HP Cloud Service Automation

Software Version 4.00

Process Definition Tool

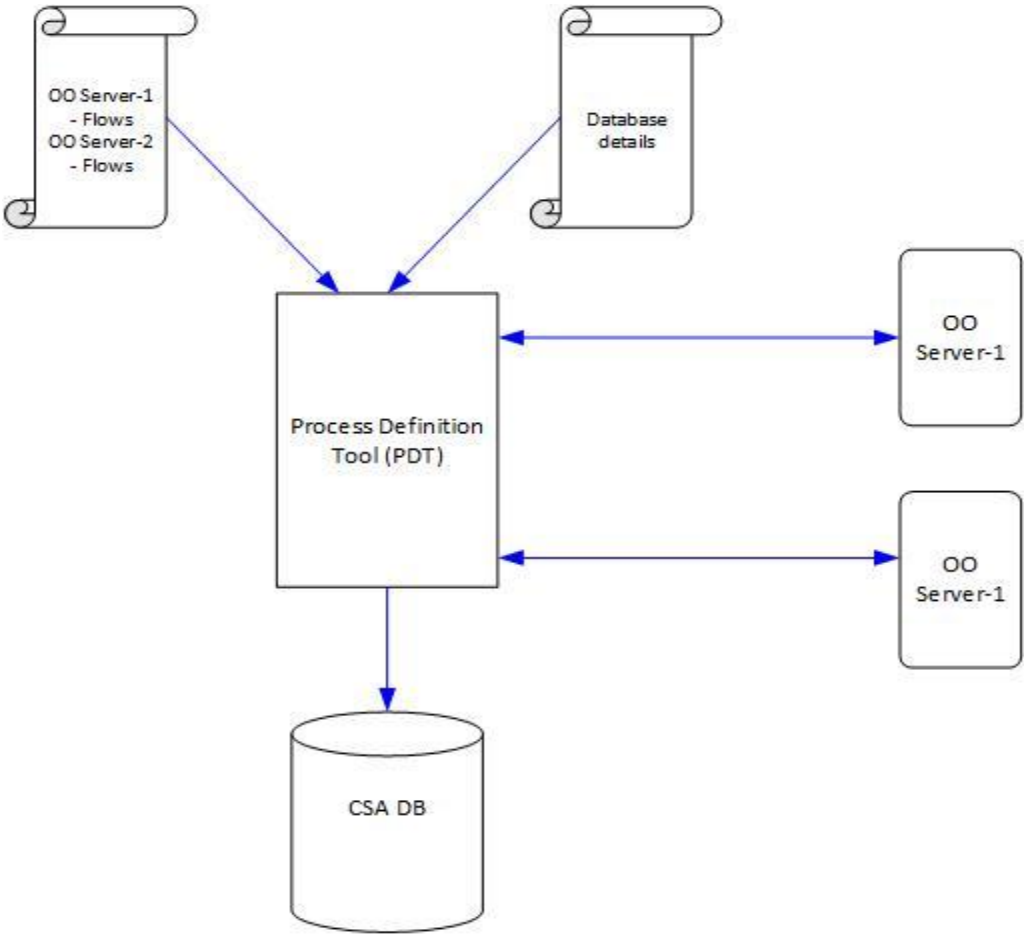
Contents

Overview

The Process Definition Tool is part of the HP Cloud Service Automation installation. For more information, see the *HP Cloud Service Automation Configuration Guide*.

The Process Definition Tool (PDT) is a command line tool used for importing HP Operations Orchestration (HP OO) flow signature-related information into the HP Cloud Service Automation Database (CSA DB) so that this information is stored locally. The imported flow signatures are used during the creation of a service design or a resource offering and when adding a resource synchronization action or an external approval type.

The PDT works by querying the flow definitions of one or more OO servers and then writing to the CSA DB. The information about the OO server, the flows, and the database is provided through an XML input file and java properties file respectively. PDT runs an initial validation of the input XML file to ensure the syntax is correct and that the database credentials are valid. The PDT then scans the OO machine listed in the input XML file sequentially and queries the OO machine for requesting information about the OO flows. The PDT then writes information about the OO flow to the CSA DB.



The PDT creates a log file titled `process-defn-tool.log` in the current working directory. This logs all the interactions between the tool and the OO server as well as the database. To find out more about how to interpret the log files, see Understanding the Log File in this document.

Basic Configuration Steps

Before running PDT for the first time, complete the following steps:

1. Import the CSA OO content to the OO server.
2. Export the OO certificate to a file.
3. Import the OO certificate information to the Java truststore.
4. Create input files for PDT using the `-g` option.
5. Update the appropriate database file with the correct credentials and URL.
6. Update the `HPOOInput.xml` file with the correct HP OO URL, user credentials, truststore location, truststore password with the correct value for all the `<ooengine/>` tags defined within the file.

For step-by-step configuration instructions, see Installing CSA Content on an OO Server.

Input Files Syntax

There are two input files needed for PDT:

1. Database Configuration File

   This database configuration file provides connection information about the CSA DB being used. The file uses the Java properties file syntax. The properties in the file describe the type of database (db.type), jdbc URL to access the database (db.url), database user (db.user), and the password (db.password). It is recommended that the database password be encrypted in CSA format. The password can be encrypted by running the PDT with the `-e/` encrypt option. If a plain text password is used, the PDT will overwrite the input properties file with the encrypted password.

   If there is uncertainty about the exact database URL, review the standalone XML file. See Troubleshooting for more information.

2. OO Input File

   The OO input file provides information about one or more OO server(s) and the OO flow information stored in the OO servers. The OO input file uses XML syntax to describe the OO server and the flows.

   The root tag is `<ooengines/>`. The direct children of the root tag use the tag `<ooengine/>`. There can be one or more of the `<ooengine/>` tags. Each of the `<ooengine/>` tags contains information about a single OO server and the OO flows that needs to be imported from them. For PDT to interact with an OO server in a secure manner, the tool needs information about the access URL of the server, user credentials, and Java truststore information (see step 3 in **Basic Configuration Steps** above).

   `<ooengine/>` Attributes

   The attributes of `<ooengine/>` tags are used for providing this information. These attributes are:

Attribute	Description
accessPointType	An optional parameter which, by default, is assumed to be of the type URL. The other possible values are `EXTERNAL_APPROVAL or RESOURCE_POOL_SYNC.`
name	The unique identifier for the process engine that was created in CSA. This value may be needed in the future if engine information needs to be updated.

Attribute	Description
url	The URL for the OO Central Web Service. Check to make sure that the port information is correct.
truststore	The truststore in which the HP OO Certificate Authority's root certificate has been imported. Having an incorrect value for the truststore or its password is a common error when running the PDT.
truststorepassword	The password of the truststore which needs to be encrypted in CSA encryption format. PDT provides a `-e/` encrypt option that allows encryption of a clear text word. If a clear text word is provided in place of an encrypted one, PDT will overwrite the clear text word with the encrypted one.
username	The OO Central user name that has access to the OO machine for reading information about the OO flows. Typically this is a user with admin rights.
password	The password for the OO Central user. This password needs to be encrypted like the truststore password. If a clear text password is provided in place of an encrypted one, PDT will overwrite the clear text word with the encrypted one.
update	Set to **true** when information about the `<ooengine/>` attribute needs to be updated in the database (url, username, password).
delete	Set to **true** when an unused process engine and all unused flows that are part of it need to be deleted.

`<oofolder/>` Attributes

Each `<ooengine/>` element contains one more `<folder/>` elements. The `<folder/>` elements specify the OO flow information that needs to be imported into the OO server. The attributes for `<folder/>` are:

- path
- flow
- recursive
- update
- delete

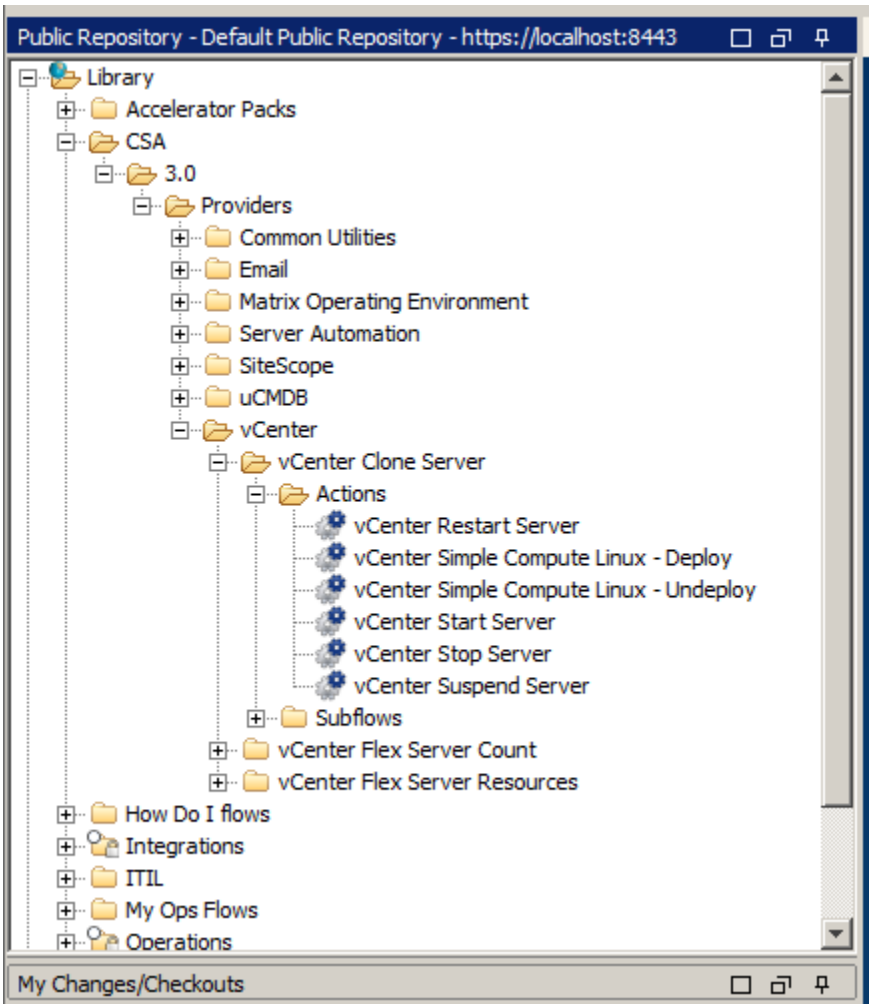
Path specifies a folder or a flow on the OO server. If the **path** points to a folder on OO server, then the PDT by default would create definitions for all the flows in that folder.

The attributes **flow**, **recursive**, **update**, and **delete** can have only **true** or **false** values. By default, these attributes are set to **false**. If the **flow** attribute is set to **true**, it means that the attribute path points to a single flow rather than a folder containing a set of flows. If the attribute **recursive** is set to **true**, then PDT will create definitions not only in the flows in the same folder but also in all subfolders of the folder specified in the path attribute.

If a definition for a flow already exists on the CSA system, the tool will leave that definition unchanged unless the **update** or **delete** attributes are set to **true**. A **true** value for the **update** attribute will result in updating the process definition for the corresponding flow. A **true** value for the **delete** attribute will result in the deletion of the process definition, provided it is unused.

Usage Examples

The following screenshot shows OO flow information stored on an OO server:



To import all the flows under the 'Actions' folder for the first time

```
<folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions"/>
```

To update the existing definitions corresponding to all the flows in the folder

The tool will update an existing definition if one already exists on the system; otherwise, it will create a new one.

```
<folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions"update="true"/>
```

To delete the definitions that already exist on the system that correspond to flows in the 'Actions' folder

```
<folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions"delete="true"/>
```

To import all the flows and folders under a root folder 'Providers,' we set the recursive flag to true

```
<folder path="/Library/CSA/3.0/Providers"recursive="true"/>
```

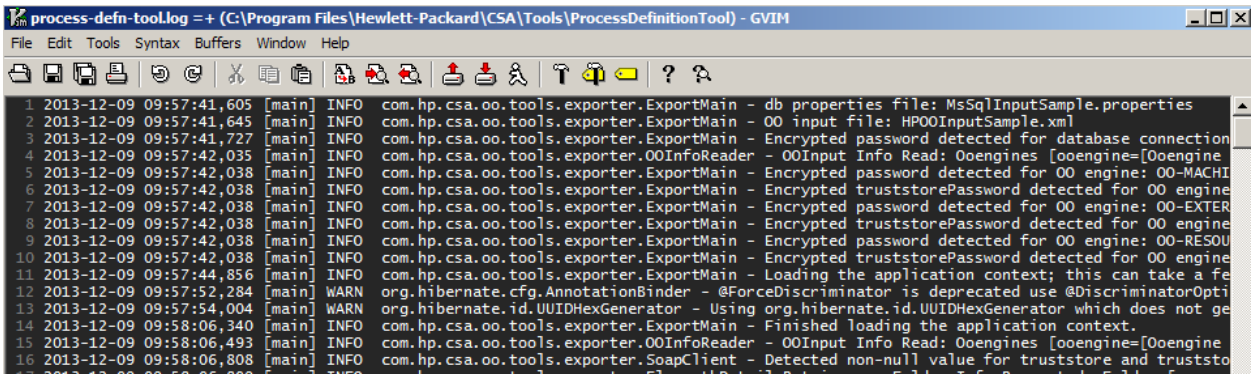
To create a definition for a single flow

In this example: `/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions/vCenter Start Server`, the flow flag is set to **true** as shown below:

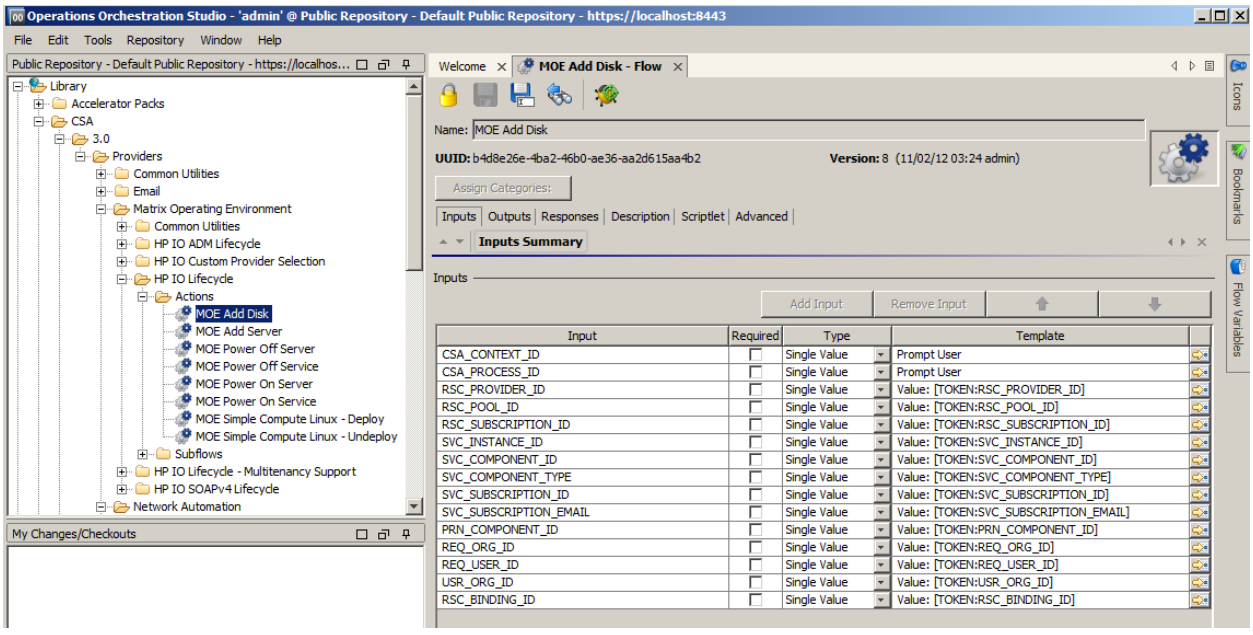
```
<folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions/vCenter Start Server"flow="true"/>
```

Understanding the Log File

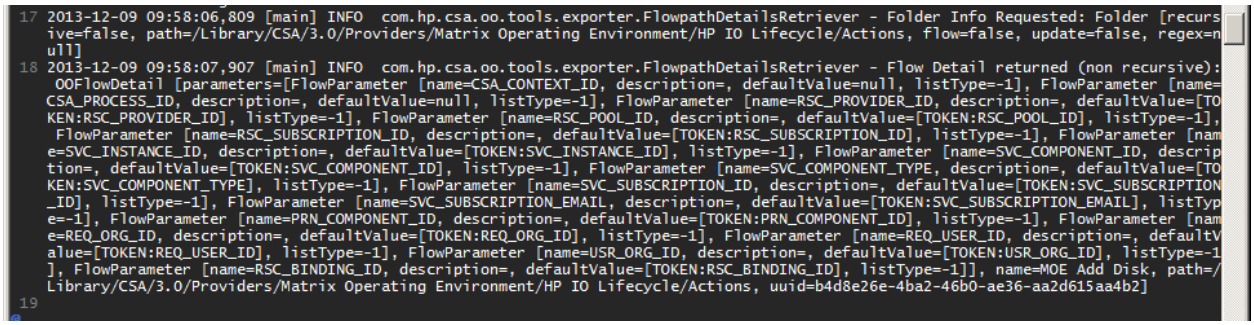
The PDT logs the interaction of the OO server and the database in the `process-defn-tool.log` file. The log file is generated in the current working directory and is useful for debugging. Typical log file content is shown below:



Further examination of the log file shows all the flows that get imported when a process definition is executed. Below is a screenshot of an OO flow and the log file contents when that flow gets imported into CSA as a process definition.

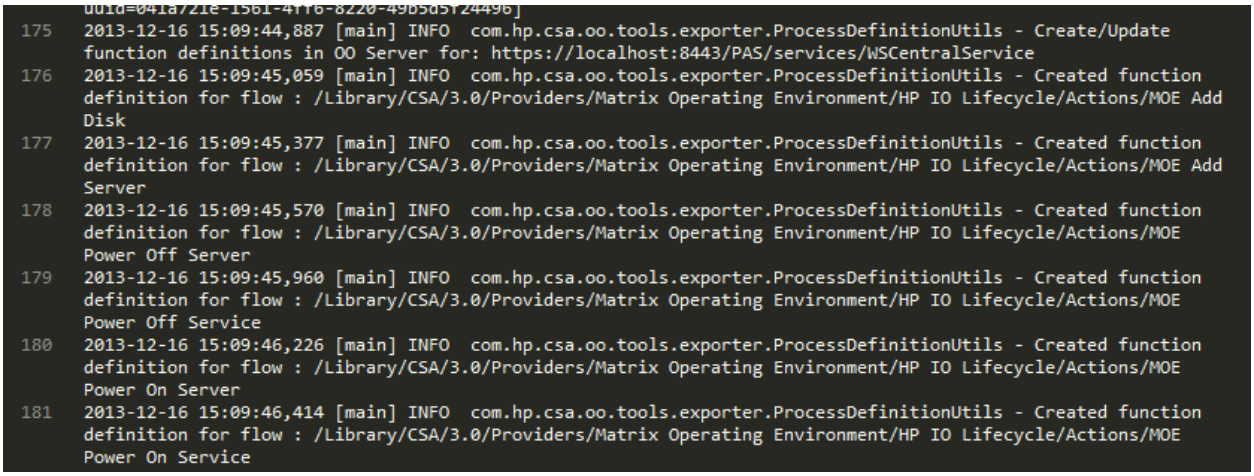


Shown below are the log file contents when that flow is retrieved from the OO server:

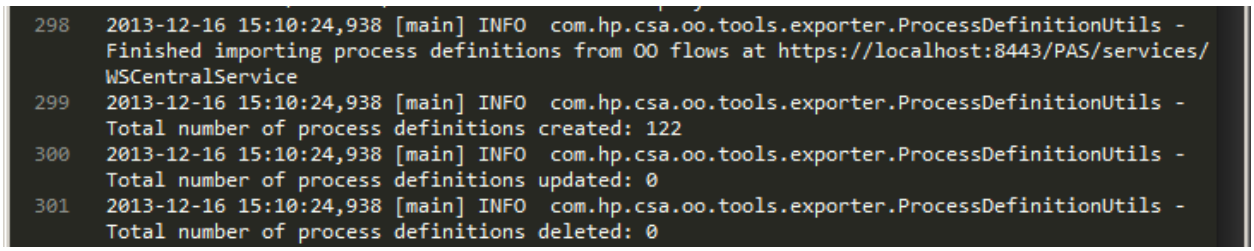


Line 18 of the log file above shows that the tool retrieved information about the flow `/Library/CSA/3.0/Providers/Matrix Operating Environment/HP IO Lifecycle/Actions/MOE Add/MOE Add Disk` and the flow name can be reconstructed by concatenating values for **path** and **name** in this line. Line 18 also contains information about the attributes that are being retrieved for this particular flow (`CSA_CONTEXT_ID, CSA_PROCESS_ID`...) and the default value of these properties. The `uuid` listed is the UUID of the flow on the OO server that CSA will use to invoke this particular flow when communicating to the OO engine.

Once the information related to all the flows in a folder is retrieved, definitions corresponding to these flows are created and written to the database. A sample screenshot of lines representing the log is shown below:



Once all the flows within an `<ooengine/>` element are processed, the PDT tool writes the statistics about the operation to the log file, including the number of definitions created, updated, and deleted based on the input. A sample screenshot of lines representing this information in the log file is shown below:



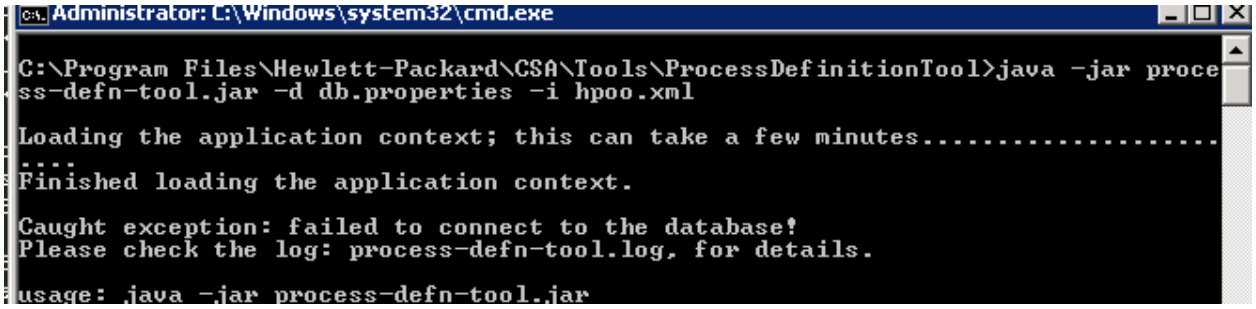
Troubleshooting

Following is a list of typical errors that cause failures when running the PDT.

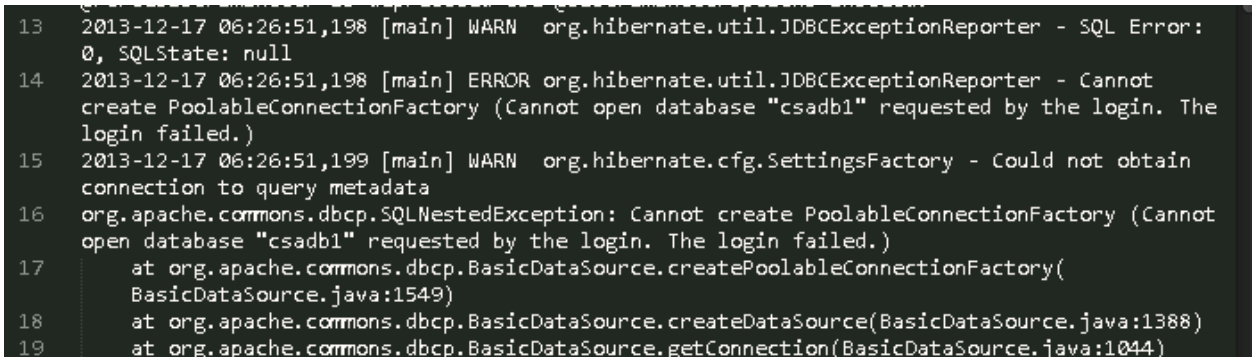
Incorrect database URL

One of the first tasks of the PDT is to make sure that the database URL and credentials are correct. In cases where a database login URL is incorrect because of an invalid database name, both the console and the log files display appropriate error messages as shown below.

Console error message:



A typical log file error message:



If you are unsure if you have the correct database URL configured, review the `standalone.xml` file for the CSA JBoss server. The database URL used for the CSA installation is usually listed under `<datasource/>` with attribute `jndi-name="java:jboss/datasources/csaDS."` A sample screenshot containing that information is shown below:

```
<datasource jndi-name="java:jboss/datasources/csaDS" pool-name="mssqlDS">
    <connection-url>jdbc:jtds:sqlserver://localhost:1433/csa;ssl=request</connection-url>
    <driver>mssqlDriver</driver>
    <pool>
        <min-pool-size>10</min-pool-size>
        <max-pool-size>200</max-pool-size>
        <prefill>true</prefill>
    </pool>
    <security>
        <security-domain>csa-encryption-sec</security-domain>
    </security>
</datasource>
```

Incorrect database user credentials

The error message displayed on the console in the case of an invalid database is similar to that of an invalid database URL. Looking at the log file allows you to differentiate a database credential error from that of an invalid database URL. A screenshot displaying this error in the log file is shown below:

```
11  2013-12-17 08:36:49,788 [main] INFO  com.hp.csa.oo.tools.exporter.ExportMain - Loading the application
    context; this can take a few minutes.
12  2013-12-17 08:36:58,702 [main] WARN  org.hibernate.cfg.AnnotationBinder - @ForceDiscriminator is deprecated
    use @DiscriminatorOptions instead.
13  2013-12-17 08:37:00,755 [main] WARN  org.hibernate.util.JDBCExceptionReporter - SQL Error: 0, SQLState: null
14  2013-12-17 08:37:00,756 [main] ERROR org.hibernate.util.JDBCExceptionReporter - Cannot create
    PoolableConnectionFactory (Login failed for user 'sa1'.)
15  2013-12-17 08:37:00,756 [main] WARN  org.hibernate.cfg.SettingsFactory - Could not obtain connection to query
    metadata
16  org.apache.commons.dbcp.SQLNestedException: Cannot create PoolableConnectionFactory (Login failed for user
    'sa1'.)
17      at org.apache.commons.dbcp.BasicDataSource.createPoolableConnectionFactory(BasicDataSource.java:1549)
```

Incorrect truststore location

```
C:\Program Files\Hewlett-Packard\CSA\Tools\ProcessDefinitionTool>java -jar proce
ss-defn-tool.jar -d db.properties -i hpoo.xml

Loading the application context; this can take a few minutes.....
....
Finished loading the application context.

Caught exception: javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpe
cted error: java.security.InvalidAlgorithmParameterException: the trustAnchors p
arameter must be non-empty
Please check the log: process-defn-tool.log, for details.
```

```
2013-12-17 09:35:03,016 [main] ERROR com.hp.csa.oo.tools.exporter.ExportMain - ; nested
exception is:
    javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error: java.
    security.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-
    empty
AxisFault
 faultCode: {http://schemas.xmlsoap.org/soap/envelope/}Server.userException
 faultSubcode:
 faultString: javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error:
 java.security.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-
 empty
 faultActor:
 faultNode:
 faultDetail:
    {http://xml.apache.org/axis/}stackTrace:javax.net.ssl.SSLException: java.lang.
    RuntimeException: Unexpected error: java.security.InvalidAlgorithmParameterException:
    the trustAnchors parameter must be non-empty
```

In most cases this truststore attribute value should be the same as the value for `csaTruststore` in the `csa.properties` file.

```
#These properties represent the SSL trust store configuration of CSA. The truststore forma
csaTruststore=C:/Program Files/Java/jre7/lib/security/cacerts
csaTruststorePassword=ENC(nN+R6mj8YZKzjsSp1fefs+2rDTEXI8AI)
```

When the issue is an invalid certificate:

```
Administrator: C:\Windows\system32\cmd.exe
C:\Program Files\Hewlett-Packard\CSA\Tools\ProcessDefinitionTool>java -jar proce
ss-defn-tool.jar -d db.properties -i hpoo.xml

Loading the application context; this can take a few minutes.....................
.....
Finished loading the application context.

Caught exception: javax.net.ssl.SSLHandshakeException: sun.security.validator.Va
lidatorException: PKIX path building failed: sun.security.provider.certpath.SunC
ertPathBuilderException: unable to find valid certification path to requested ta
rget
Please check the log: process-defn-tool.log, for details.

usage: java -jar process-defn-tool.jar
```

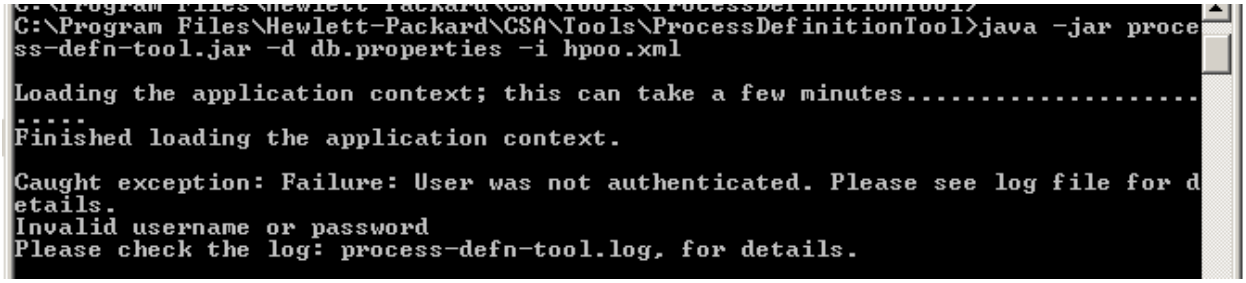
```
145  2013-12-17 09:53:28,657 [main] ERROR com.hp.csa.oo.tools.exporter.ExportMain - ; nested
     exception is:
146      javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX
         path building failed: sun.security.provider.certpath.SunCertPathBuilderException:
         unable to find valid certification path to requested target
147  AxisFault
148   faultCode: {http://schemas.xmlsoap.org/soap/envelope/}Server.userException
149   faultSubcode:
150   faultString: javax.net.ssl.SSLHandshakeException: sun.security.validator.
     ValidatorException: PKIX path building failed: sun.security.provider.certpath.
     SunCertPathBuilderException: unable to find valid certification path to requested target
151   faultActor:
152   faultNode:
153   faultDetail:
154      {http://xml.apache.org/axis/}stackTrace:javax.net.ssl.SSLHandshakeException: sun.
         security.validator.ValidatorException: PKIX path building failed: sun.security.
         provider.certpath.SunCertPathBuilderException: unable to find valid certification
         path to requested target
```

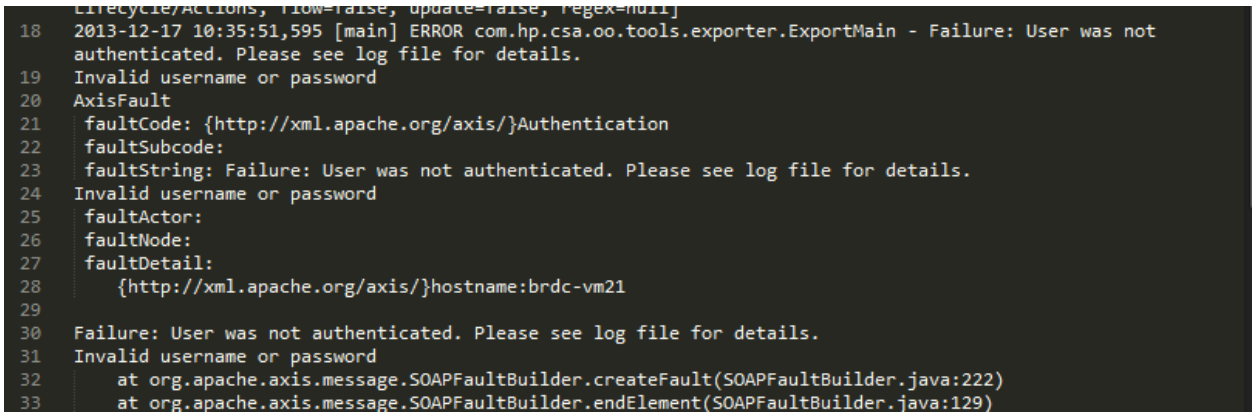
Incorrect truststore credentials

```
Administrator: C:\Windows\system32\cmd.exe

C:\Program Files\Hewlett-Packard\CSA\Tools\ProcessDefinitionTool>java -jar proce
ss-defn-tool.jar -d db.properties -i hpoo.xml

Loading the application context; this can take a few minutes.....................
....
Finished loading the application context.

Caught exception: java.net.SocketException: java.security.NoSuchAlgorithmExcepti
on: Error constructing implementation (algorithm: Default, provider: SunJSSE, cl
ass: sun.security.ssl.SSLContextImpl$DefaultSSLContext)
Please check the log: process-defn-tool.log, for details.
```

```
18  2013-12-17 10:05:47,116 [main] ERROR com.hp.csa.oo.tools.exporter.ExportMain - ; nested exception is:
19      java.net.SocketException: java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm:
        Default, provider: SunJSSE, class: sun.security.ssl.SSLContextImpl$DefaultSSLContext)
20  AxisFault
21   faultCode: {http://schemas.xmlsoap.org/soap/envelope/}Server.userException
22   faultSubcode:
23   faultString: java.net.SocketException: java.security.NoSuchAlgorithmException: Error constructing implementation (
     algorithm: Default, provider: SunJSSE, class: sun.security.ssl.SSLContextImpl$DefaultSSLContext)
24   faultActor:
25   faultNode:
26   faultDetail:
27      {http://xml.apache.org/axis/}stackTrace:java.net.SocketException: java.security.NoSuchAlgorithmException: Error
        constructing implementation (algorithm: Default, provider: SunJSSE, class: sun.security.ssl.
        SSLContextImpl$DefaultSSLContext)
28      at javax.net.ssl.DefaultSSLSocketFactory.throwException(Unknown Source)
29      at javax.net.ssl.DefaultSSLSocketFactory.createSocket(Unknown Source)
```

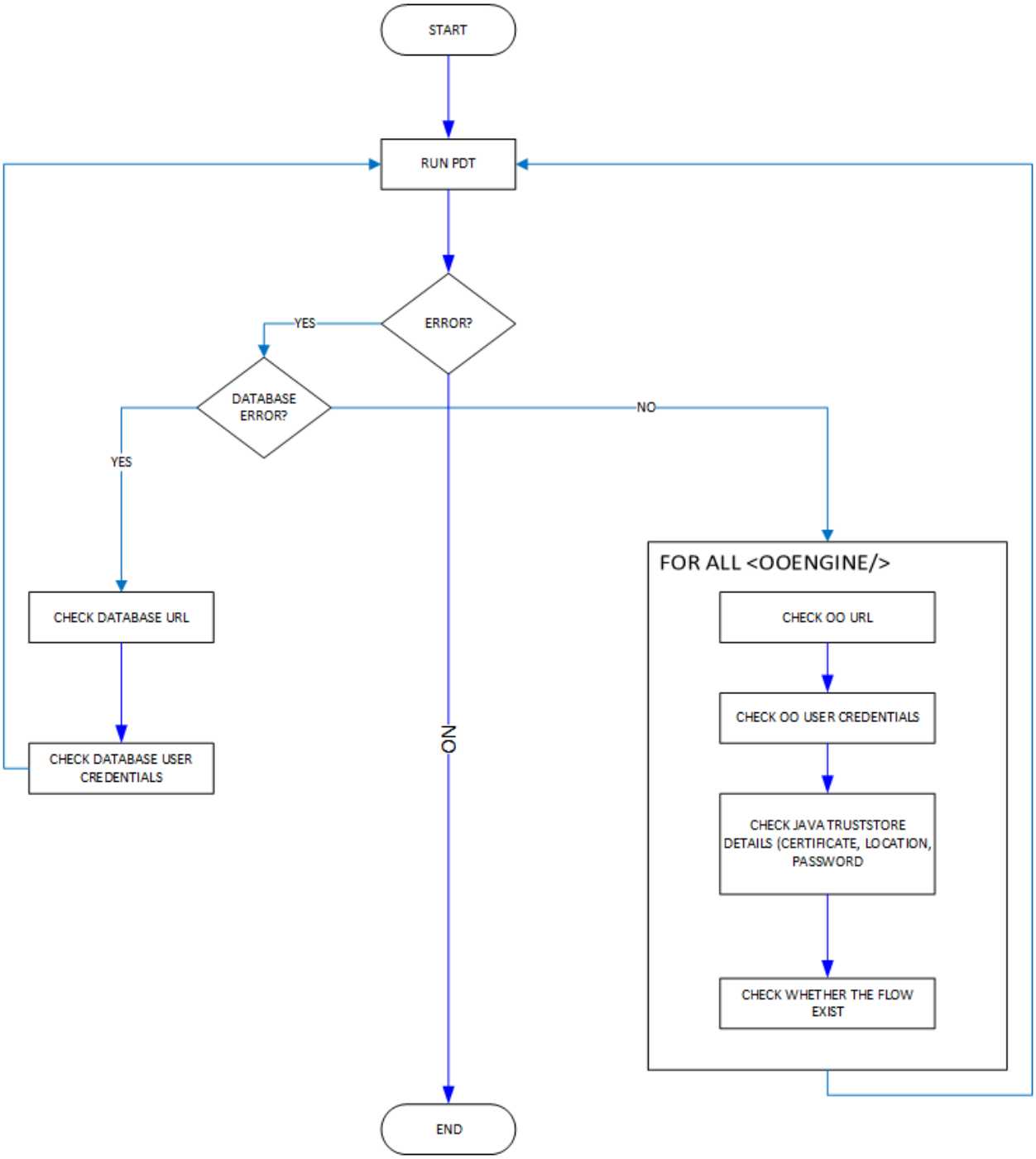
Incorrect OO credentials





Incorrect flow location

The following flowchart summarizes steps for debugging errors encountered while running the PDT:

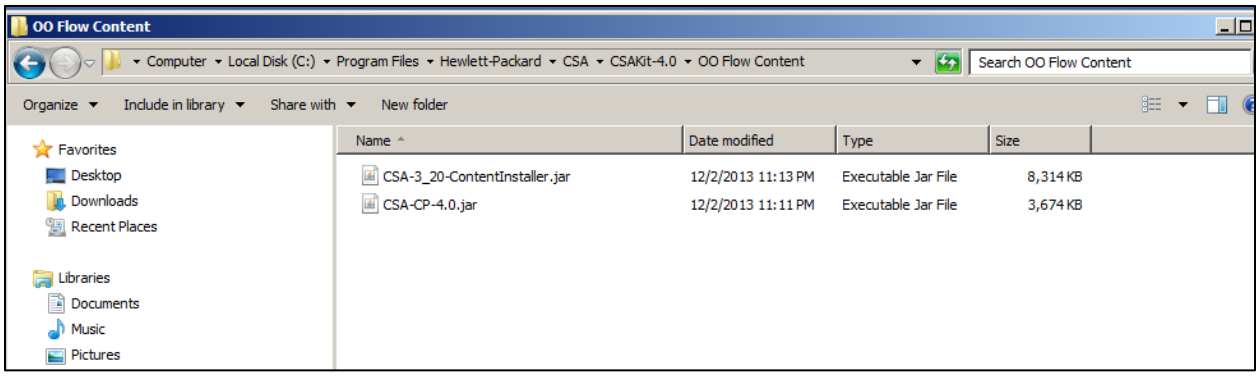


Best Practices

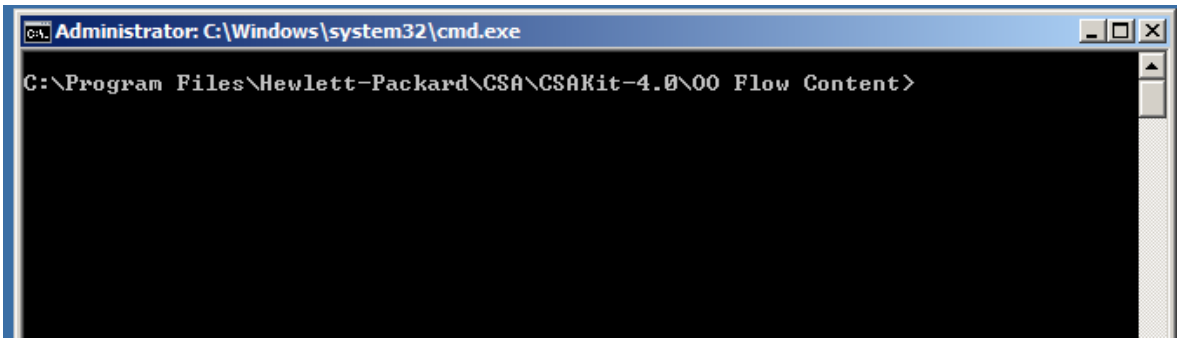
1. If your objective is to just create definitions for the OO content that comes with CSA, the sample files generated with the `-g` option will be sufficient. The user credentials for the database, database URL are the attributes that need to be updated in the database file. The attributes of `<ooengine/>` in the input XML file need to be updated, but not the contents of any of the `<folder/>` attributes within the `<ooengine/>`.
2. You can have multiple `<ooengines/>` with the same OO URL, multiple user credentials, etc., but the name attribute value for each `<ooengine/>` must be unique. In many ways it is ideal to have different OO engines configured for various types of flows.
3. The PDT can be executed to update the existing definitions. Definitions get updated only when properties of the flow (typically under the **Actions** folder) change and not when any of the sub flows invoked by the main flow change.
4. Process definitions are uniquely defined by their names. A process definition name is the entire path name for the OO flow (for ex: `/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions/vCenter Start Server`). A process definition is created with an associated process engine. To create another process definition with the same name but on a different OO server, you need to delete the existing process definition and then add a new definition as part of the new engine.
5. Review the `process-defn-tool.log` file for additional details about process definitions.

Installing CSA Content on an OO Server

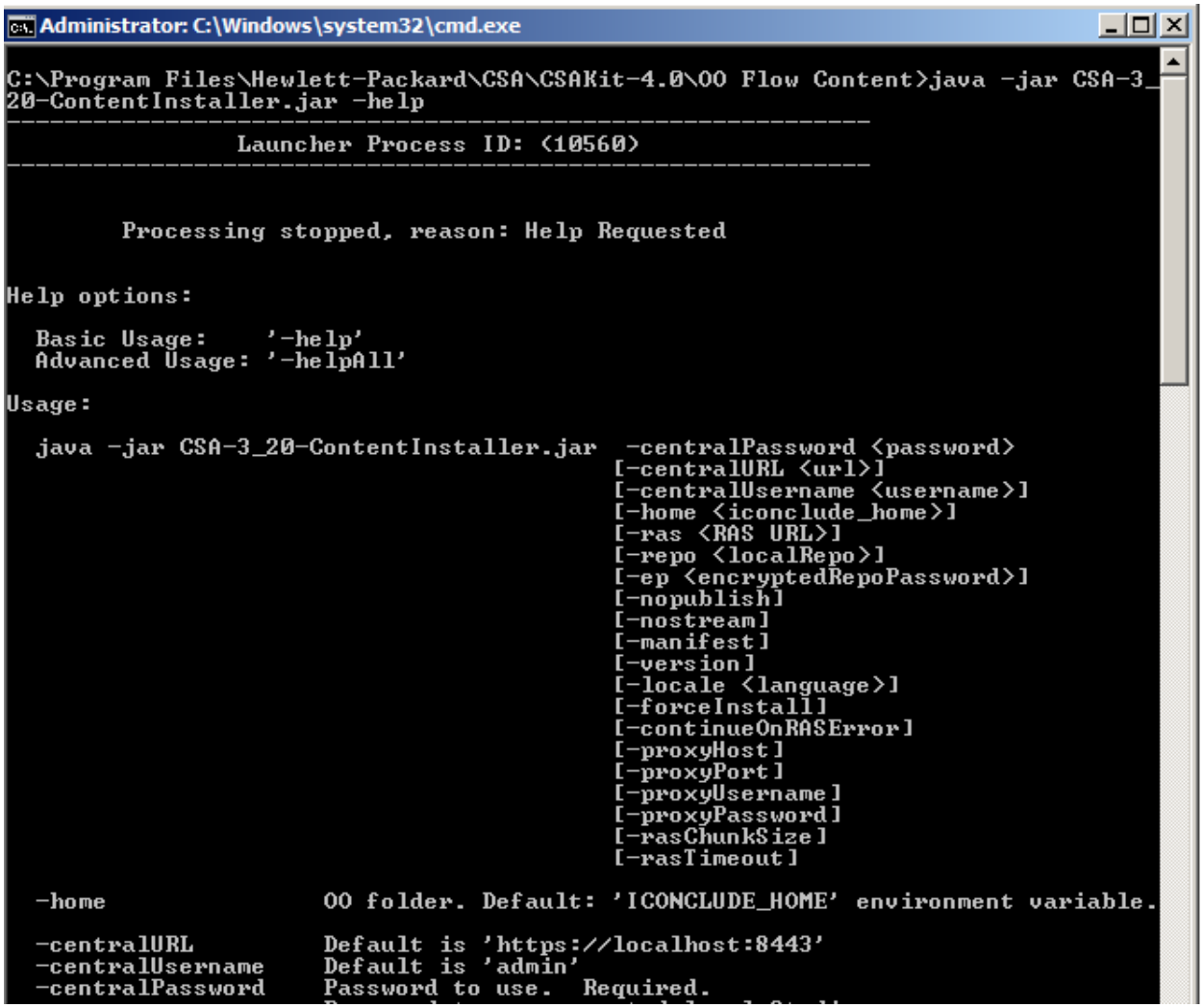
1. Change to the OO Flow Content directory under CSAKit-4.0:



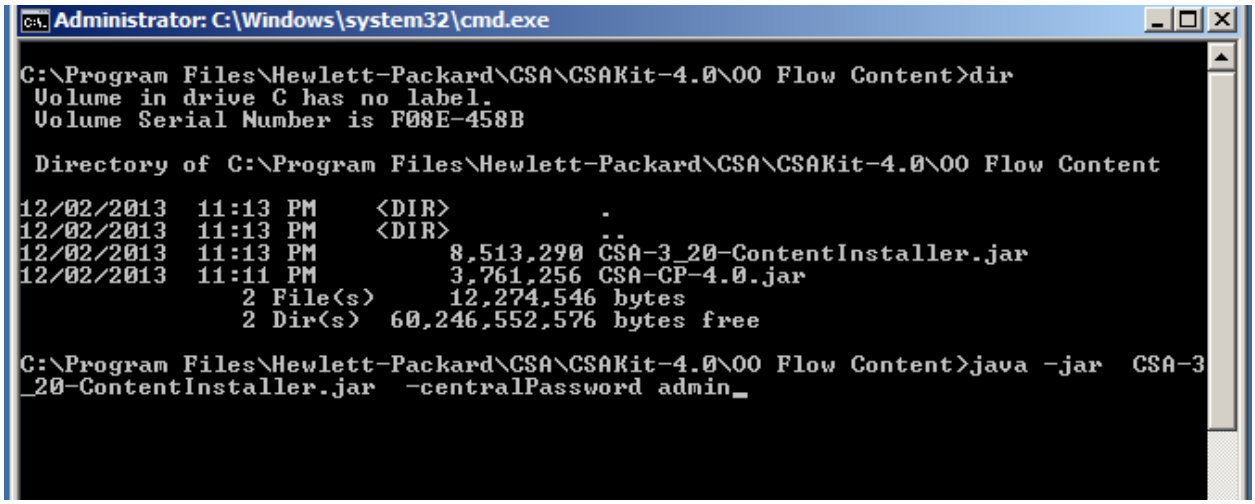
2. Open a command line window at this location:



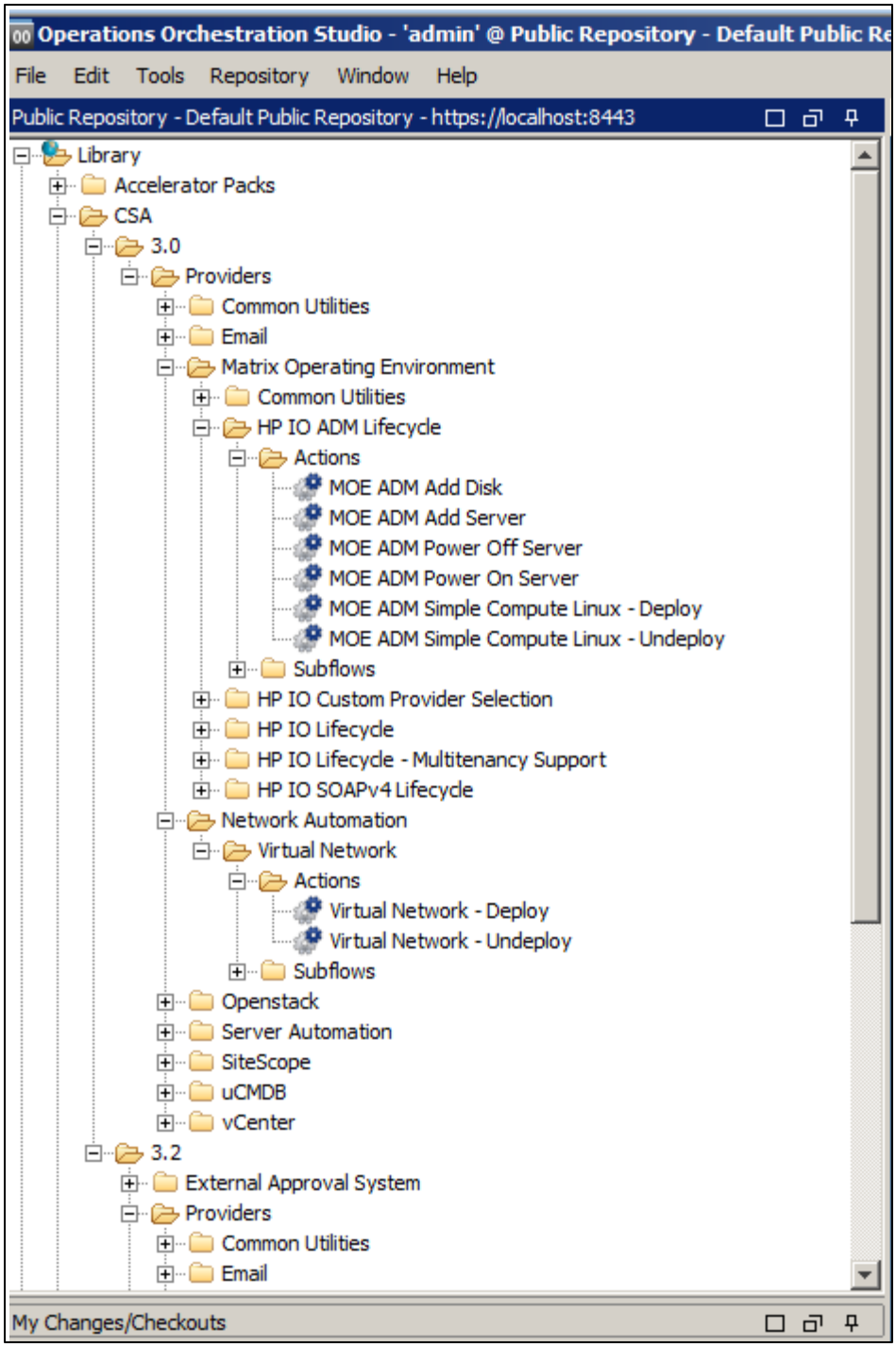
3. Run the content installer help as shown below. Doing so displays options for running the `content jar` file.



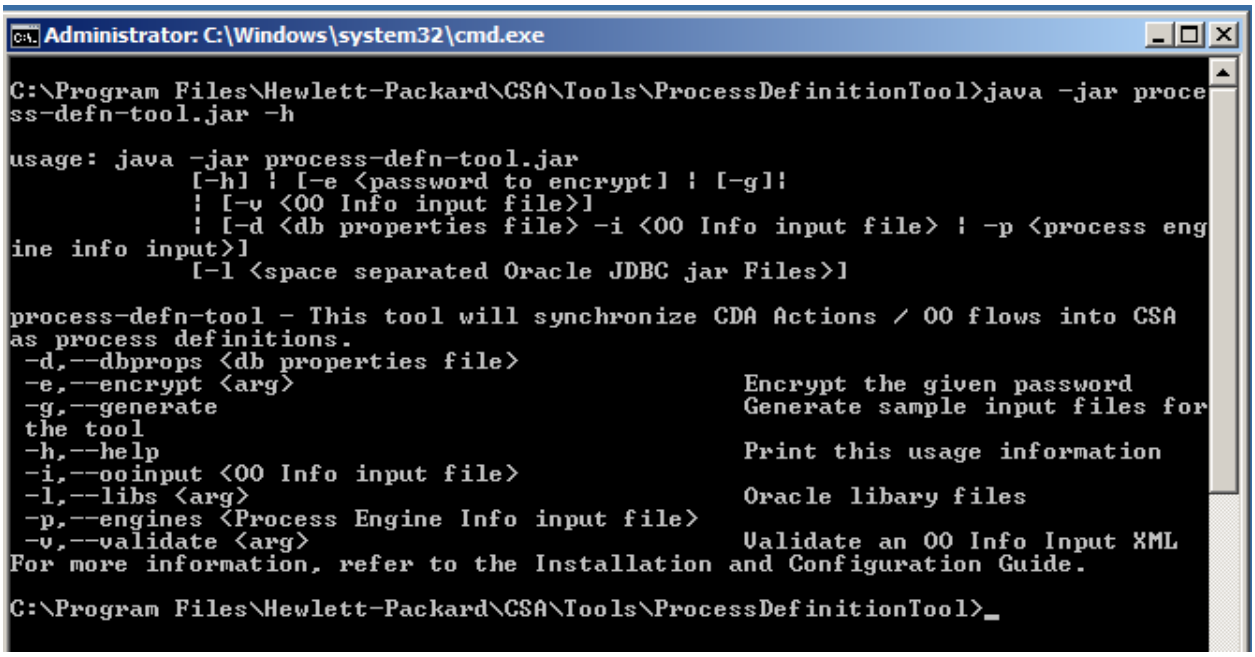
4. A typical invocation of the `content jar` for an OO server running locally is shown below (in this example the server password is **admin**).



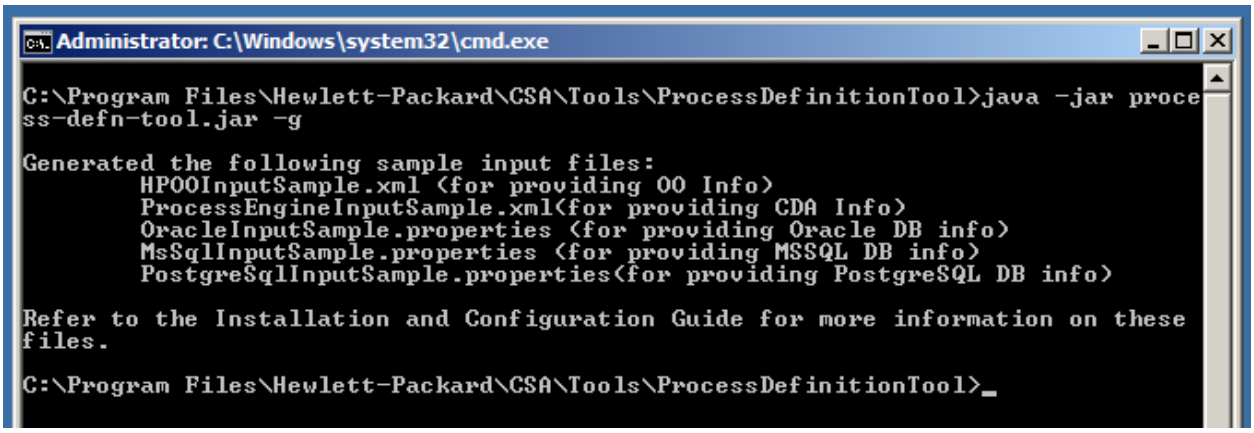
5. This action should import OO flow content to the OO server. Open HP OO Studio and you will see the HP CSA Content imported to a folder under CSA.



6. Now change to the directory where the PDT is installed and invoke PDT help as shown:



7. Running the PDT with the `-g` option generates the sample file needed for the initial run.



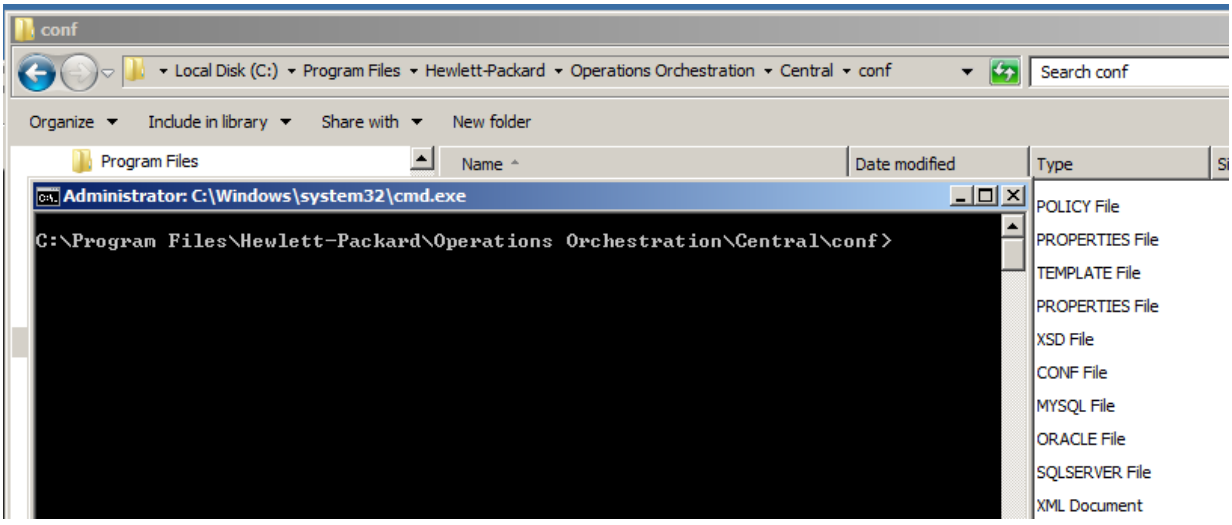
8. The file `HPOOInputSample.xml` contains information about all the OO flow contents. The sample file that gets generated contains three `<ooengine/>` instances.

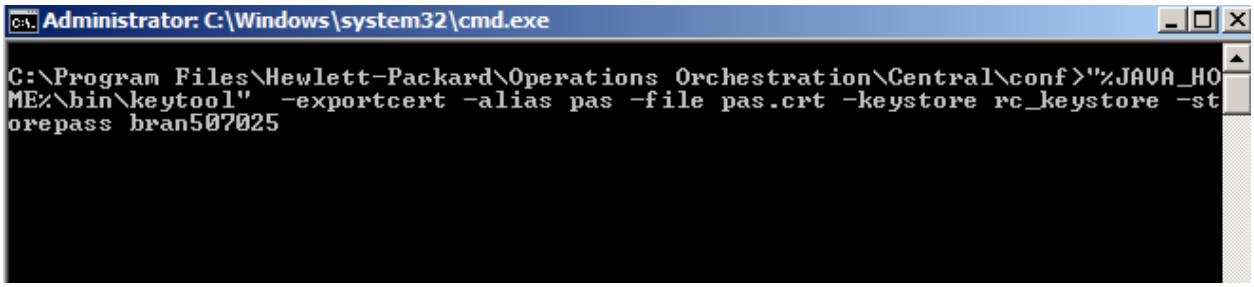
```
<ooengine
  name="OO-MACHINE-NAME"
  uri="https://localhost:8443/PAS/services/WSCentralService"
  truststore="C:/Program Files/Java/jre7/lib/security/cacerts"
  truststorePassword="ENC(q6ctyVrBrqWIp1O7ROOq58CrZh8tzPkP)"
  username="admin"
  password="ENC(OKnPim+Ox/CEVeJJMpLnIg==)">
  <!-- To import all the flows under a single folder, use the syntax
     <folder path="/path/to/folder">, as shown below
     -->
  <folder path="/Library/CSA/3.0/Providers/Matrix Operating Environment/HP IO Lifecycle/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/Matrix Operating Environment/HP IO Lifecycle - Multitenancy Support/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/Matrix Operating Environment/HP IO ADM Lifecycle/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/Server Automation/ADM Application Deployment/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/Server Automation/DMA Workflows Deployment/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/Server Automation/Manage Servers/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/SiteScope/Server Monitoring/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/uCMDB/Configuration Management/Actions"/>
  <folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Clone Server/Actions" update="true" />
  <folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Flex Server Count/Actions" update="true" />
  <folder path="/Library/CSA/3.0/Providers/vCenter/vCenter Flex Server Resources/Actions" update="true"  />
  <folder path="/Library/CSA/3.0/Providers/Matrix Operating Environment/HP IO Custom Provider Selection/Actions"/>
```

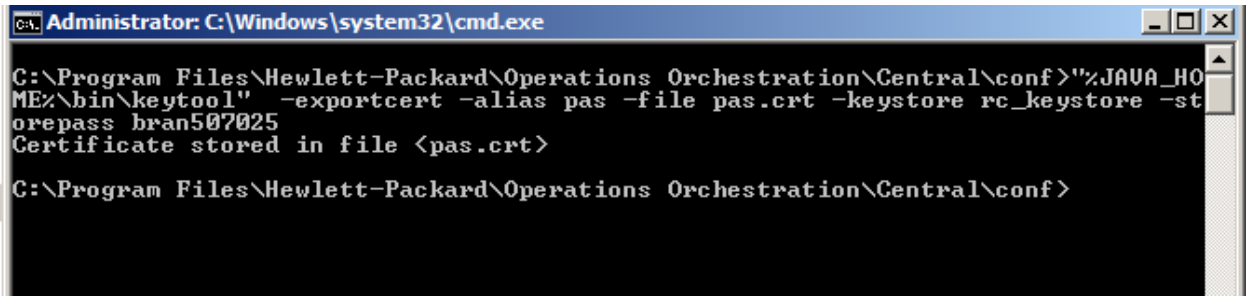
```
 <ooengine
   accessPointType="EXTERNAL_APPROVAL"
   name="OO-EXTERNAL-APPROVAL"
   uri="https://localhost:8443/PAS/services/WSCentralService"
   truststore="C:/Program Files/Java/jre7/lib/security/cacerts"
   truststorePassword="ENC(q6ctyVrBrqWIp1O7ROOq58CrZh8tzPkP)"
   username="admin"
   password="ENC(OKnPim+Ox/CEVeJJMpLnIg==)">

   <folder flow="true" path="/Library/CSA/3.2/External Approval System/Service Manager/Actions/SM Initiate Request Appro

 </ooengine>
 <ooengine
   accessPointType="RESOURCE_POOL_SYNC"
   name="OO-RESOURCE-POOL-SYNC"
   uri="https://localhost:8443/PAS/services/WSCentralService"
   truststore="C:/Program Files/Java/jre7/lib/security/cacerts"
   truststorePassword="ENC(q6ctyVrBrqWIp1O7ROOq58CrZh8tzPkP)"
   username="admin"
   password="ENC(OKnPim+Ox/CEVeJJMpLnIg==)">

   <folder path="/Library/CSA/3.2/Resource Pool Sync/Actions" recursive="true" update="true" />

 </ooengine>
```

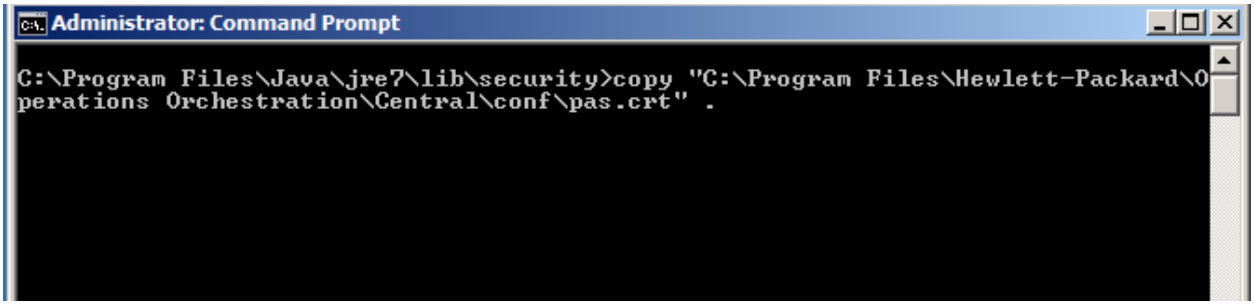
9. Before updating truststore and truststorepassword for all these `<ooengine/>`instances, make sure the OO certificate-related information is copied to the Java truststore used for running the tool.

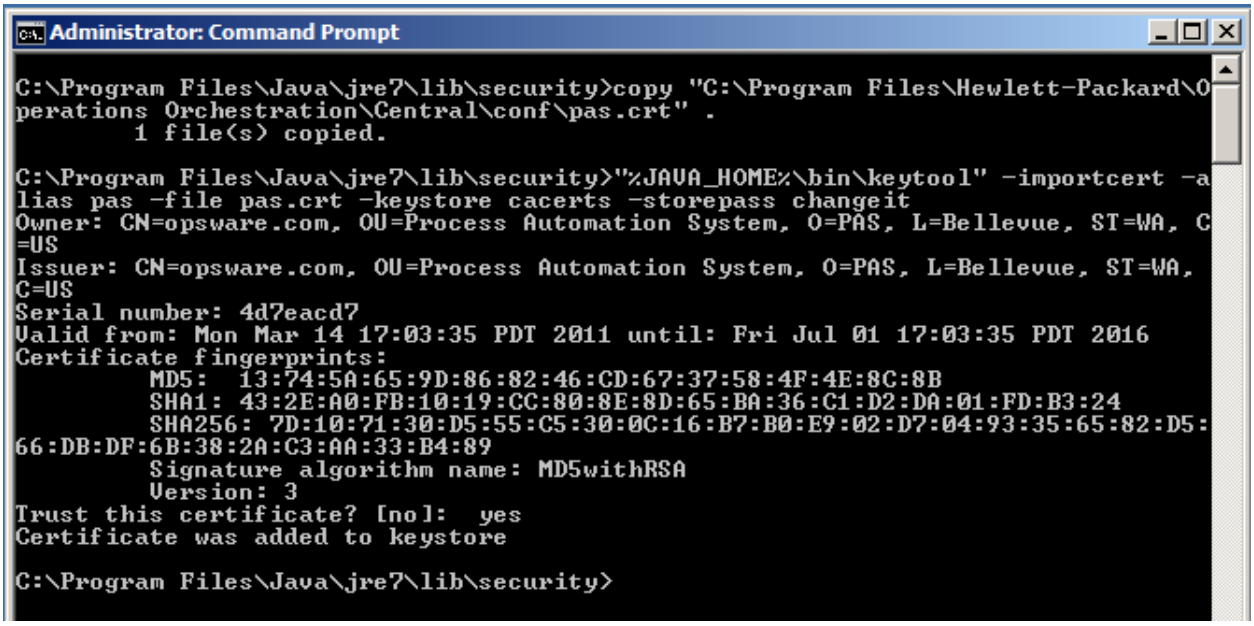
   For OO 9.x version:




Administrator: C:\Windows\system32\cmd.exe
C:\Program Files\Hewlett-Packard\Operations Orchestration\Central\conf>"%JAVA_HOME%\bin\keytool" -exportcert -alias pas -file pas.crt -keystore rc_keystore -storepass bran507025


Administrator: C:\Windows\system32\cmd.exe
C:\Program Files\Hewlett-Packard\Operations Orchestration\Central\conf>"%JAVA_HOME%\bin\keytool" -exportcert -alias pas -file pas.crt -keystore rc_keystore -storepass bran507025
Certificate stored in file <pas.crt>
C:\Program Files\Hewlett-Packard\Operations Orchestration\Central\conf>

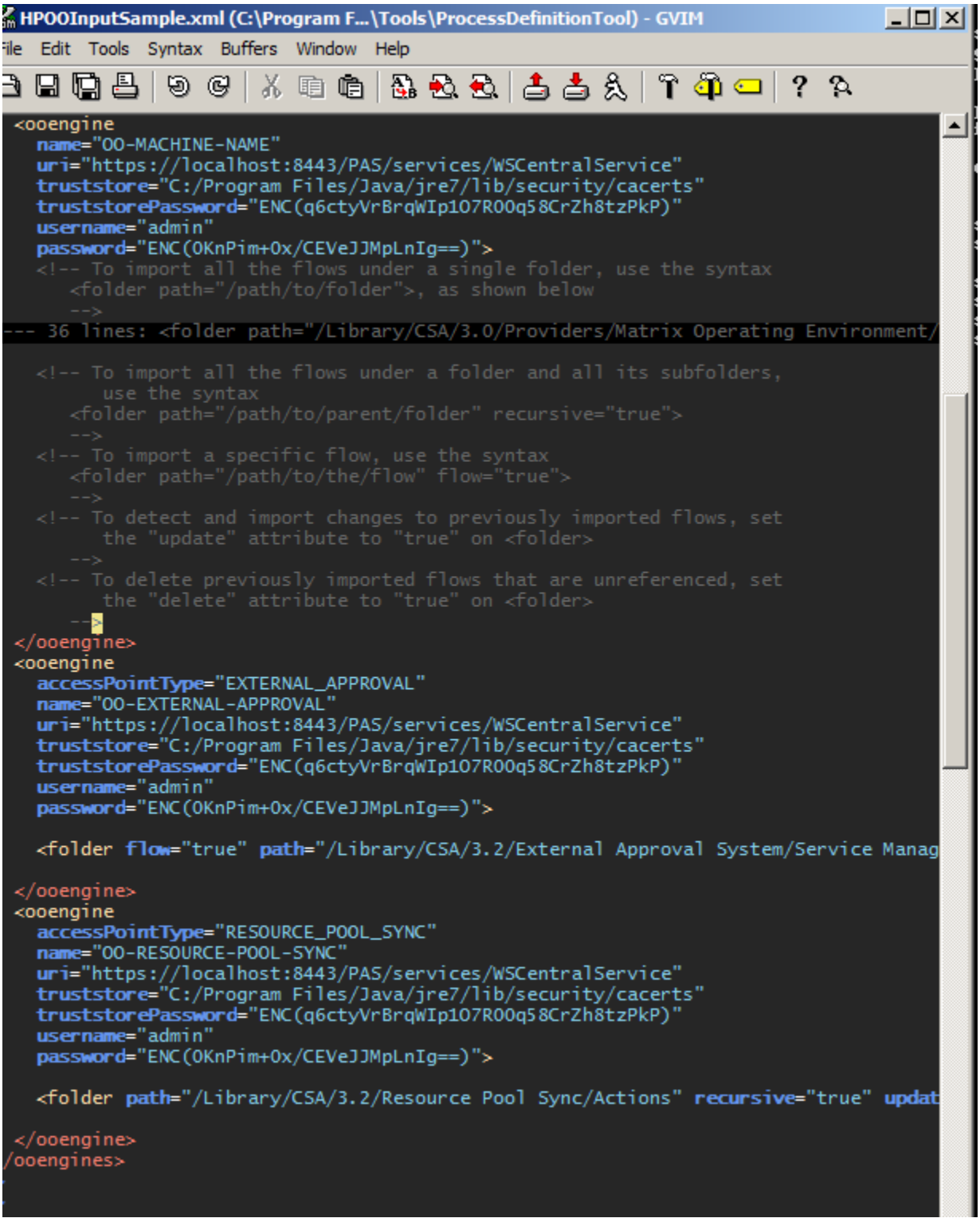

Administrator: Command Prompt
C:\Program Files\Java\jre7\lib\security>copy "C:\Program Files\Hewlett-Packard\Operations Orchestration\Central\conf\pas.crt" .


Administrator: Command Prompt
C:\Program Files\Java\jre7\lib\security>copy "C:\Program Files\Hewlett-Packard\Operations Orchestration\Central\conf\pas.crt" .
1 file(s) copied.
C:\Program Files\Java\jre7\lib\security>"%JAVA_HOME%\bin\keytool" -importcert -alias pas -file pas.crt -keystore cacerts -storepass changeit
Owner: CN=opsware.com, OU=Process Automation System, O=PAS, L=Bellevue, ST=WA, C=US
Issuer: CN=opsware.com, OU=Process Automation System, O=PAS, L=Bellevue, ST=WA, C=US
Serial number: 4d7eacd7
Valid from: Mon Mar 14 17:03:35 PDT 2011 until: Fri Jul 01 17:03:35 PDT 2016
Certificate fingerprints:
MD5: 13:74:5A:65:9D:86:82:46:CD:67:37:58:4F:4E:8C:8B
SHA1: 43:2E:A0:FB:10:19:CC:80:8E:8D:65:BA:36:C1:D2:DA:01:FD:B3:24
SHA256: 7D:10:71:30:D5:55:C5:30:0C:16:B7:B0:E9:02:D7:04:93:35:65:82:D5:66:DB:DF:6B:38:2A:C3:AA:33:B4:89
Signature algorithm name: MD5withRSA
Version: 3
Trust this certificate? [no]: yes
Certificate was added to keystore
C:\Program Files\Java\jre7\lib\security>

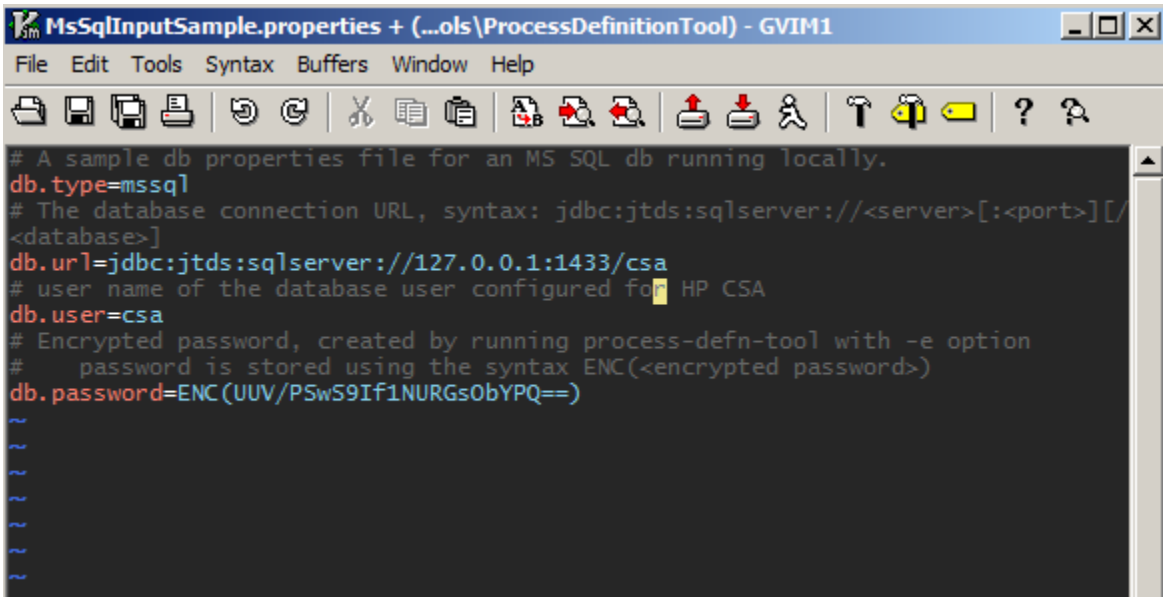
For OO 10.x version, the only differences from 9.x version are shown below:

	OO 9.x	**OO 10.x**
Alias (used to export OO SSL certificate)	pas	tomcat
Password (used to export OO SSL certificate)	bran507025	changeit
Keystore location	<ICONCLUDE_HOME>/Central/conf/rc_keystore	<ICONCLUDE_HOME>/Central/var/security/key.store

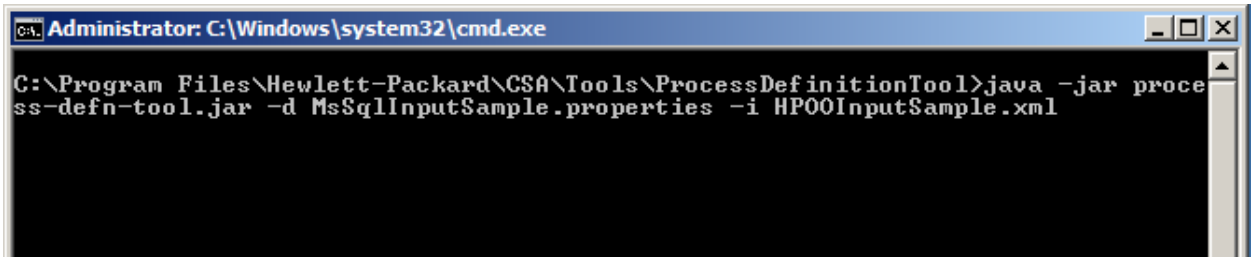
10. Now update all of the `<ooengine/>` truststore attribute locations with the value of `cacerts`.



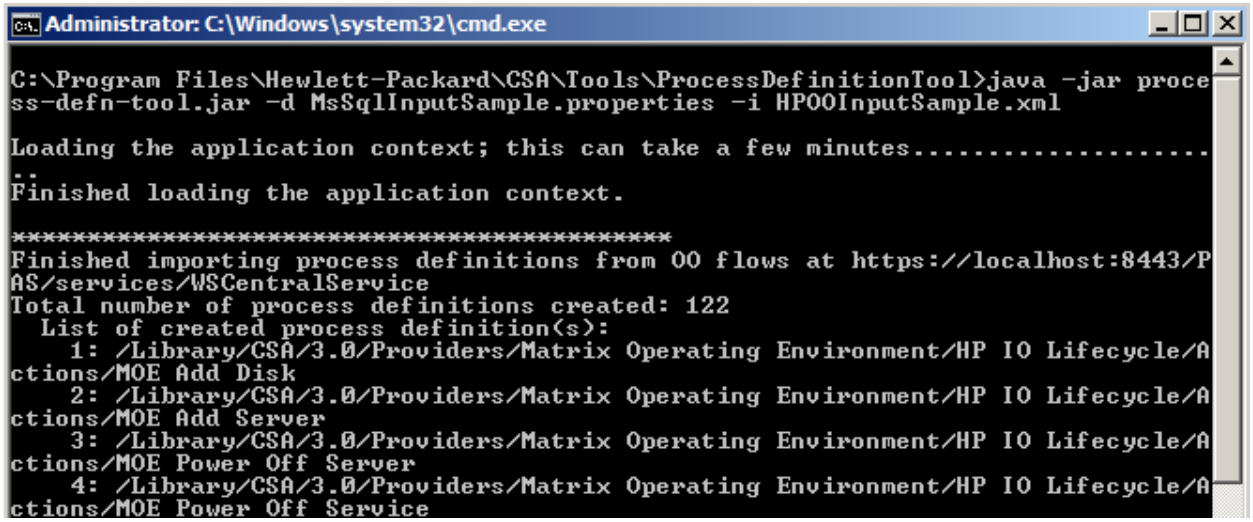
11. Now update the corresponding database file:



12. Running the PDT:

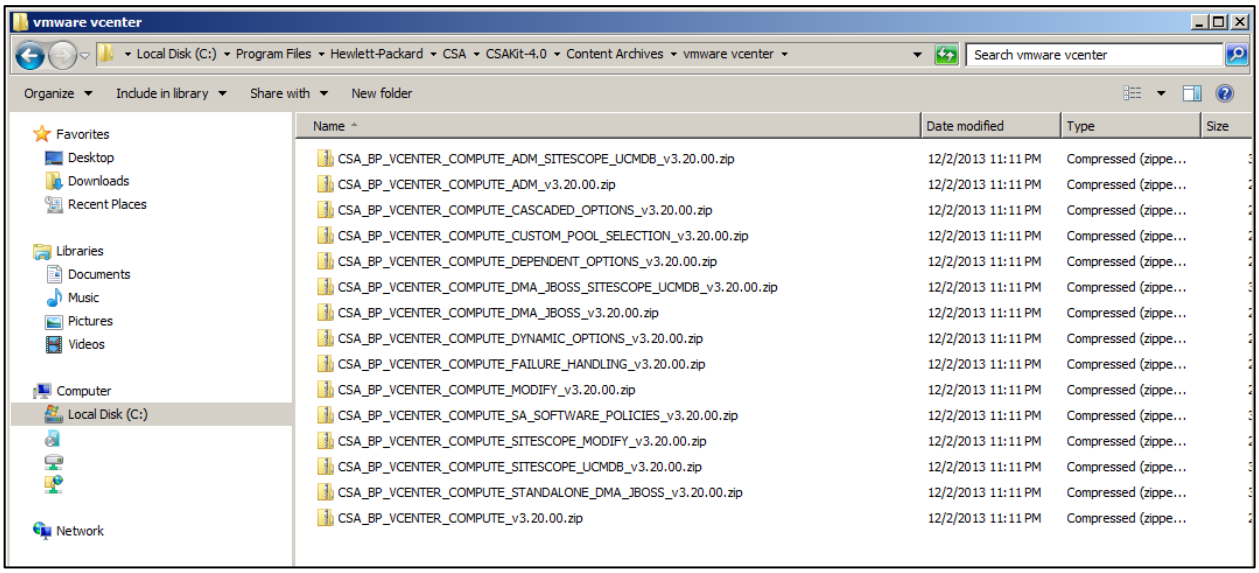


13. Typical output for a successful run of PDT:

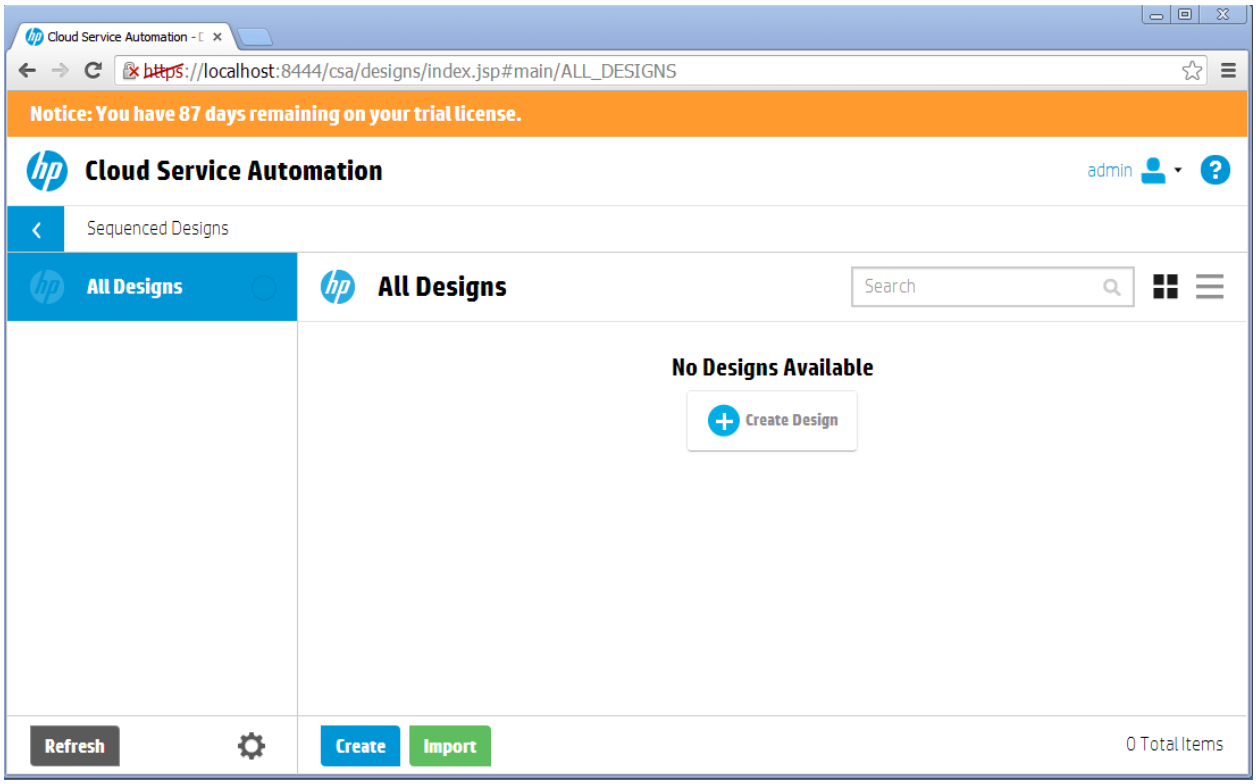


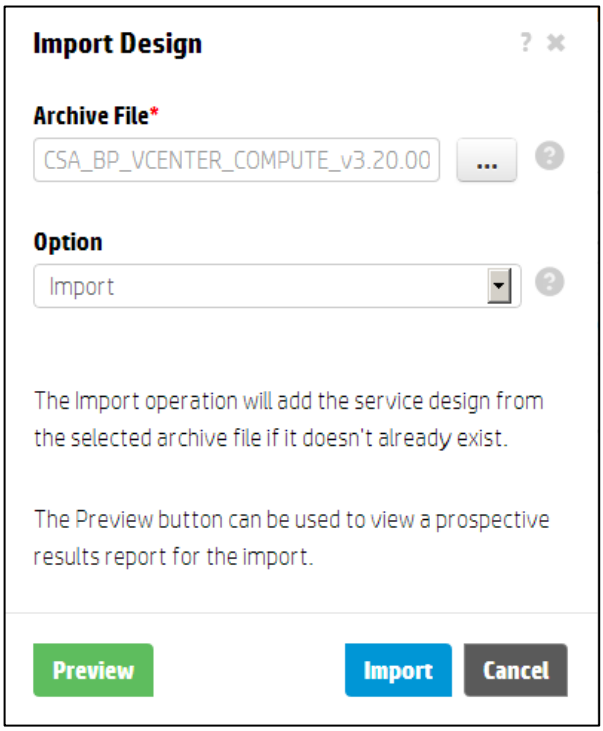
```
    120: /Library/CSA/3.2/Providers/uCMDB/Configuration Management/Actions/uCMDB
 Delete
    121: /Library/CSA/3.2/Providers/Network Automation/Virtual Network/Actions/V
irtual Network - Deploy
    122: /Library/CSA/3.2/Providers/Network Automation/Virtual Network/Actions/V
irtual Network - Undeploy

Total number of process definitions updated: 0
Total number of process definitions deleted: 0

**********************************************
Finished importing process definitions from OO flows at https://localhost:8443/P
AS/services/WSCentralService
Total number of process definitions created: 1
  List of created process definition(s):
    1: /Library/CSA/3.2/External Approval System/Service Manager/Actions/SM Init
iate Request Approval

Total number of process definitions updated: 0
Total number of process definitions deleted: 0

**********************************************
Finished importing process definitions from OO flows at https://localhost:8443/P
AS/services/WSCentralService
Total number of process definitions created: 1
  List of created process definition(s):
    1: /Library/CSA/3.2/Resource Pool Sync/Actions/VCenter Sync Resource Capacit
y

Total number of process definitions updated: 0
Total number of process definitions deleted: 0

**********************************************

C:\Program Files\Hewlett-Packard\CSA\Tools\ProcessDefinitionTool>_
```

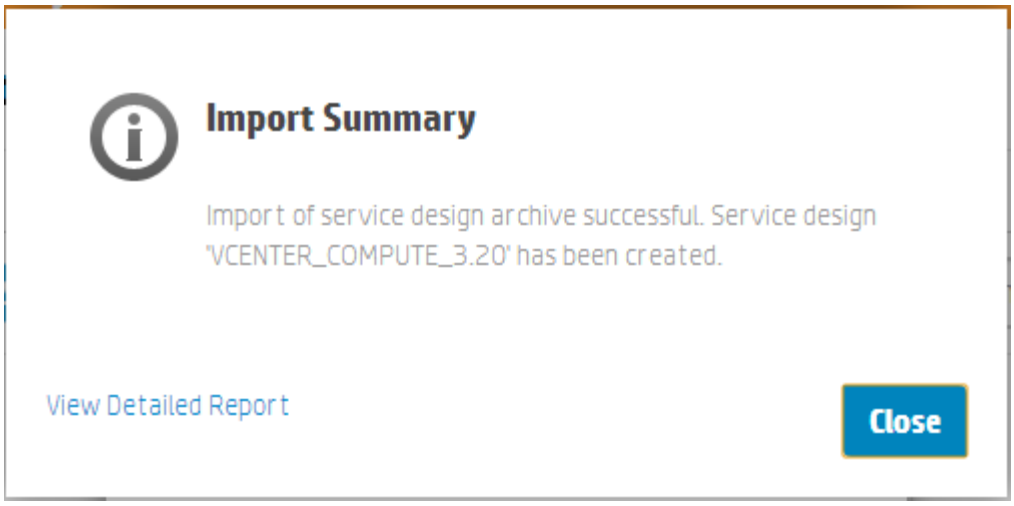
14. The following screenshot shows an example of some of the archive files that are available for import after CSA is installed.

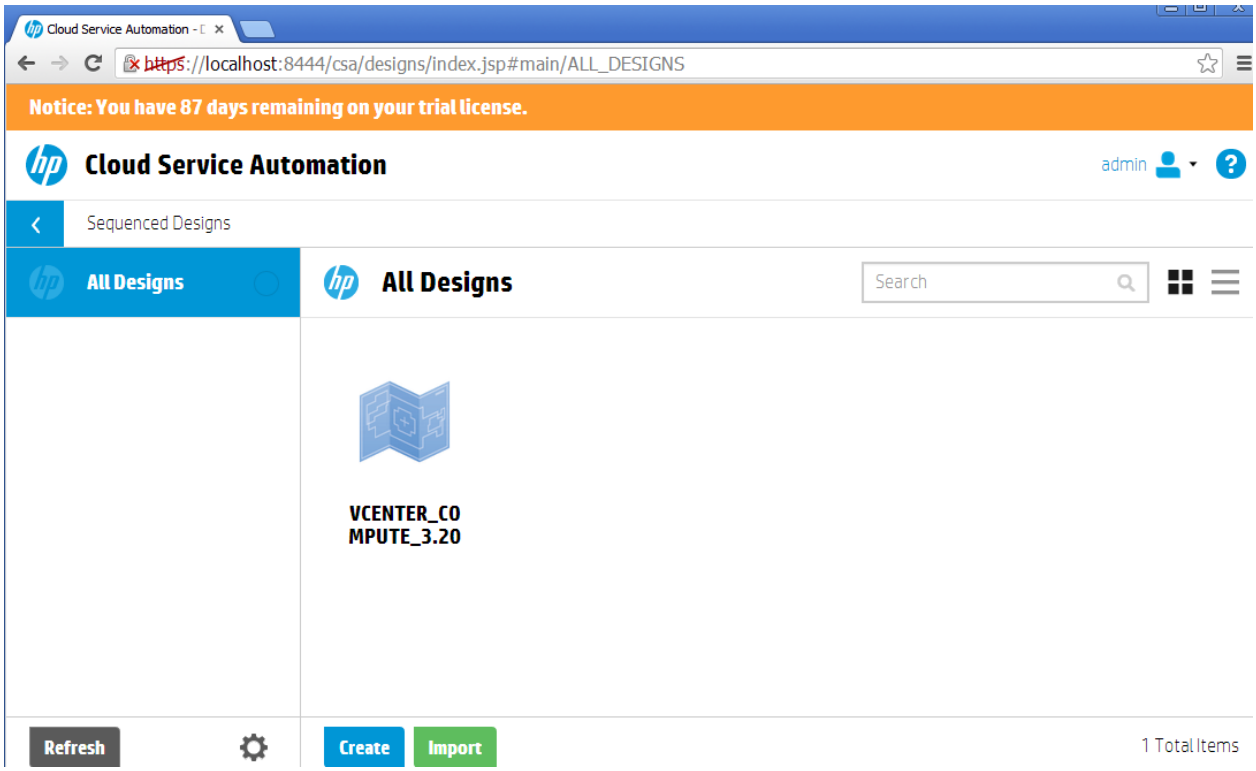


15. Importing a service design through the Console UI:






Import Summary
Import of service design archive successful. Service design 'VCENTER_COMPUTE_3.20' has been created.
View Detailed Report
Close


Cloud Service Automation
https://localhost:8444/csa/designs/index.jsp#main/ALL_DESIGNS
Notice: You have 87 days remaining on your trial license.
Cloud Service Automation
admin
Sequenced Designs
All Designs
All Designs
Search
VCENTER_COMPUTE_3.20
Refresh
Create
Import
1 Total Items