HP Cloud Service Automation

Software version: 4.10

Content Archive Tool

Table of contents

Legal Notices

Warranty

The only warranties for HP 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. HP 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 HP 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

© Copyright 2010-2014 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

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

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link 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 HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest

- Submit and track support cases and enhancement requests

- Download software patches

- Manage support contracts

- Look up HP support contacts

- Review information about available services

- Enter into discussions with other software customers

- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Overview

HP CSA allows for the exporting and importing of many of the artifacts that provide the basis for cloud automation. The export operation provides the ability to preserve the selected artifacts so they can be used to replicate the services on another system or to restore the artifacts. These exported archive files are preserved in an industry-standard zip archive file format.

Installing or replacing artifacts on the system is supported by import and update operations. The import behavior adds artifacts; the update operation overwrites matching artifacts. Note that archives created on HP CSA 3.10 or HP CSA 3.20 can be imported into HP CSA 4.x.

The HP CSA Content Archive Tool is a command line interface for exporting and importing artifacts to and from CSA archive files. An archive contains XML documents that represent the primary artifact being exported, dependent artifacts, and any customized images or dynamic property JSP files necessary to represent the entire artifact. For example, a service offering archive contains the XML documents representing the service offering, the service design associated with the service offering, resource offerings associate with the service design, and a Manifest XML document. The service offering archive will also contain any dynamic property JSP files and images used for customization of all these artifacts.

Supported Operations

Export

Use the export operation to export supported artifacts as archives. An archive name includes the primary artifact type, its display name, and its ID. A separate archive will be created for each primary, or topmost, artifact in a hierarchy being exported. A summary of the export operation is displayed after the operation completes.

Validate

Use the validate operation to preview actions that will be taken by the import operation. This operation displays a summary of artifacts that can be imported from the specified archive and artifacts that already exist on the system.

Import and Update

Installing or replacing artifacts on the system is supported by import and update operations. The default import behavior only adds artifacts, whereas the update behavior also replaces matching artifacts. A summary of the import operation is displayed after the operation completes.

Default Import process

By default, an artifact is not imported if the source and existing target artifacts are considered to be functionally equivalent which is described as follows:

- Service designs, service offerings, resource environments and component palettes with the same internal name are considered to be functionally equivalent and are not imported.

- Catalogs with the same internal name and organization ID are considered to be functionally equivalent and are not imported.

- Resource offerings with the same resource category, provider type, properties and actions are considered to be functionally equivalent and are not imported. For more information, see **Functionally Equivalent Resource Offerings** on page 9.

Important
If the import operation fails, any changes made will be rolled back so that the target system is left unchanged by the import operation.

Table 1 Values evaluated to determine functional equivalence

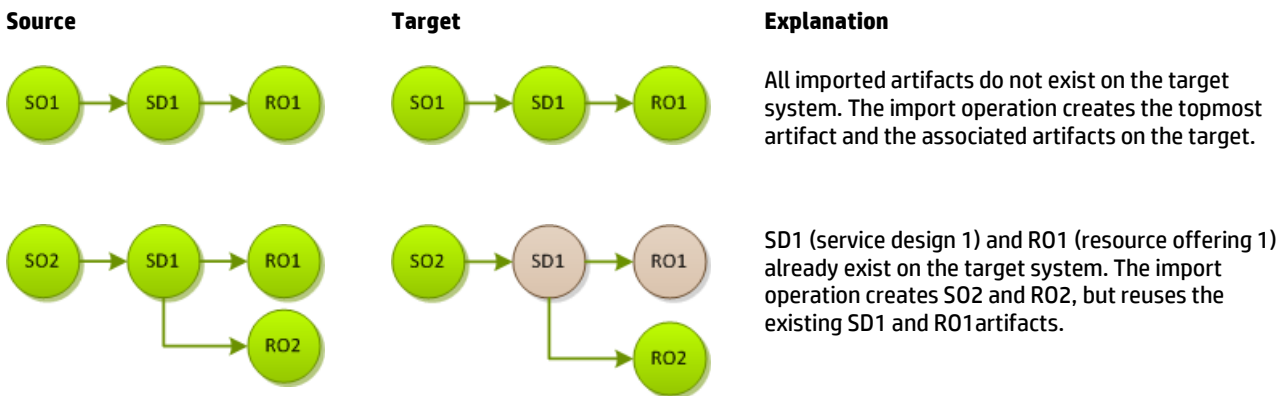| Artifact Type	Values evaluated
Component Palette	Internal name
Resource Offering	Resource category, provider type, properties and actions
Service Design	Internal name
Service Offering	Internal name
Resource Environment	Internal name
Catalog	Internal name and organization ID

Imported artifacts are associated with existing artifacts in the system. If the existing artifact is the primary artifact in a hierarchy, the import operation exits without importing any of the archive content. For example, when importing a service offering archive, if the service design in the archive already exists in the system and the service offering is not in the system, the service offering is imported and is associated with the existing service design. However, if the service offering (in this case the primary artifact) already exists in the system, the import operation exits without any artifacts being imported. Figure 1 includes additional examples of how this process works.
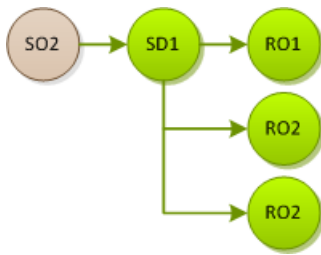
Important
Artifact import rules differ for catalogs and component palettes. If a catalog already exists on the target system, the catalog is updated with any added or removed supported artifacts. Import of a component palette by default is an update operation, which is described below. The import process imports component palettes, including their associated component types, templates, and component type constraints.

Figure 1 Example import scenarios

Service Offering (SO) = Topmost artifact in hierarchy
Service Design (SD) = Service design artifact associated with the service offering (SO)
Resource Offering (RO) = Resource offering artifact associated with the service design (SD)

Source	Target	Explanation
		All imported artifacts do not exist on the target system. The import operation creates the topmost artifact and the associated artifacts on the target.
		SD1 (service design 1) and RO1 (resource offering 1) already exist on the target system. The import operation creates SO2 and RO2, but reuses the existing SD1 and RO1 artifacts.

Operation Fails	The import operation exits without importing any content because S02 already exists on the target system.

Import using validate option
This operation combines the validate and import operations. Before importing, this operation validates the contents of the archive and displays a summary of changes that will be made during the import operation. After displaying this summary, the import with validation operation prompts the user before proceeding with the import.

Import using update option
Import using the update option is to be used with caution. Functionally equivalent artifacts that exist on the target system are updated (overwritten) with changes from the archive. New artifacts are created if they do not exist on the target system.

Import using updatePreserveExisting option
The updatePreserveExisting operation is to be used with caution. This operation imports all artifacts present in an archive, whether they exist on the target system or not. During this operation any artifact with the same name on the target system as in the archive is preserved as follows: the name, display name, and description of the existing artifact are modified internally, and the display name and the description are appended with "Superseded on" and the date. The name, display name, and description of the imported artifact remain intact.

This operation is supported for service offerings, service designs, resource offerings, and resource environments. In the case of catalogs, this process applies to the associated artifacts, but does not apply to the catalog itself. This operation does not apply to component palettes.

Important
With all methods of import and update, if a dynamic property JSP file by the same name already exists on the target system, the existing file will be used and not imported from the archive.

See **Examples** on page 16 for examples of all operations.

Note that content is added to the log file, content-archive-tool.log, for all operations.

Recommended best practice
1. Create a backup of your system or data. Before proceeding, with any import option, create a backup of any artifacts you might be affecting by using the export operation to save an archive zip file.
2. Since some of the import and update options are destructive to existing data, make sure you understand the differences between the available options to ensure you choose the one that matches your expectations.

Supported Artifacts

The following CSA artifacts are supported in CSA artifact archives: component palettes, resource offerings, service designs, service offerings, resource environments, and catalogs.

Component Palettes

Content archive for component palettes
Exporting a component palette creates a content archive (.zip) file. The content archive contains XML documents for the component palette and its associated component types, component templates, and component type constraints. If component templates in a component palette have resource bindings on resource offerings, when the component palette is exported, resource offering XML files are included in the component palette archive. The archive also contains icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

Import process for component palettes
- Import of a component palette by default is an update operation. The import process imports component palettes, including their associated component types, templates, and component type constraints.
- If the component palette already exists on the system, it is updated with any added or removed component types, component templates, and component type constraints.

- Component palettes with the same internal name are considered to be functionally equivalent and are not imported.
- Circular dependencies between palettes are not allowed. For example, Palette A cannot have components that are derived from Palette B while, at the same time, Palette B has components that are derived from Palette A.
- When importing multiple, dependent component palettes that already exist on the target system, it is recommended that you import the palettes in the order of their dependencies.
- The import process for component palettes always ensures system integrity with respect to component type derivation and property propagation. For example, if you have modified properties of a component type in a palette called "Palette A," and there are component types in other palettes that derive from this component type. When an update of 'Palette A' occurs during import, changes that have been made to the component types are propagated to component types in other palettes.

Import process for component templates in a component palette with resource bindings on resource offerings

- At component palette import, resource offerings are either created when a resource offering with same name does not exist on the system, or updated when a resource offering with the same name exists. This resource offering import behavior occurs because the import option for a component palette is always update.
- Resource offerings are not automatically associated with resource providers when component palettes are imported. The –p command line option will NOT override this behavior.

Resource Offerings

Import requirements and prerequisites
- Resource categories
  Resource categories (such as Compute) and provider types (such as VMware vCenter) are resolved first by **name** and second by **display name** during import of a resource offering. Out-of-the box resource categories and provider types have identical **name** values on all installations of CSA, and automatically resolve correctly during import. User-created resource categories and provider types do not have a **name** match on different installations of HP CSA, and instead are resolved by **display name**. For example, if a user created resource category with a display name of Auditing is used for a resource offering, when that resource offering is imported on another HP CSA installation, an attempt will be made to match a resource category with a **display name** of Auditing. This match is successful only if the user has already created this corresponding resource category on the system in which the import occurs. If a resource category or provider type cannot be resolved by either **name** or **display name**, a new resource category or provider type is automatically created during import of the resource offering. There is no need to add user-created resource categories and provider types ahead of time on the import system; however, if you have done so, ensure the **display name** values used match those on the export system.

- Process Definitions
  During an import operation, required dependencies may not exist on the system, which produces an import error that identifies the missing dependencies. For resource offerings this can include missing process definitions. These process definitions must be imported to the HP CSA installation in which the resource offering will be imported. During import, process definitions are resolved by **name**. For process definitions that are associated with the HP Operations Orchestration process engine, the **name** corresponds to the full path to the HP Operations Orchestration flow (for example, /Library/CSA/3.2/Providers/vCenter/vCenter Clone Server/Actions/vCenter Simple Compute - Deploy). Make sure that all these flows have been imported prior to importing resource offerings. Make sure that these flows have identical signatures and identical paths as the flows on the system from which the resource offering was exported. For instructions about how to import HP Operations Orchestration flows, see the section "Import HP Operations Orchestration Flows" in the *HP Cloud Service Automation Configuration Guide*.

Content archive for resource offerings
Exporting a resource offering creates a content archive (.zip) file. The content archive contains XML documents for the resource offering you are exporting, icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

Default Import process for resource offerings
Resource offerings that have the same resource category, provider type, properties, and actions are considered to be functionally equivalent and are not imported. A detailed definition of functional equivalence can be found below.
Update process for resource offerings
During the update process, resource offerings with the same internal name that exist on the target system are updated (overwritten) with changes from the archive. New resource offerings are created if they do not exist on the target system.

Note
Resource offerings are identified as equivalent by internal name only for the update operation.

UpdatePreserveExisting process for resource offerings
This process imports the resource offering, whether it exists on the target system or not. During this operation, if there is a resource offering with the same name in the system, the name, the display name, and the description of the resource offering are modified

internally; the display name and the description are appended with "Superseded on" and the date. The internal name, display name, and description of the artifact being imported remain intact.

Important

If providers of the same display name and provider type exist on the target system, use the associateToProvider option, -p, to automatically associate to these providers to imported resource offerings. Otherwise, you must manually associate providers to resource offerings.

Functionally Equivalent Resource Offerings

When a resource offering is imported, HP CSA determines if a functionally equivalent resource offering already exists on the system. If a functionally equivalent resource offering exists, then the import is skipped. Functional equivalence is determined by comparing the resource offering to be imported with other resource offerings that exist on the system, as follows:

- Resource offerings are functionally equivalent if they share the same provider type and resource category, the same set of user defined custom properties, and the same set of lifecycle actions.

- Properties are considered equivalent if they share the same type, name, and value (or values for list properties).

- Lifecycle actions are considered equivalent if they share the same process definition, lifecycle state and sub-state, execution order, and action input properties and values.

- There are additional attributes on properties and lifecycle actions that must be identical for equivalence.

For specifics about the precise requirements for resource offering equivalence as it relates to entries in the .zip file produced during resource offering export, see the following table.

Table 2 Functionally Equivalent Resource Offerings

Element	Necessary for Equivalence
property	• name
	• valueType
	• values
	• confidential – only for String property types
action	• processDefinition name
	• lifecycleState name
	• lifecycleSubstate name
	• lifecycleExecOrder
	• errorOnTimeout
	• failOnError
	• timeout
	• all properties must be identical, including the consumerVisible and consumerReadOnly elements for each property
	• consumerVisible
resourceCategory	• isCriticalSystemObject determines if this is an out -of-box resourceCategory. If true, name determines equivalence, otherwise displayName determines equivalence.
providerType	• isCriticalSystemObject determines if this is an out -of-box providerType. If true, name determines equivalence, otherwise displayName determines equivalence.

Service Designs

The import process supports the import of sequence and topology design archives.

Before you import a service design archive, complete the following prerequisites:

1. Import all process definitions that are referenced by the service design, including any referenced by resource offerings that are part of the service design. The process definitions must be imported to the HP CSA installation in which the service design will be imported. During import, process definitions are resolved by **name**. For process definitions that are associated with the HP Operations Orchestration process engine, the **name** corresponds to the full path to the HP Operations Orchestration flow (for example, /Library/CSA/3.2/Providers/vCenter/vCenter Clone Server/Actions/vCenter Simple Compute Linux – Deploy). Make sure that these flows have identical signatures and identical paths as the flows on the system from which the service design was exported. For instructions about how to import OO flows, see the section "Import HP Operations Orchestration Flows" in the *HP Cloud Service Automation Configuration Guide*.

2. When importing archives with one or more topology designs, the following process will be followed to determine the provider and resource pool associated with each design:

   - If –q/--forceCloudOSProvider or –r/--forceHPHelionOpenStackProvider, and –s/--forceResourcePool options are specified in command line,
     i. Check target system for match of provider name specified.
        1. If found, use that provider for import.
        2. If not found, fail topology design import with error provider not found.
     ii. Check target system for match of resource pool name specified.
        1. If found, use that resource pool for import.
        2. If not found, fail topology design import with error resource pool not found for provider.
   - If –q/–r and -s options are not specified in command line,
     i. Check target system for match of provider name used in archive.
        1. If provider name exists on target system, use that provider for import.
        2. If provider name does not exist on target system, check for default provider in CSA resource management.
           a. If default provider is set, use that provider for import.
           b. If no default provider set, fail topology design import with error provider not found.
     ii. If a provider is successfully matched by name used in archive, check target system for match of resource pool name used in archive.
        1. If resource pool name exists on target system, use that resource pool for import.
        2. If resource pool name does not exist on target system, check for default resource pool in CSA resource management.
           a. If default resource pool is set and exists on target system, use that resource pool for import.
           b. If no default is set, fail topology design import with error resource pool not found for provider.
     iii. If a default provider is used, check target system for default resource pool.
        1. If default resource pool is set and exists on target system, use that resource pool for import.
        2. If no default resource pool is set, Fail topology design import with error resource pool not found for provider.

3. The export operation does not export any custom service component types that the user has created on the source system. If the service design you want to import has a dependency on any custom component types, the component palettes that contain these custom component types must be imported before the service design can be imported.

Content archive for service designs
Exporting a service design creates a content archive (.zip) file. The content archive contains XML documents for the service design you are exporting, as well as associated artifacts, dynamic property JSP files, icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.
Import process for service designs
The import process imports archives of service designs and their supported artifacts. Supported artifacts for service designs include associated resource offerings. Service designs with the same internal name are considered to be functionally equivalent and are not imported.
Update process for service designs
During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.
Update preserveExisting process for service designs
This process imports all the artifacts present in an archive, whether they exist on the target system or not. During this operation, if there is an artifact with the same internal name in the system, the name, the display name, and the description of the artifact are modified internally; the display name and the description are appended with "Superseded on" and the date. The internal name, display name, and description of the artifact being imported remain intact.

During the import operation on a service design archive, if a service design has active service offerings associated with it, the subscriber options for the service design being imported must match that of the service design on the target system; otherwise, the import of the service offerings fails.

If any functionally equivalent resource offerings already exist on the target system, these resource offerings are reused for binding with the service design and these resource offerings are not imported.

Service Offerings

Content archive for service offerings
Exporting a service offering creates a content archive (.zip) file. The content archive contains XML documents for the service offering you are exporting, as well as associated artifacts, dynamic property JSP files, icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

Import process for service offerings
The import process imports archives of service offerings and their supported artifacts. Supported artifacts for service offerings include associated service designs and resource offerings. Service offerings with the same internal name are considered to be functionally equivalent and are not imported.
Update process for service offerings
During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.
UpdatePreserveExisting process for service offerings
This process imports all the artifacts present in an archive, whether they exist on the target system or not. During this operation, if there is an artifact with the same name in the system, the name, the display name, and the description of the artifact are modified internally; the display name and the description are appended with "Superseded on" and the date. The name, display name, and description of the artifact being imported remain intact.

Resource Environments

Content archive for resource environments
Exporting a resource environment creates a content archive (.zip) file. The content archive contains an XML document for the resource environment you are exporting, as well as the Manifest XML document, which contains meta-information about the archive files.

Import process for resource environments
The import process imports archives of resource environments. Resource environments with the same internal name are considered to be functionally equivalent and are not imported.
Update process for resource environments
During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.
UpdatePreserveExisting process for resource environments
This process imports the resource environment present in the archive, whether it exists on the target system or not. During this operation, if there is an artifact with the same name in the system, the name, the display name, and the description of the artifact are modified internally; the display name and the description are appended with "Superseded on" and the date. The name, display name, and description of the artifact being imported remain intact.

Important
If providers of the same display name and provider type exist on the target system, use the associateToProvider option, -p, to automatically associate to these providers to imported resource offerings. Otherwise, you must manually associate providers to resource offerings.

Catalogs

Content archive for catalogs
Exporting a service catalog creates a content archive (.zip) file. The content archive contains XML documents for the service catalog you are exporting, as well as associated artifacts, dynamic property JSP files, icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

Import process for service catalogs
The import process imports archives of service catalogs and their supported artifacts. Supported artifacts for service catalogs include associated service designs, resource offerings, service offerings, and resource environments. If the service catalog already exists, the catalog is updated with any added or removed supported artifacts. Service catalogs with the same internal name are considered to be functionally equivalent and are not imported.
Update process for service catalogs
During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.
UpdatePreserveExisting process for service catalogs

This process applies to the associated artifacts, but does not apply to the catalog itself. The process imports all the artifacts present in an archive, whether they exist on the target system or not. During this operation, if there is an artifact with the same name in the system, the name, the display name, and the description of the artifact are modified internally; the display name and the description are appended with "Superseded on" and the date. The name, display name, and description of the artifact being imported remain intact.

During the import of a catalog, an organization identifier is required. You can specify this identifier organization using the –o option. If this is not specified in the command line, the organization identifier from the ownedBy field in the catalog's XML document is used. An exception is generated if you do not supply an organization identifier during import and if the organization that the catalog is associated with on the source system does not exist on target system.

Important

- Importing the global shared catalog is not supported.

- You cannot associate a catalog with an organization with the "Provider" business role.

Configuration and Usage

The HP CSA Content Archive Tool is installed during HP CSA product installation, typically in Tools\ContentArchiveTool\ in the HP CSA installation folder.

Configuration Details

Config.properties file

The `config.properties` file must be in the same folder as the `content-archive-tool.jar` file, often under `<HP CSA install folder>\Hewlett-Packard\HP CSA\Tools\ContentArchiveTool`.

The default property filename is `config.properties`, but this file can have any name. Use the –c option to specify the name of the file.

Sample Configuration files

The `content-archive-tool.jar` can produce sample configuration files by executing the following at the command prompt:

```
java –jar content-archive-tool.jar –g
```

The following sample configuration files are created:

- `config.properties.oracle`

- `config.properties.mssql`

- `config.properties.postgresql`

Configuration Properties File Parameters

The sample configuration file not needed by the database in use by HPCSA can be deleted. For example, if you are using a Microsoft SQL Server database, retain the MS SQL configuration file and if desired rename `config.properties` or other name of your choice. The Oracle configuration file can be deleted since it is not needed.

This table lists the parameters found in the configuration file.

Table 3 Parameters in config.properties

`jdbc.driverClassName`	The database driver class. Do not change this value.
`jdbc.dialect`	The database dialect. Do not change this value.
`jdbc.databaseUrl`	When specifying an IPv6 address, it must be enclosed in square brackets (see examples below).
	Examples
	Oracle (SSL not enabled): `jdbc.databaseUrl=jdbc:oracle:thin:@127.0.0.1:1521:XE`
	Oracle (SSL not enabled, using an IPv6 address): `jdbc.databaseUrl= jdbc.databaseUrl=jdbc:oracle:thin:@ [f000:253c::9c10:b4b4]:1521:XE`

Oracle (SSL enabled, HP CSA checks the database DN):
```
jdbc.databaseUrl=jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCPS)(HOST = <host>)(PORT =1521))) (CONNECT_DATA
= (SERVICE_NAME = ORCL)) (SECURITY=(SSL_SERVER_CERT_DN =
"CCN=abc,OU=dbserver,O=xyz,L=Sunnyvale,ST=CA,C=US")))
```
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.

MS SQL (SSL not enabled):
```
jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=requi
re
```

MS SQL (SSL not enabled, using an IPv6 address):
```
jdbc.databaseUrl=jdbc:jtds:sqlserver://[::1]:1433/example;ssl=request
```

MS SQL (SSL enabled):
```
jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=authe
nticate
```

MS SQL (FIPS 140-2 compliant):
```
jdbc.databaseUrl=jdbc:jtds:sqlserver://127.0.0.1:1433/example;ssl=authe
nticate
```

PostgreSQL: `jdbc.databaseUrl=jdbc:postgresql://127.0.0.1:5432/csadb`

`jdbc.username`	User name for database
`jdbc.password`	Encrypted password for database

`csa_war.loc`	The path for `CSA.war` location in HP CSA installation. You must change this path if HP CSA is not installed in the default location.
	Example for Windows:
	  csa_war.loc=C:\\Program Files\\Hewlett-Packard\\CSA\\jboss-as-
	  7.1.1.Final\\standalone\\deployments\\csa.war
	Example for Linux:
	  csa_war.loc=/usr/local/hp/csa/jboss-as-7.1.1.Final/standalone/deployments/csa.war

Note

You must manually change the csa.war file location information in config.properties on Linux installations because it is not generated in the Linux format.

Communicating with the MS SQL or Oracle Database Using SSL

If SSL is enabled between CSA and MS SQL or the Oracle database, the URL property in the database properties file must be configured correctly, and additional command line options might be required when using an Oracle database.

Important The Content Archive Tool does not support DN verification at this time.

	Configuration options	Command line option(s)	jdbc.databaseUrl value
Oracle	CSA does not check the database DN, client authentication is enabled	`-Djavax.net.ssl.keyStore ="<certificate_key_file>"` `-Djavax.net.ssl.keyStorePassword =<certificate_key_file_password>` `-Djavax.net.ssl.keyStoreType =<certifyicate_key_file_type>` 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-`	`jdbc:oracle:thin:@(DESCRIPTION= (ADDRESS_LIST=(ADDRESS=(PROTOCO L = TCPS)(HOST = <host>)(PORT = 1521))) (CONNECT_DATA =(SERVICE_NAME = ORCL)))` where <host> is the name of the system on which the Oracle database server is installed.

			`7.1.1.Final\standalone\configura tion\standalone.xml` file (for example, `%CSA_HOME%\jboss-as- 7.1.1.Final\standalone\configura tion\.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)
	CSA does not check the database DN, client authentication is NOT enabled	<none>	`jdbc:oracle:thin:@(DESCRIPTION= (ADDRESS_LIST=(ADDRESS=(PROTOCO L = TCPS)(HOST = <host>)(PORT = 1521))) (CONNECT_DATA =(SERVICE_NAME = ORCL)))` where <host> is the name of the system on which the Oracle database server is installed.
MS SQL	SSL is enabled	<none>	`jdbc:jtds:sqlserver://127.0.0.1 :1433/example;ssl=authenticate`

Command Line Options

Invoke the Content Archive Tool from the command line as follows:

```
java -jar content-archive-tool.jar <options>
```

Usage:
```
C:\Program Files\Hewlett-Packard\CSA\Tools\ContentArchiveTool>java -jar content-archive-
tool.jar -h

usage:   -h |
         -e [[-a <Artifact Type>] [-n <Artifact Name>] | [-u <uuid of artifact>]] [-c
<config property file>] [-t <target output folder>] [-j <oracle jars>]|
         -i [-v] [-f | -w] -z <individual zip archive| zip archive folder> [-c <config
property file>] [-o <organization name>] [-r <provider name> -s <resource pool name>] [-p
true|false] [-j <oracle jars>] |
         -v -z <individual zip archive| zip archive folder> [-c <config property file>] [-
o <organization name>] [-r <provider name> -s <resource pool name>] [-j <oracle jars>] |
         -l |
         -g
content-archive-tool - This tool enables artifact information to be exported and imported
across 4.x instances of CSA.

-a,--artifact <Artifact type>       Provide Artifact type, otherwise all is exported.
-c,--config <config property file>  The config property file name.
-e,--export                         Export Artifact(s).
-f,--updatePreserveExisting         Create new artifact(s), duplicating any existing
artifact(s).
-g,--generate                       Generate sample input config properties file.
-h,--help                           Print this usage information.
-i,--import                         Import Artifact(s) archives.
-j,--jars <Oracle JARs>             Oracle JDBC Jars separated by space.
-l,--list                           List valid Artifact Types supported.
-n,--name <Artifact name>           Provide artifact display name, otherwise all is
exported.
-o,--orgName <Organization name>    Organization name.
-p,--rp <arg>                       Associate resource offerings to resource providers at
import.
-q,--forceCloudOSProvider <Force Cloud OS provider>   Force CloudOS provider name.
-r,--forceHPHelionOpenStackProvider <Force HP Helion OpenStack provider>   Force HP Helion
OpenStack provider name.
-s, -forceResourcePool <Force Resource Pool>        Force Resource Pool name.
-t,--target <Target output folder>  Target output folder.
-u,--uuid <Artifact UUID>           UUID of the artifact to be exported.
-v,--validate                       Validate before the archive is imported.
```

```
-w,--update                           Overwrite/update existing Artifact(s) without
duplication.
-z,--zip <Archive location>        Archive location.
```

The command options and sub-options are shown in the following table.

Table 4 Command Line Options

Option	Option description	Sub-options associated with the option	Sub-option description
`-h, --help`	Display syntax and usage information.	none	
`-e, --export`	Export artifacts.	`-a, --artifact <artifact_type>`	Artifact type to be exported, otherwise all are exported.
		`-n, --name <artifact_name>`	Display name of artifact to be exported.
		`-u, --uuid <artifact_id>`	UUID of artifact to be exported.
		`-c, --config <config_filename>`	Configuration property filename.
		`-t, --target <target_folder>`	Target folder for archive.
		`-j, --jars`	Load Oracle JDBC JAR files. Note jar filenames are separated by spaces.
`-i, --import`	Import artifact archives.	`-v, --validate`	Validate before the archive is imported.
		`-f, --updatePreserveExisting`	Preserve (rename) existing artifacts by the same name; then import all artifacts from archive.
		`-w, --update`	Overwrite/update existing artifact(s) without duplication.
		`-z, --zip <archive_folder>|<archive_filename>`	Required; archive folder or filename. If folder, imports all CSA archives in folder. If filename, defaults to current folder unless folder path is specified.
		`-c, --config <config_filename>`	Configuration property filename.
		`-o, --orgName <organization_id>`	Organization identifier used for catalog import.
		`-q, --forceCloudOSProvider <provider_name>`	Optional with topology design import; provider to use, must already be configured on target CloudOS system. `-s` (`--forceResourcePool`) must also be specified when using this option.
		`-r, --forceHPHelionOpenStackProvider`	Optional with topology design import; provider to use, must already be configured on target

15

		`<provider_name>`	HP Helion OpenStack system. `-s` (`--forceResourcePool`) must also be specified when using this option.
		`-s, --forceResourcePool <resource_pool_name>`	Optional with topology design import; resource pool to use, must already be defined on target CloudOS system. `-r` (`--forceCloudOSProvider`) must also be specified when using this option.
		`-p, --rp [true|false]`	Associate imported resource offerings and resource environments to resource providers on the target system with same display name and provider type.
		`-j, --jars <Oracle_JAR1 Oracle_JAR2 …>`	Load Oracle JDBC JAR files. Note jar filenames are separated by spaces.
`-v, --validate`	Validate before the archive is imported.	`-c, --config <config_filename>`	Config property filename.
		`-j, --jars <Oracle_JAR1 Oracle_JAR2 …>`	Load Oracle JDBC JAR files. Note jar filenames are separated by spaces.
		`-o, --orgName <organization_id>`	Organization identifier for export of catalog(s).
		`-z, --zip <archive_folder>|<archive_filename>`	Required; archive folder or filename. If folder, validates all CSA archives in folder. If filename, defaults to current folder unless folder path is specified.
`-l, --list`	List the supported valid artifact types.	None	
`-g, --generate`	Generate sample input config properties file.	None	

Examples

- Example 1: Display the Content Archive Tool usage/help.

```
java -jar content-archive-tool.jar -h
```

- Example 2: Generate a sample config.properties file to be used by this tool.

```
java -jar content-archive-tool.jar -g
```

- Example 3: Display a list of artifact types supported for export.

```
java -jar content-archive-tool.jar -list
```

- Example 4: Export all supported artifacts to the current folder. An archive will be created for each primary artifact.

```
java -jar content-archive-tool.jar –e
```

- Example 5: Export all service offerings to the current folder.

```
java -jar content-archive-tool.jar –export –a SERVICE_OFFERING
```

`SERVICE_DESIGN`, `RESOURCE_OFFERING`, `CATALOG`, `RESOURCE_ENVIRONMENT`, `SERVICE_OFFERING`, `COMPONENT_PALETTE` are valid values for the `–a` option.

- Example 6: Export a resource offering with the display name vCenter Resource Offering to the current folder.

```
java -jar content-archive-tool.jar –e –a RESOURCE_OFFERING –n "vCenter Resource
Offering"
```

- Example 7: Export an artifact with id `90cec2ff3b7d9a03013b7db4c6ff003f` to the current folder.

```
java –jar content-archive-tool.jar –export –uuid
90cec2ff3b7d9a03013b7db4c6ff003f
```

Note

In examples 4 to 7, artifacts could be exported to a target folder instead of the current folder by providing the `-t` option with the value of the target folder. Following is an example that exports all catalogs to folder `C:\temp\catalogs`.

```
java –jar content-archive-tool.jar –export –archive CATALOG –target
"C:\temp\catalogs"
```

- Example 8: Import all artifacts present in the specified archive.
  `SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip` from the current folder.

```
java -jar content-archive-tool.jar -i -z
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
```

- Example 9: Import all archives stored in the folder `C:\temp\archive`.

```
java -jar content-archive-tool.jar -i -z "C:\temp\archive"
```

- Example 10: Import with validation all artifacts present in an archive
  `SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip` from the current folder.

```
java -jar content-archive-tool.jar -import –v –zip
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
```

This operation validates whether or not the artifacts in the archive are already present in the system and provides a summary of the validation. You are then prompted to continue with the import operation. When `Y` (yes) is selected, the import operation continues by default. When `N` (no) is selected, the current archive is not imported. When `Q` (quit) is selected, all remaining import activity is terminated.

- Example 11: Import all artifacts present in archive
  `SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip` from the current folder regardless of whether artifacts by the same name already exist on the target system.

```
java -jar content-archive-tool.jar -i –f -z
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
```

By default the updatePreserveExisting, -f, operation validates whether or not the artifacts in the archive are already present in the system and provides a summary of validation. You are then prompted to continue with the import operation. When `Y` (yes) is selected, existing artifacts with the same name as artifacts in the archive are renamed, and all artifacts in the archive are then created on the

target system. When `N` (no) is selected, the current archive is not imported. When `Q` (quit) is selected, all remaining import activity is terminated.

- Example 12: Import a catalog and all the artifacts in the archive named `CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip`, associating the catalog with an organization named `CONSUMER_ORG`.

```
java -jar content-archive-tool.jar -i -z
CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip –o CONSUMER_ORG
```

Note

Before performing this operation on the catalog archive, the organization `CONSUMER_ORG` should be created on the target system. If this organization is not present, the Content Archive Tool will associate the catalog to the organization it was associated with on the source system. If neither organization exists on the target system, the Content Archive Tool terminates with an exception.

- Example 13: Import all artifacts and automatically associate imported resource offerings to resource providers with same display name and provider type on the target system.

```
java -jar content-archive-tool.jar -i –p true -z
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
```

- Example 14: Import all artifacts preserving original artifacts on target system. Automatically associate the resource offerings imported with existing resource providers using the same display name and provider type on the target system.

```
java -jar content-archive-tool.jar -i –f -z
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
–p true
```

- Example 15: Import all artifacts preserving original artifacts from catalog archive: `CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip`

```
java -jar content-archive-tool.jar –i –f –z CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip
```

- Example 16: Import all artifacts preserving original artifacts from catalog archive and associate resource provider to resource offering and resource environment.

```
java -jar content-archive-tool.jar –i –f –z CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip –
-rp true
```

- Example 17: Import all artifacts present in the specified archive from the current folder. Because the archive includes topology design(s), a configured CloudOS provider and CloudOS system resource pool on the target system can be specified using the –r and –s sub-options respectively to override the provider and resource pool values found in the archive.

```
java –jar content-archive-tool.jar –i –v –z
SERVICE_DESIGN_topo1_90b72c4e425ae0db01425b0c1ae40020.zip -c
c:\work\temp\exportimport\cfg.properties.oracle -r CloudOS181 -s
localResourcePool
```

- Example 18: Validate the service offering archive: `SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip`

```
java –jar content-archive-tool.jar –v –z
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
```

- Example 19: Validate catalog archive `CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip`, where the catalog in the archive is to be associated with an organization named `CONSUMER_ORG`.

```
java –jar content-archive-tool.jar –v –z
CATALOG_Consumer_Catalog_90cec2ff3c0763bf013c076fbf7000c0.zip –o CONSUMER_ORG
```

- **Example 20:** Update existing artifacts from service offering archive
  `SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip`.

```
java -jar content-archive-tool.jar -i -w  -z
SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip
```

- **Example 21:** Update existing artifacts from service offering archive
  `SERVICE_OFFERING_Sample_Service_Offering_90cec2ff3c0763bf013c076cf4f10083.zip`.

```
java -jar content-archive-tool.jar -i -j ojdbc6.jar orai18n-11.2.0.3.jar -v -z
RESOURCE_OFFERING_offerings_1_90cef59c3e682fca013e68302a240063.zip
```

Note

In examples 4 through 21, it is assumed that the information about the database and `csa.war` location is stored in the `config.properties` file in the same folder as the `content-archive-tool.jar` file. Use the –c option to specify the complete path of the configuration file.

Known Issues

1. **Problem**: Cannot import or export artifacts that include the double quote (") special character in the name. [158310]
   Cause
   The Content Archive Tool does not properly handle artifact names specified in the command line when that name includes the double quote (") character.

   Workaround

   For artifacts that include the double quote (") character in the display name, this character needs to be escaped with a backslash character. For example, to export service design Simple"Compute"Linux which contains double quotes in its display name, enter it as shown in the following:

   ```
   java -jar content-archive-tool.jar -e -a SERVICE_DESIGN -n Simple\"Compute\"Linux
   ```

2. **Problem**: Using special characters in the artifact name results in unexpected behavior of the tool during export. [158317]
   Cause
   Using characters that have a special purpose in the Windows command line is the cause of this issue.

   Workaround

   Special characters that have a specific purpose in the Windows command line need to be escaped with a caret character (^). This includes characters such as: < ! ^ ( ) = ; , >|

   For example:

       `<` should be escaped as `^<`

       `!` should be escaped as `^!`

       `^` should be escaped as `^^`

       `(` should be escaped as `^(`

3. **Problem**: Unexpected behavior during import and export when providing a folder name with trailing slashes as a value to `-z/--zip` or `-t/--target` sub-options in the command line. [158670]
   Cause
   The Content Archive Tool does not properly handle trailing slashes in folder names when provided as values to `-z/--zip or -t/--target` sub-options.

   Workaround

   Omitting trailing slashes from folder names provided as values to `-z/--zip` or `-t/--target` avoids this problem. For example, during export instead of providing:

   ```
   java -jar content-archive-tool.jar -e -t "C:\PCA Tool\output folder\"
   ```

   provide:

   ```
   java -jar content-archive-tool.jar -e -t "C:\PCA Tool\output folder"
   ```

   Similarly during import instead of:

```
java -jar content-archive-tool.jar -i -z "C:\PCA Tool\input folder\"
```

provide:

```
java -jar content-archive-tool.jar -i -z "C:\PCA Tool\input folder"
```

For more information

HP software product manuals and documentation for HP CSA can be found at http://h20230.www2.hp.com/selfsolve/manuals. You will need an HP Passport to sign in and gain access. **To register for an HP Passport ID, go to:** http://h20229.www2.hp.com/passport-registration.html

To help us improve our documents, please send feedback to csadocs@hp.com.

Table 5: Document revision history

Date or version	Changes
December 2012	HP CSA 3.10 release
June 2013	Updated for HP CSA 3.20
January 2014	Updated for HP CSA 4.00
July 2014	Updated for HP CSA 4.10