



PDTool Lab 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.

Project Name	AS Assets PDTool (Promotion and Deployment Tool)
Document Location	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 (https://github.com/TIBCOSoftware)
Purpose	User's Guide



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	08/15/2012	Mike Tinius	Initial revision
3.0	8/21/2013	Mike Tinius	Updated docs to Cisco format
3.1	2/18/2014	Mike Tinius	Prepare docs for open source.
3.2	11/17/2014	Mike Tinius	Updated license.
3.3	3/4/2015	Mike Tinius	Updated docs to Cisco format.
4.0	12/14/2017	Mike Tinius	Initial revision with Tibco

Related Documents

Name	Author
PDTool User's Guide.pdf	Mike Tinius

Supported Versions

Name	Version
TIBCO® Data Virtualization	7.0.4 or later

Table of Contents

1	Introduction.....	4
	Purpose.....	4
	Practice Goals.....	4
	Supported Platforms	4
	Audience	4
	References.....	4
2	Pre-Requisites and Assumptions	5
	Pre-Requisites	5
	Assumptions	5
3	PDTool High-Level Steps	7
	Flow Diagram.....	7
4	Procedure	8
5	Appendix A: Known Errors.....	15

1 Introduction

Purpose

The Promotion and Deployment Tool (PDTool) is a new tool offered through Professional Services engagements that gives customers an easy way of defining and executing a Data Virtualization (DV) resource promotion process. Promotion is the concept of moving DV resources from one environment to another and then performing various configurations on the target DV environment. Allow 2.5 hours for this lab which includes installation and configuration of PDTool.

Practice Goals

We will use the Promotion and Deployment Tool for executing a command line deployment of artifacts to a local Data Virtualization (DV) 7.0 server instance. If available a subversion VCS server will be used for checking out DV resources and importing into the local DV instance.

Supported Platforms

Data Virtualization 7.0 server instance.

Subversion 1.6 or higher

Audience

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

- Architects
- Developers
- Administrators
- Operations personnel

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 Pre-Requisites and Assumptions

Pre-Requisites

- Write down your assigned Student Number “NN”: _____ (e.g. 01 – 10)
 - This number will be used to refer the car file the student will import as well as various references to “NN” within the lab procedure steps.
- DV Topologies supported in this lab
 - *Local*: DV Studio and Server installed
 - *Multi-user*: Local Studio and Central Development Server installed
- DV examples “/shared/examples” must exist.
- For one section of the lab, a Version Control System (Subversion) server must exist. If one is not available, an alternate approach of importing CAR files can be used.
 - A VCS client such as subversion must be installed on the computer where the lab will be performed. TortoiseSVN 1.7.1 or higher may be used if the command line client is installed. Previous versions did not have a command line client and thus could not be used.
- Install the Promotion and Deployment Tool
 - Follow the instructions in “**PDTool Installation Guide.pdf**” to install the PDTool.
- Configure VCS Module [optional – based on VCS server availability]
 - Follow the instructions in “**PDTool Module - Version Control System.pdf**” to configure VCS for PDTool.
 - Go to the section titled “**Configuring Version Control for PDTool**”.
 - If you are running a “Multi-User/Central Dev Server” topology make sure the deploy.properties has the property set to true: VCS_MULTI_USER_TOPOLOGY=true.
 - Make sure you execute the VCS Workspace Initialization command or your lab will not work properly or the Forced Checkin in Step 3 will not work.

Assumptions

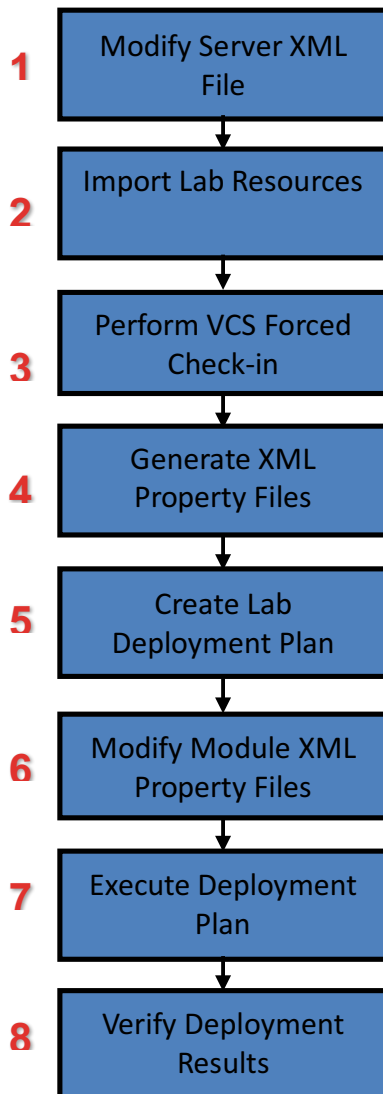
Any references to “*Location: PDTool/*” within the procedure steps will be assumed that the student understands how to use Windows Explorer or a Command Line Window to navigate to the location of PDTool and the path specified within “*Location:*”.

Starting with the 2015-03-31 release and higher, the default folder for PDTool is “PDTool7.0.0” for DV 7.0. The subfolder is called PDTool. Any reference to the location “PDTool/” should be inferred as the location of the subfolder “PDTool”.

3 PDTool High-Level Steps

Flow Diagram

The following high-level steps provide a road-map for the student as to what they will accomplish in this lab.



Lab Deployment Plan Steps

The following deployment plan shall be implemented as steps and actions within the orchestration property file

1. VCS Check Out from or Package Export if subversion not available
 - a. /services/databases/TESTNN
 - b. /services/webservices/test/testWebServiceNN
 - c. /shared/testNN
2. Create or Update Groups
 - a. group1
 - b. group2
3. Create or Update Users
 - a. user1
 - b. user2
4. Update Data Source
 - a. XML File
 - b. Database
 - c. Web Service
5. Update Privileges
 - a. /shared/testNN

4 Procedure

1 Modify the **servers.xml** property file [5 min]

Background: The servers.xml file contains the connection information to Data Virtualization (DV). It needs to be updated before any other steps can be completed.

Location: PDTool/resources/modules

1. **Open** servers.xml in a text editor
2. **Modify** the **id**: "**labhost**" to match your environment (port, user, password, cishome)
3. Save and close
4. If the password was modified to clear text then run this utility
5. Open a command line window and change directories to PDTool/bin
 - a. **Execute:**
`ExecutePDTool.bat -encrypt ../resources/modules/servers.xml`
 - b. Notice how the clear text passwords were changed to Encrypted: passwords
6. Leave the command window open. You will be using it again.

2 Execute LabPD-Deploy.dp to perform a **package import** [10 min]

Background: Package import allows the student to import the specific DV resources that will be used by this lab.

Location: PDTool/resources/plans

1. **Modify LabPD-Deploy.dp**
 - a. Uncomment the pkg_import line.
 - b. Modify "test00" to use your student number "testNN"
 - c. Everything else should be commented out.
 - d. Please note how the variable \$LAB_SERVERID is being used to indicate the serverId in the servers.xml file. This variable is set in deploy.properties as LAB_SERVERID=labhost. The variable is resolved upon execution. The reason for using variables is that it provides the user a "single" place to make a change so that all actions are performed against that designated server identifier. If a variable is not used, then the user would have to change all entries in the deployment orchestration property file. This is a good best practice going forward.
2. Use the command line window opened in Step 1 at PDTool/bin
 - a. **Execute:**
`ExecutePDTool.bat -exec ../resources/plans/LabPD-Deploy.dp`
 - b. Note: You may receive the error shown in Appendix A: Known Errors.

3. Open DV Studio and verify the following were imported:

- a. **NN** is your student number
- b. Databases: TEST**NN**
- c. Web Services: testWebService**NN**
- d. Folder: /shared/test**NN**

4. Using Studio, modify testWebService URL

- a. Open /shared/test**NN**/DataSources/testWebService
 - i. Provide the proper **Port** in the URL:
 1. [http://localhost:9400/services/testWebService**NN**?wsdl](http://localhost:9400/services/testWebServiceNN?wsdl)
 - ii. Provide the proper User/Password for your environment.

5. Leave Studio open. You will be using it again.

3 Execute LabPD-Deploy.dp to perform a **VCS Forced Checkin** [20 min]

Background: Check in the student's DV resources to the VCS server.

Assumptions: The student has configured deploy.properties for the proper VCS communication as per instructions found in the pre-requisites.

Location: PDTool/resources/plans

1. **Edit LabPD-Deploy.dp**

- a. Comment out the pkg_import line
- b. Uncomment the three vcsForcedCheckins lines

2. **Edit PDTool/resources/modules/lab/LabVCSModule.xml**

- a. **Modify references to 00 and change to NN (the student's number).**
- b. `<resourcePath>/services/databases/TEST00</resourcePath>`
- c. `<resourcePath>/services/webservices/testWebService00</resourcePath>`
- d. `<resourcePath>/shared/test00</resourcePath>`

3. Use the command line window opened in Step 1 at PDTool/bin

- a. **Execute:**
ExecutePDTool.bat -exec ..\resources\plans\LabPD-Deploy.dp

4. Verify that all three lines executed successfully (PASS) by reviewing the console output or app.log.

5. **Comment out** the three vcsForcedCheckin lines

4 Execute the deployment orchestration property file to “**Generate XML**” property files [5 min]

Background: Generate the property files for the various modules as a starting point.

Location: PDTool/resources/plans

1. **Edit** PDTool/resources/plans/LabPD-Generate.dp generation process
 - a. **Modify** all references to “00” and change to your student number “NN”
2. Use the command line window opened in Step 1 at PDTool/bin
 - a. **Execute:**
`ExecutePDTool.bat -exec ..\resources\plans\LabPD-Generate.dp`
3. Review the files generated in PDTool/resources/modules/lab
 - a. getLabGroupModule.xml
 - b. getLabUserModule.xml
 - c. getLabDataSourceModule.xml
 - d. getLabPrivilegeModule1.xml

5 Create a deployment orchestration plan file to “Execute” the deployment plan [10 min]

Location: PDTool/resources/plans

1. **Edit** LabPD-Deploy.dp
2. **Comment out** the pkg_import line if it is not already done.
3. **Comment out** the three vcsForcedCheckin lines if they are not already done.
4. **Copy** this orchestration process to the LabPD-Deploy.dp file
 - a. **Caution: After pasting the lines in the property file, each execution line will be broken up into 2 lines. Please rejoin the lines so that the execution line is all on a single line otherwise you will get “wrong number of arguments” errors.**

```
# Checkout resources from VCS
PASS TRUE ExecuteAction vcsCheckouts $LAB_SERVERID "testDB,testWS,testNN"
"$MODULE_HOME/lab/LabVCSModule.xml" "$MODULE_HOME/servers.xml"
#
# Create or Update groups
PASS TRUE ExecuteAction createOrUpdateGroups $LAB_SERVERID
"group1,group2" "$MODULE_HOME/lab/LabGroupModule.xml"
"$MODULE_HOME/servers.xml"
#
# Create or Update users
PASS TRUE ExecuteAction createOrUpdateUsers $LAB_SERVERID "user1,user2"
"$MODULE_HOME/lab/LabUserModule.xml" "$MODULE_HOME/servers.xml"
#
# Update Datasource
PASS TRUE ExecuteAction updateDataSources $LAB_SERVERID "ds1,ds2,ds3"
"$MODULE_HOME/lab/LabDataSourceModule.xml" "$MODULE_HOME/servers.xml"
#
```

Update Privileges

```
PASS TRUE ExecuteAction updatePrivileges $LAB_SERVERID "priv1"
"$MODULE_HOME/lab/LabPrivilegeModule.xml" "$MODULE_HOME/servers.xml"
```

6 Modify the Module XML property files [20 min]

Location: PDTool/resources/modules/lab

1. **Review** the existing LabGroupModule.xml **[NO CHANGES]**
 - a. This step will not use the generated file “getLabGroupModule.xml”
 - b. The following groups will get created or updated:
 - i. group1 and group2
 - c. Group 1 access rights:
 - i. privilege: ACCESS_TOOLS
 - d. Group 2 access rights:
 - i. privilege: ACCESS_TOOLS MODIFY_ALL_CONFIG
MODIFY_ALL_RESOURCES MODIFY_ALL_STATUS
MODIFY_ALL_USERS READ_ALL_CONFIG
READ_ALL_RESOURCES READ_ALL_STATUS READ_ALL_USERS
2. **Review** the existing LabUserModule.xml **[NO CHANGES]**
 - a. This step will not use the generated file “getLabUserModule.xml”
 - b. The following users will get created or updated:
 - i. user1 and user2
 - ii. encryptedPassword: password
3. **Edit** the existing LabVCSModule.xml
 - a. Rename all references of “00” to your student number “NN”
 - i. /services/databases/TEST00
 - ii. /services/webservices/testWebService00
 - iii. /shared/test00
4. **Rename** getLabDataSourceModule.xml to LabDataSourceModule.xml
5. **Edit** LabDataSourceModule.xml
 - a. Tweak the properties for the file entry -
/shared/testNN/DataSources/ServerAttributeDefinitions
 - i. id:ds1
 - ii. Make sure the value for the generic attribute <name>root</name>
matches your location

1. `<value>`
`$PROJECT_HOME/PDTool/resources/modules/generated``</value>`
 2. Note: Use forward slashes
 3. Note: \$PROJECT_HOME gets resolved at the time of execution. PROJECT_HOME is a Java environment variable that gets set with ExecutePDTool.bat or .sh.
- b. Tweak the properties for the database entry -
`/shared/testNN/DataSources/ds_orders`
 - i. id:ds2
 - ii. Make sure the values match your environment for port
 - iii. The encrypted password for the user tutorial is tutorial
 - iv. Modify the generic attribute `connPoolMaxSize=110`
 - c. Tweak the properties for the web service entry -
`/shared/testNN/DataSources/testWebService`
 - i. id:ds3
 - ii. Make sure the values for login, password and url are set appropriately for your environment. Check the port number is correct also.
 1. login=user2
 2. password=password
 3. url=http://localhost:9400/services/testWebServiceNN?wsdl
6. **Rename** getLabPrivilegeModule1.xml to LabPrivilegeModule.xml
 7. **Edit** LabPrivilegeModule.xml,
 - a. **Verify** “priv1” /services/webservices/testWebServiceNN has your “NN” student number.
 - b. For all privileges (**priv1**) replace all `<privilege>` entries with the following:


```
<privilege>
  <name>group1</name>
  <nameType>GROUP</nameType>
  <domain>composite</domain>
  <privileges>NONE</privileges>
</privilege>
<privilege>
  <name>group2</name>
  <nameType>GROUP</nameType>
  <domain>composite</domain>
  <privileges>WRITE EXECUTE SELECT</privileges>
</privilege>
```

7 Execute the deployment solution [20 min]

6. **Prepare to execute** by deleting artifacts in Studio
 - a. Open Studio and delete the folders for your student number "NN".
 - i. Databases: /TESTNN
 - ii. Web Services: /services/webservices/testWebServiceNN
 - iii. Folder: /shared/testNN
7. Use the command line window opened in Step 1 at PDTool/bin
 - a. Execution Hint:
 - i. An easy to debug the process is to comment lines of execution so that it makes it easier to focus on one issue at a time.
 - ii. Start with 1 item, get it working and then move on the next. When all items are working individually, uncomment all action lines and run the entire process.
 - b. **Execute**:
ExecutePDTool.bat -exec ..\resources\plans\LabPD-Deploy.dp
8. Review the log files:

Location: PDTool/logs

 - a. PDTool/logs/summary.log
 - b. PDTool/logs/app.log
9. Known Error:
 - a. When the VCS Module does an import of the checked out resource, it re-introspects the data sources which may throw the error in Appendix A.

8 Validate the deployment solution [15 min]

1. Use DV Studio and DV Manager (browser) to validate
2. Validate the following exist using Studio for your student number "NN":
 - a. Databases: /TESTNN
 - b. Web Services: /services/webservices/testWebServiceNN
 - c. Folder: /shared/testNN
3. Validate the users exist with the correct group membership using the web-based Manager
4. Validate the groups exist with the correct privileges using the web-based Manager
5. Validate the data source attributes are set correctly using Studio
6. Validate the privileges are set correctly using Studio
 - a. Right click on the folders specified above.
 - b. Group1: No privileges are checked
 - c. Group2: All privileges are checked (Read, Write, Execute, Select, Update, Insert, Delete, Grant)

7. Execute the testWebService using **user1/password** to log in.
 - a. Right-click on testWebServiceNN and select "Test Web Service"
 - b. Use testprocsimple('abc',0)
 - c. This should fail with insufficient privileges "401 Unauthorized"
 - d. Close the browser window
8. Execute the testWebService using **user2/password** to log in.
 - a. Use testprocsimple('abc',0)
 - b. This should execute successfully

```
<ns1:testprocsimpleResponse  
xmlns:ns1="com.compositesw/services/webservices/testWebService/testService">  
  <ns1:error xsi:nil="true"/>  
</ns1:testprocsimpleResponse>
```

5 Appendix A: Known Errors

Note: When reading this error, replace any reference to “00” with your lab number “nn”.

Cause: Reintrospection failed for datasource "/shared/test00/DataSources/testWebService".

Impact: Slight. The resources still get imported into DV. The only impact is that this web service won't be executable until it is fixed.

Resolution: Reintrospection failed due to incorrect port number, username or password for this data source.

Fix the port number in the string <http://localhost:9400/services/testWebService01?wsdl> or fix the user name (admin) or the default password for admin and save.

Complete Error:

Reintrospection failed for datasource "/shared/test00/DataSources/testWebService". Reason:

<serverexception>com.compositesw.cdms.datasource.DataSourceException: Error(s) were encountered while parsing the WSDL at

"http://localhost:9400/services/testWebService01?wsdl":

com.compositesw.xml.XmlException:

Error(s) loading "http://localhost:9400/services/testWebService00?wsdl": java.io.IOException:

HTTP

401 Unauthorized. [null-1900002] [datasrc-2990004]

at com.compositesw.cdms.ds.ws.WsIntrospector.loadDefinitions(WsIntrospector.java:199)

at com.compositesw.cdms.ds.ws.WsIntrospector.introspectNode(WsIntrospector.java:115)

at

com.compositesw.cdms.datasource.AbstractIntrospector.introspectAllDescendents(AbstractIntrospector.java:325)

at com.compositesw.cdms.ds.ws.WsIntrospector.reintrospectDataSource(WsIntrospector.java:106)

at

com.compositesw.cdms.datasource.ReintrospectOperation.performOperation(ReintrospectOperation.java:79)

at

com.compositesw.server.common.AbstractAsynchronousOperation\$OperationRunnable.run(AbstractAsynchronousOperation.java:401)

at java.util.concurrent.ThreadPoolExecutor\$Worker.runTask(ThreadPoolExecutor.java:885)

at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:907)

at java.lang.Thread.run(Thread.java:619)

Caused by: com.compositesw.xml.XmlException: Error(s) loading

"http://localhost:9400/services/testWebService00?wsdl": java.io.IOException: HTTP

401 Unauthorized. [null-1900002]

at com.compositesw.xml.wsdl.WsdlReaderJWSDLImpl.readWsdl(WsdlReaderJWSDLImpl.java:85)

at com.compositesw.cdms.ds.ws.WsIntrospector.loadDefinitions(WsIntrospector.java:197)

... 8 more

Caused by: java.io.IOException: HTTP 401 Unauthorized

at com.compositesw.common.net.URLReader.readHttpURL(URLReader.java:344)

at com.compositesw.common.net.URLReader.readURL(URLReader.java:97)

at com.compositesw.common.net.URLReader.readURL(URLReader.java:74)

at com.compositesw.xml.wsdl.WsdlReaderJWSDLImpl.readWsdl(WsdlReaderJWSDLImpl.java:83)

... 9 more

</serverexception>.