



# PDTool Installation Guide

## An Open Source Asset for use with TIBCO® Data Virtualization

**TIBCO Software** empowers executives, developers, and business users with Fast Data solutions that make the right data available in real time for faster answers, better decisions, and smarter action. Over the past 15 years, thousands of businesses across the globe have relied on TIBCO technology to integrate their applications and ecosystems, analyze their data, and create real-time solutions. Learn how TIBCO turns data—big or small—into differentiation at [www.tibco.com](http://www.tibco.com).

<b>Project Name</b>	AS Assets PDTool (Promotion and Deployment Tool)
<b>Document Location</b>	This document is only valid on the day it was printed. The source of the document will be found in the PDTool and PDToolRelease folder ( <a href="https://github.com/TIBCOSoftware">https://github.com/TIBCOSoftware</a> )
<b>Purpose</b>	User's Guide



[www.tibco.com](http://www.tibco.com)

Global Headquarters  
3303 Hillview Avenue  
Palo Alto, CA 94304

**Tel:** +1 650-846-1000  
+1 800-420-8450  
**Fax:** +1 650-846-1005

## Revision History

Version	Date	Author	Comments
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.
4.0	12/14/2017	Mike Tinius	Initial revision with Tibco
4.1	06/11/2018	Mike Tinius	Removed references to .compositesw folder.
5.0	08/27/2020	Mike Tinius	Updated documentation

## Related Documents

Name	Author
PDTool Datasheet.pdf	Mike Tinius
PDTool User's Guide.pdf	Mike Tinius
PDTool Lab Guide.pdf	Mike Tinius
PDTool Module - Archive.pdf	Mike Tinius
PDTool Module - DataSource.pdf	Mike Tinius
PDTool Module - Group.pdf	Mike Tinius
PDTool Module - Privilege.pdf	Mike Tinius
PDTool Module - Rebind.pdf	Jerry Joplin
PDTool Module - Regression.pdf	<i>Sergei Sternin</i>
PDTool Module - Resource Cache.pdf	Mike Tinius
PDTool Module - Resource.pdf	Mike Tinius
PDTool Module - Server Attribute.pdf	Mike Tinius
PDTool Module - Server Manager.pdf	Gordon Rose
PDTool Module - Trigger.pdf	Kevin O'Brien

PDTool Module - User.pdf	Mike Tinius
PDTool Module - Version Control System.pdf	Mike Tinius
PDTool Developer's Guide.docx	Gordon Rose
01 PDTool Training - Introduction v2.0.pptx	Mike Tinius
03 PDTool Training - Deployment v2.0.pptx	Mike Tinius
04 PDTool Training - Testing v2.0.pptx	Mike Tinius
05 PDTool Training - VCS Multi-Tenant v2.0.pptx	Mike Tinius

## Supported Versions

Name	Version
TIBCO® Data Virtualization	7.0.8 or later

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
	Purpose .....	5
	Audience.....	5
	OS Platforms Supported .....	6
	Data Virtualization Versions Supported .....	6
	Version Control Systems Supported .....	6
	References .....	6
<b>2</b>	<b>Distribution.....</b>	<b>7</b>
	Combined Command-Line and Ant Distribution .....	7
	PDTool Contents .....	7
<b>3</b>	<b>Installation .....</b>	<b>8</b>
	Combined Command Line and Ant Script Installation .....	8
	System Requirements .....	9
	Installation – Automated Windows Installer (new style of installation) .....	9
	Manual Installation Overview (old style of installation).....	14
	Manual Installation and Configuration Details (old style of insallation) .....	14
<b>4</b>	<b>Configure PDTool Over SSL (HTTPS) .....</b>	<b>24</b>
	How to configure PDTool over SSL .....	24
	Configuring PDTool Over SSL .....	24
<b>5</b>	<b>Configure PDTool HTTP Proxy .....</b>	<b>29</b>
	How to configure PDTool HTTP Proxy.....	29
	Configuring the HTTP Proxy for PDTool .....	29
<b>6</b>	<b>Upgrading PDTool .....</b>	<b>30</b>
	How to upgrade an instance of PDTool .....	30
	Upgrade an existing install .....	30
<b>7</b>	<b>Version Control System Installation .....</b>	<b>33</b>
	Version Control System (VCS) Installation and Configuration .....	33
<b>8</b>	<b>Conclusion .....</b>	<b>34</b>
	Concluding Remarks .....	34
	How you can help! .....	34

# 1 Introduction

## Purpose

The purpose of this document is to provide guidance on how to install the “**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 Data Virtualization (DV) resources:

1. **User** – PDTool User's Guide.pdf
2. **Training** – LabPD-DeployTool.pdf
3. **Modules** – Modularized functionality within the Promotion and Deployment Tool
  - PDTool Module - Archive.pdf
  - PDTool Module - DataSource.pdf
  - PDTool Module - Group.pdf
  - PDTool Module - Privilege.pdf
  - PDTool Module - Rebind.pdf
  - PDTool Module - Regression.pdf
  - PDTool Module - Resource Cache.pdf
  - PDTool Module - Resource.pdf
  - PDTool Module - Server Attribute.pdf
  - PDTool Module - Server Manager.pdf
  - PDTool Module - Trigger.pdf
  - PDTool Module - User.pdf
  - PDTool Module - Version Control System.pdf

For Developers building new modules to snap into the tool:

1. **Development** – PDTool Developer's Guide.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 DV 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: Windows10, Windows 7

UNIX Platforms: Linux 6 and 7

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

Java:

JRE 1.7 and 1.8 for command-line deployment

JDK 1.7 and 1.8 for Ant deployment

## Data Virtualization Versions Supported

Versions: Data Virtualization (DV) 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
- Team Foundation Server (TFS) 2010 – (TFS) 2012
- GIT

## References

Product references are shown below. Any references to CIS or DV refer to the current TIBCO® Data Virtualization.

- TIBCO® Data Virtualization was formerly known as
  - Cisco Data Virtualization (DV)
  - Composite Information Server (CIS)

## 2 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 DV 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 DV using SSL.

## 3 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 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?

- 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%\PDTool[7.0.0]\_[NOVCS|TFS|SVN|GIT|P4] 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

#### Audience?

- For DV Developers who want to deploy resources from a version control system (VCS) such as TFS or Subversion to a target DV server.
- For DV 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 DV 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 DV, 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 “***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.
- 2.3.2. Zip up the PDTool7.0\_Installer directory and post somewhere on the organizations site for developers to download and install.

### 2.4. Pre-configure configuration deployment property 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 DV 7.0 – “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\PDTool7.0\_Installer

### 3.3. Install 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] 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 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 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%\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\PDToolInstall7.0.0

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%\PDTool7.0.0\_NOVCS
- TFS: C:\Users\%USERNAME%\PDTool7.0.0\_TFS
- SVN: C:\Users\%USERNAME%\PDTool7.0.0\_SVN
- GIT: C:\Users\%USERNAME%\PDTool7.0.0\_GIT
- P4: C:\Users\%USERNAME%\PDTool7.0.0\_P4

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]

Enter I\_VCS\_HOME=[%DEF\_VCS\_HOME%]:

- This is the location of the VCS script executable. E.g.  
C:\Users\%USERNAME%\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 DV.

Enter I\_CIS\_DOMAIN=[%DEF\_CIS\_DOMAIN%]:

- This is the DV domain which is used by the I\_CIS\_USERNAME to connect to DV. 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 DV 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 installation)

1. **Project Folder** – Select a project folder preferably with no spaces.
  - 1.1. **Key definitions**
    - 1.1.1. `<ver>=7.0`
    - 1.1.2. `<vcs>=NOVCS, SVN, TFS, GIT, P4`
  - 1.2. **Windows** – e.g. `c:\users\%USERNAME%\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%\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 DV sections:

```
REM #####
REM # GENERAL GROUP ENVIRONMENT
```

```

REM #####
REM # My Java Home
set MY_JAVA_HOME=
REM # PDTool Substitute Drive Letter used for making the path shorter.
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 PDTool/resources/config
REM #   e.g. Default=deploy.properties or SVN=deploy_SVN.properties or
TFS=deploy_TFS.properties
set MY_CONFIG_PROPERTY_FILE=
REM #####
REM # DV SERVER VARIABLES
REM #####
REM # 0=Do not print this section, 1=Print this section
set CIS_PRINT=1
REM # DV Server Username
set CIS_USERNAME=
REM # DV Server Password
set CIS_PASSWORD=
REM # DV Server Domain
set CIS_DOMAIN=

```

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

```

REM #####
REM # [TFS|SVN|GIT|P4] VARIABLES
REM #####
REM # 0=Do not print this section, 1=Print this section
set [TFS|SVN|GIT|P4]_PRINT=1
REM # The VCS Home folder where VCS client exists
set [TFS|SVN|GIT|P4]_HOME=
REM # The VCS repository path at trunk or any folder designation within trunk
set [TFS|SVN|GIT|P4]_VCS_REPOSITORY_URL=
REM # The VCS folder path starting directly after the VCS repo URL and ending
where the DV base level root folders start
set [TFS|SVN|GIT|P4]_VCS_PROJECT_ROOT=
REM # VCS user name
set [TFS|SVN|GIT|P4]_VCS_USERNAME=
REM # VCS user password. See notes at top of this file to encrypt.
set [TFS|SVN|GIT|P4]_VCS_PASSWORD=
REM # Set the VCS Workspace name.
set [TFS|SVN|GIT|P4]_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 DV user list in one invocation. This option requires
storing all DV 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 DV 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 DV 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 DV Resource Type.
#   /services/webservices/system/admin/resource/operations/getResource()

```

```
#
# The DV 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 DV 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
# DV 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 **"PDTool 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] are only set when using VCS deployment. This entire section is detailed out in the document **"PDTool 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
#####
...
```

## 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.tibco.ps.deploytool.CisDeployTool=INFO
# PDTool Module Debugging for Command-line and Ant (Applies to *ALL* Modules)
log4j.logger.com.tibco.ps.deploytool=INFO
# Common Utility Framework Debugging includes:
# [CommonUtils, PropertyManager, PropertyUtil, ScriptExecutor, XMLUtils, JdbcConnector]
log4j.logger.com.tibco.ps.common.util=INFO
# VCS Diffmerger and Archive Services Debugging
log4j.logger.com.tibco.cmdline=INFO
# DV 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.tibco.ps.common.scriptutil=ERROR
```

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

9.1. This entire section is detailed out in the document “**PDTool Module - Version Control System.pdf**”

## 4 Configure PDTool Over SSL (HTTPS)

### How to configure PDTool over SSL

This section discusses the procedure for configuring both PDTool 7.0 and the target DV server. Only PDTool 7.0 supports the ability for PDTool to connect to DV over SSL using the https protocol. The minimum requirement for DV is to be patched to DV 7.0.0.00.00.

### Configuring PDTool Over SSL

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

#### 1. PDTool Modifications for Windows

- 1.1. Edit <PDTOOL\_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 TrustStore which ships with PDTool 7.0 and DV.
- 1.4. **Important:** If you your DV Server has the Strong Encryption Pack installed, you must configure PDTool to use the Strong Encryption Pack or you may see this error:

```
com.tibco.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 requested 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 Support.
  - 1.5.2. Determine which option best identifies your scenario:
  - 1.5.3. Option 1: PDTool is installed on a server with DV 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 DV 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 DV 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 <PDTOOL\_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 7.0 and DV.
- 2.4. **Important:** If you your DV Server has the Strong Encryption Pack installed, you must configure PDTool to use the Strong Encryption Pack or you may see this error:  
  
com.tibco.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 requested 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 Support.
  - 2.5.2. Determine which option best identifies your scenario:
  - 2.5.3. Option 1: PDTool is installed on a server with DV 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 DV 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 DV 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/DV/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/DV/StrongEncryption/conf/studio/security/cis\_studio\_t  
ruststore\_strong.jks -Djavax.net.ssl.trustStorePassword=changeit"

### 3. DV Modifications

- 3.1. Disable HTTP (if it hasn't been done already) in the Configuration Panel under  
"Server" → "Web Services Interface" → "Communications" → "HTTP" → "Disable  
HTTP (On server restart)" → Check "true" to disable.
- 3.2. If already at 7.0.0.00.00 or higher and patching is not desired:
  - 3.2.1. If DV has not been restarted since HTTP was disabled, restart DV so the new  
setting will take effect.
  - 3.2.2. 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):
  - 3.2.3. Windows: bin\server\_util.bat -server localhost -port 9400 -user admin -  
password <admin\_pw> -resetNamespace -encrypt
  - 3.2.4. 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 DV over SSL (https).

# CIS\_PING\_SERVER: Allows the user to control whether the server performs a ping on DV 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 DV Web Services over HTTP/S

5.2.1. The following error occurs when trying to access a new DV 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.tibco.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> <div style="font-family: sans-serif; color:

#990000; margin-top: 5px; margin-bottom: 5px; text-align: center">COMPOSITE

INFORMATION SERVER <hr style="border-style: groove;"/> </div> <div

style="font-family: sans-serif; "> <b>Error: </b>**401 Unauthorized** </div>

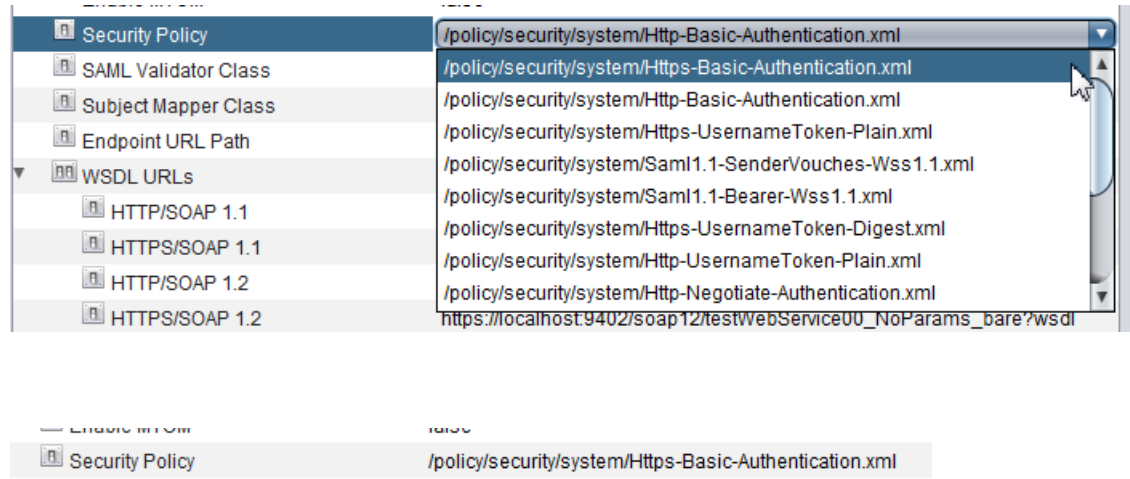
<div style="font-family: sans-serif; "> <b>**Description:** </b>**The requested**

**resource must not be accessed using the HTTP/S protocol.** </div> </body>  
</html> ]>

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

Security Policy /policy/security/system/Http-Basic-Authentication.xml

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



## 5 Configure PDTool HTTP Proxy

### How to configure PDTool HTTP Proxy

This section discusses the procedure for configuring an HTTP proxy for PDTool 7.0.

### 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"
```

## 6 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 PDTool70 or PDTool-7.0 is referenced, this may be interchanged with PDTool62 and PDTool-6.2 when working with a PDTool 6.2 or lower installation. For ease of documentation, 7.0 will be used here.

#### 1. Rename old PDTool directory (previous release)

- 2.4. Rename the original PDTool or PDTool70 directory to PDTool70-YYYY-MM-DD where YYYY-MM-DD represents the actual release of the previous version as found in PDTool-7.0-YYYY-MM-DD.r1.txt.

#### 3. Install new version of PDTool

- 3.1. Copy the PDTool zip file with the pattern "PDTool-7.0-YYYY-MM-DD.r1.zip" to the root folder where you want to unzip.

##### 3.1.1. 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 "PDTool70" sub-folder.

##### 3.1.2. 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 corresponding with this pattern: \PDTool-7.0-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 "PDTool70".

Another option is to leave the zip file where it is and simply browse to the root folder where you want to create the "PDTool70" sub-folder.

#### 4. Upgrade Module XML Files

4.1. Save the new PDTool70/resources/modules/\*.xml to a sub-folder so these files are not overwritten

4.2. Copy PDTool70-YYYY-MM-DD/resources/modules/\*.xml  
PDTool70/resources/modules

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

## 5. Upgrade Deployment Plan Files

5.1. Save the new PDTool70/resources/plans/\*.dp to a sub-folder so these files are not overwritten

5.2. Copy PDTool70-YYYY-MM-DD/resources/plans/\*.dp PDTool70/resources/plans

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

## 6. Upgrade Ant Build Files

6.1. Save the new PDTool70/resources/ant/\*.xml to a sub-folder so these files are not overwritten

6.2. Copy PDTool70-YYYY-MM-DD/resources/ant/\*.xml PDTool70/resources/ant

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

## 7. Upgrade Configuration Property Files

7.1. Save the new PDTool70/resources/config/\*.\* to a sub-folder so these files are not overwritten

7.2. Copy PDTool70-YYYY-MM-DD/config/\*.properties PDTool70/config

7.3. Copy PDTool70-YYYY-MM-DD/config/\*.xml PDTool70/config

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

## 8. Upgrade Configuration Property Files

8.1. Copy any customized batch/shell script files from PDTool70-YYYY-MM-DD/bin to PDTool70/bin

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

- 8.2.1. If no differences, simply copy PDTool70-YYYY-MM-DD/bin/setVars.bat or .sh to PDTool70/bin
- 8.2.2. 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.



## 7 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 Git, Subversion (svn), Perforce (p4) and Team Foundation Server (TFS) 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 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:

- **PDTool Module - Version Control System.pdf**

## 8 Conclusion

### Concluding Remarks

The Promotion and Deployment Tool is a set of pre-built modules intended to provide a turn-key experience for promoting DV resources from one DV 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 Professional Services for the advancement of the ***"Promotion and Deployment Tool"***.