Advanced Service Assets

“PDTool” Promotion and Deployment Tool

Installation Guide

Data Virtualization Business Unit Advanced Services

August 2015

TABLE OF CONTENTS

DOCUMENT CONTROL

Version History

Version	Date	Author	Description
1.0	07/18/2011	Mike Tinius	Initial revision
1.0.1	8/1/2011	Mike Tinius	Revisions due to Architecture changes.
1.2	7/11/2012	Mike Tinius	Added info for drive substitution and network drive mapping.
1.3	8/12/2012	Mike Tinius	Added reference to SUBST drive mapping for long file names.
1.4	10/1/2012	Mike Tinius	Added reference to server ping, retry and sleep for deploy.properties
1.5	5/9/2013	Mike Tinius	Separated out user environment variables from ExecutePDTool.bat/.sh to setVars.bat/.sh.
3.0	8/21/2013	Mike Tinius	Update docs to Cisco format.
3.1	1/31/2014	Mike Tinius	Add documentation for HTTPS configuration.
3.2	2/18/2014	Mike Tinius	Prepare docs for open source.
3.3	11/17/2014	Mike Tinius	Updated license.
3.4	3/4/2015	Mike Tinius	Updated docs to Cisco format.
3.5	08/18/2015	Mike Tinius	Added Automated Test Framework and PDTool Installer and bug fixes.

Related Documents

Composite PS Promotion and Deployment Tool User's Guide	Composite PS Promotion and Deployment Tool User's Guide v1.0.pdf	Mike Tinius
LabPD-DeployTool-v1.0	LabPD-DeployTool-v1.0.pdf	Mike Tinius
PS Promotion and Deployment Tool v1.1	PS Promotion and Deployment Tool - v1.1.ppt	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Archive	Composite PS Promotion and Deployment Tool Module - Archive.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - DataSource	Composite PS Promotion and Deployment Tool Module - DataSource.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Group	Composite PS Promotion and Deployment Tool Module - Group.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Privilege	Composite PS Promotion and Deployment Tool Module - Privilege.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Rebind	Composite PS Promotion and Deployment Tool Module - Rebind.pdf	Jerry Joplin
Composite PS Promotion and Deployment Tool Module - Regression	Composite PS Promotion and Deployment Tool Module - Regression.pdf	*Sergei Sternin*
Composite PS Promotion and Deployment Tool Module - Resource Cache	Composite PS Promotion and Deployment Tool Module - Resource Cache.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Resource	Composite PS Promotion and Deployment Tool Module - Resource.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Server Attribute	Composite PS Promotion and Deployment Tool Module - Server Attribute.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Server Manager	Composite PS Promotion and Deployment Tool Module - Server Manager.pdf	Gordon Rose
Composite PS Promotion and Deployment Tool Module - Trigger	Composite PS Promotion and Deployment Tool Module - Trigger.pdf	Kevin O'Brien

Composite PS Promotion and Deployment Tool Module - User	Composite PS Promotion and Deployment Tool Module - User.pdf	Mike Tinius
Composite PS Promotion and Deployment Tool Module - Version Control System.pdf	Composite PS Promotion and Deployment Tool Module - Version Control System.pdf	Mike Tinius
Composite PS Promotion and Deployment Developer’s Guide – Field Edition	Composite PS Promotion and Deployment Tool Developer's Guide - Field Edition.docx	Gordon Rose

Data Virtualization Business Unit (DVBU) Products Referenced

DVBU Product Name	Version
Composite Information Server	6.2, 7.0

INTRODUCTION

License

This software is released under the Eclipse Public License. The details can be found in the file LICENSE. Any dependent libraries supplied by third parties are provided under their own open source licenses as described in their own LICENSE files, generally named .LICENSE.txt. The libraries supplied by Cisco as part of the Composite Information Server/Cisco Data Virtualization Server, particularly csadmin-XXXX.jar, csarchive-XXXX.jar, csbase-XXXX.jar, csclient-XXXX.jar, cscommon-XXXX.jar, csext-XXXX.jar, csjdbc-XXXX.jar, csserverutil-XXXX.jar, csserver-XXXX.jar, cswebapi-XXXX.jar, and customproc-XXXX.jar (where -XXXX is an optional version number) are provided as a convenience, but are covered under the licensing for the Composite Information Server/Cisco Data Virtualization Server. They cannot be used in any way except through a valid license for that product.

This software is released AS-IS!. Support for this software is not covered by standard maintenance agreements with Cisco. Any support for this software by Cisco would be covered by paid consulting agreements, and would be billable work.

Purpose

The purpose of this document is to provide guidance on how to install the Composite "***PS Promotion and Deployment Tool***". Additionally, this document describes what is found in the distribution for both Command line and Ant deployment.

Upon completion of the installation, please refer to the following documentation:

For Users performing promotion and deployment of CIS resources:

1. **User** – Composite PS Promotion and Deployment Tool User's Guide v1.0.pdf
2. **Training** – LabPD-DeployTool.pdf
3. **Modules** – Modularized functionality within the Promotion and Deployment Tool
    - Composite PS Promotion and Deployment Tool Module - Archive.pdf
    - Composite PS Promotion and Deployment Tool Module - DataSource.pdf
    - Composite PS Promotion and Deployment Tool Module - Group.pdf
    - Composite PS Promotion and Deployment Tool Module - Privilege.pdf
    - Composite PS Promotion and Deployment Tool Module - Rebind.pdf
    - Composite PS Promotion and Deployment Tool Module - Regression.pdf
    - Composite PS Promotion and Deployment Tool Module - Resource Cache.pdf
    - Composite PS Promotion and Deployment Tool Module - Resource.pdf
    - Composite PS Promotion and Deployment Tool Module - Server Attribute.pdf

- Composite PS Promotion and Deployment Tool Module - Server Manager.pdf
- Composite PS Promotion and Deployment Tool Module - Trigger.pdf
- Composite PS Promotion and Deployment Tool Module - User.pdf
- Composite PS Promotion and Deployment Tool Module - Version Control System.pdf

For Developers building new modules to snap into the tool:

1. **Development** – Composite PS Promotion and Deployment Tool Developer's Guide - Field Edition - v1.0.pdf
2. **Training** – LabPD-DeployTool.pdf

Audience

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

- **Operations personnel** – provides guidance on how to execute promotion scripts.
- **Architects** – provides guidance on how CIS fits in with the version control and the deployment infrastructure.
- **Developers** – provides guidance on how to develop new Modules of functionality.

OS Platforms Supported

Windows Platforms: Windows XP, Windows 7
UNIX Platforms: Linux 5.5

Note: Porting to other operating systems that CIS runs on are available upon request

Java:

JRE 1.6 for command-line deployment
JDK 1.6 for Ant deployment

Composite Information Server Versions Supported

Composite Information Server Versions: CIS 6.2 and CIS 7.0.

Version Control Systems Supported

Version Control Systems (VCS) supported out of the box:

- Subversion 1.6 or higher
- Perforce 2010.2 or higher
- Concurrent Versions Systems (CVS) 2.0.51d or higher
- Team Foundation Server (TFS) 2010 – (TFS) 2012
- GIT

DISTRIBUTION

Combined Command-Line and Ant Distribution

The distribution file for Command Line and Ant deployment is PDTool.zip.

PDTool Contents

1. **bin** – contains the shell and batch scripts for execution.
2. **dist** – contains the PDTool distribution jar file with the necessary java classes for executing the deployment.
3. **docs** – contains documentation.
4. **ext/ant** – contains a distribution of the ant executable files such that ant is not required to be installed on the deployment server.
5. **lib** – contains libraries referenced by the java classes.
6. **resources/**
    - 6.1. **ant** – contains the ant build files for orchestration of a deployment scenario. A deployment scenario is a sequence of actions or tasks that get executed.
    - 6.2. **carfiles** – contains the test car files.
    - 6.3. **config** – contains the deployment tool configuration files for the spring and logging frameworks as well as deploy.properties for PDTool configuration.
    - 6.4. **modules** – contains the property files for CIS server instances and each Module XML property file.
    - 6.5. **plans** – contains the deployment plan files for command-line orchestration of a deployment scenario. A deployment scenario is a sequence of actions or tasks that get executed.
    - 6.6. **schema** – contains the XML Schema definition “PDToolModules.xsd” for all resource module XML property files.
    - 6.7. **vcs_initial** – contains the base folders and template folders used for doing an initial check-in into a VCS system.
7. **security** – contains the java.policy and keystore for accessing CIS using SSL.

INSTALLATION

Combined Command Line and Ant Script Installation

Installation of the command line and Ant scripts is straightforward as it is nothing more than a zip file. This is the part of PDTool that provides deployment and testing facilities. The other part of PDTool is called PDTool Studio and is focused on version control. The purpose of this installer is to allow "**pre-configuration**" of default values for your organization so that two or more users can have a consistent installation across the organization. It allows you to pre-configure variables according to your organization to make it easier for "mass-installation" of the tool. Ultimately, a consistent installation means it is easier to support the developers and testers when they have questions or issues.

Automated PDTool Installer

The automated PDTool installer only works with a Windows environment. It is not supported with UNIX. For UNIX installation, follow the "old style of installation".

What gets installed?

- Cisco PDTool 6.2 or PDTool 7.0.0
- VCS Client if the user chooses to pre-configure the \VCSClients folder

Where do the files get copied?

- Recommended Location: C:\Users\%USERNAME%\.compositesw\PDTool[6.2|7.0.0]_[NOVCS|TFS|SVN|GIT|P4|CVS] by default or a directory of your choice if provided.
    - **\PDTool** – PDTool deployment directory.
    - **\VCSClients** – Pre-configured VCS clients are optional if installed elsewhere
        - **\SVN_client** – Subversion client
        - **\TFS_TEE_client** – Team Explorer Everywhere client
        - **\GIT_client** – Git client
        - **\P4_client** – Perforce client
        - **\CVS_client** – CVS client

Audience?

- For CIS Developers who want to deploy resources from a version control system (VCS) such as TFS or Subversion to a target CIS server.
- For CIS testers who want to use PDTool Regression Module for testing but do not need to connect to a VCS.
- For a user who wants to encrypt a PDTool file which contains passwords.

System Requirements

The following system requirements must be met for executing Windows or UNIX scripts.

1. **Java 7**

   1.1. **Command-Line** - JRE 1.7 is required to be present on the system.

   1.2. **Ant** - JDK 1.7 is required for Ant Deployment and must be present on the system.

2. **Version Control System Access**.

   2.1. The user must submit a request in advance to get access to a version control system (VCS) repository prior to installation of PDTool. The user must be prepared with the correct URL and root folder to where CIS is found and credentials for login.

3. **Read/Write Access** – The script must have read/write access to the file system where PDTool is installed. The location of the VCS Workspace must have read/write access by the user. Log files will be written as well.

4. **UNIX (Bash)** – Scripts on UNIX require access to /bin/bash

   4.1. **dos2unix** – required to be installed and accessible in the path

Installation – Automated Windows Installer (new style of installation)

1. Caveat: The installer is only supported on windows. If installing on UNIX then proceed to the section "Manual Installation Overview (old style of installation)"

2. **Pre-Requisites for the PDTool Administrator**

   2.1. The following steps are performed once by a PDTool Administrator who wants to pre-configure the PDTool installation for their organization. Pre-configuring involves setting default values in the configuration file and potentially, copying the VCS binary files into the VCSClients folder. This will allow for a seamless installation by a group of developers, testers or deployers. The following steps are **performed by the PDTool Administrator**:

   2.2. Pre-configure the default variables.

   2.2.1. For VCS installation, edit: **PDTool-VCS.bat** and provide any default values that require modification. If more than one VCS is available for use in the organization or different VCS repositories are in use that require deployment for CIS, then copy PDTool-VCS.bat and give the new file name a descriptive name.

   2.2.2. For non-VCS installation, edit **PDTool-NOVCS.bat** and provide any default values that require modification.

2.2.3. Provide instructions for the user by modifying the template "***Cisco PDTool Installation.docx***" found in the \installer directory.

2.3. Pre-configure VCS client software folders (optional).

2.3.1. To insure consistency of VCS clients across the developer community, be sure to copy the correct client software into the appropriate "VCS Clients" sub-directory.

2.3.1.1. SVN – copy the subversion binaries to the VCSClients\SVN\bin directory.

2.3.1.2. TFS-TEE – copy the Team Foundation Everywhere client binaries to the TFS-TEE directory.

2.3.1.3. P4 – copy the Perforce client binaries to the P4 directory.

2.3.1.4. GIT – copy the GIT client binaries to the GIT directory.

2.3.1.5. CVS – copy the CVS client binaries to the CVS directory.

2.3.2. Zip up the Cisco_PDTool6.2_Installer or Cisco_PDTool7.0_Installer directory and post somewhere on the organizations site for developers to download and install.

2.4. Pre-configure configuration deployment propery files.

2.4.1. This step.

3. **User Installation Procedure**

3.1. **Copy** the following zip files from a pre-determined share drive folder to any directory on the **user's computer [Recommended folder: Desktop –** C:\Users\%USERNAME%\Desktop**].**

For CIS 6.2 – "Cisco_PDTool6.2_Installer.zip"

For CIS 7.0 – "Cisco_PDTool7.0_Installer.zip"

3.2. Use PKZip and select "Extract Here" which will extract them to the Desktop as shown below. If you use the Windows "Extract All", then it is recommended to remove the trailing folder:

Location: C:\Users\%USERNAME%\Desktop\Cisco_PDTool6.2_Installer

Location: C:\Users\%USERNAME%\Desktop\Cisco_PDTool7.0_Installer

3.3. Install Cisco PDTool

3.3.1. **Pre-Requisite**: The variables in **PDTool-VCS.bat** or **PDTool-NOVCS.bat** should have been pre-configured by the PDTool administrator. If you are not the PDTool Administrator, then you will need to determine if the variables have been pre-set. If not, you will need to set the variables prior to installation.

For any VCS [TFS,SVN,GIT,P4,CVS] Execute:

- Edit: **PDTool-VCS.bat** and provide any default values that require modification.
- Execute by **double-clicking** on the following batch file from windows explorer**:**

  PDTool-VCS.bat

  - Acknowledge Cisco Open Source License
  - Follow the "Prompts" section below for detailed information.

***For non*-VCS Installation such as Regression testing**:

- Edit: **PDTool-NOVCS.bat** and provide any default values that require modification..
- Execute by **double-clicking** on the following batch file from windows explorer**:**

  PDTool-NOVCS.bat

  - Acknowledge Cisco Open Source License
  - Follow the "Prompts" section below for detailed information.

3.3.2. What gets executed during installation?

- Copy source files to destination folder
- For TFS only, TFS eula –accept
- Encrypt passwords: C:\Users\%USERNAME%\.compositesw\PDTool<ver>_<vcs>\setMyPDToolVars.bat
- Initialize workspace (requires the VCS credentials and repository URL).
  - This does not get executed for PDTool-NOVCS.bat

3.3.3. User Prompts

The user is prompted for the following parameters unless this information is provided as parameters on the command line. The ***default values are provided within the square brackets***. Press enter with no input to accept the default value or type your value and press enter.

Enter I_PDTOOL_INSTALL_SCRIPTS=[%DEF_PDTOOL_INSTALL_SCRIPTS%]:

- Location of the PDTool installation scripts. E.g. C:\Users\%USERNAME%\Desktop\CiscoPDToolInstall7.0.0

License Acknowledgement

- Use space bar to page through license
- Review and acknowledge the open source license [Y].

Enter I_JAVA_HOME=[%DEF_JAVA_HOME%]:

- I_JAVA_HOME=JRE7 home folder. E.g. C:\Program Files\Java\jre7

Enter I_PDTOOL_DESTINATION_HOME=[%DEF_PDTOOL_DESTINATION_HOME%]:

- I_PDTOOL_DESTINATION_HOME - The destination home folder for the PDTool installation and associated binaries.
- NOVCS:
  C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_NOVCS
- TFS:
  C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_TFS
- SVN:
  C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_SVN
- GIT:
  C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_GIT
- P4:
  C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_P4
- CVS:
  C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_CVS

Enter I_PDTOOL_DESTINATION_DIR=[%DEF_PDTOOL_DESTINATION_DIR%]:

- The destination directory name for the PDTool installation and associated binaries. e.g. PDTool7.0.0

Enter I_OVERWRITE_DECISION=[N]:

- This prompt only comes up when the target PDTool directory exists. It allows the user to decide whether they want to overwrite that directory or not. Enter Y to overwrite the existing directory.

Enter I_VCS_TYPE=[%DEF_VCS_TYPE%]:

- I_VCS_TYPE - The version control type [TFS|SVN|GIT|P4|CVS]

Enter I_VCS_HOME=[%DEF_VCS_HOME%]:

- This is the location of the VCS script executable. E.g. C:\Users\%USERNAME%\.compositesw\PDTool7.0.0_SVN\VCSClients\SVN\bin

Enter I_VCS_REPOSITORY_URL=[%DEF_VCS_REPOSITORY_URL%]:

- Note: make sure the forward slashes are escaped with 4 slashes: https:////url
- TFS: The TFS repository URL pointing to the repository collection. e.g. http:////hostname.domain.com/tfs/DefaultCollection
- SVN: The subversion repository path at trunk or any folder designation within trunk. e.g. https:////svn.hostname.ml.com/svnrepos/myrepo/trunk/main

Enter I_VCS_PROJECT_ROOT=[%DEF_VCS_PROJECT_ROOT%]:

- E.g. TFS: Rel
- E.g. SVN: cis_objects

Enter I_RELEASE_FOLDER=[%DEF_RELEASE_FOLDER%]:

- Example: RELEASE_FOLDER=20150328

Enter I_VCS_USERNAME=[%USERNAME%]:

- This is your user name regardless of whether it is TFS or Subversion and is used to connect to the VCS repository. The user must submit a request to get access to a repository prior to installation of PDTool.

Enter I_VCS_DOMAIN=[%DEF_VCS_DOMAIN%]:

- When using TFS, include the VCS domain such as “@CORP”. The result for the I_VCS_USERNAME would look like “username@CORP”. When using subversion, leave the domain blank. The I_VCS_DOMAIN will be automatically appended to I_VCS_USERNAME.

Enter I_VCS_PASSWORD=<type-your-vcs-password>

- This is your VCS password which will be encrypted.

Enter I_WORKSPACE_NAME=[%DEF_VCS_WORKSPACE_NAME%]:

- The PDTool VCS workspace name is derived from the VCS username and Release folder so that it is unique across usernames and releases.

Enter I_CIS_USERNAME=[%USERNAME%]:

- This is your user name regardless that will be used to connect to CIS.

Enter I_CIS_DOMAIN=[%DEF_CIS_DOMAIN%]:

- This is the CIS domain which is used by the I_CIS_USERNAME to connect to Composite. E.g. ldap or composite

Enter I_CIS_PASSWORD=<type-your-cis-password>

- This is your password which will be encrypted.

Enter I_CONFIG_PROPERTY_FILE=[%DEF_CONFIG_PROPERTY_FILE%]

- This is the default PDTool configuration property file that the user will use to set the context of which CIS server to connect to.

The variables are displayed

Enter I_VARS_DECISION [Y or N] – If “Y”, then installation commences. If “N” then the user is prompted for the variables again.

3.3.4. Manage Errors

3.3.4.1. If there are any errors reported in the scripts especially during workspace initialization, and then re-execute the installation script.

3.3.4.2. If there is an error while trying to create a substitute drive then first trying removing the substitute drive which was identified as available in the script. net use <drive>: /DELETE

3.3.4.3. If there is an error during initialization, try removing the workspace directory and try again.

Manual Installation Overview (old style of installation)

4. **Select a Project Folder**
5. **Copy or FTP the PDTool.zip**
6. **Unzip PDTool.zip**
7. **Configure** UNIX permissions and property files
8. **Configure ExecutePDTool Script Environment Variable**
9. **Configure deploy.properties file**
10. **Configure log4j properties files**
11. **Configure PD Tool with version control (optional)**

Manual Installation and Configuration Details (old style of insallation)

1. **Project Folder** – Select a project folder preferably with no spaces.

1.1. **Key definitions**

1.1.1. <ver>=6.2 or 7.0

1.1.2. <vcs>=NOVCS, SVN, TFS, GIT, P4 or CVS

1.2. Windows – e.g. c:\users\%USERNAME%\.compositesw\PDTool**<ver>_<vcs>**

1.3. UNIX - e.g. /opt/PDTool**<ver>**_**<vcs>**

2. **Zip PDTool from installer** – Zip up the PDTool directory from the PDTool installer_source folder.

3. **FTP or Copy** – FTP or Copy the PDTool.zip to the project folder

4. **Unzip** – Unzip PDTool.zip to the project folder you selected in step 1.

5. **Configure scripts**

   5.1. Windows – not applicable

   5.2. UNIX - Configure permissions and dos2unix using **configureScripts.sh**. This script will iterate through a list of folders provided in the script and do the following

   - Set execution mode on all .sh files [chmod 744 filename.sh]
   - Set permissions based on the user and group: [chown user, chgrp group]
   - Remove windows linefeeds on all .sh and property files [dos2unix filename]

   Instructions:

   5.3. Starting in your PDTool folder, cd bin

   5.4. chmod 744 configureScripts.sh

   5.5. dos2unix configureScripts.sh

   5.6. ./configureScripts.sh <user> <group>

      5.6.1. e.g. ./configureScripts.sh composite composite

6. **Configure Environment Variables**

   6.1. Windows

      6.1.1. Modify "**setVars.bat**"

         6.1.1.1. Location: \PDTool<ver>_<vcs>\bin\setVars.bat

         6.1.1.2. Edit the location of MY_VARS_HOME.

         6.1.1.3. e.g. set MY_VARS_HOME= c:\users\%USERNAME%\.compositesw\PDTool<ver>_<vcs>\bin

         6.1.1.4. Save and exit.

      6.1.2. Modify "**setMyPrePDToolVars.bat**"

         6.1.2.1. Location: \PDTool<ver>_<vcs>\bin\setMyPrePDToolVars.bat

         6.1.2.2. Provide default values for the General and CIS sections:

            REM ###################################

REM # GENERAL GROUP ENVIRONMENT
REM ###################################
REM # My Java Home
set MY_JAVA_HOME=
REM # PDTool Studio Substitute Drive Letter used for making the path shorter
REM # PDTool and PDTool Studio must use different drive letters.
set PDTOOL_SUBSTITUTE_DRIVE=
REM # PDTool Installation Home Directory
set PDTOOL_INSTALL_HOME=
REM # PDTool Home directory
set PDTOOL_HOME=
REM # Name of the configuration property file located in PDToolStudio62/resources/config
REM # e.g. Default=studio.properties or SVN=studio_SVN.properties or TFS=studio_TFS.properties
set MY_CONFIG_PROPERTY_FILE=
REM ###################################
REM # COMPOSITE SERVER VARIABLES
REM ###################################
REM # 0=Do not print this section, 1=Print this section
set CIS_PRINT=1
REM # Composite Server Username
set CIS_USERNAME=
REM # Composite Server Password
set CIS_PASSWORD=
REM # Composite Server Domain
set CIS_DOMAIN=

6.1.2.3. Provide default values for one of the VCS sections [TFS|SVN|GIT|P4|CVS]:

REM ###################################
REM # [TFS|SVN|GIT|P4|CVS] VARIABLES
REM ###################################
REM # 0=Do not print this section, 1=Print this section
set [TFS|SVN|GIT|P4|CVS]_PRINT=1
REM # The VCS Home folder where VCS client exists
set [TFS|SVN|GIT|P4|CVS]_HOME=
REM # The VCS repository path at trunk or any folder designation within trunk
set [TFS|SVN|GIT|P4|CVS]_VCS_REPOSITORY_URL=
REM # The VCS folder path starting directly after the VCS repo URL and ending where the Composite base level root folders start
set [TFS|SVN|GIT|P4|CVS]_VCS_PROJECT_ROOT=
REM # VCS user name
set [TFS|SVN|GIT|P4|CVS]_VCS_USERNAME=
REM # VCS user password. See notes at top of this file to encrypt.

set [TFS|SVN|GIT|P4|CVS]_VCS_PASSWORD=
REM # Set the VCS Workspace name.
set [TFS|SVN|GIT|P4|CVS]_VCS_WORKSPACE_NAME=

6.1.2.4. Save and exit.

6.2. UNIX –

6.2.1. Modify **setVars.sh**

6.2.2. Edit JAVA_HOME variable to point to either a JRE 1.7 when using Command-line deployment or JDK 1.7 home when using Ant deployment.

6.2.3. Edit MIN_MEMORY and set the minimum java heap memory to use. The default is "-Xms256m".

6.2.4. Edit MAX_MEMORY and set the maximum java heap memory to use. The default is "-Xmx512m".

6.2.5. Save and exit

7. **Configure deploy.properties**

7.1. Note: If not using a Version Control System (VCS) for deployment, the default values should be acceptable for deploy.properties.

Instructions for modifying deploy.properties

7.2. This step is applicable for both UNIX and Windows

7.3. This step is applicable for both Command-Line and Ant deployment.

7.4. **General Instructions for deploy.properties**

7.4.1. **PROJECT_HOME** is automatically set upon invocation of the ExecutePDTool script based on PDTool/bin relative location

7.4.2. **Edit** PDTool/resources/config/**deploy.properties** using a text editor.

7.4.3. Always use forward slashes for both Windows and Unix paths and URLs.

7.4.4. Variables may use $ or % notations. It is not operating system specific.

7.4.5. Variables may resolve to this property file, Java Environment (-DVAR=val) or the System Environment variables

7.4.6. Surround variables with two $ or two % when concatenating strings (e.g. VCS_TYPE_cisVcsTemp)

7.5. **Guidance**

7.5.1. If not using VCS deployment, you should be able to take all of the defaults.

7.5.2. If using VCS deployment then pay close attention to section 6.7.

7.6. **Customer-Defined** Environment Variables [Optionally set]

7.6.1. Set environment variables as needed for use within the PDTool.dp orchestration file. The idea with these properties is to use them within the property orchestration file as variables. An Example is shown below:

```
#=====================================================================
# Customer-defined Environment Variables [Optional]
#   Note: May be used in either the orchestration property file.
#   This section may not be used with ANT.
#=====================================================================
CARFILE=testout.car
#
# Provides an easy way to customize which server to the deployment
orchestration property file will connect to
SERVERID=localhost
#
# Server environment for the LabPD-Deploy
LAB_SERVERID=labhost
#
```

7.7. **Behavior** Environment Variables [Optionally modified]

7.7.1. It is recommended that these properties be left as defaults unless more debug information is required.

```
#=====================================================================
# Behavior Environment Variables [default values set]
#=====================================================================
# Suppress [true|false] printing PDTool orchestration comments to the log
SUPPRESS_COMMENTS=true
#
# DEBUG=true|false :: Turn on when debugging this script
# Debug Level 1: Debug PDTool script only
DEBUG1=false
# Debug Level 2: Debug ExecuteAction, ExecuteVCS
DEBUG2=false
# Debug Level 3: Debug 3rd level scripts invoked from ExecuteAction and
ExecuteVCS
DEBUG3=false
```

```
# Diffmerger Verbose allows the VCS Diffmerger process to output more
information when set to true [Default=true]

DIFFMERGER_VERBOSE=true

#

# Used when parsing property file and processing the moduleId list

allResourcesIndicator=*

exculdeResourcesIndiator=-

#

# userOptionThreshold - The threshold of the number of users where it is more
efficient to retrieve all users at once (option 1).

# If the number of users to be processed in the UserModule XML file exceeds
the threshold then use option 1.

# Retrieve the entire CIS user list in one invocation. This options requires
storing all CIS user and domain in memory.

# If there are 1000's of users, this could take a lot of memory.

# To use this option, determine how many users are in your composite domain
and set the threshold to that value or greater.

# vs.

# calling the API to retrieve a user for each user found in the UserModule.xml
(option 2)

# If the number of users to be processed in the UserModule XML file is less
than the threshold then use option 2.

# Retrieve the user info on each invocation of a user found in the
UserModule.xml.

# This requires a web service invocation to the CIS server for each user to be
processed.  A threshold of 0 will cause the invocation of the user API one at
a time.

userOptionThreshold=0

#

# Listing of DataSourceModule generic attributes that are not updateable

DataSourceModule_NonUpdateableAttributes=url

#

# Listing of ServerAttributeModul generic attributes that are not updateable

ServerAttributeModule_NonUpdateableAttributes=

#

# Listing of DomainModule generic attributes that are not updateable

DomainModule_NonUpdateableAttributes=

#

# This provides an externalized mechanism to teach PD Tool about new Resource
Types and how they are associated with the basic VCS Resource Types.  The
basic VCS Resource Types include:

#    FOLDER, definitions, link, procedure, table, tree, trigger

#
```

```
# Each Studio Resource contains an info tab with a resource path and a display
type.  Use the Resource path in the info tab as input into the following
Studio Web Service API to discover the CIS Resource Type.

#   /services/webservices/system/admin/resource/operations/getResource()

#

# The CIS Resource Type is mapped to one of the basic VCS Resource Types
provided above.  Finally, provide the name value pair in the form of "VCS
Resource Type=Studio Display Resource Type".  Create a comma separate list of
these name=value pairs.  For example:

# folder=Data Source,folder=Composite Database,procedure=Basic Transformation

#

VCSModule_ExternalVcsResourceTypeList=

#

# CIS_PING_SERVER:  Allows the user to control whether the server performs a
ping on CIS prior to executing the actual command.

# The default is true if this variable is not set or does not exist.  This
will override all settings.

CIS_PING_SERVER=false

#

# CIS_CONNECT_RETRY: Allows the user to set the number of retries to connect
to CIS before throwing an error.

# The default is 1 if this variable is not set or does not exist.

CIS_CONNECT_RETRY=1

#

# CIS_CONNECT_RETRY_SLEEP: Number of milliseconds to sleep on connection retry

CIS_CONNECT_RETRY_SLEEP_MILLIS=5000

#----------------------------------------------------------------------
```

7.8. **Version Control System (VCS)** Environment Variables [Optionally set]

7.8.1. Only set when using VCS deployment. This entire section is detailed out in the document "**Composite PS Promotion and Deployment Tool Module - Version Control System.pdf"**

```
#======================================================================

# VCS Environment Variables [Optional]

#======================================================================

...
```

7.9. **VCS-Specific** Environment Variables [Optionally set]

7.9.1. The following three sections [Subversion, Perforce, CVS] are only set when using VCS deployment. This entire section is detailed out in the document "**Composite PS Promotion and Deployment Tool Module - Version Control System.pdf"**

```
#=====================================================================
#### [SUBVERSION] USER MODIFIES [OPTIONAL] #####
# Subversion [svn] specific environment variables are set here
#=====================================================================
...
#=====================================================================
#### [PERFORCE] USER MODIFIES [OPTIONAL] #####
# Perforce [p4] specific environment variables are set here
#=====================================================================
...
#=====================================================================
#### [CVS] USER MODIFIES [OPTIONAL] #####
# Concurrent Versions System [cvs] specific environment variables are set here
#=====================================================================#
...
```

7.10. DeployTool System Environment Variables [Do Not Modify]

7.10.1. This section defaults to preset values based on the location of PROJECT_HOME. The variable PROJECT_HOME is automatically derived upon execution of the ExecutePDTool script. These values should not be modified.

```
#---------------------------------------------------------------------------
#### DO NOT MODIFY #####
#============================================================================
# DeployTool System Environment Variables
#============================================================================
# PDTool configuration directory locations
# Note:  Relative paths outside of the project home are permitted as shown
here:  $PROJECT_HOME/../modules.  This path gets resolved to a sibling
directory to $PROJECT_HOME.
MODULE_HOME=$PROJECT_HOME/resources/modules
SUMMARY_LOG=$PROJECT_HOME/logs/summary.log
#
# SCHEMA_LOCATION - Provides a copy of the XML Schema which is used for
validation purposes.
# For command line deployment, use of "../resources/schema/PDToolModules.xsd"
will work.
SCHEMA_LOCATION=$PROJECT_HOME/resources/schema/PDToolModules.xsd
#
```

```
# Comma separated list of Resource Sub Types that are not considered
rebindable
nonRebindableResourceSubTypeList=DATABASE_TABLE
#-------------------------------------------------------------------------
-
```

8. **Configure log4j.properties**

8.1. Note: The default values should be acceptable for log4j.properties.

Instructions for modifying log4j.properties

8.2. This step is applicable for both UNIX and Windows

8.3. This step is applicable for both Command-Line and Ant deployment.

8.4. **Edit** PDTool/resources/config/**log4j.properties** using a text editor.

8.5. It is up to the user to determine what levels of logging are required. This rule is at the heart of log4j. It assumes that levels are ordered. For the standard levels in order, we have DEBUG < INFO < WARN < ERROR < FATAL.

8.6. The following describes the “app.log” settings. The app.log is used for output of the invoked Java Actions.

```
log4j.rootCategory=WARN,stdout,FileAppender

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ISO8601} %t %p [%c] - <%m>%n
log4j.appender.FileAppender=org.apache.log4j.RollingFileAppender
# Forward slashes (/) must be used when defining log file path
log4j.appender.FileAppender.File=../logs/app.log
log4j.appender.FileAppender.MaxFileSize=1MB
log4j.appender.FileAppender.MaxBackupIndex=1
log4j.appender.FileAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.FileAppender.layout.ConversionPattern=%d{ISO8601} %t %p [%c] - <%m>%n
log4j.appender.FileAppender.Threshold=DEBUG
```

8.7. The following describes logging general behavior settings.

```
# This rule is at the heart of log4j. It assumes that levels are ordered.
# For the standard levels in order, we have DEBUG < INFO < WARN < ERROR < FATAL.
# Set to DEBUG if there are script issues and you want to get more output information
# PDTool Command-line Orchestration Debugging
log4j.logger.com.cisco.dvbu.ps.deploytool.CisDeployTool=INFO
# PDTool Module Debugging for Command-line and Ant (Applies to *ALL* Modules)
```

```
log4j.logger.com.cisco.dvbu.ps.deploytool=INFO
# Common Utility Framework Debugging includes:
# [CommonUtils, PropertyManager, PropertyUtil, ScriptExecutor, XMLUtils, JdbcConnector]
log4j.logger.com.cisco.dvbu.ps.common.util=INFO
# VCS Diffmerger and Archive Services Debugging
log4j.logger.com.cisco.dvbu.cmdline=INFO
# Composite Archive [INFO, DEBUG, WARN]
#   INFO: required for any of the print options such as printinfo
log4j.logger.com.compositesw.cmdline.archive=INFO
# Spring Framework Debugging
log4j.logger.org.springframework=WARN
```

Note: This should be left at ERROR

```
# Note: This should be left at ERROR
log4j.logger.com.cisco.dvbu.ps.common.scriptutil=ERROR
```

9. **Configure PD Tool with version control (optional)**

9.1. This entire section is detailed out in the document "**Composite PS Promotion and Deployment Tool Module - Version Control System.pdf"**

CONFIGURE PDTOOL OVER SSL (HTTPS)

How to configure PDTool over SSL

This section discusses the procedure for configuring both PDTool 6.2 and the target CIS server. Only PDTool 6.2 supports the ability for PDTool to connect to CIS over SSL using the https protocol. The minimum requirement for CIS is to be patched to CIS 6.2.4.00.46.

Configuring PDTool Over SSL

The following steps outline what a deployment administrator or developer should do to modify PDTool and CIS for communicating over SSL using HTTPS.

1. **PDTool Modifications for Windows**

   1.1. Edit <PDTOOOL_HOME>\bin\setVars.bat

   1.2. Find the section labeled: PDTool Over SSL (https)

   1.3. The default setting is for PDTool to point to the weak TustStore which ships with PDTool 6.2 and CIS.

   1.4. **Important**: If you your CIS Server has the Strong Encryption Pack installed, you must configure PDTool to use the Strong Encryption Pack or you may see this error:

   com.cisco.dvbu.ps.deploytool.DeployManagerUtil.main(DeployManagerUtil.java:103) Caused by: javax.xml.ws.WebServiceException: **Failed to access the WSDL at: https://localhost:9402/services/system/admin?wsdl.** It failed with: Got sun.security.validator.ValidatorException: **PKIX path building failed**: sun.security.provider.certpath.SunCertPathBuilderException: **unable to find valid certification path to requsted target while opening stream from https://localhost:9402/services/system/admin?wsdl**.

   1.5. To point to the strong encryption pack:

      1.5.1. Acquire the strong encryption pack from Cisco Support.

      1.5.2. Determine which option best identifies your scenario:

      1.5.3. Option 1: PDTool is installed on a server with CIS Server present.

         1.5.3.1. Unzip the Strong Encryption Pack into <CIS_HOME>

         1.5.3.2. Modify the CERT_ARGS path to point to cis_server_truststore_strong.jks

      1.5.4. Option 2: PDTool is installed on a client with CIS Studio present.

         1.5.4.1. Unzip the Strong Encryption Pack into <CIS_HOME>

1.5.4.2. Modify the CERT_ARGS path to point to cis_studio_truststore_strong.jks

1.5.5. Option 3: PDTool is installed stand-alone with no CIS Server or Studio present on the machine.

1.5.5.1. Unzip the Strong Encryption to any folder.

1.5.5.1.1. E.g. C:\Composite\StrongEncryption\conf\studio\security\cis_studio_truststore_strong.jks

1.5.5.2. Modify the CERT_ARGS path to point to cis_studio_truststore_strong.jks

1.5.5.3. SET CERT_ARGS="-Djavax.net.ssl.trustStore=C:\Composite\StrongEncryption\conf\studio\security\cis_studio_truststore_strong.jks" -Djavax.net.ssl.trustStorePassword=changeit

2. **PDTool Modifications for Linux**

2.1. Edit <PDTOOOL_HOME>/bin/setVars.sh

2.2. Find the section labeled: PDTool Over SSL (https)

2.3. The default setting is for PDTool to point to the weak TustStore which ships with PDTool 6.2 and CIS.

2.4. **Important**: If you your CIS Server has the Strong Encryption Pack installed, you must configure PDTool to use the Strong Encryption Pack or you may see this error:

com.cisco.dvbu.ps.deploytool.DeployManagerUtil.main(DeployManagerUtil.java:103) Caused by: javax.xml.ws.WebServiceException: **Failed to access the WSDL at: https://localhost:9402/services/system/admin?wsdl.** It failed with: Got sun.security.validator.ValidatorException: **PKIX path building failed**: sun.security.provider.certpath.SunCertPathBuilderException: **unable to find valid certification path to requsted target while opening stream from https://localhost:9402/services/system/admin?wsdl**.

2.5. To point to the strong encryption pack:

2.5.1. Acquire the strong encryption pack from Cisco Support.

2.5.2. Determine which option best identifies your scenario:

2.5.3. Option 1: PDTool is installed on a server with CIS Server present.

2.5.3.1. Unzip the Strong Encryption Pack into <CIS_HOME>

2.5.3.2. Modify the CERT_ARGS path to point to cis_server_truststore_strong.jks

2.5.4. Option 2: PDTool is installed on a client with CIS Studio present.

2.5.4.1. Unzip the Strong Encryption Pack into <CIS_HOME>

2.5.4.2. Modify the CERT_ARGS path to point to cis_studio_truststore_strong.jks

2.5.5. Option 3: PDTool is installed stand-alone with no CIS Server or Studio present on the machine.

2.5.5.1. Unzip the Strong Encryption to any folder.

2.5.5.1.1. E.g. /opt/Composite/StrongEncryption/conf/studio/security/cis_studio_truststore_strong.jks

2.5.5.2. Modify the CERT_ARGS path to point to cis_studio_truststore_strong.jks

2.5.5.3. export CERT_ARGS="-Djavax.net.ssl.trustStore=/opt/Composite/StrongEncryption/conf/studio/security/cis_studio_truststore_strong.jks -Djavax.net.ssl.trustStorePassword=changeit"

3. **CIS Modifications**

3.1. Disable HTTP (if it hasn't been done already) in the Configuration Panel under "Composite Server" → "Web Services Interface" → "Communications" → "HTTP" → "Disable HTTP (On server restart)" → Check “true” to disable.

3.2. If using CIS at version 6.2.4.00.45 or lower:

3.2.1. Shut down CIS.

3.2.2. Update to the latest CIS 6.2 patch.

3.2.3. Restart CIS.

3.3. If already at 6.2.4.00.46 or higher and patching is not desired:

3.3.1. If CIS has not been restarted since HTTP was disabled, restart CIS so the new setting will take effect.

3.3.2. Run the following command from a command prompt in the <CIS_HOME> folder (update the base port from 9400 to your instance's base port if necessary. Do NOT specify the HTTPS port, 9402; the -encrypt flag tells server_util to do this for you):

3.3.3. Windows: bin\server_util.bat -server localhost -port 9400 -user admin -password <admin_pw> -resetNamespace -encrypt

3.3.4. Linux: . ./bin/server_util.sh -server localhost -port 9400 -user admin -password <admin_pw> -resetNamespace -encrypt

4. **PDTool Configuration:**

4.1. For the Archive module, each module entry where HTTPS needs to be used requires the <encrypt> flag to be turned on in the ArchiveModule.xml file.

4.1.1. See /resources/modules/ArchiveModule.xml

4.2. For all other modules, each server in servers.xml that will be using HTTPS will need to have the <useHttps> flag turned on.

4.2.1. See /resources/modules/servers.xml

5. Known Issues

5.1. The deploy.properties parameter CIS_PING_SERVER must be set to “false” or an exception will be thrown when connecting to CIS over SSL (https).

CIS_PING_SERVER: Allows the user to control whether the server performs a ping on CIS prior to executing the actual command.

The default is true if this variable is not set or does not exist. This will override all settings.

CIS_PING_SERVER=false

5.2. Executing Composite Web Services over HTTP/S

5.2.1. The following error occurs when trying to access a new Composite web service over HTTPS and the correct security policy is not set.

Error: 500 Internal Server Error
Description: The requested resource must not be accessed using the HTTP/S protocol.

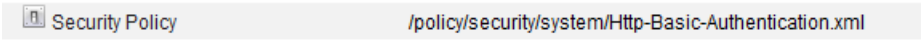
Example of an error using Regression Module to execute the web service:

The regression input file encrypt=false has been overridden by useHttps=true for path=/soap11/testWebService00_NoParams_bare

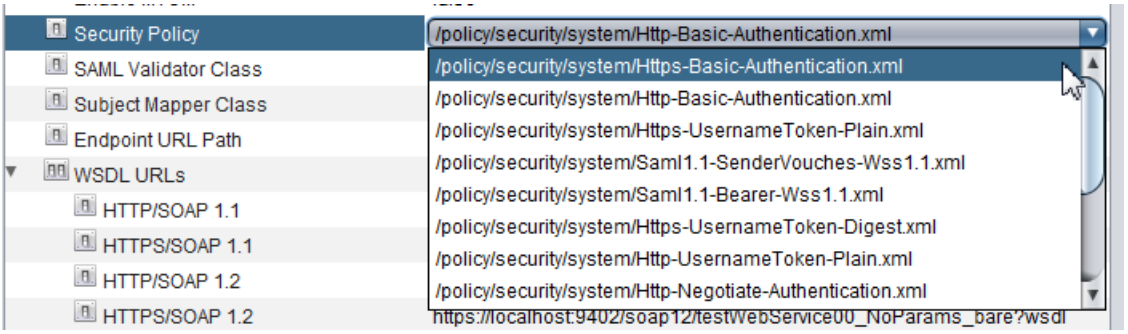
2014-02-03 20:59:44,050 main ERROR [com.cisco.dvbu.ps.deploytool.dao.jdbcapi.RegressionPubTestJdbcDAOImpl] - <executeWs(): Server returned HTTP response code: 401 for URL: https://localhost:9402/soap11/testWebService00_NoParams_bare DETAILED_MESSAGE=[executeWs(): <?xml version="1.0" encoding="ISO-

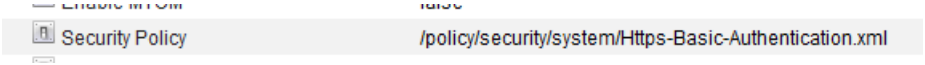
8859-1"?> <html> <head/> <body> COMPOSITE INFORMATION SERVER <hr style="border-style: groove;"/>   Error: **401 Unauthorized**   **Description:** **The requested resource must not be accessed using the HTTP/S protocol.** </body> </html>]>

5.2.2. **Issue**: The security policy is Http-Basic-Authentication



5.2.3. **Resolution**: Change the security policy to Https-Basic-Authentication





CONFIGURE PDTOOL HTTP PROXY

How to configure PDTool HTTP Proxy

This section discusses the procedure for configuring an HTTP proxy for PDTool 6.2.

Configuring the HTTP Proxy for PDTool

The following steps outline what a deployment administrator or developer should do to modify PDTool to use an HTTP Proxy.

1. **PDTool Modifications for Windows**

   1.1. Edit <PDTOOOL_HOME>\bin\setVars.bat

   1.2. Find the section labeled: PDTool Proxy Settings

   1.3. Set the HTTP proxy settings for PDTool. Determine if a proxyUser and proxyPassword is required and set accordingly. The following line provides an example:

   ```
   set HTTP_PROXY=-DproxySet=true -Dhttp.proxyHost=wwwproxy.mydomain.com -Dhttp.proxyPort=80 -Dhttp.proxyUser=mydomain\myuser -Dhttp.proxyPassword=mypassword
   ```

2. **PDTool Modifications for Linux**

   2.1. Edit <PDTOOOL_HOME>/bin/setVars.sh

   2.2. Find the section labeled: PDTool Proxy Settings

   2.3. Set the HTTP proxy settings for PDTool. Determine if a proxyUser and proxyPassword is required and set accordingly. The following line provides an example:

   ```
   export HTTP_PROXY="-DproxySet=true -Dhttp.proxyHost=wwwproxy.mydomain.com -Dhttp.proxyPort=80 -Dhttp.proxyUser=mydomain\myuser -Dhttp.proxyPassword=mypassword"
   ```

UPGRADING PDTOOL

How to upgrade an instance of PDTool

This section discusses the procedure for upgrading an instance of PDTool to a new one.

Upgrade an existing install

The following steps outline what a deployment administrator or developer should do to upgrade PDTool from one version to another. Regarding the instructions below, wherever PDTool62 or PDTool-6.2 is referenced, this may be interchanged with PDTool61 and PDTool-6.1 when working with a PDTool 6.1 or lower installation. For ease of documentation, 6.2 will be used here.

3. **Rename old PDTool directory (previous release)**

3.1. Rename the original PDTool or PDTool62 directory to PDTool62-YYYY-MM-DD where YYYY-MM-DD represents the actual release of the previous version as found in PDTool-6.2-YYYY-MM-DD.r1.txt.

4. **Install new version of PDTool**

4.1. Copy the PDTool zip file with the pattern "PDTool-6.2-YYYY-MM-DD.r1.zip" to the root folder where you want to unzip.

4.1.1. OPTION 1:

Install the new version by unzipping the PDTool zip file using "WinZip" or "7-Zip"

If you using WinZip or 7-Zip then select the option to "Extract Here"

Another option is to leave the zip file where it is and simply browse to the root folder where you want to create the "PDTool62" sub-folder.

4.1.2. OPTION 2:

Install the new version by unzipping the PDTool zip file using "Extract All"

If using Windows "Extract All", make sure that you remove the folder name at the end corrensponding with this pattern: \PDTool-6.2-YYYY-MM-DD.r1

The Extract All will unzip into the root of the directory where the zip file lives and create the sub-folder "PDTool62".

Another option is to leave the zip file where it is and simply browse to the root folder where you want to create the "PDTool62" sub-folder.

5. **Upgrade Module XML Files**

5.1. Save the new PDTool62/resources/modules/*.xml to a sub-folder so these files are not overwritten

5.2. Copy PDTool62-YYYY-MM-DD/resources/modules/*.xml PDTool62/resources/modules

5.3. Edit XML files as needed or as directed by the release notes to take advantage of new functionality

6. **Upgrade Deployment Plan Files**

6.1. Save the new PDTool62/resources/plans/*.dp to a sub-folder so these files are not overwritten

6.2. Copy PDTool62-YYYY-MM-DD/resources/plans/*.dp PDTool62/resources/plans

6.3. Edit .dp files as needed or as directed by the release notes to take advantage of new functionality

7. **Upgrade Ant Build Files**

7.1. Save the new PDTool62/resources/ant/*.xml to a sub-folder so these files are not overwritten

7.2. Copy PDTool62-YYYY-MM-DD/resources/ant/*.xml PDTool62/resources/ant

7.3. Edit .xml files as needed or as directed by the release notes to take advantage of new functionality

8. **Upgrade Configuration Property Files**

8.1. Save the new PDTool62/resources/config/*.* to a sbu-folder so these files are not overwritten

8.2. Copy PDTool62-YYYY-MM-DD/config/*.properties PDTool62/config

8.3. Copy PDTool62-YYYY-MM-DD/config/*.xml PDTool62/config

8.4. Edit .properties files as needed or as directed by the release notes to take advantage of new functionality

9. **Upgrade Configuration Property Files**

9.1. Copy any customized batch/shell script files from PDTool62-YYYY-MM-DD/bin to PDTool62/bin

9.2. Review any differences between the old and new setVars.bat.

9.2.1. If no differences, simply copy PDTool62-YYYY-MM-DD/bin/setVars.bat or .sh to PDTool62/bin

9.2.2. If there are differences, then copy your modifications to the same environment variables in the new setVars.bat or .sh. Open both the old and new setVars.bat and copy the environment variable settings as needed.

VERSION CONTROL SYSTEM INSTALLATION

Version Control System (VCS) Installation and Configuration

The PDTool contains a version control module called VCSModule and is used for integrating with version control systems. This is an optional module and not required to be configured if you are not using any version control system. VCS integration with Subversion (svn), Perforce (p4) and Concurrent Versions System (cvs) are provided out-of-the-box. If other version control systems are being used, then the VCS Module will need to be enhanced to support that VCS. Please contact Composite Professional Services for a PS engagement.

All documentation regarding setup and configuration of the VCSModule can be found in PDTool/docs. Please refer to the following documents:

- **Composite PS Promotion and Deployment Tool Module - Version Control System.pdf**

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***".