



# PDTool Developer's Startup 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>	Developer'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	12/06/2017	Mike Tinius	Initial revision with Tibco

## Related Documents

Name	Version

## Supported Versions

Name	Version
TIBCO® Data Virtualization	7.0.4 or later

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
	Purpose .....	4
	Audience.....	4
	References .....	4
<b>2</b>	<b>Github Repository Structure.....</b>	<b>5</b>
	Recommended Development Tools.....	5
	How Can I Download a Copy of the PDTool and PDToolRelease GitHub Repository? .....	5
	Repository Folder Structure .....	5
<b>3</b>	<b>Configure the Eclipse Development Environment.....</b>	<b>6</b>
	Downloading Eclipse .....	6
	Configure Eclipse Variables .....	7
	Checking Out the PDTool Project .....	9
	Clone the PDTool Git repository to your local machine .....	9
	Create a General project from the Git repository .....	12
	Build the PDTool jar files .....	15
	Debugging PDTool (Java) Source Code .....	16
	Checking Out the PDToolRelease Project .....	18
	Clone the PDToolRelease Git repository to your local machine .....	18
	Create a General project from the Git repository .....	21

# 1 Introduction

## Purpose

The purpose of this document is to provide guidance on how develop the AS Assets PDTool (Promotion and Development Tool).

## Audience

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

- Developers

## 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 Github Repository Structure

### Recommended Development Tools

CIS itself can (and should) be used for developing the SQL Script procedures. Any robust Java IDE can be used to develop the CJP's, however up to this point, the CJP's have been developed using Eclipse. The CJP artifacts in the open source repository (GitHub) are therefore geared towards an Eclipse project.

Git is used as the version control system for the ASAssets open source project. For those new to Git, a visit to <https://help.github.com/articles/set-up-git> will provide an overview of Git and instructions on downloading and setting up the basic Git tools.

### How Can I Download a Copy of the PDTool and PDToolRelease GitHub Repository?

The GitHub master repository is located at <https://github.com/TIBCOSoftware/PDTool>. The PDTool Release master repository is located at <https://github.com/TIBCOSoftware/PDToolRelease>.

### Repository Folder Structure

The folder structure of the GitHub repository contains Java source, documentation source, and distribution resources.

### 3 Configure the Eclipse Development Environment

#### Downloading Eclipse

##### 1. Download the latest Eclipse from here:

<http://www.eclipse.org/platform>

Eclipse IDE for Java Developers (includes Incubating components)

Version: 2020-06 (4.16.0)

Build id: 20200615-1200

(c) Copyright Eclipse contributors and others 2000, 2020. All rights reserved. Eclipse and the Eclipse logo are trademarks of the Eclipse Foundation, Inc., <https://www.eclipse.org/>.

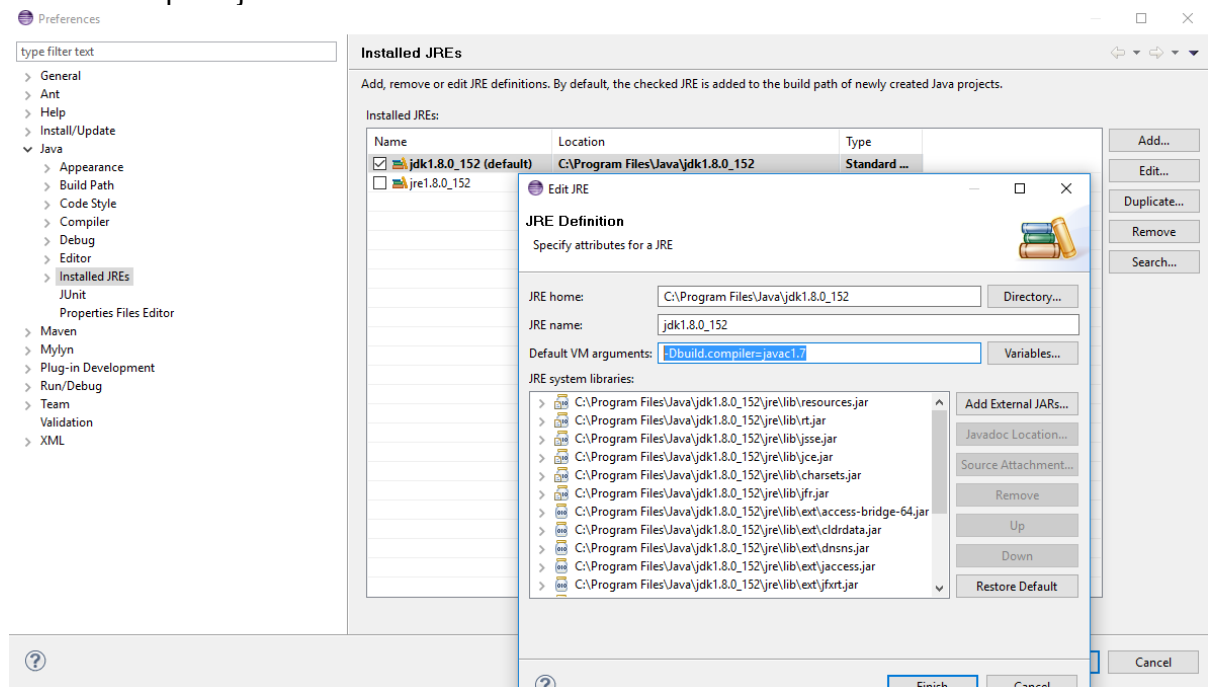
The Eclipse logo cannot be altered without Eclipse's permission. Eclipse logos are provided for use under the Eclipse logo and trademark guidelines, <https://www.eclipse.org/logotm/>. Oracle and Java are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This product includes software developed by other open source projects including the Apache Software Foundation, <https://www.apache.org/>.

Set the Installed JREs Default VM arguments

From eclipse, open Windows → Preferences → Java → Click on Installed JREs → Edit jdk 1.8.0 and set the VM arguments.

-Dbuild.compiler=javac1.7



## 2. Open Eclipse, Install New Software, enter this URL:

<http://eclipse.org/egit/download/>

### Configure Eclipse Variables

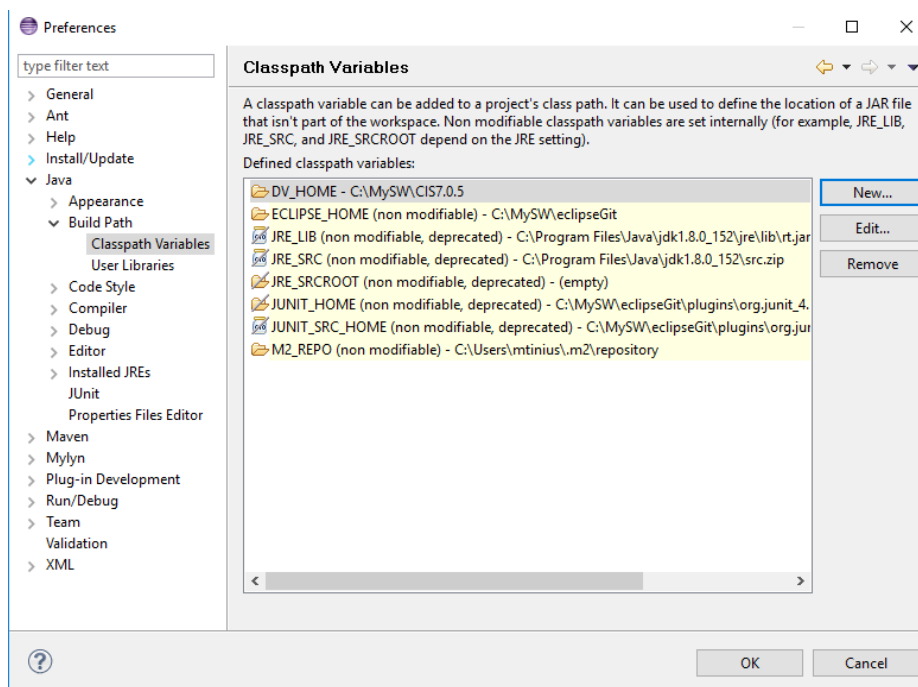
There are two variables that need to be created. One is for Eclipse and one is for Ant.

Create DV\_HOME for Eclipse.

In Eclipse, select Window > Preferences > Java > Build Path > Classpath Variables > New

Enter Name: DV\_HOME

Path: Browse to your data virtualization home folder

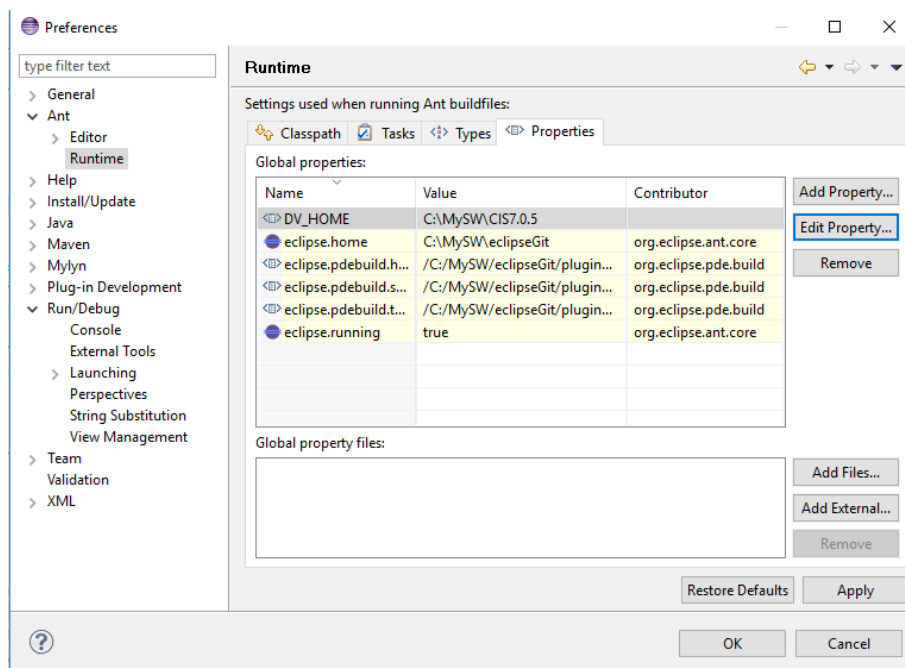
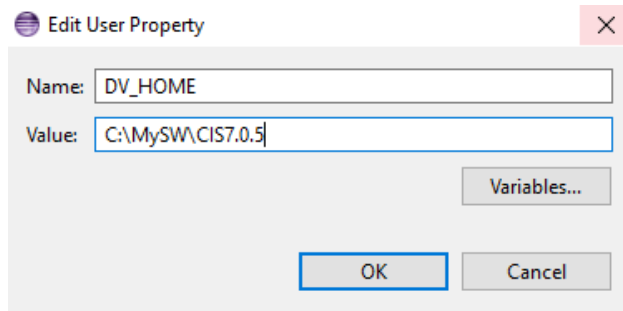


Create DV\_HOME for Ant.

In Eclipse, select Window > Preferences > Ant > Runtime > Properties > Add Property

Enter Name: DV\_HOME

Path: Locate your data virtualization home folder and type it or copy and paste it.





## Checking Out the PDTool Project

PDTool is currently developed in Eclipse. These instructions are based on the Luna distribution of Eclipse (if not using this release then the following steps may need to be modified. For instance, Luna includes the Git client, whereas earlier releases of Eclipse do not.) If installing Eclipse for the first time, the "Eclipse Standard" (or "Eclipse IDE for Java Developers", if space is tight) should be used.

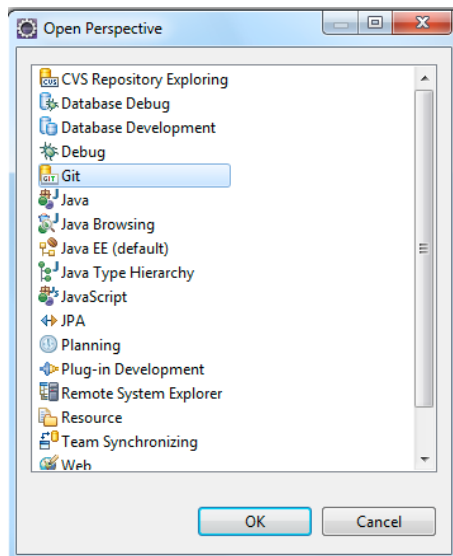
### Clone the PDTool Git repository to your local machine

For the next steps, start Eclipse and make sure to apply any updates (Help -> Check for Updates).

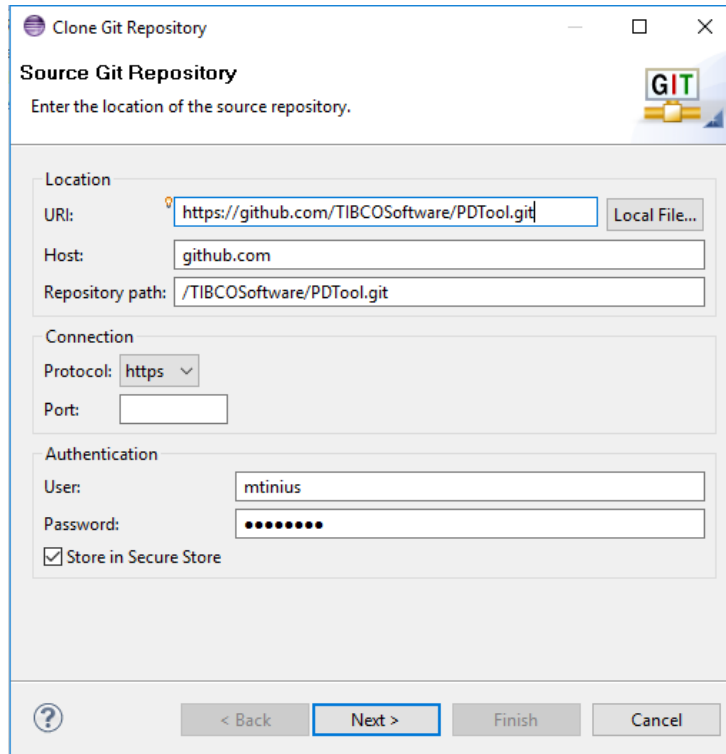
1. Open the Git perspective using the "Open Perspective" panel in the upper right of the Eclipse screen:



2. Choose the Git perspective:

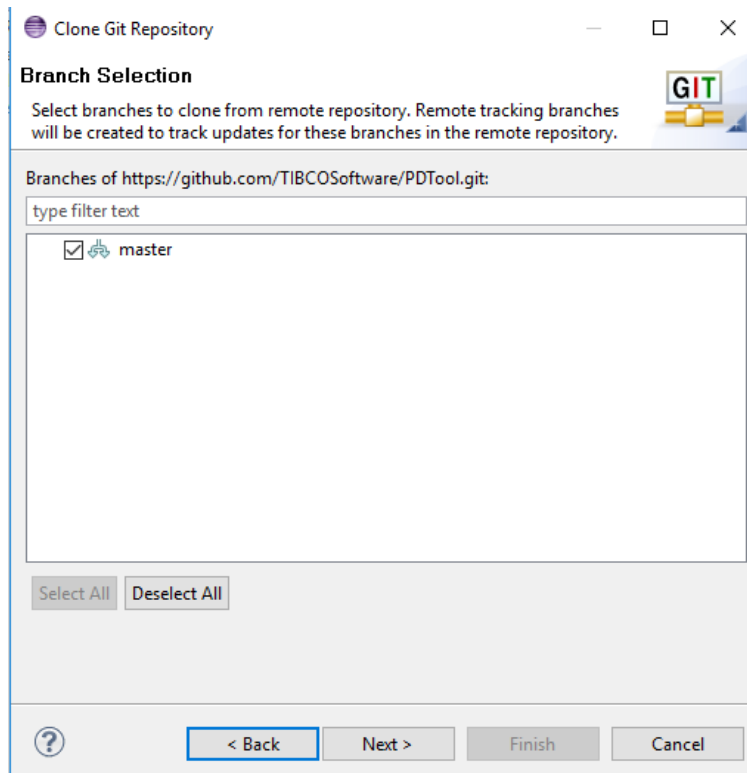


3. In the "Git Repositories" panel, click the "Clone a Git repository" link. Alternatively, go to the "File" menu and select "New" -> "Other..." Choose "Git" -> "Git Repository" from the resulting dialog.
4. In the resulting dialog, paste the Git repository URL **<https://github.com/TIBCOSoftware/PDTool.git>** into the "URI" field. The "Host" and "Repository Path" fields should auto-populate. Enter your Git user name and password and click "Next >".



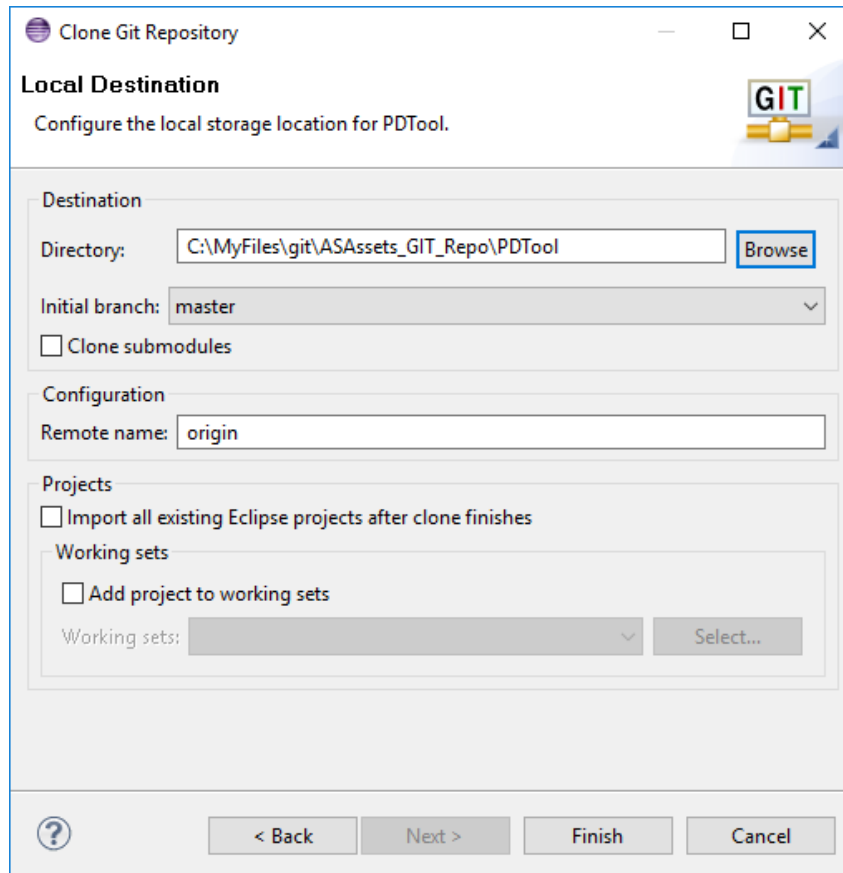
The "Clone Git Repository" dialog box is shown. It has a title bar with a question mark icon, a minimize button, a maximize button, and a close button. The main title is "Clone Git Repository". Below it is the subtitle "Source Git Repository" and the instruction "Enter the location of the source repository." There is a "GIT" logo in the top right corner. The dialog is divided into several sections: "Location" with fields for "URI" (containing "https://github.com/TIBCOSoftware/PDTool.git"), "Host" (containing "github.com"), and "Repository path" (containing "/TIBCOSoftware/PDTool.git"); "Connection" with a "Protocol" dropdown set to "https" and an empty "Port" field; and "Authentication" with a "User" field (containing "mtinius"), a "Password" field (containing dots), and a checked checkbox "Store in Secure Store". At the bottom are buttons: a question mark icon, "< Back", "Next >" (highlighted with a blue border), "Finish", and "Cancel".

5. The "Branch Selection" screen should show one branch called "master". This should already be selected so click "Next >".

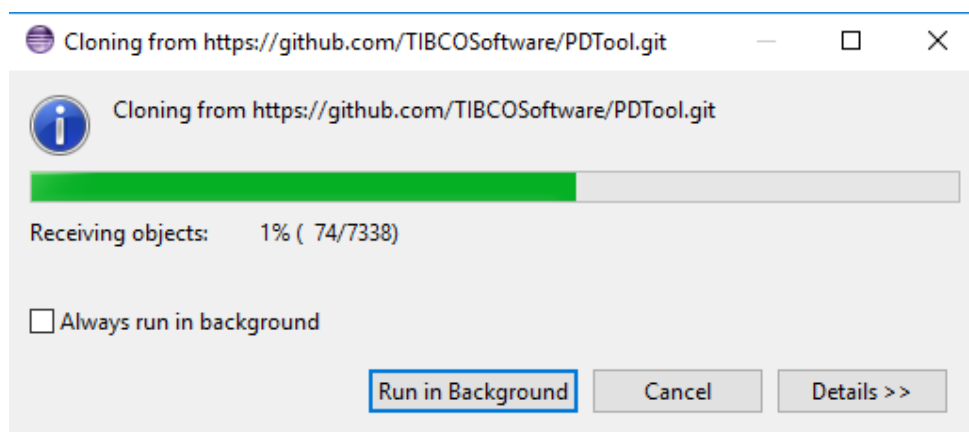


The "Branch Selection" dialog box is shown. It has a title bar with a question mark icon, a minimize button, a maximize button, and a close button. The main title is "Branch Selection". Below it is the instruction "Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository." There is a "GIT" logo in the top right corner. The dialog is divided into several sections: "Branches of https://github.com/TIBCOSoftware/PDTool.git:" with a "type filter text" input field; a list of branches with a checked checkbox and a branch icon next to "master"; and buttons "Select All" and "Deselect All". At the bottom are buttons: a question mark icon, "< Back" (highlighted with a blue border), "Next >", "Finish", and "Cancel".

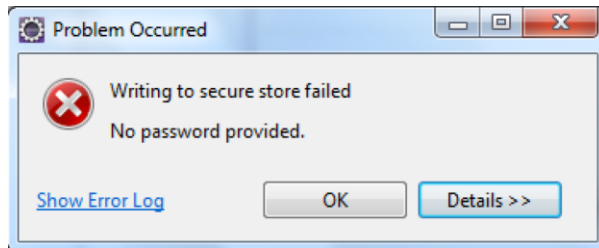
6. Choose the location where the downloaded source code should be stored. Click "Finish".



7. A progress dialog will appear indicating how far along the clone process is. Once done, the PDTool clone should then appear in your Git Repositories list.

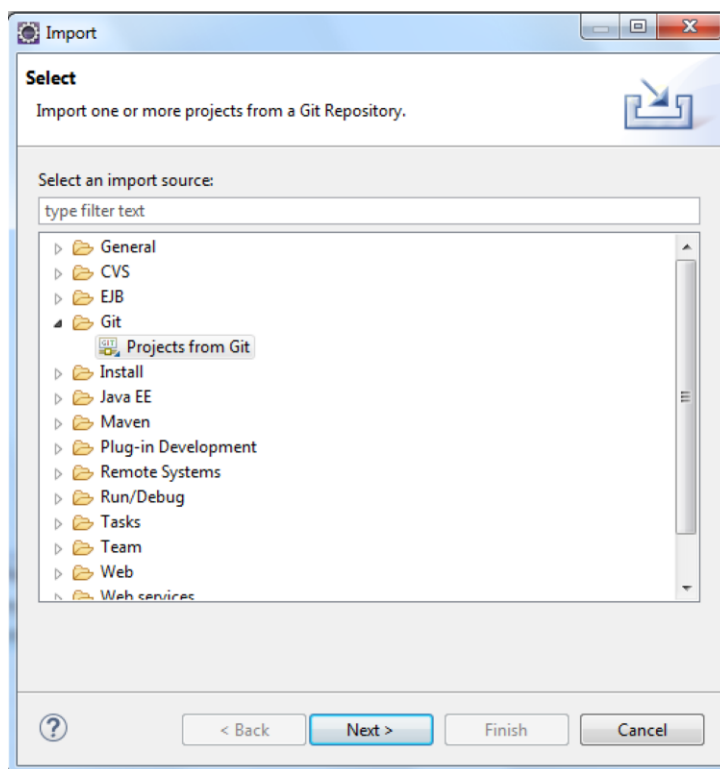


8. You may get an error stating that your secure store password has not been set. This can be safely ignored but it's probably a good idea to go into the Eclipse preferences and set a password for your secure storage.

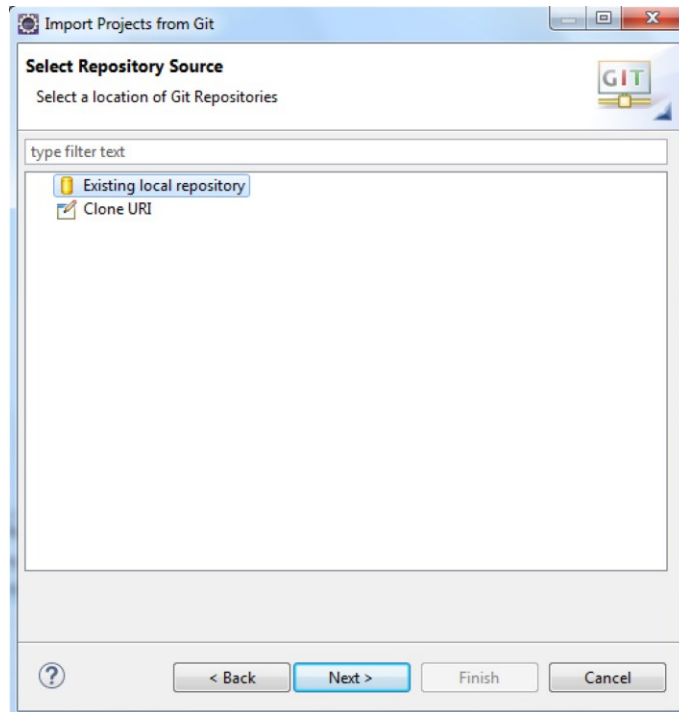


### Create a General project from the Git repository

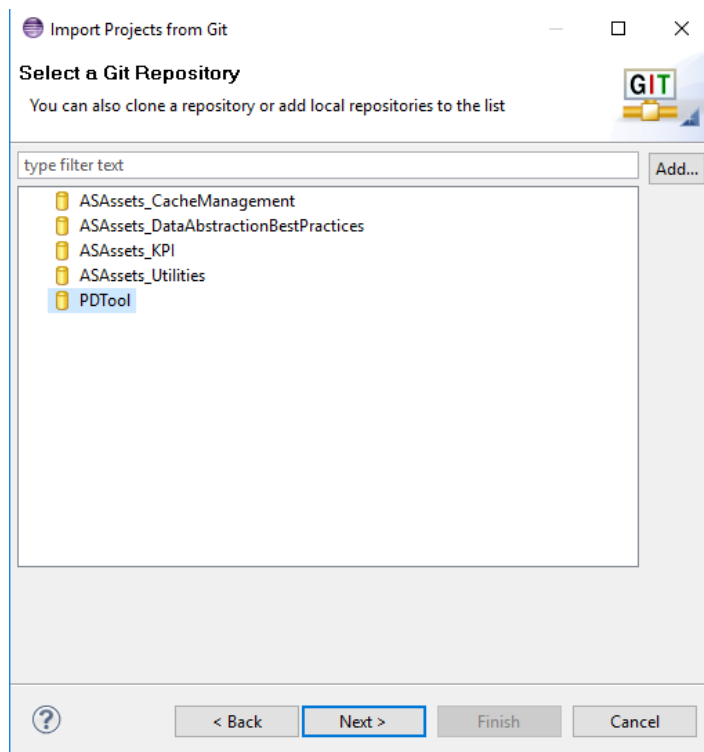
1. Open the Java perspective clicking the "Java" perspective button in the upper right of the Eclipse screen. Since the entire PDTool project is a java project, it comes into Eclipse as a Java project and is ready to start using as is.
2. Right-click in the "Package Explorer" panel on the left side of Eclipse. Select "Import ..."
3. Drill into "Git" and select "Projects from Git". Click "Next >".



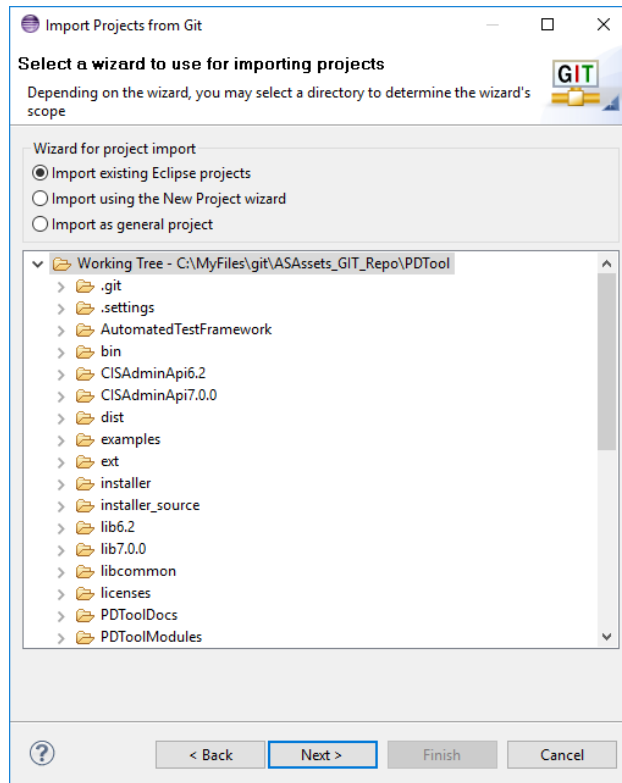
4. In the next panel choose "Existing local repository". Click "Next >".



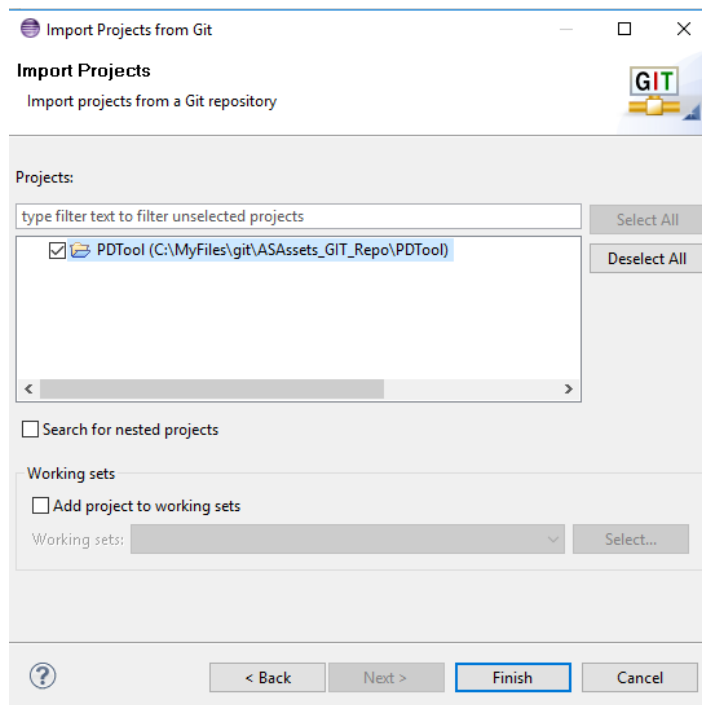
5. Choose the "PDTool" repository. Click "Next >".



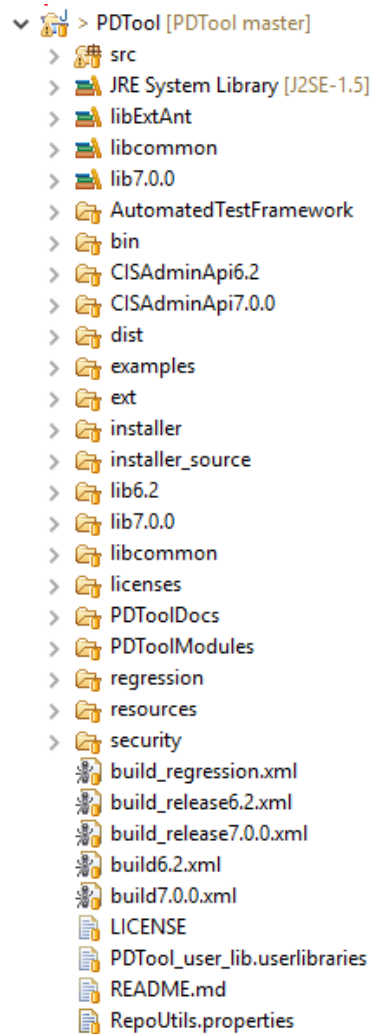
6. Select **"Import existing Eclipse projects"** as the root folder to import. Click **"Next >"**.



7. Select only the **"PDTool"** project and click **"Finish"**.



## 8. The imported project:



### Build the PDTool jar files

Build just the jar distribution file:

- Right click on build7.0.0.xml and select “Run As” → “Ant Build”
- The jar file is created as /dist/PDTool7.0.0.jar

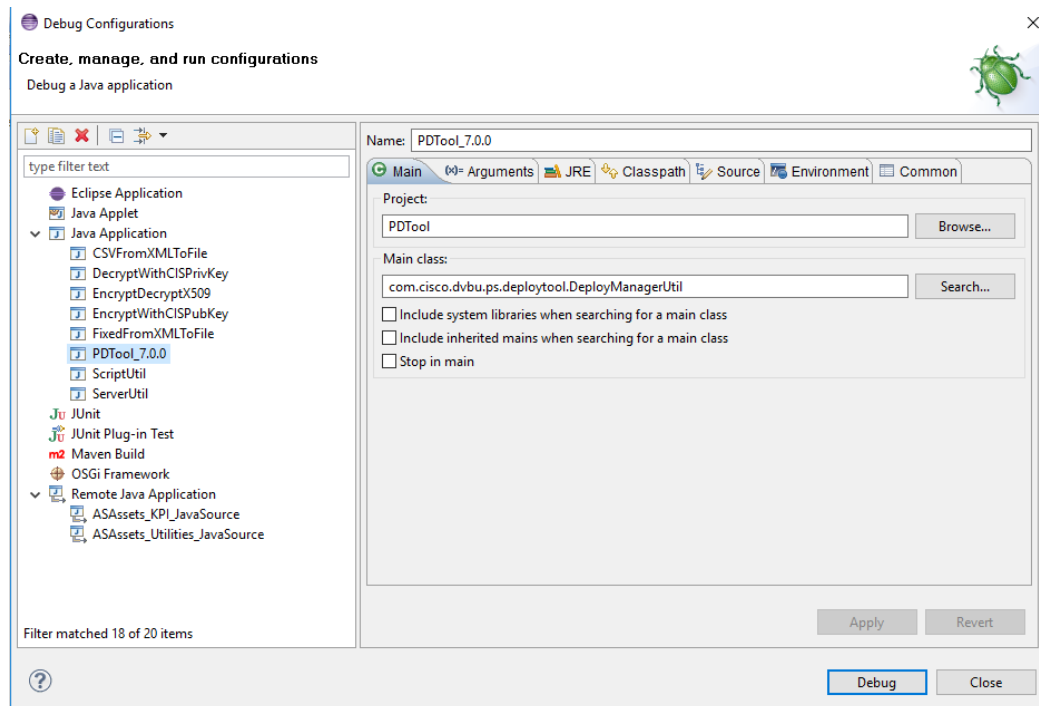
Build the entire release distribution:

- Right click on build\_release7.0.0.xml and select “Run As” → “Ant Build”
- Right click on build\_regression.xml and select “Run As” → “Ant Build”
- The zip files are created in the PDToolRelease project

## Debugging PDTool (Java) Source Code

These steps discuss how you can debug the PDTool Java code.

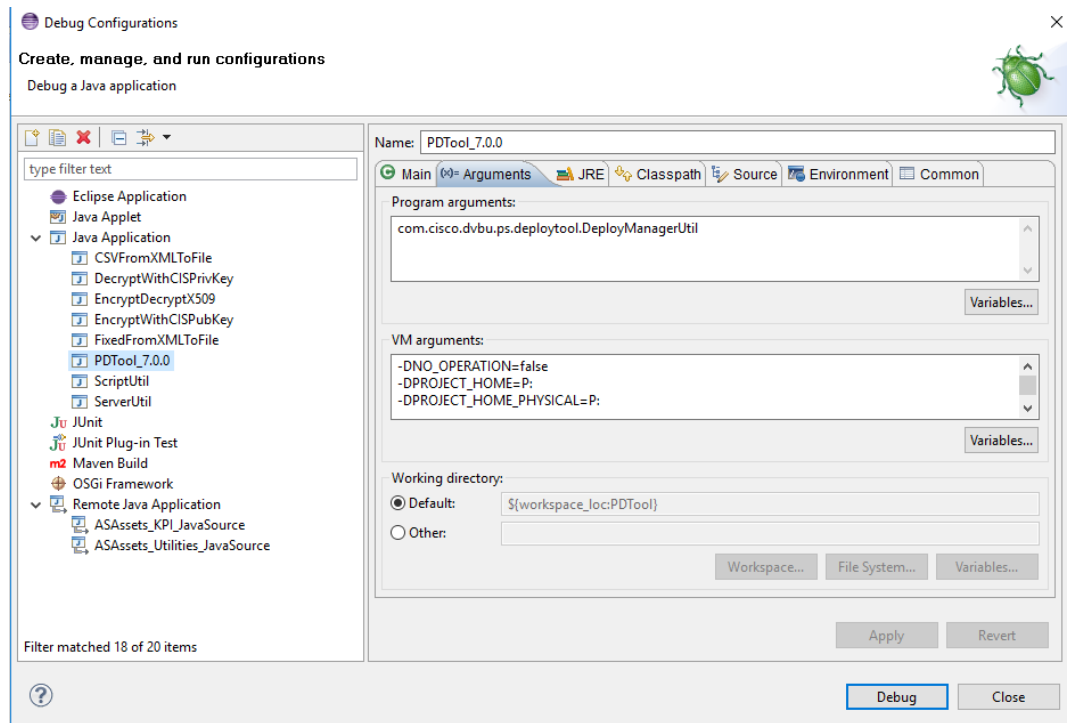
1. Open your Eclipse workspace **PDTool** java project.
  - a. Open the Debug Configurations and create a “Java Application” configuration for the PDTool Java Project
  - b. Set the parameters as shown below for Host and Port.
  - c. Use this when debugging and setting breakpoints.



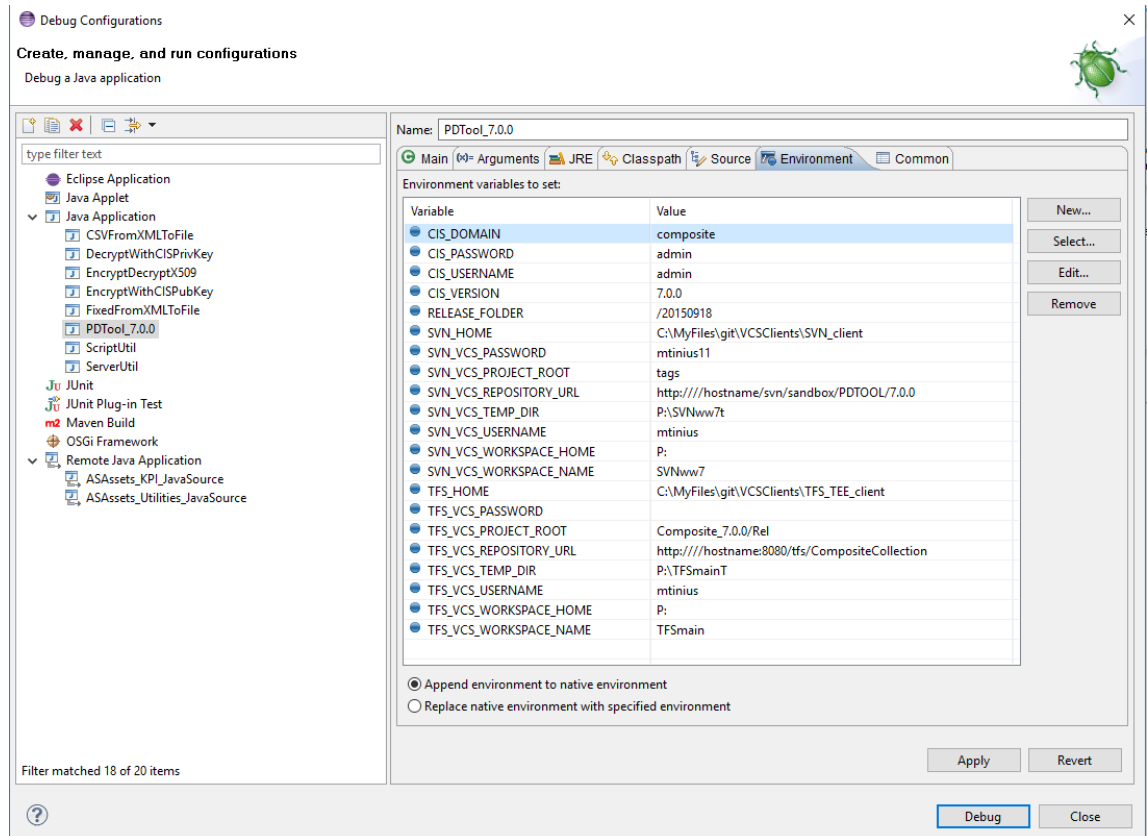
- d. Set the arguments as follows:

```
-DNO_OPERATION=false
-DPROJECT_HOME=P:
-DPROJECT_HOME_PHYSICAL=P:
-Dlog4j.configuration="file:resources\config\log4j.properties"
-Dcom.tibco.ps.configroot=resources\config
-Djava.endorsed.dirs=lib\endorsed
-
Djavax.net.ssl.trustStore="C:\Users\mtinius\git\PDToolGitTest\PDTool\security\cis_
studio_truststore_strong.jks"
-Djavax.net.ssl.trustStorePassword=changeit
-DSTUDENTID=01
```





e. Set the Environment variables as follows:



## Checking Out the PDToolRelease Project

PDToolRelease is used for the synchronization between the PDTool Java project and the PDToolRelease project and Github master repository. This is where PDTool build releases are stored and then checked in to Github.

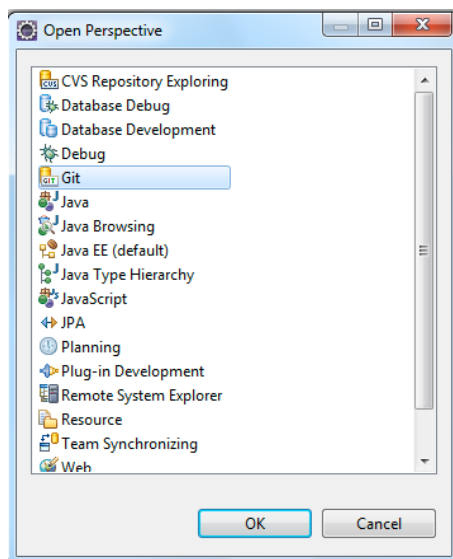
### Clone the PDToolRelease Git repository to your local machine

For the next steps, start Eclipse and make sure to apply any updates (Help -> Check for Updates).

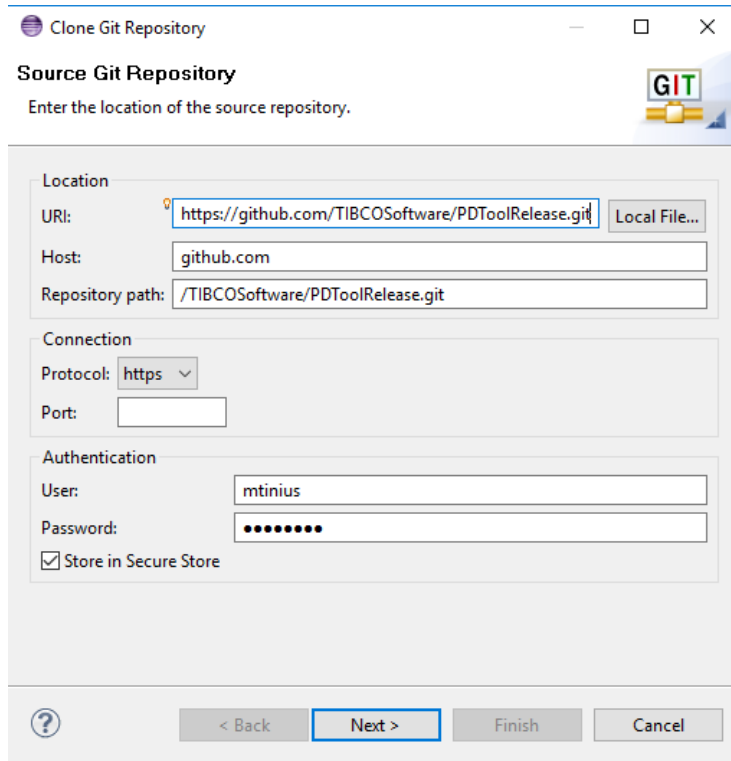
9. Open the Git perspective using the "Open Perspective" panel in the upper right of the Eclipse screen:



10. Choose the Git perspective:

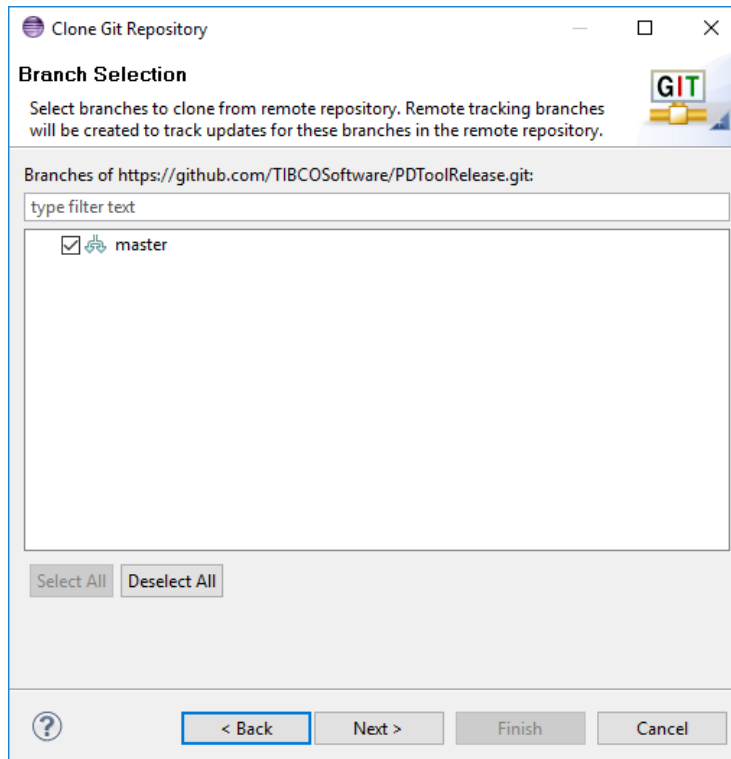


11. In the "Git Repositories" panel, click the "Clone a Git repository" link. Alternatively, go to the "File" menu and select "New"->"Other..." Choose "Git"->"Git Repository" from the resulting dialog.
12. In the resulting dialog, paste the Git repository URL **<https://github.com/TIBCOSoftware/PDToolRelease.git>** into the "URI" field. The "Host" and "Repository Path" fields should auto-populate. Enter your Git user name and password and click "Next >".



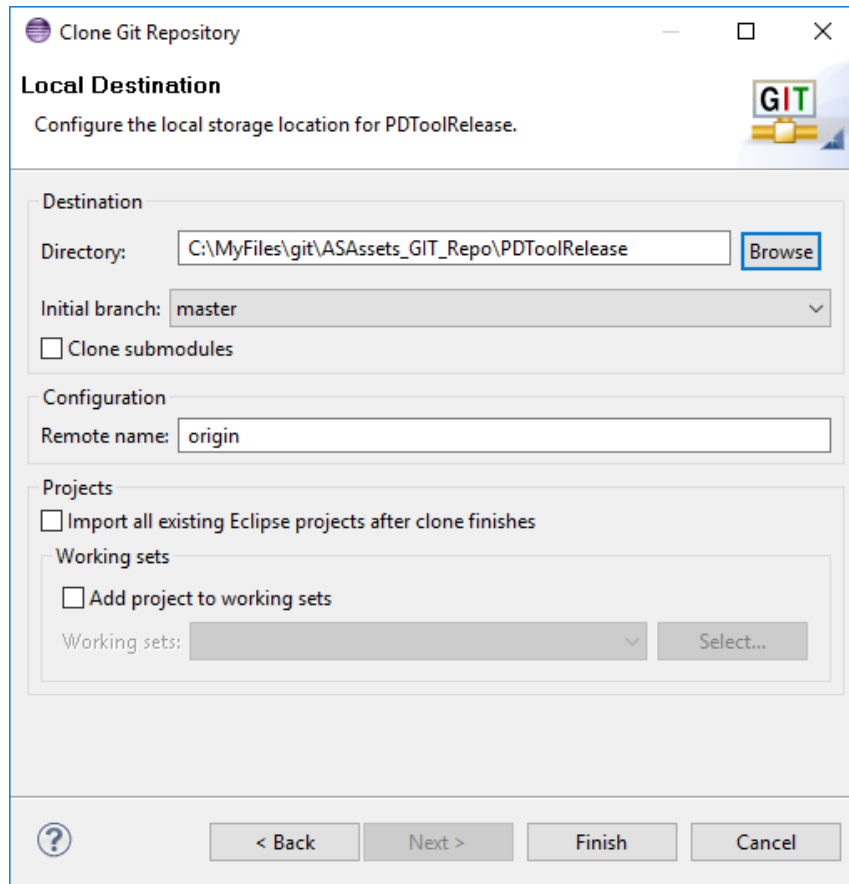
The "Clone Git Repository" dialog box is shown. It has a title bar with a question mark icon, a minimize button, a maximize button, and a close button. The main title is "Clone Git Repository". Below it is the subtitle "Source Git Repository" and a prompt "Enter the location of the source repository." To the right is a Git logo. The dialog is divided into sections: "Location" with fields for "URI" (containing "https://github.com/TIBCOSoftware/PDToolRelease.git"), "Host" (containing "github.com"), and "Repository path" (containing "/TIBCOSoftware/PDToolRelease.git"); "Connection" with a "Protocol" dropdown set to "https" and an empty "Port" field; and "Authentication" with a "User" field (containing "mtinius"), a "Password" field (filled with dots), and a checked checkbox "Store in Secure Store". At the bottom are buttons: a question mark icon, "< Back", "Next >" (highlighted with a blue border), "Finish", and "Cancel".

13. The "Branch Selection" screen should show one branch called "master". This should already be selected so click "Next >".

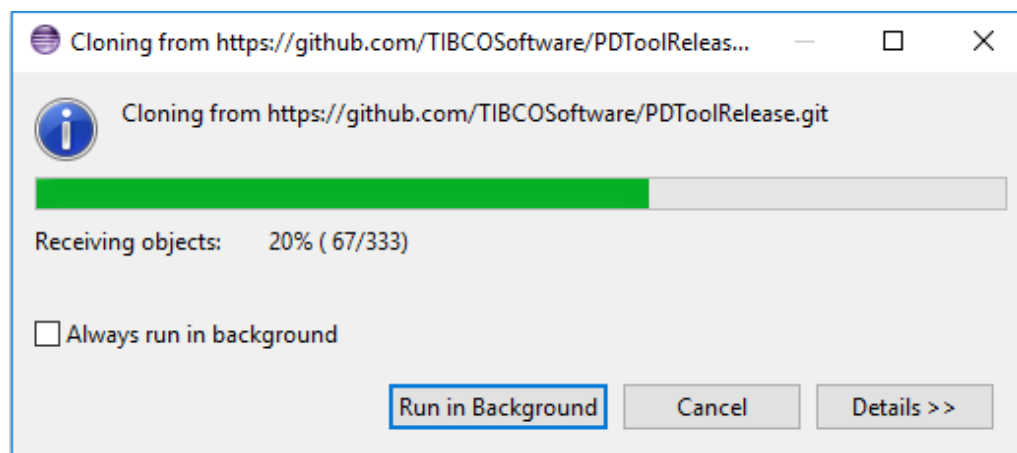


The "Branch Selection" dialog box is shown. It has a title bar with a question mark icon, a minimize button, a maximize button, and a close button. The main title is "Branch Selection". Below it is a prompt "Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository." To the right is a Git logo. The dialog is divided into sections: "Branches of https://github.com/TIBCOSoftware/PDToolRelease.git:" with a "type filter text" input field; a list box containing one item "master" with a checked checkbox and a branch icon; and buttons "Select All" and "Deselect All". At the bottom are buttons: a question mark icon, "< Back" (highlighted with a blue border), "Next >", "Finish", and "Cancel".

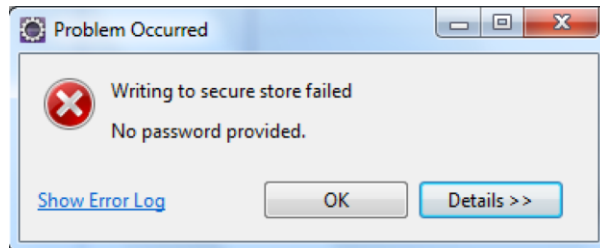
14. Choose the location where the downloaded source code should be stored. Click "Finish".



15. A progress dialog will appear indicating how far along the clone process is. Once done, the PDToolRelease clone should then appear in your Git Repositories list.

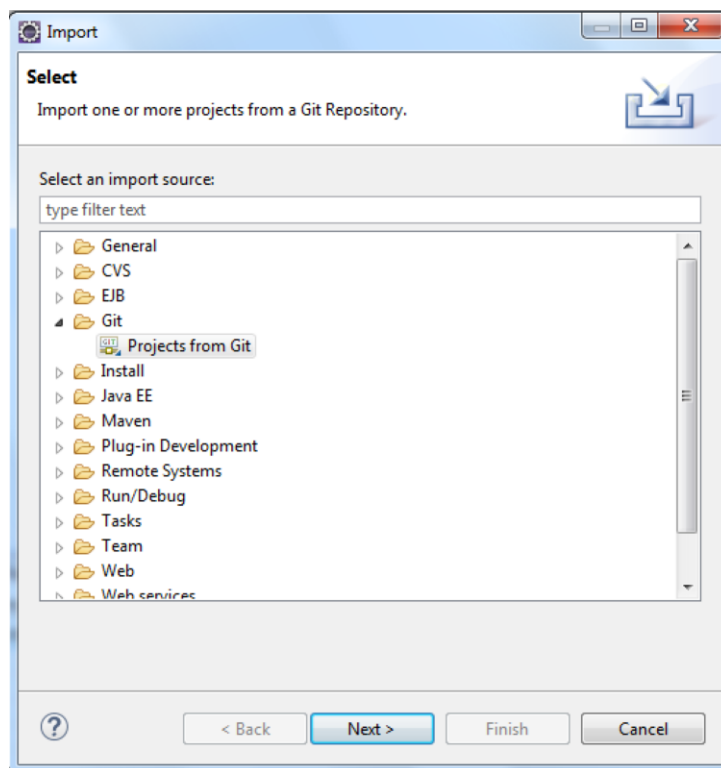


16. You may get an error stating that your secure store password has not been set. This can be safely ignored but it's probably a good idea to go into the Eclipse preferences and set a password for your secure storage.

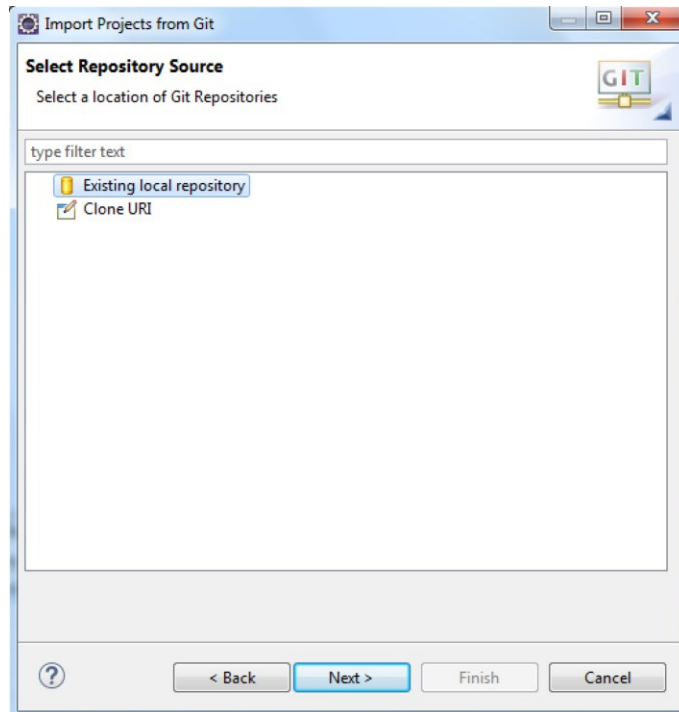


### Create a General project from the Git repository

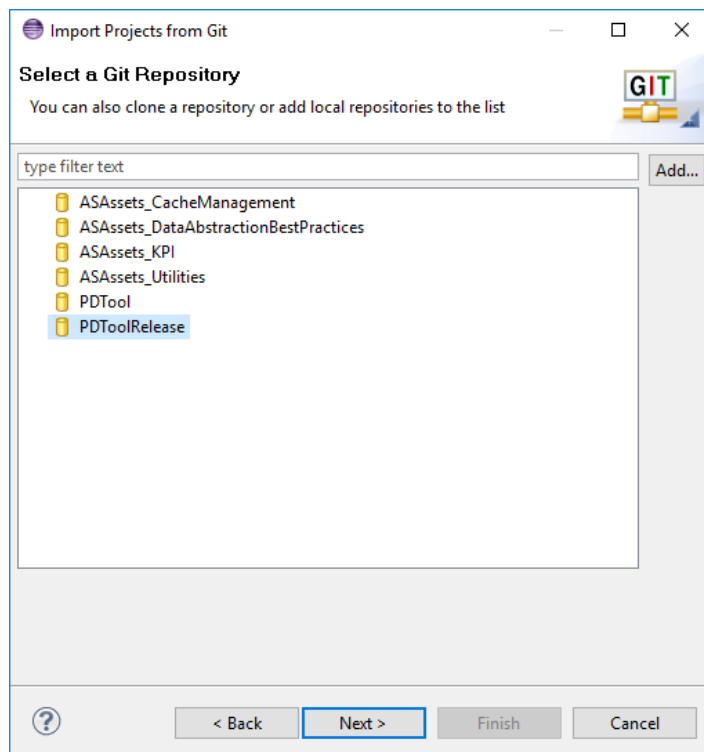
9. Open the Java perspective clicking the "Java" perspective button in the upper right of the Eclipse screen.
10. Right-click in the "Package Explorer" panel on the left side of Eclipse. Select "Import ..."
11. Drill into "Git" and select "Projects from Git". Click "Next >".



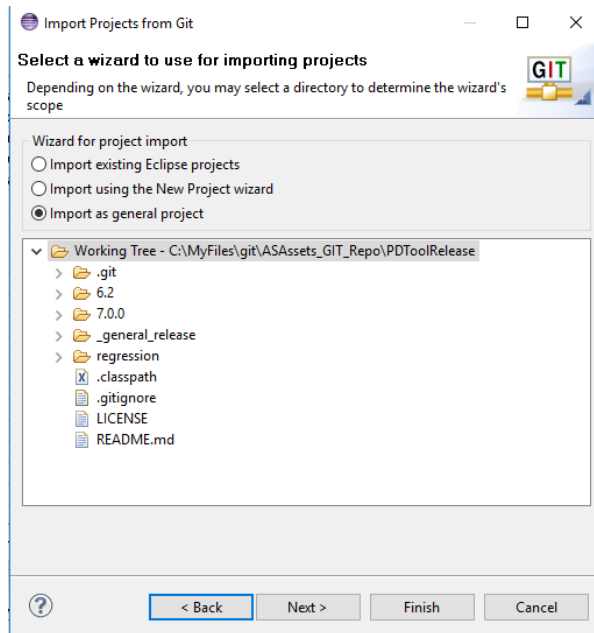
12. In the next panel choose "Existing local repository". Click "Next >".



13. Choose the "PDToolRelease" repository. Click "Next >".



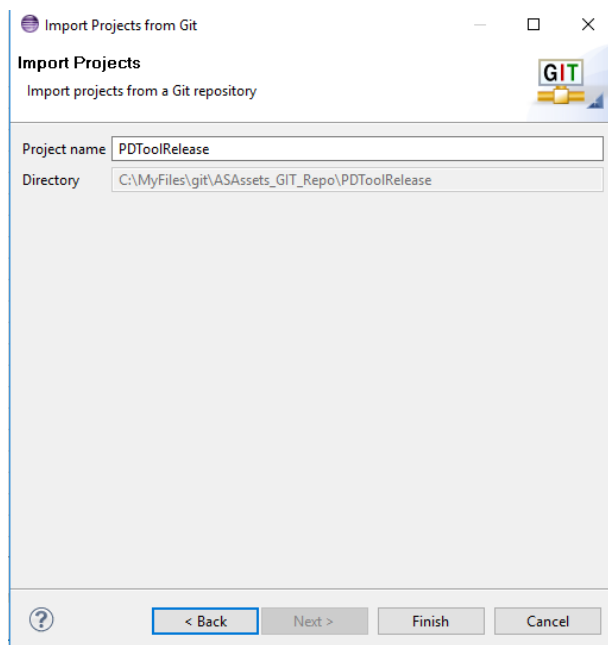
14. Select **"Import as general project"** as the root folder to import. Click **"Next >"**.










15. Handling errors.

- a. If an error message is shown that the ".project" already exists, then remove the .project file from the "Git" perspective and the PDToolRelease directory.
- b. Try again.

16. Select only the "PDToolRelease" project and click **"Finish"**.



17. The imported project:

- ▼  > PDToolRelease [PDToolRelease master]
  - >  \_general\_release
  - >  6.2
  - >  7.0.0
  - >  regression
  -  LICENSE
  -  README.md