></p>	<p>Required. The encrypted password of the CSA keystore (see <a href="#">"Encrypt a password" on page 172</a> for instructions on encrypting passwords). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>Default: ENC(&lt;encrypted_value&gt;) where the default value of &lt;encrypted_value&gt; is the encrypted value of "changeit".</p>

### Service request processor scheduler

These properties are used to configure the service request processor scheduler. The service request processor scheduler validates a consumer's requests, initiates the approval process, if configured, and maintains a request's status.

These properties are configured in `csa.properties`.

Property	Description
<code>serviceRequestProcessorScheduler.maxInstancesToProcess</code>	Optional. The maximum number of service requests the service request processor can process when it checks the start and end dates of submitted subscriptions.  <b>Default:</b> 100
<code>serviceRequestProcessorScheduler.period</code>	Optional. How often, in milliseconds, the service request processor checks the start and end dates of submitted subscriptions.  <b>Default:</b> 5000 (5 seconds)

## Auditing

These properties are used to configure auditing.

These properties are configured in `csa.properties`.

Property	Description
<code>csaAuditEnabled</code>	<p>Optional. Enable or disable auditing, which tracks user activities and system-generated events. Messages are logged to the <code>CSA_AUDIT_EVENT</code> table in the database.</p> <p><b>Default:</b> true (enabled)</p>
<code>jboss.shutdown.log.location</code>	<p>Required. This property is set during installation and <i>must not be changed</i>. The location of the JBoss log file that records when the CSA service was stopped. Used for auditing purposes.</p> <p><b>Default:</b> <code>CSA_HOME/jboss-as/bin/shutdown.log</code></p> <p><b>Note:</b> On Windows, use only forward slashes (/) as your path separators.</p>
<code>csa.origin.ip.header</code>	<p>Optional. Defines a custom HTTP header used to capture the originating IP address of a REST API call. If this property is disabled (commented out) or not set to a value, the standard HTTP header <code>X-Forwarded-For</code> is used to capture the originating IP address. If the originating IP address is not captured by either this custom or the standard header, CSA fetches the originating IP address from the incoming request. The originating IP address is used for auditing.</p> <p>CSA sets the following precedence when capturing the originating IP address of a REST API call:</p> <ol style="list-style-type: none"> <li>1. Uses the custom HTTP header (if defined)</li> <li>2. Uses the <code>X-Forwarded-For</code> header</li> <li>3. Fetches from the incoming request</li> </ol> <p>If this property is set to a custom HTTP header, CSA checks if this custom HTTP header is defined (set to the originating IP address) in the REST API call. If this property is not set or if the custom header is not defined, CSA checks if the <code>X-Forwarded-For</code> header is defined in the REST API call. If the <code>X-Forwarded-For</code> header is not defined, CSA fetches the originating IP address from the incoming request. CSA does not validate the captured value (if the value is an IP address and if it is a valid IP address).</p> <p>The following is a list of CSA REST API types and which ones do and do not capture the originating IP address:</p>

Property	Description
	<ul style="list-style-type: none"><li>• Legacy CSA 3.x APIs: originating IP address IS CAPTURED</li><li>• Consumer (Consumption) APIs that include onBehalf parameter in the Response Content Type (i.e. Consumer APIs that use the POST, PUT, or DELETE methods): originating IP address IS CAPTURED</li><li>• Consumer (Consumption) APIs that do not include onBehalf parameter in the Response Content Type (i.e. Consumer APIs that use the GET method): originating IP address IS NOT CAPTURED</li><li>• Management (Consumption) APIs: originating IP address IS NOT CAPTURED</li></ul> <p>The originating IP address is stored in the ORIGIN_IP field of the RPT_AUDIT_EVENT_V view and the ORIGIN_IP column of the CSA_AUDIT_EVENT table. If the originating IP address is not captured, the field or column is empty.</p> <p><b>Default:</b> (disabled)</p>

**Process execution manager**

These properties are used to configure the process execution manager. The process execution manager starts internal actions and Operations Orchestration flow actions, checks the status of process instances, and performs callback once the actions are completed.

These properties are configured in `csa.properties`.

Property	Description
<code>com.hp.csa.ProcessExecutor.THREAD_WAKEUP_TIME</code>	Optional. How often, in milliseconds, the process execution manager starts new process instances (which start Operations Orchestration flows) and checks the status of process instances.  <b>Default:</b> 5000 (5 seconds)
<code>com.hp.csa.ProcessExecutor.THREAD_POOL_CORE_SIZE</code>	Optional. The maximum number of threads used to run process instances.  <b>Default:</b> 2
<code>com.hp.csa.PEM.PARAM_PROCESS_INSTANCE_ID</code>	Optional. The token that stores the process instance ID and is used when CSA starts an Operations Orchestration flow.  <b>Default:</b> <code>CSA_PROCESS_ID</code>
<code>com.hp.csa.PEM.PARAM_CONTEXT_ID</code>	Optional. The token that stores the artifact ID of the artifact that owns the action that executes the Operations Orchestration flow.  <b>Default:</b> <code>CSA_CONTEXT_ID</code>



## Lifecycle engine

These properties are used to configure the lifecycle engine. The lifecycle engine processes service instances and executes lifecycle actions.

These properties are configured in `csa.properties`.

Property	Description
<code>com.hp.csa.LifecycleExecutor.THREAD_WAKEUP_TIME</code>	Optional. How often, in milliseconds, the lifecycle engine checks for service components that it needs to transition. <b>Default:</b> 5000 (5 seconds)
<code>com.hp.csa.LifecycleExecutor.THREAD_POOL_SIZE</code>	Optional. The maximum number of threads used to transition service components. <b>Default:</b> 2

### Approval engine scheduler

These properties are used to configure the approval engine scheduler. The approval engine scheduler checks each approver's response to a pending approval process to see if the process can be marked as completed and updates the decision and status of an approval process, as needed.

This property is configured in `csa.properties`.

Property	Description
<code>com.hp.csa.ApprovalDecisionMaker.THREAD_POOL_SIZE</code>	Optional. The maximum number of threads used to process approvals. <b>Default:</b> 4
<code>com.hp.csa.ApprovalDecisionMaker.THREAD_WAKEUP_TIME</code>	Optional. How often, in milliseconds, the approval engine scheduler checks for completion of an approval process to determine if an approval process should be approved or denied. <b>Default:</b> 5000 (5 seconds)

### LDAP cache scheduler

These properties are used to configure the LDAP cache scheduler. The LDAP cache scheduler checks the age of the user group cache and deletes it if it has expired.

For users who can log in to the Cloud Service Management Console or Marketplace Portal, certain actions require authorization (verification if the user belongs to a group). When authorization is requested for a user, CSA checks for group membership by using the cache. If the cache does not exist, LDAP is queried for the user's user groups which are temporarily cached to the database. After a configured expiration time, the cache is deleted. During a single session, the cache may be deleted and refreshed as needed.

These properties are configured in `csa.properties`.

Property	Description
<code>com.hp.csa.UserGroupExecutor.THREAD_WAKEUP_TIME</code>	Optional. How often, in minutes, the LDAP cache scheduler checks for user group caches that have expired. This number should be less than the value configured for <code>com.hp.csa.UserGroupExecutor.CACHE_EXPIRATION_TIME</code> .  <b>Default: 20</b>
<code>com.hp.csa.UserGroupExecutor.CACHE_EXPIRATION_TIME</code>	Optional. How long, in minutes, LDAP user groups for a user are temporarily cached in the database before they are deleted. This time should be greater than the value configured for <code>com.hp.csa.UserGroupExecutor.THREAD_WAKEUP_TIME</code> .  <b>Default: 30</b>
<code>com.hp.csa.UserGroupExecutor.UserGroupDeletionBatchSize</code>	Optional. The maximum number of user IDs that are deleted in a single batch from the cache. This number cannot be larger than 1,000.  <b>Default: 250</b>

## Clustering

This property is used to configure clustering.

This property is configured in `csa.properties`.

Property	Description
deploymentMode	<p>Required. The mode in which CSA is running (single or clustered). When set to <code>single</code>, CSA runs in standalone mode (on a single instance) and all CSA services are run on this instance. When set to <code>clustered</code>, CSA runs in a clustered environment and all CSA services run on only one node (which is selected by the cluster as the singleton-service provider).</p> <p><b>Default:</b> <code>single</code></p>
com.hp.csa.LockMonitorService.LOCK_TIMEOUT	<p>Default timeout in milliseconds for the background thread that checks if processes have stale locks. Individual entities may have their own timeout.</p>
com.hp.csa.LockMonitorService.NODE_TIMEOUT	<p>Default timeout in milliseconds for entities that have been locked by a cluster node that is no longer responsive (such as the locking node has shut down or cannot connect to the cluster).</p>

### Dynamic property

These configuration properties are used to limit the amount of time to retrieve data and the amount of data retrieved when using a dynamic property. A dynamic property is a Dynamic Query value entry method for a subscriber option property that defines what information is retrieved. A dynamic property allows the Service Designer to list a dynamic set of values that change based on the user context (for example, the organization to which the user belongs).

These properties are configured in `csa.properties`.

Property	Description
DynamicPropertyFetch.READ_TIMEOUT	Optional. How long, in milliseconds, CSA attempts to fetch or retrieve data for dynamic properties. <b>Default:</b> 30000 (30 seconds)
DynamicPropertyFetch.RESPONSE_SIZE	Optional. The maximum amount of data, in bytes, that can be retrieved for dynamic properties. <b>Default:</b> 50000

**Group approval**

This configuration property is used when configuring a group approval template.

This property is configured in `csa.properties`.

Property	Description
csa.group.numberOfApprovers	Optional. The maximum number of members in an LDAP group used for approvals. For reasonable performance, do not specify more than ten (10) members.  <b>Default: 10</b>

## Marketplace Portal

These properties are the default values displayed in the Cloud Service Management Console that are used to configure the Marketplace Portal for an organization. The values configured in the Cloud Service Management Console take precedence over the values set in this properties file. See ["Appendix: Marketplace Portal Attributes" on page 349](#) for descriptions of the attributes that can be configured for the Marketplace Portal.

These properties are configured in `csa.properties`.

Property	Description
csa.consumer. featuredCategory	<p>Optional. The default value of the Featured Category field displayed in the Cloud Service Management Console of a selected organization. This value may be overwritten in the Cloud Service Management Console. The value configured in the Cloud Service Management Console takes precedence over this value.</p> <p>This is the category that is used when displaying service offerings in the Marketplace Portal.</p> <p>The value entered for this attribute is the name of a category configured in the Cloud Service Management Console but is in all capitalized letters and replaces any spaces with an underscore (_). For example, if you configure a category named <b>e-mail Servers</b> and want to feature this category, you would set this attribute to <b>E-MAIL_SERVERS</b>.</p> <ul style="list-style-type: none"> <li>• ACCESSORY</li> <li>• APPLICATION_SERVERS - Default.</li> <li>• APPLICATION_SERVICES</li> <li>• BACKUP_SERVICES</li> <li>• CRM</li> <li>• DATABASE_SERVERS</li> <li>• FILE_SERVERS</li> <li>• HARDWARE</li> <li>• MAIL_SERVICES</li> <li>• NETWORK_SERVICES</li> <li>• PLATFORM_SERVICES</li> <li>• SIMPLE_SYSTEM</li> <li>• SOFTWARE</li> <li>• WEB_HOSTING_SERVICES</li> </ul> <p>For more information about the featured services, refer to the</p>

Property	Description
	<p><i>Marketplace Portal Help.</i></p> <p><b>Default:</b> APPLICATION_SERVERS</p>
csa.consumer.endDatePeriod	<p>Optional. The default value of the Subscription End Date field displayed in the Cloud Service Management Console of a selected organization. This value may be overwritten in the Cloud Service Management Console by a lower value. The value configured in the Cloud Service Management Console takes precedence over this value.</p> <p>This is the maximum length of a subscription, in months, if a requested end date is specified. When a subscriber selects a requested start date and requests an end date, the length of the subscription cannot be longer than the value of this property. The maximum allowed value is 12 months. For example, if the subscriber selects a requested start date of June 15, 2015, based on the default value of this property, the requested end date cannot be later than June 14, 2016. If <b>no end date</b> is selected, this value is ignored.</p> <p><b>Default:</b> 12 (months)</p>
csa.consumer.legalNoticeUrl	<p>Optional. The default value of the Privacy Statement Link field displayed in the Cloud Service Management Console of a selected organization. This value may be overwritten in the Cloud Service Management Console. The value configured in the Cloud Service Management Console takes precedence over this value.</p> <p>This is a link to an organization's privacy statement and, when enabled in the Cloud Service Management Console, appears on the login page below the copyright statement.</p> <p><b>Default:</b> The online privacy statement.</p>
csa.consumer.termsOfUseUrl	<p>Optional. The default value of the Terms and Conditions Link field displayed in the Cloud Service Management Console of a selected organization. This value may be overwritten in the Cloud Service Management Console. The value configured in the Cloud Service Management Console takes precedence over this value.</p> <p>This is a link to an organization's terms and conditions statement and, when enabled in the Cloud Service Management Console, appears when a subscriber is ordering a service.</p> <p><b>Default:</b> The terms of use statement.</p>

### FIPS 140-2 configuration on Windows

These configuration properties are used to configure CSA on Windows to be compliant with FIPS 140-2.



**Note:** The `csaTruststore` and `csaTruststorePassword` properties are repeated here because you may need to update them for FIPS 140-2 configuration. These properties are configured in a different section of the `csa.properties` file.

These properties are configured in `csa.properties`.

Property	Description
<p><code>useExternalProvider</code></p>	<p>Required if enabling FIPS 140-2 compliance mode. To enable, set this property to true. To disable, set this property to false or comment it out.</p> <p>When enabled, CSA uses the RSA BSAFE libraries to encrypt and decrypt passwords. If a password was encrypted using different libraries (for example, if the password was encrypted before this property is enabled), the resulting decrypted password will not be valid.</p> <p>If you cannot connect to the database after you have configured CSA for FIPS 140-2 compliance, try re-encrypting the database password in the database properties file.</p> <p><b>Default:</b> commented out/disabled</p>
<p><code>securityProviderName</code></p>	<p>Required if FIPS 140-2 compliance mode is enabled. The name of the FIPS 140-2 compliant provider. By default, CSA uses the RSA BSAFE provider and this property should be set to <code>JsafeJCE</code>.</p>
<p><code>keySize</code></p>	<p>Optional. The key size used for CSA encryption. By default, the key size is 128. If you manually enter a different key size when encrypting a password, uncomment this property and configure the value to the key size used to encrypt the passwords.</p> <p><b>Note:</b> All passwords must be encrypted using the same key size.</p> <p>By default, the password encryption utility encrypts all passwords using a key size of 128 (even if you do not specify a key size when running the utility).</p>
<p><code>keystore</code></p>	<p>Required if FIPS 140-2 compliance mode is enabled. The absolute path to and file name of the CSA encryption keystore. This is the keystore that supports PKCS #12 and stores the key used by CSA to encrypt and decrypt data in CSA.</p> <p><b>Example</b> (this example uses the same example name from the <i>Create a CSA Encryption Keystore</i> section in the <i>Cloud Service Automation FIPS 140-2 Compliance Configuration Guide</i>):</p> <p><code>CSA_HOME/jboss-as/standalone/configuration/csa_encryption_keystore.p12</code></p>

Property	Description
	<p><b>Note:</b> On Windows, use only forward slashes (/) as your path separators.</p>
keyAlias	<p>Required if FIPS 140-2 compliance mode is enabled. The alias used to identify the CSA encryption key in the CSA encryption keystore.</p> <p><b>Example</b> (this example uses the same example name from the <i>Create a CSA Encryption Keystore</i> section in the <i>Cloud Service Automation FIPS 140-2 Compliance Configuration Guide</i>):</p> <pre>csa_encryption_key</pre>
keystorePasswordFile	<p>Required if FIPS 140-2 compliance mode is enabled. The absolute path to and file name of the CSA encryption keystore password. This is a temporary file that stores the CSA encryption keystore password in clear text. This file is required to start the CSA service and is automatically deleted when the service is started.</p> <p>The password file must contain only the following content:</p> <pre>keystorePassword=&lt;CSA encryption keystore password&gt;</pre> <p>where &lt;CSA encryption keystore password&gt; is the CSA encryption keystore password in clear text.</p> <p><b>Note:</b> On Windows, use only forward slashes (/) as your path separators.</p>
encryptedKeyFile	<p>Required if FIPS 140-2 compliance mode is enabled. The location of the CSA encrypted symmetric key.</p> <p><b>Example</b> (this example uses the same example name from the <i>Create a CSA Encryption Keystore</i> section in the <i>Cloud Service Automation FIPS 140-2 Compliance Configuration Guide</i>):</p> <pre>CSA_HOME/jboss-as/standalone/configuration/key.dat</pre> <p><b>Note:</b> On Windows, use only forward slashes (/) as your path separators.</p>
csaTruststore	<p>Required. The CSA keystore that stores trusted Certificate Authority certificates.</p> <p><b>Note:</b> This property is located in another section of the <code>csa.properties</code> file. Its description is repeated here as its value should be updated when CSA has been configured to be compliant with FIPS 140-2.</p> <p><b>Example</b> (this example uses the same example name of the CSA server truststore from the <i>Create a CSA Encryption Keystore</i></p>

Property	Description
	<p>section in the <i>Cloud Service Automation FIPS 140-2 Compliance Configuration Guide</i>):</p> <p>CSA_HOME/jboss-as/standalone/configuration/csa_server_truststore.p12</p> <p><b>Note:</b> On Windows, use only forward slashes (/) as your path separators.</p>
<p>csaTruststorePassword</p>	<p>Required. The encrypted password of the CSA keystore (see <a href="#">"Encrypt a password" on page 172</a> for instructions on encrypting passwords). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p>Default: ENC(&lt;encrypted_value&gt;)            where the default value of &lt;encrypted_value&gt; is the encrypted value of "changeit".</p> <p><b>Note:</b> This property is located in another section of the <code>csa.properties</code> file. Its description is repeated here as its value should be updated when CSA has been configured to be compliant with FIPS 140-2.</p> <p>This is the &lt;CSA server truststore password&gt; from the <i>Create a CSA Encryption Keystore</i> section in the <i>Cloud Service Automation FIPS 140-2 Compliance Configuration Guide</i>):</p>

## Common Access Card

These properties are used to enable integration between Common Access Card (CAC) and CSA and to extract a user name from the subjectDN X.509 attribute.

These properties are configured in `csa.properties`.

Property	Description
<code>enableCAC</code>	Optional. Enable integration between CAC and CSA, where the CAC is used as an approval mechanism. To enable, this property must be uncommented and set to <code>true</code> . To disable, either comment out the property or set it to <code>false</code> .  Default: <code>false</code> (disabled)
<code>csa.cac.regex</code>	The regular expression used to extract a user name from the subjectDN X.509 attribute. If this property is not set, then the default for <code>regex</code> is <code>CN=(.*)</code> . This property need not be set if the property <code>csa.cac.x509Attribute</code> is set to "san".  <b>Note:</b> To retrieve the data between the parentheses from the subjectDN X.509 attribute, use the filter <code>csa.cac.regex=\\((.*)\\)</code> .
<code>idm.cac.regex</code>	The regular expression used to extract a user name from the subjectDN X.509 attribute. If this property is not set, then the default for <code>regex</code> is <code>CN=(.*)</code> . This property need not be set if the property <code>idm.cac.x509Attribute</code> is set to "san".  <b>Note:</b> To retrieve the data between the parentheses from the subjectDN X.509 attribute, use the filter <code>csa.cac.regex=\\((.*)\\)</code> .

## Single Sign-On

This property is used to enable integration between CA SiteMinder and CSA. SSO can be used when launching an application, such as HPE IT Business Analytics, from the Cloud Service Management Console.

This property is configured in `csa.properties`.

Property	Description
<code>enableSSO</code>	Enables SSO post install if you want to enable Siteminder SSO.  This property must be uncommented and set to <code>true</code> to enable integration between CA SiteMinder and CSA, where SiteMinder is

Property	Description
	<p>used for single sign-on. In all other cases, either comment out this property or set it to <code>false</code> to disable it.</p> <p><b>Default:</b> <code>false(disabled)</code></p>

### Process executor delegate

These properties are used to configure the process executor delegate. The process executor delegate handles processing of the process instances. It discovers the ready instances, submits them to different thread pools for processing based on process definition and model type (sequenced or topology).

These properties are configured in `csa.properties`.

Property	Description
com.hp.csa.service.process.ProcessExecutorDelegate. INTERNAL_POOL_SIZE	Optional. The maximum number of threads used for processing internal executors (for example, clone patterns).  <b>Default: 2</b>
com.hp.csa.service.process.ProcessExecutorDelegate. EXTERNAL_POOL_SIZE	Optional. The maximum number of threads used for processing external executors (for example, Operations Orchestration).  <b>Default: 2</b>
com.hp.csa.service.process.ProcessExecutorDelegate. CALLBACK_POOL_SIZE	Optional. The maximum number of threads used by the callback pool.  <b>Default: 2</b>
com.hp.csa.service.process.ProcessExecutorDelegate. MONITOR_POOL_SIZE	Optional. The maximum number of threads used by the monitor pool.  <b>Default: 2</b>

### Miscellaneous

The following are miscellaneous properties that do not fall under any specific category.

These properties are configured in `csa.properties`.

Property	Description
com.hp.csa.aosMonitor. THREAD_WAKEUP_TIME	Optional. How often, in milliseconds, the background thread monitors plug-in processes.  <b>Default: 20000</b>
com.hp.csa.TimeoutChecker. THREAD_WAKEUP_TIME	Optional. How often, in milliseconds, the background thread monitors for processes that have timed out.  <b>Default: 300000</b>

Property	Description
com.hp.csa.ExportSvcOffering.THREAD_WAKEUP_TIME	<p>Defines the background service wakeup time to export non-posted offerings, subscriptions and instances into elasticsearch. When the CSA service starts, the background service wakes up. If there are no records to be exported to elasticsearch then the background services dies immediately. Otherwise the background service exports records into elasticsearch in the batches of the property defined in com.hp.csa.ExportSvcOffering.FETCH_SIZE. The background service continues to run until it processes all the non-posted records available in the CSA database.</p> <p>If the background service is not running, it wakes-up again according to the time defined in this property. The value of this property should be in milliseconds.</p>
com.hp.csa.ExportSvcOffering.FETCH_SIZE	<p>Defines the number of records to be processed at a time. The SQL used to fetch the records from the CSA database, uses this property value to limit the number of records that can be fetched from the database and then exported to elasticsearch.</p>
com.hp.csa.plugin.cloudos.util.TokenCache.TIMEOUT	<p>Identity Management component token cache timeout, in milliseconds.</p> <p>Every REST call to CSA (such as for provisioning) is authenticated by Identity Management. CSA uses trustId to get the authentication token from Identity Management. Because these REST calls can be more frequent, this property allows you to define the cache timeout to prevent enormous sizes during the REST call's authentication lifecycle.</p> <p><b>Default</b> value: 300000 (5 minutes)</p> <p>Value 0 disables cache</p>
com.hp.csa.import.BUILD_ARTIFACT_RELATIONSHIP	<p>Disables the artifact relationship section of the import/preview results.</p>
loggerEnabled	<p>Enables the logging filter for the legacy REST APIs, so that the requesting user and artifact information is logged.</p>

Property	Description
csa.productPerspective	Determines which version of CSA has been installed: Enterprise or Codar.
jdbc.dialect	<p>Holds explicitly set Hibernate dialect for a given database. Recommended values for the databases are:</p> <ul style="list-style-type: none"><li data-bbox="878 506 1377 596">• MSSQL: org.hibernate.dialect.SQLServer2008Dialect</li><li data-bbox="878 621 1349 680">• Oracle: org.hibernate.dialect.Oracle10gDialect</li><li data-bbox="878 705 1373 764">• PostgreSQL: org.hibernate.dialect.PostgreSQLDialect</li></ul>



## Operations Orchestration

These properties are used to integrate with Operations Orchestration.

These properties are configured in `csa.properties`.

The following properties configure the interaction between the Cloud Service Management Console and Operations Orchestration. In the subscription event overview section of the **[[[Undefined variable CSAVariables.tabOperations]]]** area in the Cloud Service Management Console, selecting the Process ID opens Operations Orchestration to the detailed page of the selected process when these properties are configured.

Property	Description
OOS_URL	<p>The URL used to access Operations Orchestration Central. This is the Operations Orchestration used for provisioning topology designs. For example, <code>https://&lt;hostname&gt;:8445</code>.</p> <p>This property is automatically set during installation. If you are using the embedded Operations Orchestration that is included with CSA, this property is set using the values entered for the <b>Fully qualified domain name</b> on Windows or the <b>Fully Qualified Hostname</b> on Linux and <b>HPE OO Port</b> fields during installation. If you are using a standalone/external Operations Orchestration, this property is set using the values entered for the <b>HPE OO Hostname</b> and <b>HPE OO Port</b> fields during installation.</p>
OOS_USERNAME	<p>The username used to log in to Operations Orchestration Central.</p> <p>This property is automatically set during installation using the value entered for the <b>HPE OO User Name</b> field during installation.</p>
OOS_PASSWORD	<p>The encrypted password used by the user defined in OOS_USERNAME to log in to Operations Orchestration Central.</p> <p>This property is automatically set during installation using the value entered for the <b>HPE OO Password</b> field during installation.</p>
embedded.oo.root.dir	<p>Location of the embedded Operations Orchestration when it is installed with CSA. This property is generated when embedded Operations Orchestration is installed during the CSA installation.</p>

Property	Description
	This property is the only indicator of embedded Operations Orchestration, which is important mainly for uninstallation and upgrades. This property cannot be edited.

The following properties configure background services to monitor Operations Orchestration.

Property	Description
com.hp.csa.oo.OOClient.SOCKET_TIMEOUT	Optional. How long, in milliseconds, CSA keeps a socket open for SOAP-based communication with Operations Orchestration.  <b>Default:</b> 60000
com.hp.csa.OosMonitor.THREAD_WAKEUP_TIME	Optional. How often, in milliseconds, the background thread monitors Operations Orchestration processes.  <b>Default:</b> 60000
com.hp.csa.service.process.OosMonitorDelegate.MONITOR_POOL_SIZE	Optional. The maximum number of threads used by the monitor pool.  <b>Default:</b> 2
OOS_MASTER_OOFLOW_CONTENT_LOCATION	The location in Operations Orchestration where CSA generates topology design-based master Operations Orchestration flows and related subflows. The folder structure must use forward slashes.  <b>Default:</b> Library/CSA/Topology_Generated_Flows

### CSA 3.x API authentication

These properties are used to configure authentication for the CSA 3.x API.

These properties are configured in `csa.properties`.

Property	Description
xAuthToken	<p>Optional. An optional token in the Authorization header used for HTTP basic authentication by the CSA 3.x API. If the token is sent, it is used to authenticate the <code>userIdentifier</code> parameter in the REST API. For more information about the CSA API, see the <i>Cloud Service Automation API Quick Start Guide</i>.</p> <p><b>Default:</b> X-Auth-Token</p>
integrationAccountUserList	<p>Required. A comma-delimited list of users who are authorized to exercise the CSA 3.x API. The username in the Authorization header used for HTTP basic authentication must match one of the users in this list.</p> <p>By default, the following CSA built-in users are configured: <code>admin</code>, <code>csaCatalogAggregationTransportUser</code>, <code>csaReportingUser</code>, <code>csaTransportUser</code>, <code>oolInboundUser</code>, and <code>cdalInboundUser</code>. You can also add LDAP users (identified by the User ID) to this list. For example, if you use email addresses for the User ID, you could add <code>user1@xyz.com</code> to the list.</p> <p>For more information about the CSA API, see the <i>Cloud Service Automation API Quick Start Guide</i>.</p> <p><b>Default:</b>  <code>admin,csaReportingUser,oolInboundUser,cdalInboundUser,csaTransportUser,csaCatalogAggregationTransportUser</code></p>

## Topology Designer

These properties are used to configure the features of topology designs.

These properties are configured in `csa.properties`.

Property	Description
TopologyDesignProvisioning.TIMEOUT	<p>Optional. The amount of time, in seconds, CSA attempts to provision or de-provision a topology design that is not based on an Helion OpenStack® provider (topology design provisioning and de-provisioning is orchestrated by interacting with resource providers corresponding to the components used in the design).</p> <p>If the time is exceeded, in the Operations area of the Cloud Service Management Console, the subscription (to a service offering that is created from a topology design that is not based on an Helion OpenStack® provider) will show a Subscription Status of <code>Failed</code> and a Service Instance Status of <code>Failed</code>. If you select the Events tab of the subscription, the event will show a Status of <code>Timeout</code>. If you select the Topology tab of the subscription, the topology view will show the status of the components in the service instance as their respective status just before the timeout occurred.</p> <p>It is recommended that this value be set to the same value as the Operations Orchestration flow timeout value.</p> <p><b>Default:</b> 7200 (2 hours)</p>
OrchestratedTopologyDesignProvisioning.ProviderSelection.Enabled	<p>Optional. Enable or disable resource environment and provider selection by the subscriber in the Marketplace Portal for service offerings based on topology designs that are not based on an Helion OpenStack® provider. For more information, refer to the <i>Cloud Service Management Console Help</i>.</p> <p><b>Default:</b> true (enabled)</p>

## Elasticsearch

These properties are used to integrate global search with CSA.

These properties are configured in `csa.properties`.

Property	Description
csa.provider.es.exists	<p>Required. Enable or disable the global search feature on this CSA node. If enabled, additional <a href="#">microservice properties</a> may be configured.</p> <p>To enable the global search feature, set this property to <b>yes</b>.</p> <p>In a FIPS 140-2 compliant environment on Windows, this property must be set to <b>no</b>.</p> <p><b>Default:</b> yes (enabled)</p>
csa.provider.es.authUser	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The user used by the Elasticsearch service to authenticate requests coming from CSA. It is recommended that you create a user specifically for this purpose.</p> <p>If the CSA built-in <code>consumer</code> user is disabled or another user is used, either another built-in user or LDAP user must be configured. If using a built-in user, this user must have the <code>SERVICE_CONSUMER</code> role configured. If using an LDAP user, this user must be assigned to the Service Consumer role.</p> <p><b>Default:</b> consumer</p>
csa.provider.es.authPassword	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The encrypted password of the <code>csa.provider.es.authUser</code> user.</p> <p>The password should be encrypted (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses.</p> <p><b>Default:</b> <i>&lt;encrypted password of the consumer user&gt;</i></p>
csa.provider.es.authOrganization	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The name of the organization to which the <code>csa.provider.es.authUser</code> user belongs.</p> <p>The organization is used only for authentication purposes. The Elasticsearch service will index the service offerings, service instances, or subscriptions for all organizations. However, global search results for a Marketplace Portal user will be limited to the service offerings, service instances, or subscriptions of the organization to which the user belongs and to which the user has access.</p> <p>If the CSA built-in <code>CONSUMER</code> organization is disabled or removed, the <code>csa.provider.es.authUser</code>, <code>csa.provider.es.authPassword</code>, and</p>

Property	Description
	<p><code>csa.provider.es.auth0organization</code> properties must be updated to use a valid user and organization.</p> <p><b>Default:</b> CONSUMER</p>
<code>csa.provider.es.idmURL</code>	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The URL used to generate Identity Management component tokens for Elasticsearch service authentication. If a CSA cluster is configured for high availability using a load balancer, <code>localhost</code> must be changed to the hostname or IP address of the system on which the load balancer is running.</p> <p><b>Default:</b> <code>https://localhost:8444/idm-service</code></p>

### Microservices

These properties are used to configure the HPE Search Service, which creates the indices for Elasticsearch. The Elasticsearch property, `csa.provider.es.exists`, must be enabled for these properties to take effect.

These properties are configured in `csa.properties`.

Property	Description
<code>csa.provider.msvc.hostname</code>	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The fully-qualified domain name of the system on which the HPE Search Service is running or <code>localhost</code>.</p> <p><b>Default:</b> <code>localhost</code></p>
<code>csa.provider.msvc.port</code>	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The port used to connect to the system on which the HPE Search Service is running.</p> <p><b>Default:</b> 9000</p>
<code>csa.provider.msvc.rest.protocol</code>	<p>Required if <code>csa.provider.es.exists</code> is enabled (set to yes). The protocol used by the REST API to connect to the system on which the HPE Search Service is running.</p> <p><b>Default:</b> <code>https</code></p>

### LDAP access point

This property is used to enable or disable access to the LDAP access point configuration in the Cloud Service Management Console.

This property is configured in `csa.properties`.

Property	Description
csa.IdapReadOnly	<p>Required. Enable or disable access to the LDAP access point configuration in the Cloud Service Management Console.</p> <p>By default, the property is set to false and the CSA administrator can configure the LDAP access point of any organization from the Cloud Service Management Console (the LDAP access point is typically configured when an organization is created in the Cloud Service Management Console). LDAP configuration includes fields for the LDAP Server Information, LDAP Attributes, and User Login Information in the Cloud Service Management Console. The LDAP access point is used by CSA for authentication and authorization.</p> <p>For security reasons, you may not want to allow the CSA administrator to configure the LDAP access point from the Cloud Service Management Console. You can disable access to the LDAP access point fields for all organizations from the Cloud Service Management Console by setting this property to true (disabling access makes the LDAP configuration fields read-only in the Cloud Service Management Console). By disabling this access, only the system administrator or other privileged users on the CSA system can update the LDAP access point using the LDAP Configuration Tool. Refer to the <i>LDAP Configuration Tool</i> guide for more information about the LDAP Configuration Tool.</p> <p>To enable access to the LDAP access point configuration in the Cloud Service Management Console, set this property to false. To disable access to the LDAP access point configuration in the Cloud Service Management Console, set this property to true.</p> <p><b>Default:</b> false</p>

**Service Design, Service Offering, and Catalog Content archive verification**

This property is used to enable or disable service design, service offering, and catalog content archive verification.

This property is configured in `csa.properties`.

Property	Description
csa.security.enable	<p>Required. Enable or disable service design, service offering, and catalog content archive verification.</p> <p>By default, the property is set to false (verification is disabled), allowing the Cloud Service Management Console or Content Archive Tool to import a service design, service offering, or catalog content archive directly without verification.</p> <p>When the property is set to true (verification is enabled), CSA verifies the digital signature of the content archive, validates the date of the certificate used to sign the content archive, and verifies that the content in the content archive has not been modified after it was signed. If the content archive fails one of these validation or verification checks, the content archive will not be imported into CSA.</p> <p>When enabled, all imported service design, service offering, or catalog content archives must be signed. Refer to <a href="#">"Signing the Content Archive" on page 162</a> for the steps required to sign a content archive.</p> <p><b>Note:</b> Verifying service designs and catalogs before they are imported is done using the Cloud Service Management Console or the Content Archive Tool. Verifying service offerings before they are imported is done using the Content Archive Tool.</p> <p><b>Caution:</b> Verification cannot be enabled for importing a service design, service offering, or catalog content archive using the REST APIs. A service design, service offering, or catalog content archive imported using the REST APIs will always be imported directly. Verification can only be enabled for the Cloud Service Management Console or the Content Archive Tool.</p> <p><b>Default:</b> false</p>



**HPE CSA-ITOC Integration version 1 (v1)**

These properties are used to enable integration between CSA and IT Operations Compliance (ITOC).

These properties are configured in `csa.properties`.

Property	Description
<code>csa.ITOC.Integration.enabled</code>	Optional. Enable or disable integration between CSA and ITOC. To enable, this property must be uncommented and set to true. To disable, either comment out the property or set it to false.  <b>Default:</b> (disabled)
<code>csa.ITOC.Notification.BaseUri</code>	Required if integration between CSA and ITOC is enabled. To enable, this property must be uncommented and set to the endpoint of the ITOC instance. The endpoint is the URL for connecting to the ITOC instance where <code>&lt;protocol&gt;</code> is the protocol used to communicate with the ITOC instance (for example, http or https), <code>&lt;itoc_host&gt;</code> is the hostname of the ITOC instance, and <code>&lt;port&gt;</code> is the port used to connect to the system on which ITOC is running.  <b>Default:</b> (disabled)
<code>csa.ITOC.Notification.username</code>	Required if integration between CSA and ITOC is enabled. To enable, this property must be uncommented and set to the username used to log in to the ITOC instance.  <b>Default:</b> (disabled)
<code>csa.ITOC.Notification.password</code>	Required if integration between CSA and ITOC is enabled. To enable, this property must be uncommented and set to the encrypted password used by the user defined in <code>csa.ITOC.Notification.username</code> to log in to the ITOC instance. (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.  <b>Default:</b> (disabled)
<code>csa.ITOC.Notification.tenant</code>	Required if integration between CSA and ITOC is enabled. To enable, this property must be uncommented and set to the tenant group to which the user defined in <code>csa.ITOC.Notification.username</code> belongs.  <b>Default:</b> (disabled)

**HPE CSA-ITOC Integration version 2 (v2)**

Property	Description
csa.ITOC.IntegrationV2.enabled	<p>Optional. Enable or disable integration v2 between CSA and ITOC. To enable, this property must be uncommented and set to true. To disable, either comment out the property or set it to false.</p> <p><b>Default:</b> (disabled)</p>
csa.ITOC.IntegrationV2.ITOCBaseUri	<p>Required if integration v2 between CSA and ITOC is enabled. To enable, this property must be uncommented and set to the endpoint of the ITOC instance. The endpoint is the URL for connecting to the ITOC instance where <i>&lt;protocol&gt;</i> is the protocol used to communicate with the ITOC instance (for example, http or https), <i>&lt;itoc_host&gt;</i> is the hostname of the ITOC instance, and <i>&lt;port&gt;</i> is the port used to connect to the system on which ITOC is running.</p> <p><b>Default:</b> (disabled)</p>
csa.ITOC.IntegrationV2.public.<user>=<password>	<p>Required if integration v2 between CSA and ITOC is enabled. To enable, this property must be uncommented and set to the ITOC integration credentials. It is used to access ITOC public tenant to get the list of policies and maintenance windows.</p> <p>Username is part of the configuration key and encrypted password is value of the configuration key.</p> <p>(see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p> <p><b>Default:</b> (disabled)</p>
csa.ITOC.IntegrationV2.<tenant>.<user>=<password>	<p>Required if integration v2 between CSA and ITOC is enabled. To enable, this property must be uncommented and set to integration credentials for each consumer tenant of ITOC. Tenant is part of the configuration key, username is part of the configuration key and encrypted password is value of the configuration</p>

Property	Description
	key. (see <a href="#">"Encrypt a password" on page 172</a> for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.  <b>Default:</b> (disabled)
csa.ITOC.IntegrationV2.timeout=10000	Required if integration v2 between CSA and ITOC is enabled. To enable, this property must be uncommented and set to connection timeout in milliseconds.  <b>Default:</b> (disabled)

### Session Timeout

This property is used to configure the Cloud Service Management Console session.

This property is configured in `web.xml`.

Property	Description
session-timeout	Optional. The amount of inactivity, in minutes, that causes the Cloud Service Management Console session to time out.  <b>Default:</b> 60

### REST

These properties are used to configure the REST response.

These properties are configured in `csa.properties`.

Property	Description
rest.restrict.fields	A comma separated list of the fields that are not included in the REST response. By default the <code>rest.restrict.fields</code> property includes these fields: <code>createdBy</code> , <code>updatedBy</code> , <code>createdOn</code> , <code>updatedOn</code> , <code>description</code> , <code>iconUrl</code> , and <code>categoryType</code> .  For details see "Values for the restrict parameter" in the <i>Cloud Service Automation API Guide</i> .
rest.restrict	Enable or disable the fields specified in the <code>rest.restrict.fields</code>

Property	Description
	<p>property to be excluded/included in the output of the REST response.</p> <p>If set to true, the fields are excluded in the output of the REST response.</p> <p>If set to false, the fields are included in the output of the REST response.</p> <p><b>Default:</b> false</p> <p>For details see "Values for the restrict parameter" in the <i>Cloud Service Automation API Guide</i>.</p>
rest.excludedoc	<p>Enable or disable the document field to be excluded/included in the output of the REST response.</p> <p>If set to true, the document field is excluded in the output of the REST response.</p> <p>If set to false, the document field is included in the output of the REST response.</p> <p><b>Default:</b> false</p> <p>For details see "Values for the excludedoc parameter" in the <i>Cloud Service Automation API Guide</i>.</p>

# Appendix: Marketplace Portal Attributes

This section lists and describes the attributes that can be configured for the Marketplace Portal. Recommended modifications to the values can be found in the related feature's section in this guide or other documentation (for example, see ["Appendix: Identity Management component Configuration" on page 362](#) for more information about the Identity Management component-related attributes).

The attributes are located in the following files:

- `CSA_HOME/portal/conf/mpp.json`
- `CSA_HOME/portal/node_modules/mpp-consumption/dist/offerings/config.json`

where

`CSA_HOME` is the directory in which CSA is installed

.

The following areas contain attributes that can be configured (for many attributes, default values are provided):

- [General Marketplace Portal Attributes](#)
- [Shopping Cart Attributes](#)
- [Provider Attributes](#)
- [Identity Management component Attributes](#)
- [Security Attributes](#)
- [High availability Attributes](#)
- [Logging Attributes](#)
- [Proxy Server Attributes](#)

## General Marketplace Portal Attributes

These attributes are general purpose attributes that can be configured for the Marketplace Portal.

Attribute	Description
uid	A unique identifier of the Marketplace Portal process used only on Linux systems.

Attribute	Description
	Default: ccue_mpp
port	<p>The port used to connect to the system on which the Marketplace Portal is running.</p> <p>The port configured for the Marketplace Portal in this attribute should match the port value configured for the <code>csa.subscriber.portal.url</code> property in the <code>CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties</code> file.</p> <p>Default: 8089</p>
defaultOrganizationName	<p>The organization identifier of the organization that is accessed by the Marketplace Portal when the Marketplace Portal is launched from a URL that does not specify the organization. The organization identifier is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the <b>Organizations</b> menu of the Cloud Service Management Console).</p> <p>Default: CONSUMER</p>
defaultHelpLocale	<p>The language in which the online help is presented. Available languages can be found in the <code>CSA_HOME/portal/node_modules/mpp-ui/dist/ccue-marketplaceportal-help/help/&lt;defaultHelpLocale&gt;</code> directory.</p> <p>Default: en_US (English)</p>
defaultHelpPage	<p>The name of the help file that is launched if there is no context-sensitive help available for a topic.</p> <p>The page is relative to <code>CSA_HOME/portal/node_modules/mpp-ui/dist/ccue-marketplaceportal-help/help/&lt;defaultHelpLocale&gt;</code> and uses the <code>defaultHelpLocale</code> to determine which language to use.</p> <p>Default: MarketplacePortal_Help_CSA.htm</p>
keyfile	<p>The file that contains the Marketplace Portal's encrypted symmetric key and is used by the Marketplace Portal to encrypt and decrypt data in the Marketplace Portal. The path to the file can be absolute or relative to the <code>CSA_HOME/portal/bin</code> directory.</p> <p>If this file does not exist, it can be generated using the <code>CSA_HOME/portal/bin/passwordUtil</code> utility (see <a href="#">"Encrypt a Marketplace Portal Password" on page 180</a> for more information).</p> <p>Default: ../conf/keyfile</p>

Attribute	Description
rejectUnauthorized	<p>Allows the Marketplace Portal to accept or reject requests based on the type of certificate passed. If enabled (set to true), the Marketplace Portal will only accept requests that use a Certificate Authority-signed or subordinate Certificate Authority-signed certificate and it will reject requests that use a self-signed certificate.</p> <p>If disabled (set to false), the Marketplace Portal will accept requests that use a Certificate Authority-signed, subordinate Certificate Authority-signed certificate, or a self-signed certificate.</p> <p>Default: false</p>
session: cookieSecret	<p>The authentication cookie used to verify if a user is logged in and to encrypt the user's identification.</p> <p>The cookie/password should be encrypted (see "<a href="#">Encrypt a Marketplace Portal Password</a>" on page 180 for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p>
session: timeoutDuration	<p>The amount of inactivity, in seconds, that causes the Marketplace Portal session to time out.</p> <p>Default: 1800 (30 minutes)</p>
session: cleanupInterval	<p>How often, in seconds, a background process is run to clean up expired sessions.</p> <p>Default: 3600 (1 hour)</p>

### Shopping Cart Attributes

These attributes are used to configure the shopping cart for the Marketplace Portal.

Attribute	Description
thresholdQuantity	<p>The minimum number of items in a shopping cart that, upon submission, may delay response time of the submission.</p> <p>Default: 20</p>
maximumQuantity	<p>The maximum number of items in a shopping cart that can be submitted.</p> <p>Default: 100</p>

### Provider Attributes

These attributes are used to configure how the Marketplace Portal interacts with CSA.

Attribute	Description
url	<p>The URL to access CSA.</p> <p>Default: <code>https://localhost:8444</code></p>
contextPath	<p>The context path to access CSA.</p> <p>Default: <code>/csa/api/mpp</code></p>
strictSSL	<p>When enabled, when the Marketplace Portal establishes a secure connection to CSA, the following occurs:</p> <ul style="list-style-type: none"> <li>• The connection will be encrypted</li> <li>• Certificate validation - Checks that the certificate used by CSA has not expired</li> <li>• Hostname verification - Checks that the certificate hostname matches the URL hostname of the CSA system to which the Marketplace Portal is connecting</li> <li>• Certificate authentication - Checks that the certificate or the root certificate used to sign the certificate is listed in the file defined by the <code>ca</code> attribute</li> </ul> <p>When enabled, if the hostname configured for the certificate is not valid, access is denied to the Marketplace Portal. To check if this is causing access problems to the Marketplace Portal, look for the following error message in the <code>CSA_HOME/portal/logs/mpp.log</code> file:</p> <pre>ERROR GetPost : java.security.cert.CertificateException: No name matching &lt;csa.provider.hostname&gt; found</pre> <p>When disabled, when the Marketplace Portal establishes a secure connection to CSA, the connection will be encrypted. Certificate validation, hostname verification, and certificate authentication do not occur.</p> <p>Default: <code>true</code> (enabled)</p>
TLSVersions	<p>Used to specify TLS versions directly. Multiple comma-separated values are accepted. Versions accepted are “1.0” (alternatively “1”), “1.1”, and “1.2”. Change values only in coordination with other TLS Version configurations to ensure client-server compatibility. Using only latest version(s) increases security, but it may prevent compatibility.</p> <p>Example: “1.2”</p> <p>By default, CSA supports secure connections using TLSv1.2 and to enable support for TLSv1.2, you must configure the load balancer. CSA configuration can be manually changed to support TLSv1.1 or TLSv1.0, to work with older load balancers or other HTTPS clients that do not support TLSv1.2. However it is not recommended to enable TLSv1.1 or TLSv1.0 for security reasons.</p>
ca	<p>Used only when <code>strictSSL</code> is enabled. The path to and name of the file that is an actual certificate or contains a comma-delimited list of certificates for CSA, which may include Certificate Authority-signed and self-signed certificates. If you are using</p>



Attribute	Description
	<p>a self-signed certificate, it must be listed in this file. The path to the file can be absolute or relative to the <code>CSA_HOME/portal/bin</code> directory.</p> <p>The certificates must be in a PEM or DER format.</p> <p>To use the self-signed certificate generated during the installation of CSA, set this attribute's value to <code>CSA_HOME\jboss-as/standalone/configuration/jboss.cer</code> where <code>CSA_HOME</code> is the directory in which CSA is installed</p>

### Identity Management component Attributes

These attributes are used to configure how the Marketplace Portal interacts with the Identity Management component.

Attribute	Description
url	<p>The URL to access the Identity Management component.</p> <p>Default: <code>https://localhost:8444</code></p>
returnUrl	<p>If proxy configuration is enabled, this is the URL to which the Identity Management component is redirected after authentication has succeeded.</p> <p>Default: <code>https://localhost:8089</code></p>
contextPath	<p>The context path to access the Identity Management component.</p> <p>Default: <code>/idm-service</code></p>
username	<p>The name of the account used by CSA to authenticate REST API calls.</p> <p>Default: <code>idmTransportUser</code></p>
password	<p>The encrypted password for the username (see <a href="#">"Encrypt a Marketplace Portal Password" on page 180</a> for instructions). An encrypted password is preceded by <code>ENC</code> without any separating spaces and is enclosed in parentheses. See <a href="#">"Change CSA Built-In User Accounts" on page 205</a> for more information about this account.</p>
strictSSL	<p>When enabled, when the Marketplace Portal establishes a secure connection to the Identity Management component, the following occurs:</p> <ul style="list-style-type: none"> <li>• The connection will be encrypted</li> <li>• Certificate validation - Checks that the certificate used by the Identity Management component has not expired</li> <li>• Hostname verification - Checks that the certificate hostname matches the URL hostname of the Identity Management component system to which the</li> </ul>

Attribute	Description
	<p>Marketplace Portal is connecting</p> <ul style="list-style-type: none"> <li>• Certificate authentication - Checks that the certificate or the root certificate used to sign the certificate is listed in the file defined by the <code>ca</code> attribute</li> </ul> <p>When enabled, if the hostname configured for the certificate is not valid, access is denied to the Marketplace Portal. To check if this is causing access problems to the Marketplace Portal, look for the following error message in the <code>CSA_HOME/portal/logs/mpp.log</code> file:</p> <pre>ERROR GetPost : java.security.cert.CertificateException: No name matching &lt;csa.provider.hostname&gt; found</pre> <p>When disabled, when the Marketplace Portal establishes a secure connection to the Identity Management component, the connection will be encrypted. Certificate validation, hostname verification, and certificate authentication do not occur.</p> <p>Default: true (enabled)</p>
TLSVersions	<p>Used to specify TLS versions directly. Multiple comma-separated values are accepted. Versions accepted are “1.0” (alternatively “1”), “1.1”, and “1.2”. Change values only in coordination with other TLS Version configurations to ensure client-server compatibility. Using only latest version(s) increases security, but it may prevent compatibility.</p> <p>Example: “1.2”</p> <p>By default, CSA supports secure connections using TLSv1.2 and to enable support for TLSv1.2, you must configure the load balancer. CSA configuration can be manually changed to support TLSv1.1 or TLSv1.0, to work with older load balancers or other HTTPS clients that do not support TLSv1.2. However it is not recommended to enable TLSv1.1 or TLSv1.0 for security reasons.</p>
ca	<p>Used only when <code>strictSSL</code> is enabled. The path to and name of the file that is an actual certificate or contains a comma-delimited list of certificates for the Identity Management component, which may include Certificate Authority-signed and self-signed certificates. If you are using a self-signed certificate, it must be listed in this file. The path to the file can be absolute or relative to the <code>CSA_HOME/portal/bin</code> directory.</p> <p>The certificates must be in a PEM or DER format.</p> <p>To use the self-signed certificate generated during the installation of CSA, set this attribute's value to <code>CSA_HOME/jboss-as/standalone/configuration/jboss.cer</code> where <code>CSA_HOME</code> is the directory in which CSA is installed</p>

## Security Attributes

These attributes are used to configure security settings for the Marketplace Portal.

Attribute	Description
enabled	<p>Determines the protocol used by the Marketplace Portal. If enabled (set to true), the Marketplace Portal uses the HTTPS protocol. If disabled (set to false), the Marketplace Portal uses the HTTP protocol.</p> <p>The options listed below are used only when this attribute is enabled. Additional options may be specified and are defined at <a href="http://nodejs.org/api/tls.html#tls_tls_createserver_options_secureconnectionlistener">http://nodejs.org/api/tls.html#tls_tls_createserver_options_secureconnectionlistener</a>.</p> <p>Default: true</p>
options: pfx	<p>The file that contains the Marketplace Portal's private key, self-signed certificate, and Certificate Authority-signed certificates (also known as a PKCS #12 archive). The path to the file can be absolute or relative to the CSA_HOME/portal/bin directory.</p> <p>Default: ../conf/.mpp_keystore</p>
options: passphrase	<p>The encrypted password used to access the pfx (see "<a href="#">Encrypt a Marketplace Portal Password</a>" on page 180 for instructions). An encrypted password is preceded by ENC without any separating spaces and is enclosed in parentheses.</p>
options: TLSVersions	<p>Used to specify TLS versions directly. Multiple comma-separated values are accepted. Versions accepted are "1.0" (alternatively "1"), "1.1", and "1.2". Change values only in coordination with other TLS Version configurations to ensure client-server compatibility. Using only latest version(s) increases security, but it may prevent compatibility.</p> <p>Example: "1.2"</p> <p>By default, CSA supports secure connections using TLSv1.2 and to enable support for TLSv1.2, you must configure the load balancer. CSA configuration can be manually changed to support TLSv1.1 or TLSv1.0, to work with older load balancers or other HTTPS clients that do not support TLSv1.2. However it is not recommended to enable TLSv1.1 or TLSv1.0 for security reasons.</p>
enableSecurityWarning	<p>Enables/disables the security warning messages for files that are uploaded or downloaded in the Marketplace Portal. Value is true or false.</p> <p>enableSecurityWarning is in the CSA_HOME\portal\node_modules\mpp-consumption\dist\offerings\config.json file.</p> <p>Default: true</p>

## High Availability Attributes

These attributes are used to configure the Marketplace Portal in a clustered environment. For more information about how to configure CSA in a clustered environment, (which disables these attributes), see the *Cloud Service Automation Cluster Configuration Guide Using a Load Balancer* guide.

Attribute	Description
enabled	Determines the environment in which the Marketplace Portal is running. If enabled (set to true), the Marketplace Portal is running in a clustered environment. If disabled (set to false), the Marketplace Portal is running in a standalone environment.  Default: false
numWorkers	The number of workers on which to deploy the Marketplace Portal. Each worker is deployed on each CPU and is therefore bound by the number of CPUs on the host.  Default: 2
redis: options: host	The hostname of the system on which the Redis data structure server is running.  Default: localhost
redis: options: port	The port to connect to the Redis data structure server.  Default: 6379

### Logging Attributes

These attributes are used to configure logging.

Attribute	Description
console: enabled	Determines if messages are written to the console. If enabled (set to true), messages are displayed in the console. If disabled (set to false), messages are not displayed in the console.  Default: false
console: level	The level of logging. For example, error, warn, info, debug, or trace.  Default: info
file: enabled	Determines if messages are written to a log file. If enabled (set to true), messages are logged to a file (CSA_HOME/portal/logs/mpp.log). If disabled (set to false), messages are not logged to a file.  Default: true
file: level	The level of logging. For example, error, warn, info, debug, or trace.  Default: info
file:	The maximum size to which the log file can grow, in megabytes, before it is archived.

Attribute	Description
maxSizeMB	Default: 10
file: maxFile	The maximum number of archived log files. Default: 10
cef: enabled	If the Marketplace Portal logging has been integrated with ArcSight Logger, determines if log events are sent and stored in ArcSight Logger. If enabled (set to true), log events are sent and stored in ArcSight Logger. If disabled (set to false), log events are not sent and stored in ArcSight Logger.  For information about CSA and ArcSight Logger integration, see the <i>Integration with ArcSight Logger</i> technical white paper.  Default: false
cef: host	The hostname of the system on which the ArcSight Logger is installed. Default: localhost
cef: port	The port used to connect to the system on which the ArcSight Logger is installed. Default: 9876
cef: level	The level of logging. For example, error, warn, info, or debug. Default: warn

### Proxy Attributes

These attributes are used to configure proxy settings for the Marketplace Portal.

Attribute	Description
enabled	Determines if a proxy (an alternate URL using a different port and context path) is used to access the Marketplace Portal (for example, you may need to use a proxy, such as <code>http://localhost:8090/mpp</code> instead of <code>http://localhost:8089</code> , when the Marketplace Portal is integrated with a single sign-on solution). If enabled (set to true), the Marketplace Portal uses a proxy. If enabled, you must update the <code>returnUrl</code> attribute to use the proxy for the Identity Management component (this attribute is also located in the <code>mpp.json</code> file).  If disabled (set to false), the Marketplace Portal does not use a proxy.  Default: false
port	The port used for proxying. Default: 8090

Attribute	Description
contextPath	The mount path to which the Marketplace Portal is forwarded. Default: /mpp

# Appendix: OO Settings

This section is provided as a reference only.

The following areas contain settings that can be configured from OO Studio:

- [Remote Action Services](#)
- [System Accounts](#)
- [System Properties](#)

## Remote Action Services

Setting	Description
RAS_ Operator_ Path	<p>Required. The name and URL that accesses the RAS used by OO Central.</p> <p>Recommend the following value:</p> <p><code>https://&lt;FQDN&gt;:9004/RAS/services/RCAgentService</code></p> <p>where &lt;FQDN&gt; is the fully qualified domain name or IP address of the Operations Orchestration host. Do not use localhost in the URL. Using localhost does not work correctly even though it appears to work when you run Operations Orchestration Studio on the same machine as the RAS.</p> <p>RAS must be run on the same system as Operations Orchestration Studio. Running Operations Orchestration Studio on another machine produces errors and turns flows red with a cryptic error message about result assignments to result variables that do not exist.</p>

## System Accounts

Setting	Description
CSA_REST_ CREDENTIALS	<p>Required. Credentials for CSA REST authentication.</p> <p>It is recommended that you set the Credentials to the following values:</p> <ul style="list-style-type: none"><li>• <b>User Name:</b> oolnboundUser</li><li>• <b>Password:</b> cloud</li></ul> <p><b>Note:</b> The <b>User Name</b> configured for the CSA_REST_CREDENTIALS</p>

### System Accounts, continued

Setting	Description
	System Account setting must match the <b>Property Value</b> (Operations Orchestration version 9.07) or <b>Override Value</b> (Operations Orchestration version 10.50) configured for the CSA_OO_USER System Property setting.

### System Properties

Setting	Description
CSA_DMA_WorkflowTimeout	Required. The amount of time, in seconds, to wait for a DMA workflow to complete.  Default Property Value:  3600
CSA_NA_CreateVlanScript	Required. The name of the Network Automation command script to create a VLAN that was imported when you integrated Network Automation with CSA.  Default Property Value:  HPN Create Vlan
CSA_NA_DeleteVlanScript	Required. The name of the Network Automation command script to delete a VLAN that was imported when you integrated Network Automation with CSA.  Default Property Value:  HPN Delete Vlan
CSA_OO_USER	Required. The user that communicates with CSA using the REST API.  Default Property Value:  oolnboundUser  <b>Note:</b> The <b>Property Value</b> (Operations Orchestration version 9.07) or <b>Override Value</b> (Operations Orchestration version 10.50) configured for the CSA_OO_USER System Property setting must match the <b>User Name</b> configured for the CSA_REST_CREDENTIALS System Account setting.
CSA_REST_URI	Required. The URI used to communicate with CSA using the REST API.



**System Properties, continued**

Setting	Description
	<p>Recommend the following Property Value:</p> <p><code>https://&lt;csa_hostname&gt;:8444/csa/rest</code></p>
<p>CSA_SiteScope_MonitoringLockId</p>	<p>Required. SiteScope monitoring lock ID.</p> <p>Default Property Value:</p> <p>SiteScope Lock for Deploying Monitors</p>
<p>CSA_SiteScope_RootMonitorGroup</p>	<p>Required. The default name of the SiteScope root monitor group path.</p> <p>Default Property Value:</p> <p>CSA Monitors</p>
<p>CSA_SiteScope_MonitoringSleepTime</p>	<p>Required. The amount of time, in seconds, to wait before acquiring the SiteScope monitoring lock. This time may be increased if there are a large number of subscription requests.</p> <p>Default Property Value:</p> <p>30</p>
<p>CSA_vCenterPropertyCollectionTimeout</p>	<p>Required. How often, in seconds, properties are collected about a deployed virtual machine.</p> <p>Default Property Value:</p> <p>1800</p>

# Appendix: Identity Management component Configuration

If you are using the Identity Management component, the identity service and its components require configuration. Because it is a Spring Framework application, most of its configuration is defined in the `applicationContext.xml` file, although key attributes are externalized to the `applicationContext.properties` file. Both files are in `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/`.

You should make most common configuration changes to the `applicationContext.properties` file. To avoid service disruptions, only advanced users who understand the Spring Framework should change the `applicationContext.xml` file.

You must also configure the Java Relying Party Library.

**Note:** You should always make a copy of a configuration file before editing it.

## External Configuration

Selected settings are pulled from the `applicationContext.properties` file, which you can override by an external properties file set as a JVM argument: `-Didm.properties=<external_properties_filename>`. You can add this JVM argument to the `JAVA_OPTS` environment variable or by editing the `standaloneconf.bat` file on Windows or the `standalone.conf` file on Linux in `CSA_HOME/jboss-as/bin/` to add it to `JAVA_OPTS` for the CSA JBoss container.

The table below describes the properties that are set in the properties file. These properties are required.

If you are integrating with Keystone, the `idm.keystone*` properties must match the Keystone network location, transport user credentials, and so on. All `idm.csa*` properties and all `ConvergedLdapAuthConfig` properties (which are listed in the "[ConvergedLdapAuthConfig](#)" on page 367 section below) must match the CSA network location and transport user credentials.

Property Name	Description
<code>idm.ssl.requireValidCertificate</code>	Flag indicating whether valid certificates are required: true or false

Property Name	Description
<code>idm.csa.protocol</code>	The protocol used to access the CSA instance: <code>http</code> or <code>https</code>
<code>idm.csa.hostname</code>	The hostname or IP address of the CSA server
<code>idm.csa.port</code>	The port number used by the CSA server
<code>idm.csa.username</code>	The username for the CSA integration account
<code>idm.csa.password</code>	The password for the CSA integration account. For improved security, this value should be encrypted.
<code>idm.encryptedSigningKey</code>	The shared signing key for all token factory objects. For improved security, this value should be encrypted.
<code>idm.keystone.enabled</code>	<p>Flag indicating whether secondary authentication through Keystone is enabled: <code>true</code> or <code>false</code>.</p> <ul style="list-style-type: none"> <li>Set to <code>true</code> when you want to generate secondary authentication tokens via keystone authentication from your registered OpenStack providers.</li> <li>Set to <code>false</code> when attempting to use CAC or Siteminder for authentication. If <code>idm.keystone.enabled</code> is set to <code>true</code> in these cases, the keystone authentication will not function correctly.</li> </ul> <p>The following <code>idm.keystone</code> properties need not be configured as these values will be ignored:</p> <pre> idm.keystone.protocol = http idm.keystone.hostname = keystone idm.keystone.port = 5000 idm.keystone.servicePath = v3 idm.keystone.domainName = Default idm.keystone.transportUsername = consumer idm.keystone.transportPassword = encrypted idm.keystone.transportProject = CSA_CONSUMER </pre>

## Configure Seeded Authentication

The top-level configuration file for seeded authentication in the Identity Management component is specified by the `configFile` property of the `SeededAuthenticationProvider` bean defined in the

`applicationContext.xml` configuration file.

In the default configuration, this file is `seededorgs.properties`, but it can be changed. Each line in this file contains a key-value pair. The key is a CSA organization ID, and the value is the name of another properties file Identity Management component users for that organization.

**Notes:**

- This seeded authentication configuration only applies to Identity Management component seeded users.
- CSA does not support adding new seeded users to the Identity Management component configuration for any CSA organizations.
- CSA does not support modifying the Identity Management component configuration to use existing seeded users with any new CSA organizations.

By default, the following organizations in the Identity Management component are configured to use the specified files.

Organization	User File
CONSUMER	<code>consumer-users.properties</code>

You can define additional Identity Management component organizations or change the user file associated with any organization. Each line in each user file contains a key-value pair. The key is the username, and the value is a comma-separated list of the password, granted authorities, and an optional flag indicating whether the account is enabled. For improved security, the *entire* value should be encrypted. Following is an example of a line from an Identity Management component user file that defines a user named `consumer` with the password `c1oud` and granted the `SERVICE_CONSUMER` and `ROLE_REST` authorities.

```
consumer=c1oud,SERVICE_CONSUMER,ROLE_REST,enabled
```

## Configure the Java Relying Party Library

The Java Relying Party Library is a set of classes provided by the identity service that abstract and simplify invoking the service from Java applications, such as CSA. You modify the properties listed in this section in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/applicationContext-security.xml` file. The `tokenFactory` property value should be the same for all `AuthProvider` beans (listed in the *Internal Configuration* section below) in the identity service and in the Java Relying Party library.

## IdentityServiceConfig

Configures the connection to the identity service.

**Class:** `com.hp.ccue.identity.rp.IdentityServiceConfig`

Property Name	Description
<code>protocol</code>	The protocol ( <code>http</code> or <code>https</code> ) to use to connect to the identity service
<code>hostname</code>	The hostname or IP address of the server running the identity service
<code>port</code>	The port number where the identity service is running, typically <code>8444</code>
<code>servicePath</code>	The path on the server to the identity service, typically <code>idm-service</code>

## IdentityAuthenticationProvider

Abstracts the invocation of the identity service to perform authentication.

**Class:** `com.hp.ccue.identity.rp.IdentityAuthenticationProvider`

Property Name	Description
<code>templateFactory</code>	Creates the <code>RestTemplate</code> object that facilitates performing REST calls
<code>configuration</code>	Network configuration of the identity service to connect to perform authentication: an <code>IdentityServiceConfig</code> object
<code>tokenFactory</code>	The token factory to validate returned tokens
<code>tenantHeaderName</code>	The name of the HTTP header where the tenant name is passed. The default is <code>HPE-Tenant-Name</code>

## HeaderAuthenticationProvider

Performs authentication based on a token passed in an HTTP header.

**Class:** `com.hp.ccue.identity.rp.HeaderAuthenticationProvider`

Property Name	Description
headerName	The name of the HTTP header where the token is transferred
tokenValidator	The TokenValidator object to use to validate tokens

## Internal Configuration

The `applicationContext.xml` file defines the configuration of the classes in the identity service. The `tokenFactory` property value should be the same for all `AuthenticationProvider` beans (listed in the sections below) in the identity service and in the Java Relying Party library.

**Note:** Modify this file only if you cannot express the necessary configuration change in the `applicationContext.properties` file. The `applicationContext.xml` file must follow the syntax rules specified by the Spring Framework. In the following tables, the default values are used if no values are provided in the configuration file. You can configure items marked as externalized in the `applicationContext.properties` file.

## JwtTokenFactory

Defines how tokens are created.

**Class:** `com.hp.ccue.identity.domain.JwtTokenFactory`

Property Name	Description
lifetimeMinutes	Required. The lifetime of the token, in minutes. The lifetime as installed is 30 minutes. Reducing this value will render tokens invalid faster and thus requires a more-frequent token refresh, which might reduce performance. Increasing this value allows tokens to last longer, which might allow someone who has intercepted a valid token to access the system for a period of time.  Default value: (None)  Externalized: No
defaultTypeName	Optional. Default type of JWT token to create: PLAINTEXT, SIGNED, or ENCRYPTED  Default value: PLAINTEXT  Externalized: No
signingKey	Required if <code>defaultTypeName</code> is set to SIGNED. This is a Base64-encoded byte

Property Name	Description
	<p>array representing the key used to sign signed tokens. If <code>defaultTypeName</code> is set to <code>SIGNED</code>, this value must be the same for all components that validate tokens. For improved security, this item should be encrypted.</p> <p>Default value: (None)</p> <p>Externalized: <code>idm.encryptedSigningKey</code></p>
<code>refreshEnabled</code>	<p>Optional. Boolean value indicating whether token refresh is enabled: <code>true</code> or <code>false</code>. The recommended value is <code>true</code>.</p> <p>Default value: <code>true</code></p> <p>Externalized: No</p>

## ConvergedLdapAuthConfig

Defines the configuration for connecting to a CSA server to get LDAP configuration information. The `idm.csa*` external properties (which are listed in the *External Configuration* section above) and all `ConvergedLdapAuthConfig` properties must match the CSA network location and transport user credentials.

**Class:** `com.hp.ccue.identity.ldap.ConvergedLdapAuthConfig`

Property Name	Description
<code>providerProtocol</code>	<p>Required if using ActiveDirectory or LDAP. <code>http</code> or <code>https</code>, depending on the protocol used by the CSA instance</p> <p>Default value: (None)</p> <p>Externalized: <code>idm.csa.protocol</code></p>
<code>providerHostname</code>	<p>Required if using ActiveDirectory or LDAP. Hostname or IP address of the CSA server</p> <p>Default value: (None)</p> <p>Externalized: <code>idm.csa.hostname</code></p>
<code>providerPort</code>	<p>Required if using ActiveDirectory or LDAP. Port number used by the CSA server</p> <p>Default value: (None)</p> <p>Externalized: <code>idm.csa.port</code></p>
<code>securityTransportUsername</code>	<p>Required if using ActiveDirectory or LDAP. Username for the CSA integration account</p>

Property Name	Description
	Default value: (None) Externalized: idm.csa.username
securityTransportPassword	Required if using ActiveDirectory or LDAP. Password for the CSA integration account Default value: (None) Externalized: idm.csa.password

## ConvergedActiveDirectoryAuthenticationProvider and ConvergedLdapAuthenticationProvider

Performs authentication with Active Directory and LDAP authentication mechanisms.

**Class:** com.hp.ccue.identity.ldap.ConvergedActiveDirectoryAuthenticationProvider, com.hp.ccue.identity.ldap.ConvergedLdapAuthenticationProvider

Property Name	Description
config	Required if using ActiveDirectory or LDAP. The ConvergedLdapAuthConfig that represents the CSA server to use to get the LDAP configuration for each organization Default value: (None) Externalized: No
tokenFactory	Required if using ActiveDirectory or LDAP. The token factory for creating identity tokens in response to successful authentications Default value: (None) Externalized: No

## SeededAuthenticationProvider

Performs seeded authentication.

**Class:** com.hp.ccue.identity.seeded.SeededAuthenticationProvider



Property Name	Description
configFile	Required if using seeded authentication. Typically <code>seededorgs.properties</code> , which is the file that defines the seeded organizations  Default value: (None)  Externalized: No
tokenFactory	Required if using seeded authentication. The token factory for creating identity tokens in response to successful authentications  Default value: (None)  Externalized: No

## IdentityAuthenticationProvider

Performs integration account authentication.

**Class:** `com.hp.ccue.identity.seeded.IntegrationAuthenticationProvider`

Property Name	Description
configFile	Required. Typically <code>integrationusers.properties</code> , which is the file that defines the seeded organizations  Default value: (None)  Externalized: No
tokenFactory	Required. The token factory for creating identity tokens in response to successful authentications  Default value: (None)  Externalized: No

## MultiTenantAuthenticationProvider

Connects to mechanism-specific authentication providers.

**Class:** `com.hp.ccue.identity.authn.MultiTenantAuthenticationProvider`

Property Name	Description
providers	Required. List of AuthenticationProvider objects that provide mechanism-specific authentication  Default value: (None)  Externalized: No
secondaryEnabled	Required if using Keystone. Flag that indicates whether the secondary authentication path (Keystone) is enabled  Default value: false  Externalized: idm.keystone.enabled
secondaryProvider	Required if using Keystone. Reference to Authentication provider bean to use for secondary authentication path. The Keystone authentication provider is the only one that supports this type of usage.  Default value: (None)  Externalized: No

## IdentityServiceImpl

The identity service implementation object.

**Class:** com.hp.ccue.identity.service.IdentityServiceImpl

Property Name	Description
provider	Required. Reference to the AuthenticationProvider bean to use to perform authentication. This is the MultiTenantAuthenticationProvider  Default value: (None)  Externalized: No
tokenFactory	Required. The token factory for creating identity tokens in response to successful authentications  Default value: (None)  Externalized: No
queryService	Required. The persistence service that provides all persistence operations.  Default value: (None)  Externalized: No

Property Name	Description
trustFactory	Required. The TrustFactory for validating all Trust objects. Default value: (None) Externalized: No

## IdentityController

The controller object that provides the REST API for the identity service.

**Class:** `com.hp.ccue.identity.service.IdentityController`

Property Name	Description
identityService	Required. The IdentityService object that implements the identity service. You must set the value of this to the IdentityServiceImpl instance. Default value: (None) Externalized: No

## KeystoneAuthenticationProvider

Uses Keystone (if used) to perform authentication.

**Class:** `com.hp.ccue.identity.keystone.KeystoneAuthenticationProvider`

Property Name	Description
templateFactory	Required. Creates the RestTemplate object that facilitates performing REST calls Default value: (None) Externalized: No
tokenFactory	Required. The token factory to validate returned tokens Default value: (None) Externalized: No

## KeystoneSecondaryAuthenticationProvider

Uses Keystone (if used) to perform authentication.

**Class:** com.hp.ccue.identity.keystone.KeystoneSecondaryAuthenticationProvider

Property Name	Description
keystoneConfigurations	Required. Associative array mapping configuration identifiers to KeystoneConfig objects defining network configurations to connect to one or more Keystone services.  Default value: (None)  Externalized: No
configurationFile	Required. Filename for properties file that contains Keystone configurations.  Default value: (None)  Externalized: No
tokenFactory	Required. The token factory to validate returned tokens.  Default value: (None)  Externalized: No
templateFactory	Required. Creates the RestTemplate object that facilitates performing REST calls.  Default value: (None)  Externalized: No

## RestTemplateFactoryImpl

Configures how REST services are invoked.

**Class:** com.hp.ccue.identity.rest.RestTemplateFactoryImpl

Property Name	Description
fipsEnabled	A flag that indicates whether the template factory should ignore settings that interfere with FIPS 140-2 compliance  Default value: false

Property Name	Description
	Externalized: No
wrapEnabled	<p>A flag that indicates whether the template factory should wrap JSON output in its specified root value or assume that incoming JSON is wrapped in the root value. This setting depends on the REST service being invoked. For template factories used to invoke CSA REST APIs, it should be set to <code>false</code>; for template factories used to invoke Keystone REST APIs, it should be set to <code>true</code>.</p> <p>Default value: <code>true</code></p> <p>Externalized: No</p>
requireValidCertificate	<p>A flag that indicates whether the template factory should perform certificate validation and hostname verification (<code>true</code>) or ignore them (<code>false</code>). If this value is set to <code>true</code>, then the corresponding server host names for all beans that use that template factory must be given in a way that matches the certificate for that server (a fully-qualified domain name is generally required).</p> <p>Default value: <code>true</code></p> <p>Externalized: <code>idm.ssl.requireValidCertificate</code></p>

## TrustFactory

Configures how the Identity Management component trusts are created and validated.

**Class:** `com.hp.ccue.identity.domain.impersonation.TrustFactory`

Property Name	Description
lifetime	<p>Required. The lifetime of a trust.</p> <p>Default value: 90 (days)</p> <p>Externalized: No</p>
lifetimeMinutes	<p>Required. Alternate setter for trust lifetime, expressed in minutes (write only).</p> <p>Default value: (None)</p> <p>Externalized: No</p>
lifetimeHours	<p>Required. Alternate setter for trust lifetime, expressed in hours (write only).</p> <p>Default value: (None)</p> <p>Externalized: No</p>

Property Name	Description
lifetimeDays	Required. Alternate setter for trust lifetime, expressed in days (write only). Default value: (None) Externalized: No

## Token Store Cleanup Service

This background service is used to cleanup expired tokens in the Identity Management component `token_store` database table. The service runs twice a day ( 7 PM and 7 AM ) every day by default and cleans up tokens older than one hour from the time the service starts up.

The default settings should work for most workloads. However under heavy workloads such as OO callbacks to CSA in sequenced designs, the table may need to be cleaned up more often. When there are over 50K rows in the `token_store` database table in a single day, the overall performance of the database and eventually CSA and Marketplace Portal are greatly impacted. This is particularly true for the Microsoft SQLserver 2012 database.

If you see the `token_store` table growing beyond 50K rows in a single day, modify the following cron setting in the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-services.xml` file to run the service more often.

To modify the cron setting, complete the following steps:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-services.xml` file.
2. Look for the following line:

```
<task:scheduled ref="tokenStoreCleaner" method="cleanDbTokenStore" cron="0 0
7/12 * * *" />
```

3. Change the cron setting to have the service run every 2 hours after the start of CSA:

Change: `cron="0 0 7/12 * * *`

To: `cron="0 0 */2 * * *"`

# Appendix: Operations Orchestration Manual Configuration for Designs

The CSA solution includes a number of OO flows that perform CSA operations. This appendix describes how to configure OO for topology and sequential designs without using the Cloud Content Capsule Installer.

**Note:** If you followed the instructions in the *Cloud Service Automation Installation Guide* or *Cloud Service Automation Upgrade Guide* to configure Operations Orchestration, you should have already completed the tasks in this section.

In this release, you can install Operations Orchestration with CSA using the CSA installer or you can install Operations Orchestration externally. Only one instance of Operations Orchestration is required for both topology and sequential designs. If you have upgraded from an earlier version of CSA that uses multiple instances of Operations Orchestration for sequential designs, you can continue to use the multiple instances of Operations Orchestration for sequential designs. If you have upgraded from an earlier version of CSA that uses only a single instance of Operations Orchestration or are installing CSA for the first time, only one configured instance of Operations Orchestration is supported.

This appendix describes the following tasks:

- ["Configure OO for Topology Designs" on page 77](#)
- ["Configure OO for Sequential Designs" on page 87](#)

**Note:** If you are configuring OO for both topology and sequential designs, complete the configuration for topology designs before the configuration for sequential designs.

## Manually Configure Operations Orchestration for Topology Designs

The following tasks are to configure OO for topology designs. Configure only one instance of OO for topology designs without using the Cloud Content Capsule Installer.

**Note:** If you followed the instructions in the *Cloud Service Automation Installation Guide* or *Cloud*

*Service Automation Upgrade Guide* to configure Operations Orchestration, you should have already completed the tasks in this section.

Complete the following tasks to configure OO to integrate with CSA:

- Configure a secure connection between CSA and OO
- Configure an internal user
- Deploy content packs
- Update the Service Manager base content pack
- Configure properties in CSA
- Configure Single Sign-On
- Obscure passwords in OO flows (optional)

**Note:** In the following instructions, `CSA_HOME` is the directory in which CSA is installed and `OO_HOME` is where you installed Operations Orchestration.

Be sure all the latest patches for Operations Orchestration have been installed. See the *Cloud Service Automation System and Software Support Matrix* for more information.

Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com> (this site requires a Passport ID). Select **Dashboards > Manuals**.

## Configure a Secure Connection between CSA and OO

Export Operations Orchestration's certificate from Operations Orchestration's truststore. If Operations Orchestration and CSA are not installed on the same system, copy the certificate to the CSA system and import the certificate into CSA's truststore. TLS must be configured between CSA and Operations Orchestration.

Do the following:

1. On the system running Operations Orchestration, open a command prompt and change to the directory where Operations Orchestration is installed.
2. Run the following command:



**Windows:**

```
.\java\bin\keytool -export -alias tomcat -file C:\oo.cer
-keystore .\Central\var\security\key.store -storepass changeit
```

**Linux:**

```
./java/bin/keytool -export -alias tomcat -file /tmp/oo.cer
-keystore ./Central/var/security/key.store -storepass changeit
```

where C:\oo.cer on Windows and /tmp/oo.cer on Linux are examples is an example of a filename and location used to store the exported root certificate (you can choose a different filename and location).

3. If Operations Orchestration is not running on the same system as CSA, copy oo.cer from the Operations Orchestration system to the system running CSA.
4. On the system running CSA, open a command prompt.
5. Run the following command:

**Windows:**

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias tomcat -file C:\oo.cer -
trustcacerts -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

**Linux:**

```
CSA_JRE_HOME/bin/keytool -importcert -alias tomcat -file /tmp/oo.cer -
trustcacerts -keystore CSA_JRE_HOME/lib/security/cacerts
```

where CSA\_JRE\_HOME is the directory in which the JRE that is used by CSA is installed

6. When prompted for the keystore password, enter changeit.
7. Enter yes when prompted to trust the certificate.

## Configure an Internal User

Internal users can be used to configure Operations Orchestration for CSA.

This user is used for provisioning topology designs.

1. From the system on which CSA is installed (the system on which the content packs are installed), log in to OO Central.
2. Click **System Configuration**.

3. Select **Security > Internal Users**.
4. Click the **+** (Add) icon.
5. Enter the following information:

Field	Recommended Value
User Name	admin
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The admin user is used with HP Single Sign-On (HPSSO). When Operations Orchestration is launched from the Cloud Service Management Console, this user allows access to Operations Orchestration without having to log in. If you are using topology designs, the admin user can also be used for provisioning topology designs.

6. Click **Save**.

## Deploy Content Packs

The following groups of content packs must be deployed in the order described below:

- Base content packs
- Component Tool content packs
- CSA content packs
- Codar content packs (optional)

**Note:** You can also deploy content packs from the **Content Store**. Refer to online help for more details.

**Note:** Do not deploy the Component Tool and CSA content packs until after you have deployed the base content packs. These content packs must be deployed separately from the base content packs and after you have deployed the base content packs.

1. a. From Operations Orchestration Central, click **Content Management**.
  - b. Click the **Content Packs** tab.
  - c. Click the **Deploy New Content** icon.

- d. In the **Deploy New Content** dialog, in the upper left corner, click the + (Add files for deployment) icon.
- e. Deploy the Component Tool content packs. From Operations Orchestration Central, navigate to the `tools/oo/content-packs` directory. Add and deploy the content packs (after each successful deployment, to add and deploy the next content pack without closing the dialog, click the Reset icon in the upper left corner to clear the dialog and enable the + (Add files for deployment) icon).

The deployment may take a few minutes and the dialog will show a progress bar. When you have finished deploying all the content packs, click **Close** to close the dialog.

## Update and Redeploy the Service Manager Base Content Pack

Update and redeploy the `oo10-sm-cp-1.0.3.jar` base content pack. If you deployed an earlier version of the Service Manager base content pack, you must do the following (if this is a fresh installation of Operations Orchestration and you did not deploy an earlier version of the Service Manager base content pack, you do not have to complete these steps):

1. Stop the Operations Orchestration services:

### Windows:

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the HPE Operations Orchestration Central service and select **Stop**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Stop**.

### Linux:

- a. On the server that hosts Operations Orchestration, run the following command:  

```
<HPE00installation>/central/bin/central stop
```

  
For example, `/usr/local/hpe/csa/00/central/bin/central stop`
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost),

run the following command: `<HPE00installation>/ras/bin/ras stop`.

For example, `/usr/local/hpe/csa/00/ras/bin/ras stop`

2. Clear the Operations Orchestration Central cache by deleting the following folder:

`<HPE00installation>/central/var/cache`

For example,

**Windows:** `C:\Program Files\HPE\HPE Operations Orchestration\central\var\cache`

**Linux:** `/usr/local/hpe/csa/oo/central/var/cache`

3. If RAS is installed, clear the RAS artifact cache by deleting the following folder (on all RAS systems, including localhost):

`<HPE00installation>/ras/var/cache`

For example,

**Windows:** `C:\Program Files\HPE\HPE Operations Orchestration\ras\var\cache`

**Linux:** `/usr/local/hpe/csa/oo/ras/var/cache`

4. Run the following SQL command against the Operations Orchestration database:

```
DELETE from OO_ARTIFACTS where NAME =
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.pom' or NAME =
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.jar'
```

5. Start the Operations Orchestration services:

**Windows:**

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click the HPE Operations Orchestration Central service and select **Start**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Start**.

**Linux:**

- a. On the server that hosts Operations Orchestration, run the following command:

```
<HPE00installtion>/central/bin/central start
```

For example, /usr/local/hpe/csa/00/central/bin/central start

- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: <HPE00installtion>/ras/bin/ras start.

For example, /usr/local/hpe/csa/00/ras/bin/ras start

6. Redeploy the oo10-sm-cp-1.0.3.jar base content pack:
  - a. Log in to Operations Orchestration Central and click **Content Management**.
  - b. Click the **Content Packs** tab.
  - c. Click the **Deploy New Content** icon.
  - d. In the Deploy New Content dialog, in the upper left corner, click the + (Add files for deployment) icon.
  - e. Navigate to the CSA\_HOME/oo/ooContentPack directory and select **oo10-sm-cp-1.0.3.jar**.
  - f. Click **Deploy**.

The deployment may take a few minutes and the dialog will show a progress bar.

- g. Click **Close**.

## Configure OO Properties in the csa.properties File

If you integrated with OO using the installer (during the installation or upgrade process), you do not need to configure these properties (they are already configured). These properties are used to integrate with Operations Orchestration. In the subscription event overview section of the **[[[Undefined variable CSAVariables.tabOperations]]]** area in the Cloud Service Management Console, selecting the Process ID opens Operations Orchestration to the detailed page of the selected process when these properties are configured.

To configure the OO properties:

1. Edit the CSA\_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties file and configure the following properties:

Property	Description
OOS_URL	The URL used to access Operations

Property	Description
	<p>Orchestration Central. This is the Operations Orchestration used for provisioning topology designs. For example, <code>https://&lt;hostname&gt;:8445</code>.</p> <p>This property is automatically set during installation. If you are using the embedded Operations Orchestration that is included with CSA, this property is set using the values entered for the <b>Fully qualified domain name</b> on Windows or the <b>Fully Qualified Hostname</b> on Linux and <b>HPE OO Port</b> fields during installation. If you are using a standalone/external Operations Orchestration, this property is set using the values entered for the <b>HPE OO Hostname</b> and <b>HPE OO Port</b> fields during installation.</p>
OOS_USERNAME	<p>The username used to log in to Operations Orchestration Central.</p> <p>This property is automatically set during installation using the value entered for the <b>HPE OO User Name</b> field during installation.</p>
OOS_PASSWORD	<p>The encrypted password used by the user defined in OOS_USERNAME to log in to Operations Orchestration Central.</p> <p>This property is automatically set during installation using the value entered for the <b>HPE OO Password</b> field during installation.</p>
embedded.oo.root.dir	<p>Location of the embedded Operations Orchestration when it is installed with CSA. This property is generated when embedded Operations Orchestration is installed during the CSA installation.</p> <p>This property is the only indicator of embedded Operations Orchestration, which is important mainly for uninstallation and upgrades. This property cannot be edited.</p>

## 2. Restart CSA.

See ["Restart CSA" on page 170](#) for instructions.

## Configure Single Sign-On between CSA and OO

If HP Single Sign-On (HP SSO) was enabled during installation of CSA, SSO can be configured between CSA and Operations Orchestration. Configuring HP SSO allows you to launch Operations Orchestration from the Cloud Service Management Console without having to log in to Operations Orchestration.

CSA provides a login user (admin) and password (cloud) and, earlier in this guide, you configured an internal user for Operations Orchestration with the same user name and password. When Single Sign-On is configured between CSA and Operations Orchestration, this user can be used for single sign-on. That is, if you are logged in to CSA as the admin user, you can launch Operations Orchestration from the Cloud Service Management Console and not have to log in to Operations Orchestration.

You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and the embedded Operations Orchestration to use the same LDAP source or, if CSA and the embedded Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the embedded Operations Orchestration user must be assigned any role that allows flows to be viewed.

**Note:** To use HP SSO between CSA and Operations Orchestration, the systems on which CSA and Operations Orchestration are installed must be in the same domain.

## Configure and Enable HP Single Sign-On

To configure and enable HP SSO on Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > SSO**.
4. Select the **Enable** checkbox.
5. Enter the **InitString**. The `initString` setting for CSA and Operations Orchestration must be configured to the same value. In CSA, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/idm.war/WEB-INF/hpsssoConfiguration.xml` file. The `initString` value represents a secret key and should be treated as such in your environment (this string is used to encrypt and decrypt the `LWSSO_COOKIE_KEY` cookie that is used

to authenticate the user for single sign-on).

6. Enter the **Domain**. This is the domain name of the network of the servers on which CSA and Operations Orchestration are installed.
7. Click **Save**.

## Configure LDAP Users for Single Sign-On

To enable single sign-on for LDAP users, you must either configure CSA and Operations Orchestration to use the same LDAP source or, if CSA and Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the Operations Orchestration user must be assigned any role that allows flows to be viewed.

To configure LDAP for Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > LDAP**.
4. Enter the information to configure LDAP.
5. Click **Save**.

## Obscure Passwords in OO Flows (Optional)

Some Operations Orchestration flows included with CSA may show passwords in clear text when viewed in Operations Orchestration Central. You can obscure these passwords by modifying the flow in Operations Orchestration Studio.

**Note:** You must have Operations Orchestration Studio installed. Operations Orchestration Studio is supported on Windows platforms only and is not part of the embedded Operations Orchestration that is included with CSA. See the Operations Orchestration documentation, such as the *Operations Orchestration System Requirements*, for more information about Operations Orchestration Studio.

To obscure passwords in Operations Orchestration flows:



1. Open Operations Orchestration Studio.
2. Locate the flow to update.
3. Right-click on the flow and select **References > What uses this?**.

A list of flows that use the flow is displayed (that is, the flow to update is a subflow of the flows displayed in the list).

4. Select a flow from the list of flows.
5. Locate the subflow (the flow to update).
6. Right-click on the subflow and select **Properties**.
7. Located the property to obscure (such as a password), enable it, but do not assign a value to it.
8. Save the flow.
9. Repeat this procedure for every flow from the list of flows.

## Manually Configure Operations Orchestration for Sequential Designs

The following tasks are to configure OO for sequential designs. If you are installing CSA for the first time, configure only one instance of OO. If you have upgraded from an earlier version of CSA that has multiple instances of OO configured for sequential designs, you can continue to use multiple instances of OO, including OO 9.07.

**Note:** If you followed the instructions in the *Cloud Service Automation Installation Guide* or *Cloud Service Automation Upgrade Guide* to configure Operations Orchestration, you should have already completed the tasks in this section.

Complete the following tasks to configure OO to integrate with CSA:

**Note:** If you have manually configured OO for topology designs, you have already completed some of these tasks. Skip the tasks that you have already completed.

- Add a JRE to the system path
  - Install the CSA content pack
- Configure internal users
- Deploy content packs

- Update the Service Manager base content pack
- Set up system accounts for the CSA content pack
- Set up system properties
- Import OO flows
- Configure a secure connection between CSA and OO
- Configure Single Sign-On
- Obscure passwords in OO flows (optional)

**Note:** In the following instructions, `CSA_HOME` is the directory in which CSA is installed and `OO_HOME` is where you installed Operations Orchestration.

Be sure all the latest patches for Operations Orchestration have been installed. See the *Cloud Service Automation System and Software Support Matrix* for more information.

Guides are available on the HPE Software Support web site at: <https://softwaresupport.hpe.com> (this site requires a Passport ID). Select **Dashboards > Manuals**.

## Add a JRE to the System Path

The CSA flows that are imported require that a JRE be included in the system path on the system running CSA.

**To add a JRE to the system path on Windows, complete the following steps:**

1. Open the **Environment Variables** dialog:
  - a. Right-click **Computer** and select **Properties**.
  - b. Select **Advanced System Settings**.
  - c. Click **Environment Variables**.
2. Select the **Path** system variable.
3. Click **Edit**.
4. At the end of the value for **Variable value**, add a semicolon (;) and the following path:

If Operations Orchestration and CSA are installed on the same system:

```
OO_HOME/java/bin
```

or

If Operations Orchestration and CSA are installed on different systems:

```
CSA_JRE_HOME/bin
```

5. Click **OK** and close all windows.

### To add a JRE to the system path on Linux, complete the following steps:

Open a shell and enter one of the following commands:

- If Operations Orchestration and CSA are installed on the same system:

```
export PATH=$PATH:$OO_HOME/java/bin
```

- If Operations Orchestration and CSA are installed on different systems:

```
export PATH=$PATH:$CSA_JRE_HOME/bin
```

**Note:** By setting the system path, all applications (that require a JRE) use the JRE that is installed with Operations Orchestration or CSA (depending on the path you configured and if it is the only path or the first path set to a JRE in the system path). If you need to run another JRE with an application, you must type in the relative path to that JRE to run it (for example, when you configure TLS).

## Install the CSA Content Pack

(missing or bad snippet)

## Configure Internal Users

Internal users can be used to configure Operations Orchestration for CSA.

1. From the system on which CSA is installed (the system on which the content packs are installed), log in to Operations Orchestration Central.
2. Click **System Configuration**.
3. Select **Security > Internal Users**.

4. Click the **+** (Add) button.
5. Enter the following information:

Field	Recommended Value
User Name	csaouser
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The csaouser user is used to import the Operations Orchestration flows. When importing flows, this user is configured in the Operations Orchestration input file.

6. Click **Save**.
7. Click the **+** (Add) button.
8. Enter the following information:

Field	Recommended Value
User Name	csaouser
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The admin user is used with HP Single Sign-On (HP SSO). When Operations Orchestration is launched from the Cloud Service Management Console, this user allows access to HPE Operations Orchestration without having to log in. If you are using topology designs, the admin user can also be used for provisioning topology designs.

9. Click **Save**.
10. Click the **+** (Add) icon.
11. Enter the following information:

Field	Recommended Value
User Name	admin
Password	cloud
Roles	ADMINISTRATOR, SYSTEM_ADMIN

The admin user is used with HP Single Sign-On (HPSSO). When Operations Orchestration is launched from the Cloud Service Management Console, this user allows access to Operations

Orchestration without having to log in. If you are using topology designs, the admin user can also be used for provisioning topology designs.

12. Click **Save**.
13. Log out of Operations Orchestration Central and log back in as the csaoouser.

## Deploy Content Packs

Deploy the following content packs:

- oo10-base-cp.jar - 1.9.0
- oo10-cloud-cp.jar - 2.1.0
- oo10-hpe-solutions-cp.jar - 1.10.0
- oo10-virtualization-cp.jar - 1.9.1

**Note:** You can also deploy content packs from the **Content Store**. Refer to online help for more details.

1. From Operations Orchestration Central, click **Content Management**.
2. Click the **Content Packs** tab.
3. Click the **Deploy New Content** icon.
4. In the **Deploy New Content** dialog, in the upper left corner, click the + (Add files for deployment) icon.
5. Deploy the Component Tool content packs. From Operations Orchestration Central, navigate to the `tools/oo/content-packs` directory. Add and deploy the content packs (after each successful deployment, to add and deploy the next content pack without closing the dialog, click the Reset icon in the upper left corner to clear the dialog and enable the + (Add files for deployment) icon).

The deployment may take a few minutes and the dialog will show a progress bar. When you have finished deploying all the content packs, click **Close** to close the dialog.

# Update and Redeploy the Service Manager Base Content Pack

Update and redeploy the `oo10-sm-cp-1.0.3.jar` base content pack. If you deployed an earlier version of the Service Manager base content pack, you must do the following (if this is a fresh installation of Operations Orchestration and you did not deploy an earlier version of the Service Manager base content pack, you do not have to complete these steps):

1. Stop the Operations Orchestration services:

**Windows:**

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click on the HPE Operations Orchestration Central service and select **Stop**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Stop**.

**Linux:**

- a. On the server that hosts Operations Orchestration, run the following command:  
`<HPE00installation>/central/bin/central stop`  
For example, `/usr/local/hpe/csa/00/central/bin/central stop`
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: `<HPE00installation>/ras/bin/ras stop`.  
For example, `/usr/local/hpe/csa/00/ras/bin/ras stop`

2. Clear the Operations Orchestration Central cache by deleting the following folder:

`<HPE00installation>/central/var/cache`

For example,

**Windows:** `C:\Program Files\HPE\HPE Operations Orchestration\central\var\cache`

**Linux:** `/usr/local/hpe/csa/00/central/var/cache`

3. If RAS is installed, clear the RAS artifact cache by deleting the following folder (on all RAS systems, including localhost):

```
<HPE00installation>/ras/var/cache
```

For example,

**Windows:** C:\Program Files\HPE\HPE Operations Orchestration\ras\var\cache

**Linux:** /usr/local/hpe/csa/oo/ras/var/cache

4. Run the following SQL command against the Operations Orchestration database:

```
DELETE from OO_ARTIFACTS where NAME =
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.pom' or NAME =
'org/apache/ws/security/wss4j/1.5.7/wss4j-1.5.7.jar'
```

5. Start the Operations Orchestration services:

**Windows:**

- a. On the server that hosts Operations Orchestration, navigate to **Start > Administrative Tools > Services**.
- b. Right-click the HPE Operations Orchestration Central service and select **Start**.
- c. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), navigate to **Start > Administrative Tools > Services**.
- d. Right-click on the Operations Orchestration RAS service and select **Start**.

**Linux:**

- a. On the server that hosts Operations Orchestration, run the following command:  

```
<HPE00installation>/central/bin/central start
```

For example, /usr/local/hpe/csa/oo/central/bin/central start
- b. If you installed the Remote Action Server (RAS), on all RAS systems (including localhost), run the following command: 

```
<HPE00installation>/ras/bin/ras start
```

For example, /usr/local/hpe/csa/oo/ras/bin/ras start

6. Redeploy the oo10-sm-cp-1.0.3.jar base content pack:

- a. Log in to Operations Orchestration Central and click **Content Management**.
- b. Click the **Content Packs** tab.
- c. Click the **Deploy New Content** icon.

- d. In the Deploy New Content dialog, in the upper left corner, click the **+** (Add files for deployment) icon.
- e. Navigate to the `CSA_HOME/oo/ooContentPack` directory and select **oo10-sm-cp-1.0.3.jar**.
- f. Click **Deploy**.  
The deployment may take a few minutes and the dialog will show a progress bar.
- g. Click **Close**.

## Set Up System Accounts for the Content Packs

Set up system accounts for the content packs:

1. Log in to Operations Orchestration Central.
2. Click **Content Management**.
3. Select **Configuration Items > System Accounts**.
4. Click the **Add** icon.
5. Enter the following information if it is not already configured:

Field	Recommended Value
System Account Name	CSA_REST_CREDENTIALS
User Name	oolnboundUser
Password	cloud

**Note:** The **User Name** configured for the `CSA_REST_CREDENTIALS` System Account setting must match the **Override Value** configured for the `CSA_00_USER` System Property setting.

6. Click **Save**.
7. Click the **Add** icon.



8. Enter the following information if it is not already configured:

Field	Recommended Value
System Account Name	CSA_SERVICEMANAGER_CREDENTIALS
User Name	falcon
Password	<leave_blank>

9. Click **Save**.

## Set Up System Properties for the Content Packs

Set up the following system properties for the content packs:

1. Log in to Operations Orchestration Central.
2. Click **Content Management**.
3. Select **Configuration Items > System Properties**.
4. Click the **Add** icon.
5. Enter the following information if it is not already configured:

Field	Recommended Value
Name	CSA_REST_URI
Override Value	https://<csa_hostname>:8444/csa/rest

6. Click **Save**.

## Configure a Secure Connection between CSA and OO

Export Operations Orchestration's certificate from Operations Orchestration's truststore. If Operations Orchestration and CSA are not installed on the same system, copy the certificate to the CSA system and import the certificate into CSA's truststore. TLS must be configured between CSA and Operations Orchestration.

Do the following:

1. On the system running Operations Orchestration, open a command prompt and change to the directory where Operations Orchestration is installed.

2. Run the following command:

**Windows:**

```
.\java\bin\keytool -export -alias tomcat -file C:\oo.cer
-keystore .\Central\var\security\key.store -storepass changeit
```

**Linux:**

```
./java/bin/keytool -export -alias tomcat -file /tmp/oo.cer
-keystore ./Central/var/security/key.store -storepass changeit
```

where C:\oo.cer on Windows and /tmp/oo.cer on Linux are examples is an example of a filename and location used to store the exported root certificate (you can choose a different filename and location).

3. If Operations Orchestration is not running on the same system as CSA, copy oo.cer from the Operations Orchestration system to the system running CSA.
4. On the system running CSA, open a command prompt.
5. Run the following command:

**Windows:**

```
"CSA_JRE_HOME\bin\keytool" -importcert -alias tomcat -file C:\oo.cer -
trustcacerts -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

**Linux:**

```
CSA_JRE_HOME/bin/keytool -importcert -alias tomcat -file /tmp/oo.cer -
trustcacerts -keystore CSA_JRE_HOME/lib/security/cacerts
```

where CSA\_JRE\_HOME is the directory in which the JRE that is used by CSA is installed

6. When prompted for the keystore password, enter changeit.
7. Enter yes when prompted to trust the certificate.

## Configure HP Single Sign-On between CSA and OO

If HP Single Sign-On (HP SSO) was enabled during installation of CSA, SSO can be configured between CSA and Operations Orchestration. Configuring HP SSO allows you to launch Operations Orchestration from the Cloud Service Management Console without having to log in to Operations Orchestration.

CSA provides a login user (admin) and password (cloud) and, earlier in this guide, you configured an internal user for Operations Orchestration with the same user name and password. When Single Sign-On is configured between CSA and Operations Orchestration, this user can be used for single sign-on. That is, if you are logged in to CSA as the admin user, you can launch Operations Orchestration from the Cloud Service Management Console and not have to log in to Operations Orchestration.

You can also configure LDAP users for single sign-on. To enable single sign-on for LDAP users, you must either configure CSA and the embedded Operations Orchestration to use the same LDAP source or, if CSA and the embedded Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the embedded Operations Orchestration user must be assigned any role that allows flows to be viewed.

**Note:** To use HP SSO between CSA and Operations Orchestration, the systems on which CSA and Operations Orchestration are installed must be in the same domain.

## Configure and Enable HP Single Sign-On

To configure and enable HP SSO on Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > SSO**.
4. Select the **Enable** checkbox.
5. Enter the **InitString**. The `initString` setting for CSA and Operations Orchestration must be configured to the same value. In CSA, `initString` is configured in the `crypto` element in the `CSA_HOME/jboss-as/standalone/deployments/idm.war/WEB-INF/hpssoConfiguration.xml` file. The `initString` value represents a secret key and should be treated as such in your environment (this string is used to encrypt and decrypt the `LWSSO_COOKIE_KEY` cookie that is used to authenticate the user for single sign-on).
6. Enter the **Domain**. This is the domain name of the network of the servers on which CSA and Operations Orchestration are installed.
7. Click **Save**.

## Configure LDAP Users for Single Sign-On

To enable single sign-on for LDAP users, you must either configure CSA and Operations Orchestration to use the same LDAP source or, if CSA and Operations Orchestration use different LDAP sources, configure the same users in both sources. In either case, the CSA user must be assigned to the CSA Administrator or Service Operations Manager role and the Operations Orchestration user must be assigned any role that allows flows to be viewed.

To configure LDAP for Operations Orchestration, do the following:

1. Log in to Operations Orchestration Central.
2. Click the **System Configuration** button.
3. Select **Security > LDAP**.
4. Enter the information to configure LDAP.
5. Click **Save**.

## Obscure Passwords in OO Flows (Optional)

Some Operations Orchestration flows included with CSA may show passwords in clear text when viewed in Operations Orchestration Central. You can obscure these passwords by modifying the flow in Operations Orchestration Studio.

**Note:** You must have Operations Orchestration Studio installed. Operations Orchestration Studio is supported on Windows platforms only and is not part of the embedded Operations Orchestration that is included with CSA. See the Operations Orchestration documentation, such as the *Operations Orchestration System Requirements*, for more information about Operations Orchestration Studio.

To obscure passwords in Operations Orchestration flows:

1. Open Operations Orchestration Studio.
2. Locate the flow to update.
3. Right-click on the flow and select **References > What uses this?**.

A list of flows that use the flow is displayed (that is, the flow to update is a subflow of the flows displayed in the list).

4. Select a flow from the list of flows.
5. Locate the subflow (the flow to update).
6. Right-click on the subflow and select **Properties**.
7. Located the property to obscure (such as a password), enable it, but do not assign a value to it.
8. Save the flow.
9. Repeat this procedure for every flow from the list of flows.

## Appendix: Hubot Notifications Integration with CSA

CSA provides the ability to integrate with Slack and Hubot so that subscription-related messages can be sent from a Slack chat room. After you subscribe or cancel a subscription within the chat room, you will receive a Hubot notification within the chat room of the subscription's status.

As part of this integration, an example Coffee Script and environment file are shipped with CSA. You can use these files to quickly integrate CSA and Hubot. These files are shipped in the `CSA_HOME/extras` directory.

The following properties must be added and configured in the `csa.properties` file to enable CSA to send pre-defined messages through the Slack channel. Notifications are sent back to the chat room with the subscription ID when the subscription becomes active, is successful, or canceled.

Property	Description
<code>csa.HUBOT.Integration.enabled</code>	Set this property to <code>true</code> if you want to enable integration between CSA and Hubot notifications.  Set this property to <code>false</code> to disable CSA and Hubot integration.
<code>csa.HUBOT.Notification.BaseUri</code>	Use <code>http://&lt;hubot_host&gt;:&lt;port&gt;</code> to enable CSA and Hubot integration. Replace the <code>&lt;hubot_host&gt;</code> and <code>&lt;port&gt;</code> with the hostname and port used by the Hubot instance.
<code>csa.HUBOT.Notification.ChatPlatform</code>	Sets the Slack chat platform with which Hubot will be interacting. Slack is the only chat platform that is supported. Slack must be installed to use Hubot .  Example: <code>csa.HUBOT.Notification.ChatPlatform=slack</code>
<code>csa.HUBOT.Notification.BotName</code>	Sets the <code>BotName</code> that Hubot listens to when sending messages. The <code>BotName</code> is configured both in <code>slack</code> as well as in the Hubot environment file.
<code>csa.HUBOT.Notification.Room</code>	Sets the channel name for Hubot to post messages back into room that deployment or cancel actions were called from. Example format is <code>#&lt;channel-name&gt;</code> .  Example: <code>csa.HUBOT.Notification.Room=#qa-channel</code>

# Send documentation feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

## **Feedback on Configuration Guide (Cloud Service Automation 4.80)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [clouddocs@hpe.com](mailto:clouddocs@hpe.com).

We appreciate your feedback!