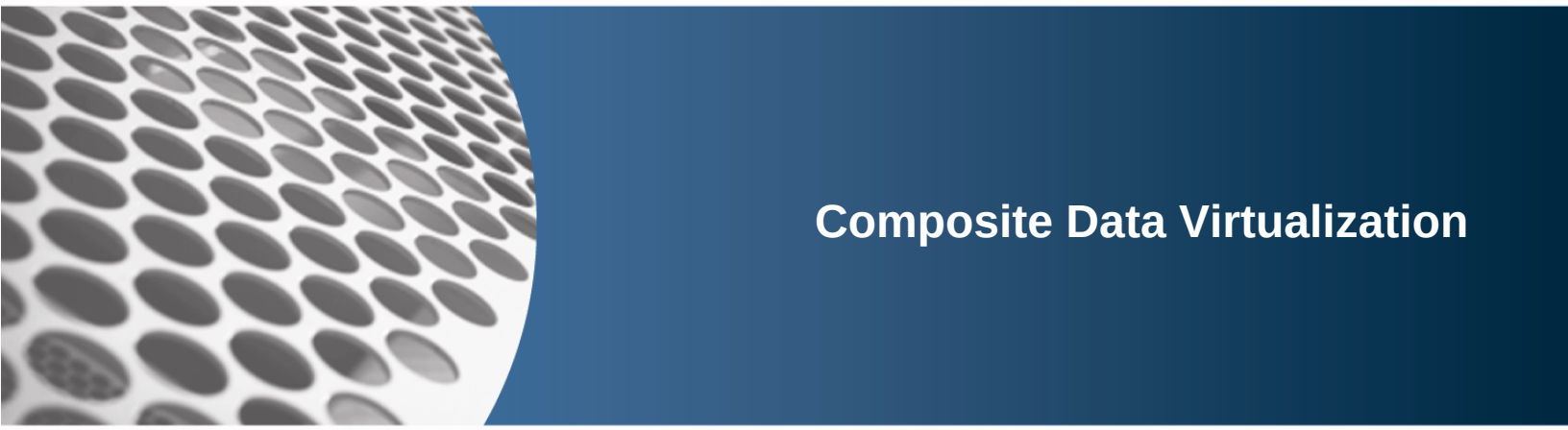
WHITE PAPER

COMPOSITE SOFTWARE



Composite PS Promotion and Deployment Tool

Archive Module User Guide

Composite Professional Services

February 2014

TABLE OF CONTENTS

DOCUMENT CONTROL

Version History

Version	Date	Author	Description
1.0	6/9/2011	Mike Tinius	Initial revision for Archive User Guide
1.0.1	8/1/2011	Mike Tinius	Revision due to Architecture changes.
2.0	7/13/2012	Alex Dedov	Document enhancements to support full set of command-line options
2.1	8/24/2012	Mike Tinius	Updated the archive attribute options.
3.0	8/21/2013	Mike Tinius	Updated docs to Cisco format
3.1	1/31/2014	Mike Tinius	Updated documentation for Includesourceinfo attribute and the use of encrypt flag and https. See PDTool Installation Guide for https configuration.
3.2	2/18/2014	Mike Tinius	Added section on debugging. Fixed some of the options that were upper case and made them lower case. Prepare docs for open source.

Related Documents

Document	File Name	Author
Composite PS Promotion and Deployment Tool User's Guide v1.0	*Composite PS Promotion and Deployment Tool User's Guide v1.0.pdf*	Mike Tinius
CIS Administration Guide	*AdministrationGuide.pdf*	Composite Software

Composite Products Referenced

Composite Product Name	Version
Composite Information Server	5.2, 6.0, 6.1, 6.2

INTRODUCTION

Purpose

The purpose of the Archive User Guide is to demonstrate how to effectively use the Archive Module for importing, exporting, backing up and restoring CIS resources. The scripts in this module will interact directly with a CIS server. There is no version control associated with this module. For version control, review the VCS Module documentation. This will module will focus on exporting and importing resources via Composite Archive files (.car). It will also show how to use backup and restore features which also use the Composite Archive files. The Archive Module is completely shell/batch script based.

Audience

This document is intended to provide guidance for the following users:

- Architects
- Developers
- Administrators.
- Operations personnel.

ARCHIVE MODULE DEFINITION

Method Definitions and Signatures

1. **pkg_import**

   Import a CAR file into a local or remote CIS instance with options.

```
@param serverId - target server id from servers config xml
@param archiveIds - list of comma separate archive Ids containing the
archive instructions
@param pathToArchiveXML - path to the archive module xml
@param pathToServersXML - path to the server values xml
@throws CompositeException


public void pkg_import(String serverId, String archiveIds, String
pathToArchiveXML, String pathToServersXML) throws CompositeException
```

2. **pkg_export**

   Export a CAR file from a local or remote CIS instance by designating a list of resources to export.

```
@param serverId - target server id from servers config xml
@param archiveIds - list of comma separate archive Ids containing the
archive instructions
@param pathToArchiveXML - path to the archive module xml
@param pathToServersXML - path to the server values xml
@throws CompositeException


public void pkg_export(String serverId, String archiveIds, String
pathToArchiveXML, String pathToServersXML) throws CompositeException]
```

3. **backup_export**

   Perform a complete backup of a local or remote CIS server instance.

```
Restore a full server backup from a backup car file
@param serverId - target server id from servers config xml
@param archiveIds - list of comma separated archive Ids containing the
archive instructions
@param pathToArchiveXML - path to the archive module xml
@param pathToServersXML - path to the server values xml
@throws CompositeException
```

```
public void backup_export(String serverId, String archiveIds, String
pathToArchiveXML, String pathToServersXML) throws CompositeException
```

4. **backup_import**

   Perform a complete restore of a local or remote CIS server instance.

```
Create a full server backup of a CIS instance to a car file

@param serverId - target server id from servers config xml

@param archiveIds - list of comma separated archive Ids containing the
archive instructions

@param pathToArchiveXML - path to the archive module xml

@param pathToServersXML - path to the server values xml

@throws CompositeException


public void backup_import(String serverId, String archiveIds, String
pathToArchiveXML, String pathToServersXML) throws CompositeException
```

General Notes:

The arguments pathToArchiveXML and pathToServersXML will be located in PDTool/resources/modules. The value passed into the methods will be the fully qualified path. The paths get resolved when executing the property file and evaluating the $MODULE_HOME variable.

ARCHIVE MODULE XML CONFIGURATION

A full description of the PDToolModule XML Schema can be found by reviewing /docs/PDToolModules.xsd.html.

Description of the Module XML

The Archive Module performs various CIS archiving operations with CAR (Composite Archive) files such as backup, restore, import and export.

```
<?xml version="1.0"?>
<p1:ArchiveModule xmlns:p1="http://www.cisco.dvbu.com/ps/deploytool/modules">
    <!--Element archive is optional, maxOccurs=unbounded-->
    <archive>
        <id>example_id</id>
        <archiveMethod>CAR</archiveMethod>
        <archiveFileName>$PROJECT_HOME/test01.car</archiveFileName>
        <!--Element includeDependencies is optional-->
        <includeDependencies>false</includeDependencies>
        <!--Element resources is optional-->
        <resources>
            <!--'Choice' block, maxOccurs=unbounded-->
            <export>/shared/examples/CompositeView</export>
            <relocate>
                <oldPath>/shared/examples/CompositeView</oldPath>
                <newPath>/shared/examples/newdir/CompositeView</newPath>
            </relocate>
            <rebind>
                <oldPath>/shared/examples/ViewSales</oldPath>
                <newPath>/shared/examples/newds/ViewSales</newPath>
            </rebind>
        </resources>
        <!--Element encrypt is optional-->
        <encrypt>false</encrypt>
        <!--Element description is optional-->
        <description>Export CompositeView</description>
        <!--Element includeaccess is optional-->
        <includeaccess>true</includeaccess>
        <!--Element includeallusers is optional-->
        <includeallusers>false</includeallusers>
        <!--Element includerequiredusers is optional-->
        <includerequiredusers>false</includerequiredusers>
        <!--Element includecaching is optional-->
        <includecaching>true</includecaching>
        <!--Element includejars is optional-->
        <includejars>true</includejars>
        <!--Element includesourceinfo is optional-->
        <includesourceinfo>false</includesourceinfo>
        <!--Element includestatistics is optional-->
        <includestatistics>false</includestatistics>
        <!--Element messagesonly is optional-->
        <messagesonly>false</messagesonly>
        <!--Element overridelocks is optional-->
        <overridelocks>false</overridelocks>
        <!--Element overwrite is optional-->
        <overwrite>false</overwrite>
        <!--Element pkgName is optional-->
        <pkgName>Composite Server Archive File via PD Tool</pkgName>
        <!--Element printinfo is optional-->
        <printinfo>false</printinfo>
```

```
        <!--Element printroots is optional-->
        <printroots>false</printroots>
        <!--Element printusers is optional-->
        <printusers>false</printusers>
        <!--Element printcontents is optional-->
        <printcontents>false</printcontents>
        <!--Element printreferences is optional-->
        <printreferences>false</printreferences>
        <!--Element setAttributes is optional-->
        <setAttributes>
            <!--Element resourceAttribute, maxOccurs=unbounded-->
            <resourceAttribute>
                <resourcePath>/shared/examples/ds_inventory</resourcePath>
                <resourceType>DATA_SOURCE</resourceType>
                <attribute>annotation</attribute>
                <value>sample data source</value>
            </resourceAttribute>
        </setAttributes>
        <!--Element users is optional-->
        <users>
            <!--'Choice' block is optional, maxOccurs=unbounded-->
            <!--Element export, maxOccurs=unbounded-->
            <export>
                <domain>composite</domain>
                <!--Element group is optional-->
                <group>group1</group>
            </export>
            <export>
                <domain>composite</domain>
                <!--Element user is optional-->
                <user>user1</user>
            </export>
            <import>merge</import>
         </users>
   </archive>
</p1:ArchiveModule>
```

Attributes of Interest

id – a uniquely named value identifying an archive operation.

archiveMethod – export archive file (CAR).

archiveFileName – full path to the archive file name and location. The value may contain variables defined via deploy.dp, JAVA Environment or System Environment. A variable is defined with the format of VAR=myvalue and referenced as $VAR. For example the entry archiveFileName may look like this:

<archiveFileName>$PROJECT_HOME/resources/carfiles/test00.car</archiveFileName>

includeDependencies – (true/false) Include resource dependencies on export.

resources

export – unbound list of CIS resources to export

relocate

oldPath – the old path to the resource

newPath – the new path to the resource

rebind

oldPath – the old path to the resource to be rebound

newPath – the new path the resource to be rebound

encrypt – (true/false) Encrypts communication between the command line and CIS using SSL sent over the dedicated HTTPS port. The PDTool 6.2 Installation Guide provides setup information for both PDTool 6.2 and CIS. Encryption is not supported in PDTool 6.1. For additional information, see Composite Administration guide chapter 11 for complete details.

description – Package description of the archive file. Notation appears when Composite Studio is used to import the CAR.

includeaccess – (true/false)

- ON IMPORT: The -includeaccess option must be used if you want to preserve ownership and privilege information. This option is ignored if you are not logged in as a member of the admin group.
- ON EXPORT: Includes the current user access controls (privilege specifications) on the resources in the export file. Default setting does NOT include access control even when exported by an administrator. An error can occur if this option is used and the exporting user is not a member of the admin group with the Read All Resources right and include access on export.

includerequiredusers – (true/false)

- ON PACKAGE EXPORT: Exports all domains, groups, and users to the export file. Requires the Read All Users right.

includecaching – (true/false)

- ON IMPORT: Caching configuration is imported by default. The -nocaching option must be used if you want to ignore cache configurations.
- ON EXPORT: Includes the details of caching on views and procedures in the export file. This option must be specified to include cached data from materialized views, or configurations that include scheduling for cache refreshes.

includejars – (true/false)

- ON PACKAGE EXPORT: Exports any included custom Java procedure data source's JAR.

- ON BACKUP EXPORT: Can suppress export of custom Java procedure data source JARs.
- ON PACKAGE IMPORT: Can suppress import of any custom Java procedure data source's JAR within the CAR file.

includesourceinfo – (true/false)

- ON PACKAGE EXPORT: Data source connection details such as user name, password, host name, and port are not included by default. When the Includesourceinfo flag is set to true, (equivalent to the -includesourceinfo flag for pkg_export) then include the data source connection details. If passwords are included, they are encrypted. If the Includesourceinfo flag is set to false, (equivalent to the -nosourceinfo flag for pkg_export), then the data source connection information is not included. The minimum requirement is for Hotfix 6.2.5.00.24 or higher for pkg_import -nosourceinfo to work properly.
- ON PACKAGE IMPORT: An overwrite safeguard. Suppresses import of the following pre-existing connection attributes when an otherwise identical resource is already present at the target: Driver, ConnectionURL, Port, Database Name, Login, Password, and Pass-through Login.

  Supports re-import without need for explicit "-set" options and without altering original data source attributes.

includestatistics – (true/false)

- ON PACKAGE EXPORT and BACKUP EXPORT: Includes any known resource cardinality statistics about data source table boundaries, column boundaries, and configurations when statistics gathering is enabled.

messagesonly – (true/false)

- ON PACKAGE IMPORT ONLY: Prints the messages generated in a package import to a log4j file without actually performing the import.

overridelocks – (true/false)

- ON PACKAGE IMPORT ONLY: Overrides Studio locks when importing resources from a CAR file.

overwrite – (true/false)

- ON BACKUP IMPORT: This option does a destructive import clearing all resources prior to importing the backup CAR. Using the backup_import utility with the -overwrite option ensures that CIS matches the contents of the imported CAR file.

- ON PACKAGE IMPORT: Use of the -overwrite option ensures that CIS will exactly match the directories present in the CAR file. The -overwrite option clears targeted folder directories before copying the CAR file contents to those cleared directories. The pkg_import utility clears only those folders that have representative resources in the CAR file from those directories. CIS directories that do not have representative resources in the CAR file remain untouched even if the -overwrite option is specified.

pkgName – Names an attribute in the contents.xml within the exported backup file.

printinfo – (true/false) The "printinfo" option causes the archive file to be examined and for information about it to be printed to the display. The archive file is NOT imported when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO.

printroots – (true/false) Prints the new paths to the imported resources. The archive file is NOT imported when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO.

printusers – (true/false) Prints the user names of the owners of the imported resources and their associated user groups. The archive file is NOT imported when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO.

printcontents – (true/false) This option disables actual import, and prints properties of the CAR file to the command window. The archive file is NOT imported when this option is is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO.

printreferences – (true/false) Prints a list of resources referred to by the imported resources. The archive file is NOT imported when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO.

setAttributes – Enables resource attribute changes during import. The set option can be repeated to set different attributes. Multiple class paths can be set with a single statement.

```
<setAttributes>
    <!--Element resourceAttribute, maxOccurs=unbounded-->
    <resourceAttribute>
        <resourcePath>/shared/examples/ds_inventory</resourcePath>
        <resourceType>DATA_SOURCE</resourceType>
        <attribute>annotation</attribute>
        <value>sample data source</value>
    </resourceAttribute>
</setAttributes>
```

- **resourcePath** – The Composite resource location and name.

- The "**resourceType"** is equal to “DATA_SOURCE” when attribute is classpath, host, port, database, user, or password.
- The "**attribute**" can be one of the following in the restriction list:
    - *Restriction List*: user, password, user2, password2, host, port, database, path, annotation
    - **user** "login" or "username" or error depending on source type
    - **password** "password" or error depending on source type
    - **user2** "appUserName" or error if not Oracle EBS
    - **password2** "appPassword" or error if not Oracle EBS
    - **host** "urlIP" or "dsn" or "server" or "appServer" or "url" or "root" or error depending on the source type
    - **port** "urlPort" or "port" or error depending on source type
    - **database** "urlDatabaseName" or "enterprise" or "appServer" or error depending on the source type
    - **path** "root" or "url" or error depending on source type
    - **annotation** “description of view”
- Set "**value**" to a valid entry for the selected attribute. String values can be enclosed with double quotes to allow for spaces.
    - For Windows systems, use the semicolon delimiter: C:\DevZone\ATeam\Jars\my.jar;D:\Current\Ref\classes.
    - For UNIX systems, use colons as the delimiter: /lib/ext/classes:/lib/src/jars

users – Provides the ability to set users for export and import.

(See Composite Admin Guide for details)

```
<users>
    <!--'Choice' block is optional, maxOccurs=unbounded-->
    <!--Element export, maxOccurs=unbounded-->
    <export>
        <domain>composite</domain>
        <!--Element group is optional-->
        <group>group1</group>
    </export>
    <export>
        <domain>composite</domain>
        <!--Element user is optional-->
        <user>user1</user>
```

```
            </export>
            <import>merge</import>
         </users>
```

- Archive Security Type for package **export** command: Used to include specified user, group and domain info in the export file. This option can be repeated to export multiple entities.

  export

  domain – domain such as composite

  group – group within composite

  user – user within composite

- Archive Security Type for package **import** command:
  - "overwrite" option imports all users present in the exported package car. By default domain, groups, and user information are not included in export or import packages.
  - "merge" imports all users present in the CAR file who are not already present in the server target. This option takes precedence over "overwrite" option.

  import – one of overwrite or merge.

Complete List of Attributes

The below table summarizes the attributes supported within ***ArchiveModule.xml*** file. A brief explanation of each attribute is also provided. Please refer to the CIS Admin Guide, Chapter 11 "CIS Command-Line Utilities" for their usage details and complete description.

(Common node prefix */ArchiveModule/archive/* is omitted from the XPath definitions in the table in order to save space, i.e. XPath "`archiveFileName`" should be actually read as *"/ArchiveModule/archive/archiveFileName"*.)

		Applicable Archive Module operation			
Attribute XPath	**Description**	**BACKUP**	**RESTORE**	**EXPORT**	**IMPORT**
id	ID is a unique name identifier within the list of archive actions (backup, import, export, etc.).	X	X	X	X
archiveFileName	Name of a CAR file to be used	X	X	X	X
archiveMethod	Archive Method must be 'CAR' (future support for VCS is planned)	X	X	X	X
description	Package description of the archive file. Notation appears when Composite Studio is used to import the CAR.	X		X	
pkgName	Names an attribute in the contents.xml within the exported backup file.	X		X	
resources/export	Defines the resources to be exported.			X	
resources/relocate/oldPath resources/relocate/newPath	This option can be used to change the location for the import.		X		X
resources/rebind/oldPath resources/rebind/newPath	Sets a new resource path for a dependency resource.				X
encrypt	Encrypts communication between the command line and CIS using SSL sent over the dedicated HTTPS port. The PDTool 6.2 Installation Guide provides setup information for both PDTool 6.2 and CIS. Encryption is not supported in PDTool 6.1. For additional information, see Composite Administration guide chapter 11 for complete details.	X	X	X	X
setAttributes/resourceAttribute/resourcePath setAttributes/resourceAttribute/resourceType setAttributes/resourceAttribute/attribute setAttributes/resourceAttribute/value	Enables resource attribute changes during import. The set option can be repeated to set different attributes. Multiple class paths can be set with a single statement. • The Composite resource name is the <resourcePath>. • The <resourceType> is equal to “DATA_SOURCE” when attribute is classpath, host, port, database, user, or password. • The <attribute> can be: classpath, host, port, database, user, or password, with associated value dependent on source type: o user <login> or <username> or error o password <password> or error o user2 <appUserName> or error if not Oracle EBS o password2 <appPassword> or error if not Oracle EBS o host <urlIP> or <dsn> or <server> or <appServer> or <url> or <root> or error o port <urlPort> or <port> or error o database <urlDatabaseName> or <enterprise> or <appServer> or error o path <root> or <url> or error		X		X

	o annotation				
printcontents	This option disables actual import, and prints properties of the CAR file to the command window. The archive file is **NOT imported** when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO				X
printinfo	Causes the archive file to be examined and for information about it to be printed to the log file. The archive file is **NOT imported** when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO		X		X
printroots	Prints the new paths to the imported resources. The archive file is **NOT imported** when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO				X
printusers	Prints the user names of the owners of the imported resources and their associated user groups. The archive file is **NOT imported** when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO				X
printreferences	Prints a list of resources referred to by the imported resources. The archive file is **NOT imported** when this option is given. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO				X
messagesonly	Prints the messages generated in a package import to a log4j file without actually performing the import. This requires the following log4j.property setting: log4j.logger.com.compositesw.cmdline.archive=INFO				X

overwrite	ON RESTORE: This option does a destructive import clearing all resources prior to importing the backup CAR. Using it ensures that CIS matches the contents of the imported CAR file. ON IMPORT: Ensures that CIS will exactly match the directories present in the CAR file. This option clears targeted folder directories before copying the CAR file contents to those cleared directories. Only those folders that have representative resources in the CAR file from those directories. CIS directories that do not have representative resources in the CAR file remain untouched even if the -overwrite option is specified.		X		X
overrideLocks	Overrides Studio locks when importing resources from a CAR file.				X
includeaccess	ON IMPORT: This option must be used if you want to preserve ownership and privilege information. This option is ignored if does not belong to admin group. ON EXPORT: Includes the current user access controls (privilege specifications) on the resources in the export file. Default setting does NOT include access control even when exported by an administrator.			X	X
includecaching	ON IMPORT: Caching configuration is imported by default. The includecaching must be explicitly set to 'false' if you want to ignore cache configurations. ON EXPORT: Includes the details of caching on views and procedures in the export file. This option must be specified to include cached data from materialized views, or configurations that include scheduling for cache refreshes.			X	X
includeDependencies	On export, gathers and includes all dependent resources for the resources specified in the <resources> section.			X	X
includesourceinfo	ON EXPORT: Data source connection details such as user name, password, host name, and port are not included by default. Specify -includesourceinfo to include these. If passwords are included, they are encrypted. ON IMPORT: Used as an overwrite safeguard. Suppresses import of the following pre-existing connection attributes when an otherwise identical resource is already present at the target: o Driver, o ConnectionURL, o Port, o Database Name, o Login, Password, and o Pass-through Login. Supports re-import without need for explicit "-set" options and without altering original data source attributes.			X	X

includejars	ON EXPORT: Exports any included custom Java procedure data source's JAR. ON BACKUP: Can be used to suppress export of custom Java procedure data source JARs. ON IMPORT: Can suppress import of any custom Java procedure data source's JAR within the CAR file.	X		X	X
includeallusers	Exports all domains, groups, and users to the export file. Requires the Read All Users right.			X	
users/export/domain	Used to include specified user, group and domain info in the export file.			X	
users/export/group	Used to include specified user, group and domain info in the export file.			X	
users/export/user	Used to include specified user, group and domain info in the export file.			X	
users/import	Allows to update target server with user information from the exported package CAR. o "overwrite" option imports all users that are present in the car-file o "merge" imports all users present in the CAR file who are not already present in the server target. This option takes precedence over "overwrite" option. By default domain, groups, and user information are not included in export or import packages. See Composite Admin Guide for complete details of how these parameters work in combination with Resource overwrite option.				X
includerequiredusers	Includes the information about the required users in the export file.			X	
includestatistics	DURING EXPORT and BACKUP: Includes any known resource cardinality statistics about data source table boundaries, column boundaries, and configurations when statistics gathering is enabled.	X		X	

Attribute Value Restrictions

```
<xs:element name="import" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="overwrite"/>
          <xs:enumeration value="merge"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="attribute">
    <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="user"/>
          <xs:enumeration value="password"/>
          <xs:enumeration value="user2"/>
```

```
            <xs:enumeration value="password2"/>
            <xs:enumeration value="host"/>
            <xs:enumeration value="port"/>
            <xs:enumeration value="database"/>
            <xs:enumeration value="path"/>
            <xs:enumeration value="annotation"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

HOW TO EXECUTE

The following section describes how to setup a property file for both command line and Ant and execute the script. This script will use the ArchiveModule.xml that was described in the previous section.

Script Execution

The full details on property file setup and script execution can be found in the document "*Composite PS Promotion and Deployment Tool User's Guide v1.0.pdf*". The abridged version is as follows:

Windows: ExecutePDTool.bat -exec ../resources/plans/UnitTest-Archive.dp

Unix: ./ExecutePDTool.sh -exec ../resources/plans/UnitTest-Archive.dp

Properties File (UnitTest-Archive.dp):

Property File Rules:

```
# ----------------------------
# UnitTest-Archive.dp
# ----------------------------
#   1. All parameters are space separated.  Commas are not used.
#         a. Any number of spaces may occur before or after any parameter and are
trimmed.
#
#   2. Parameters should always be enclosed in double quotes according to these
rules:
#         a. when the parameter value contains a comma separated list:
#                                     ANSWER: "ds1,ds2,ds3"
#
#         b. when the parameter value contain spaces or contains a dynamic variable
that will resolve to spaces
#            i.   There is no distinguishing between Windows and Unix variables.
Both UNIX style variables ($VAR) and
#                 and Windows style variables (%VAR%) are valid and will be parsed
accordingly.
#            ii.  All parameters that need to be grouped together that contain
spaces are enclosed in double quotes.
#            iii. All paths that contain or will resolve to a space must be enclosed
in double quotes.
#                 An environment variable (e.g. $MODULE_HOME) gets resolved on
invocation PDTool.
#                       Paths containing spaces must be enclosed in double quotes:
#                              ANSWER: "$MODULE_HOME/LabVCSModule.xml"
#                       Given that MODULE_HOME=C:/dev/Cis Deploy
Tool/resources/modules, PDTool automatically resolves the variable to
#                       "C:/dev/Cis Deploy Tool/resources/modules/LabVCSModule.xml".
#
#         c. when the parameter value is complex and the inner value contains spaces
```

```
#                    i. In this example $PROJECT_HOME will resolve to a path that
contains spaces such as C:/dev/Cis Deploy Tool
#                       For example take the parameter -pkgfile
$PROJECT_HOME$/bin/carfiles/testout.car.
#                       Since the entire command contains a space it must be
enclosed in double quotes:
#                                ANSWER: "-pkgfile
$PROJECT_HOME/bin/carfiles/testout.car"
#
#   3. A comment is designated by a # sign preceding any other text.
#         a. Comments may occur on any line and will not be processed.
#
#   4. Blank lines are not processed
#         a. Blank lines are counted as lines for display purposes
#         b. If the last line of the file is blank, it is not counted for display
purposes.
#
```

Property File Parameters:

```
# ----------------------------
# Parameter Specification:
# ----------------------------
# Param1=[PASS or FAIL]  :: Expected Regression Behavior.  Informs the script
whether you expect the action to pass or fail.  Can be used for regression testing.
# Param2=[TRUE or FALSE] :: Exit Orchestration script on error
# Param3=Module Batch/Shell Script name to execute (no extension).  Extension is
added by script.
# Param4=Module Action to execute
# Param5-ParamN=Specific space separated parameters for the action.  See Property
Rules below.
```

Property File Example:

```
# ----------------------------------------
# Begin task definition list for UNIX:
# ----------------------------------------
# Backup
PASS   TRUE   ExecuteAction       backup_export   $SERVERID backup
$MODULE_HOME/ArchiveModule.xml $MODULE_HOME/servers.xml
# Restore
PASS   TRUE   ExecuteAction       backup_import   $SERVERID restore
$MODULE_HOME/ArchiveModule.xml $MODULE_HOME/servers.xml
# Export
PASS   TRUE   ExecuteAction       pkg_export      $SERVERID export
$MODULE_HOME/ArchiveModule.xml $MODULE_HOME/servers.xml
# Import
PASS   TRUE   ExecuteAction       pkg_import      $SERVERID import
$MODULE_HOME/ArchiveModule.xml $MODULE_HOME/servers.xml
```

Ant Execution

The full details on build file setup and ant execution can be found in the document "*Composite PS Promotion and Deployment Tool User's Guide v1.0.pdf*". The abridged version is as follows:

Windows: ExecutePDTool.bat -ant ../resources/ant/build-Archive.xml

Unix: ./ExecutePDTool.sh -ant ../resources/ant/build-Archive.xml

Build File:

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="PDTool" default="default" basedir=".">
  <description>description</description>

  <!-- Default properties -->
  <property name="SERVERID"                  value="localhost"/>
  <property name="noarguments"               value="&amp;quot;&amp;quot;"/>

  <!-- Default Path properties -->
  <property name="RESOURCE_HOME"             value="${PROJECT_HOME}/resources"/>
  <property name="MODULE_HOME"               value="${RESOURCE_HOME}/modules"/>
  <property name="pathToServersXML"          value="${MODULE_HOME}/servers.xml"/>
  <property name="pathToArchiveXML"          value="${MODULE_HOME}/ArchiveModule.xml"/>
  <property name="pathToDataSourcesXML"      value="${MODULE_HOME}/DataSourceModule.xml"/>
  <property name="pathToGroupsXML"           value="${MODULE_HOME}/GroupModule.xml"/>
  <property name="pathToPrivilegeXML"        value="${MODULE_HOME}/PrivilegeModule.xml"/>
  <property name="pathToRebindXML"           value="${MODULE_HOME}/RebindModule.xml"/>
  <property name="pathToRegressionXML"       value="${MODULE_HOME}/RegressionModule.xml"/>
  <property name="pathToResourceXML"         value="${MODULE_HOME}/ResourceModule.xml"/>
  <property name="pathToResourceCacheXML"    value="${MODULE_HOME}/ResourceCacheModule.xml"/>
  <property name="pathToServerAttributeXML"  value="${MODULE_HOME}/ServerAttributeModule.xml"/>
  <property name="pathToTriggerXML"          value="${MODULE_HOME}/TriggerModule.xml"/>
  <property name="pathToUsersXML"            value="${MODULE_HOME}/UserModule.xml"/>
  <property name="pathToVCSModuleXML"        value="${MODULE_HOME}/VCSModule.xml"/>

  <!-- Custom properties -->
  <property name="archiveIds"                value="import1"/>

  <!-- Default Classpath [Do Not Change] -->
  <path id="project.class.path">
       <fileset dir="${PROJECT_HOME}/lib"><include name="**/*.jar"/></fileset>
       <fileset dir="${PROJECT_HOME}/dist"><include name="**/*.jar"/></fileset>
       <fileset dir="${PROJECT_HOME}/ext/ant/lib"><include name="**/*.jar"/></fileset>
  </path>
  <taskdef name="executeJavaAction" description="Execute Java Action"
classname="com.cisco.dvbu.ps.deploytool.ant.CompositeAntTask"
classpathref="project.class.path"/>

<!-- ================================
 target: default
```

```
================================= -->
  <target name="default" description="Update CIS with environment specific parameters">
  <!-- Windows / UNIX -->
        <executeJavaAction action="backup_export"
arguments="${SERVERID}^backup1^${pathToArchiveXML}^${pathToServersXML}"
endExecutionOnTaskFailure="TRUE"/>
        <executeJavaAction action="pkg_import"
arguments="${SERVERID}^import1^${pathToArchiveXML}^${pathToServersXML}"
endExecutionOnTaskFailure="TRUE"/>
        <executeJavaAction action="pkg export"
arguments="${SERVERID}^export1^${pathToArchiveXML}^${pathToServersXML}"
endExecutionOnTaskFailure="TRUE"/>
  </target>
</project>
```

EXAMPLES

The following are common scenarios when using the Archive Module.

Scenario 1 – Export resources from a CIS instance

Description:

Utilize the Archive Module to export a series of folders from a CIS instance into a Composite Archive (.car) file.

XML Configuration Sample:

Use the ArchiveModule XML file found in the /resources/module directory. Modify the option nodes to use paths that are pertinent to your environment. The entry for export should look like something this:

```
<?xml version="1.0" encoding="UTF-8"?>
<p1:ArchiveModule  xmlns:p1="http://www.cisco.dvbu.com/ps/deploytool/modules">
   <archive>
       <!--  <ArchiveType>EXPORT</ArchiveType> -->
       <id>export</id>
       <archiveMethod>CAR</archiveMethod>
       <archiveFileName>$PROJECT_HOME/resources/carfiles/testout.car</archiveFileName>
       <includeDependencies>false</includeDependencies>
       <overwrite></overwrite>
       <resources>
              <resourcePath>/shared/examples/CompositeView</resourcePath>
              <resourcePath></resourcePath>
       </resources>
   </archive>
</p1:ArchiveModule>
```

Execution Sample:

Unix: ./ExecutePDTool.sh -exec ../resources/plans/UnitTest-Archive.dp

Property file setup for UnitTest-Archive.dp:

```
# ----------------------------------------
# Begin task definition list:
# ----------------------------------------
# Export
PASS   TRUE   ExecuteAction       pkg_export      $SERVERID export
$MODULE_HOME/ArchiveModule.xml $MODULE_HOME/servers.xml
```

Results Expected:

The script will report “PASS” for the execution of this action. Check timestamp of the output file (PDTool/bin/carfiles) to make sure it is current.

Scenario 2 – Import a Composite Archive (.car)

Description:

Utilize the Archive Module to import resources from a Composite Archive (.car) file into a target CIS instance.

XML Configuration Sample:

Use the ArchiveModule XML file found in the /resources/module directory. To prove that the import worked, use Composite Studio to remove a folder resource prior to executing the import. The entry for export should look like something this:

```
<?xml version="1.0" encoding="UTF-8"?>
<p1:ArchiveModule  xmlns:p1="http://www.cisco.dvbu.com/ps/deploytool/modules">
   <archive>
       <id>import</id>
       <archiveMethod>CAR</archiveMethod>
        <archiveFileName>$PROJECT_HOME/resources/carfiles/test00.car</archiveFileName>
       <includeDependencies>false</includeDependencies>
       <overwrite>true</overwrite>
       <resources/>
   </archive>
</p1:ArchiveModule>
```

Execution Sample:

Unix: ./ExecutePDTool.sh -exec ../resources/plans/UnitTest-Archive.dp

Property file setup for UnitTest-Archive.dp:

```
# ----------------------------------------
# Begin task definition list for UNIX:
# ----------------------------------------
# Import
PASS   TRUE   ExecuteAction       pkg_import      $SERVERID import
$MODULE_HOME/ArchiveModule.xml $MODULE_HOME/servers.xml
```

Results Expected:

The script will report “PASS” for the execution of this action. Use Composite Studio to verify that the resources that were deleted previously are not present again. You may have to refresh Studio to see the imported resources.

DEBUGGING

The following describes how to setup debugging.

Set the log4j.properties file.

INFO setting (default): Required for any of the archive print options such as “printinfo”.

# Composite Archive [INFO, DEBUG, WARN]
# INFO: required for any of the print options such as printinfo
log4j.logger.com.compositesw.cmdline.archive=INFO

DEBUG setting: print out debug information in console window

log4j.logger.com.compositesw.cmdline.archive=DEBUG

WARN setting: print out warnings in console window (suppress INFO)

log4j.logger.com.compositesw.cmdline.archive=WARN

EXCEPTIONS AND MESSAGES

The following are common exceptions and messages that may occur.

CONCLUSION

Concluding Remarks

The PS Promotion and Deployment Tool is a set of pre-built modules intended to provide a turn-key experience for promoting CIS resources from one CIS instance to another. The user only requires system administration skills to operate and support. The code is transparent to operations engineers resulting in better supportability. It is easy for users to swap in different implementations of a module using the Spring framework and configuration files.

How you can help!

Build a module and donate the code back to Composite Professional Services for the advancement of the "***PS Promotion and Deployment Tool***".

ABOUT COMPOSITE SOFTWARE

Composite Software, Inc. ® is the only company that focuses solely on data virtualization.

Global organizations faced with disparate, complex data environments, including ten of the top 20 banks, six of the top ten pharmaceutical companies, four of the top five energy firms, major media and technology organizations as well as government agencies, have chosen Composite's proven data virtualization platform to fulfill critical information needs, faster with fewer resources.

Scaling from project to enterprise, Composite's middleware enables data federation, data warehouse extension, enterprise data sharing, real-time and cloud computing data integration.

Founded in 2002, Composite Software is a privately held, venture-funded corporation based in Silicon Valley. For more information, please visit www.compositesw.com.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA CXX-XXXXXX-XX 10/11