Utilities Developer's 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 Utilities
Document Location	This document is only valid on the day it was printed. The source of the document will be found in the ASAssets_Utilities folder (https://github.com/TIBCOSoftware)
Purpose	Developer'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	06/22/2014	Mike Tinius	Initial revision
1.1	12/06/2017	Mike Tinius	Transitioned to Tibco for release 2017Q4

Related Documents

Name	Version
How To Use AS Utilities.pdf	2017Q4

Supported Versions

Name	Version
TIBCO® Data Virtualization	7.0 or later

Table of Contents

1 Introduction 4
Purpose 4
Audience 4
References 4

2 Utilities Overview 5
What are the AS Assets Utilities? 5
Recommended Development Tools 5

3 Github Repository Structure 6
How Can I Download a Copy of the Utilities GitHub Repository? 6
Repository Folder Structure 6
DocumentationSource 6
DVSource 6
DVSource/cis_objects 6
DVSource/scripts 6
JavaSource 6
JavaSource/lib 6
Release 6
Release/archive 6

4 Configuring the Eclipse Development Environment 7
Checking Out the AS Utilities DV Source Code 7
Configure Eclipse Variables 7
Checking Out the AS Utilities Project 9
Clone the AS Utilities Git repository to your local machine 9
Create a General project from the Git repository to control check-in 12
Create a Java project from the Git repository JavaSource folder 14
Build the AS Utilities jar files 16
Debugging AS Utilities CJP (Java) Source Code 16

1 Introduction

Purpose

The purpose of this document is to provide detailed usage of the core Utilities. Utilities are building blocks of functionality that, in some cases, provide a wrapper on top of the TIBCO® Data Virtualization API and in other cases simply encapsulate a capability deemed to be repeatable and useful.

This document covers the following topics:

- Advanced Services (AS) Assets Utilities Overview - Describes the history and use of the AS Assets Utilities distribution.
- Extending and/or Contributing to the AS Assets Utilities distribution.

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 Utilities Overview

What are the AS Assets Utilities?

The AS Assets Utilities, or simply "The Utilities", were started as an internal project to collect the ever-increasing number of simple SQL Scripts and custom Java procedures (CJPs) that ended up being reusable at multiple customer sites. Rather than reinvent the wheel at each customer, it was proposed that these simple procedures should be gathered and made available to *all* customers free of charge as an open source offering.

The Utilities are made up of a number of procedures that perform relatively simple tasks, such as string, date/time, and XML manipulation; interactions with the CIS repository or CIS host filesystem; and helpers for working with another AS Asset known as "PDTool".

These procedures are not intended to be full custom solutions (though many end up being used in such solutions) but are simple timesavers that make the life of a CIS developer easier.

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 CJPs, 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 Utilities 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.

3 Github Repository Structure

How Can I Download a Copy of the Utilities GitHub Repository?

The GitHub master repository for the Utilities is located at https://github.com/TIBCOSoftware/ASAssets_Utilities.

Repository Folder Structure

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

DocumentationSource

This folder contains the source versions of the user's guide and developer's guide (this document) in Microsoft Word format. When a release version of a document is ready for distribution, it should be saved in PDF format and placed in the Release folder.

DVSource

This folder contains the source code for the Utilities DV resources.

DVSource/cis_objects

This folder contains the exported CIS resources in the version control export format. Instead of a single .CAR file, the resources are exported in a folder tree structure that matches the container structure in CIS. A .CMF file named for the resource with a resource type suffix contains the resource's source code, model, ownership information, annotation, permissions, etc. A container will have a corresponding .CMF file contained inside it.

DVSource/scripts

This folder will contain (still in development) a number of Windows batch files and Linux shell scripts that can be used to import the resources in the cis_objects folder into a CIS instance and export resources from a CIS instance into the cis_objects folder structure.

JavaSource

This folder contains the source code for the Utilities DV resources. It contains an Eclipse project that can be used to build individual CJP data source jars or all of them in one build. Each CJP collection has a child folder here.

JavaSource/lib

This folder contains library jar files used to build the CJP jars. Open source libraries have corresponding LICENSE.txt files. CIS libraries are named with a "cs" prefix.

Release

This folder contains the current release of the Utilities and corresponding documentation.

Release/archive

This folder contains past releases of the Utilities and corresponding documentation.

4 Configuring the Eclipse Development Environment

Checking Out the AS Utilities DV Source Code

TBD. For now, the best way to make sure a new or updated SQL Script gets into the next release is to send the script (plus documentation and examples!) in a CAR file to Mike Tinius.

Configure Eclipse Variables

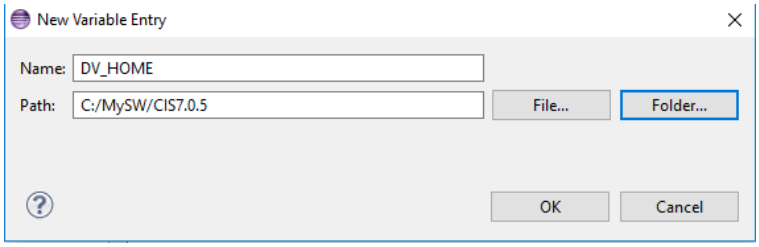
There are two variables that need to be created. One is for Eclipse and one is for Ant.

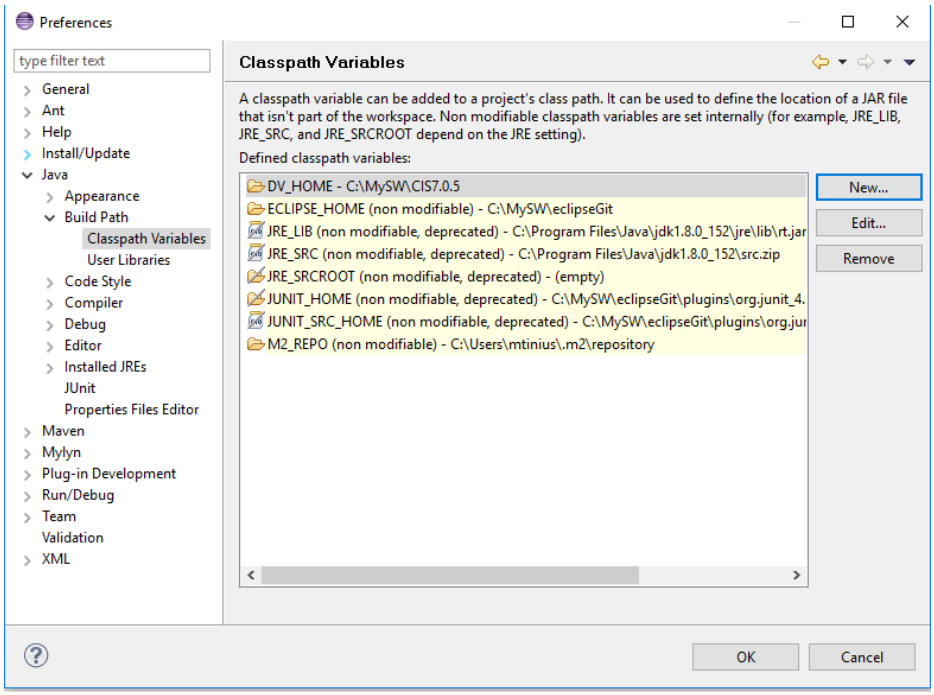
Create DV_HOME for Eclipse.

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

Enter Name: DV_HOME

Path: Browse to your data virtualization home folder



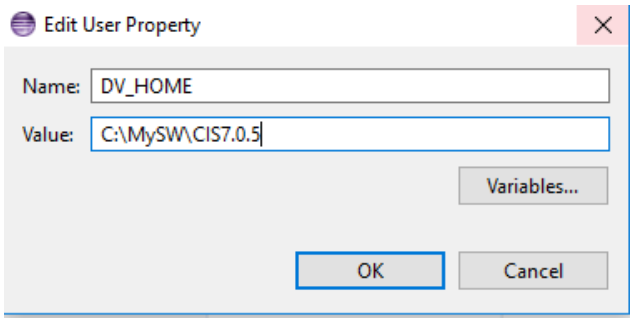


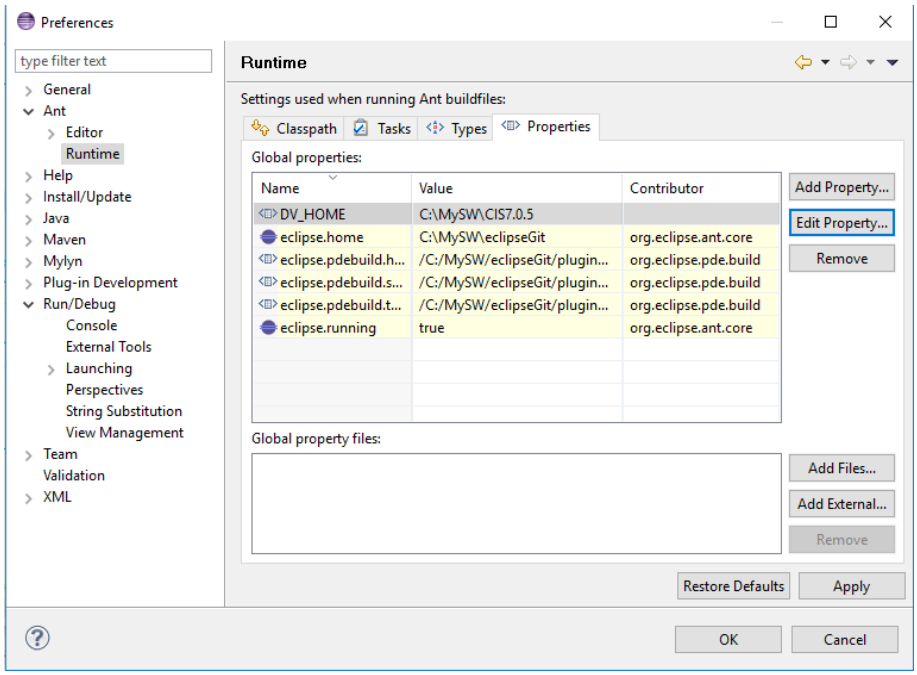
Create DV_HOME for Ant.

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 AS Utilities Project

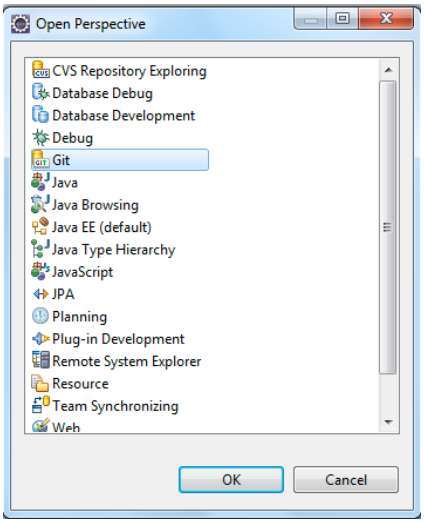
CJPs for the AS Utilities are currently developed in Eclipse. These instructions are based on the Luna distribution of Eclipse (if not using this release then the following steps may need to be modified. For instance, Luna includes the Git client, whereas earlier releases of Eclipse do not.) If installing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

Clone the AS Utilities Git repository to your local machine

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

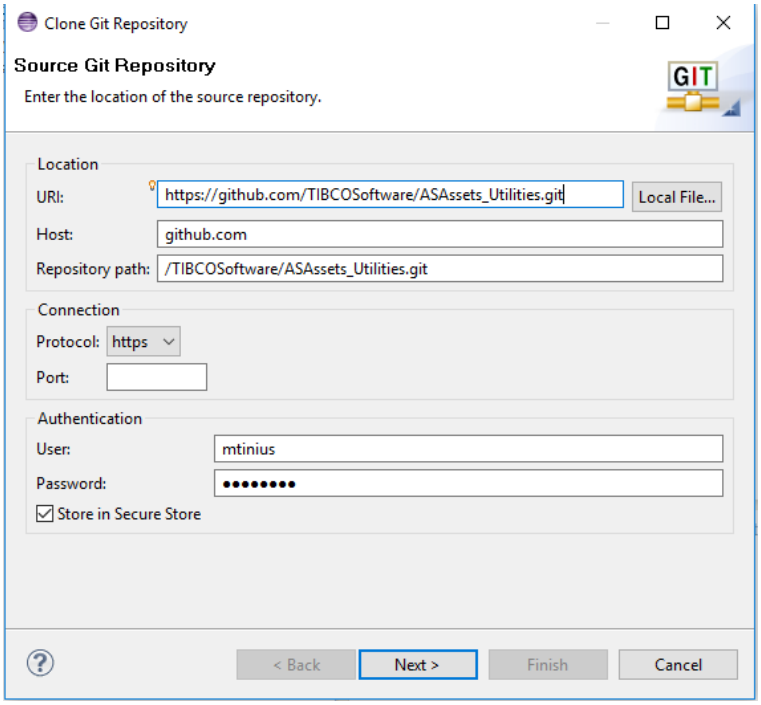
1. Open the Git perspective using the "Open Perspective" panel in the upper right of the Eclipse screen:

2. Choose the Git perspective:

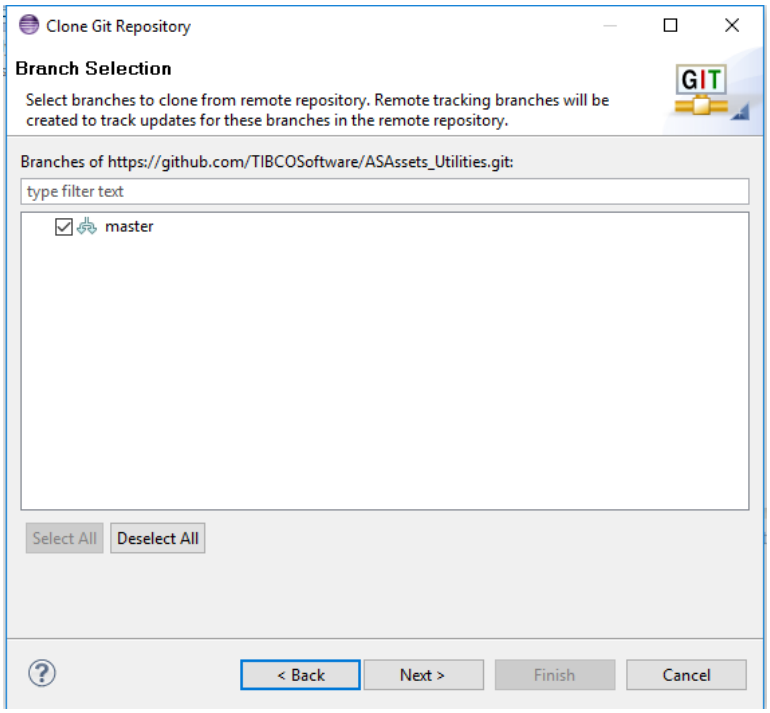


3. In the "Git Repositories" panel, click the "**Clone a Git repository**" link. Alternatively, go to the "File" menu and select "New"->"Other…" Choose "Git"->"Git Repository" from the resulting dialog.

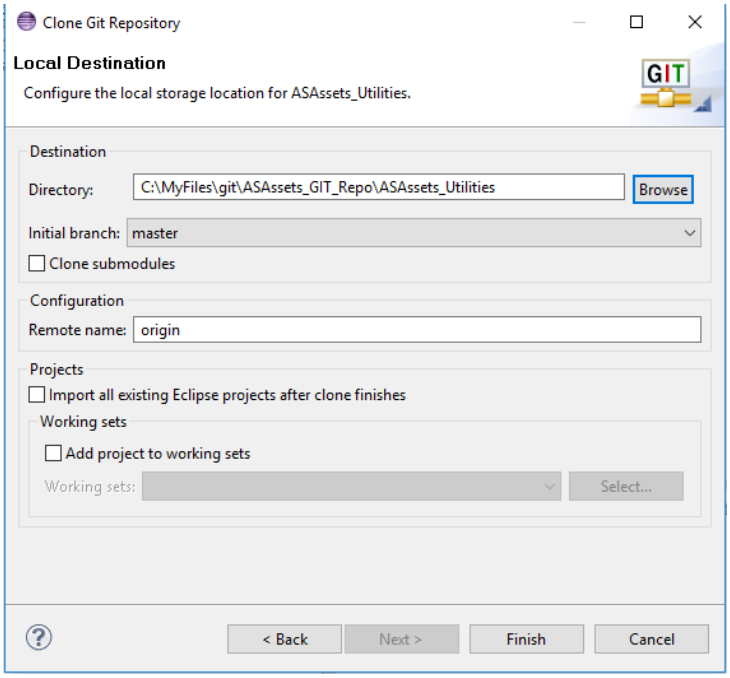
4. In the resulting dialog, paste the Git repository URL ***https://github.com/TIBCOSoftware/ASAssets_Utilities.git*** into the "URI" field. The "Host" and "Repository Path" fields should auto-populate. Enter your Git user name and password and click "Next >".



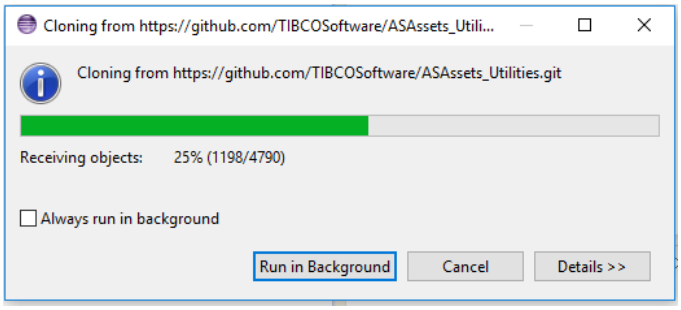
5. The "Branch Selection" screen should show one branch called "master". This should already be selected so click "Next >".



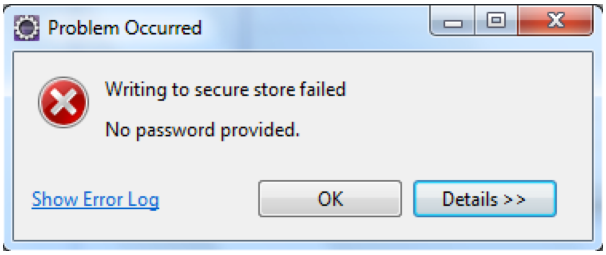
6. Choose the location where the downloaded source code should be stored. Click "Finish".



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

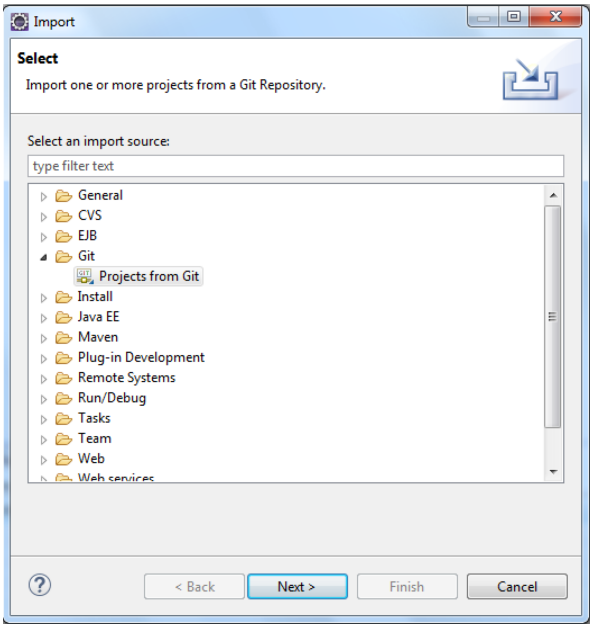


8. You may get an error stating that your secure store password has not been set. This can be safely ignored but it's probably a good idea to go into the Eclipse preferences and set a password for your secure storage.

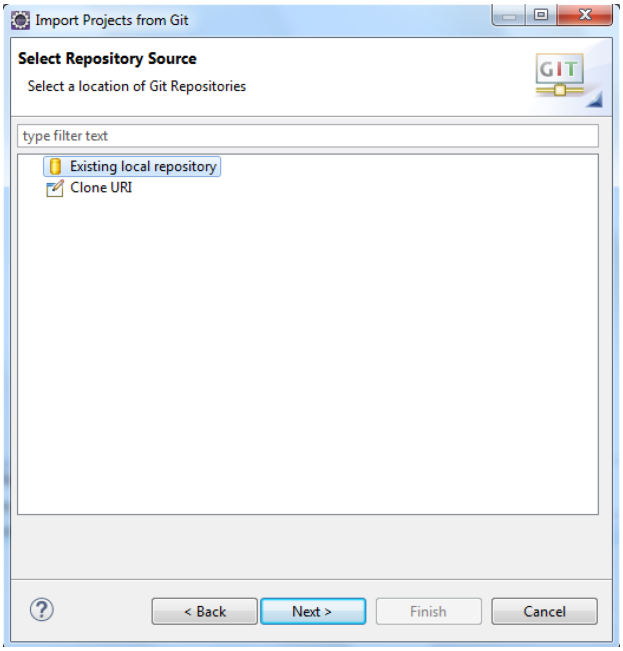


Create a General project from the Git repository to control check-in

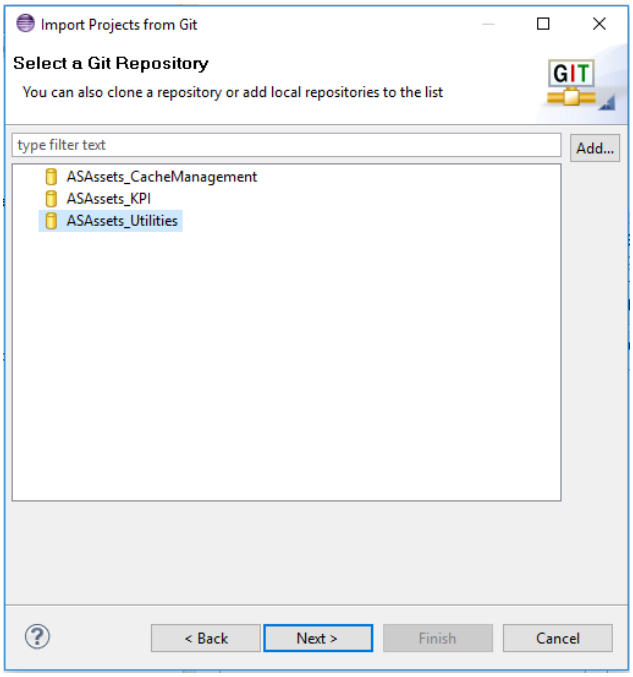
1. Open the Java perspective clicking the "Java" perspective button in the upper right of the Eclipse screen.
2. Right-click in the "Package Explorer" panel on the left side of Eclipse. Select "Import …"
3. Drill into "Git" and select "Projects from Git". Click "Next >".



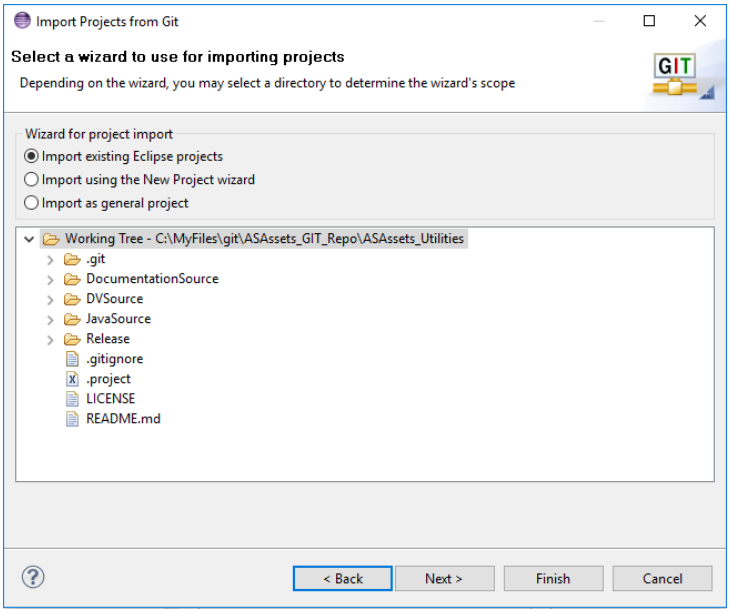
4. In the next panel choose "Existing local repository". Click "Next >".



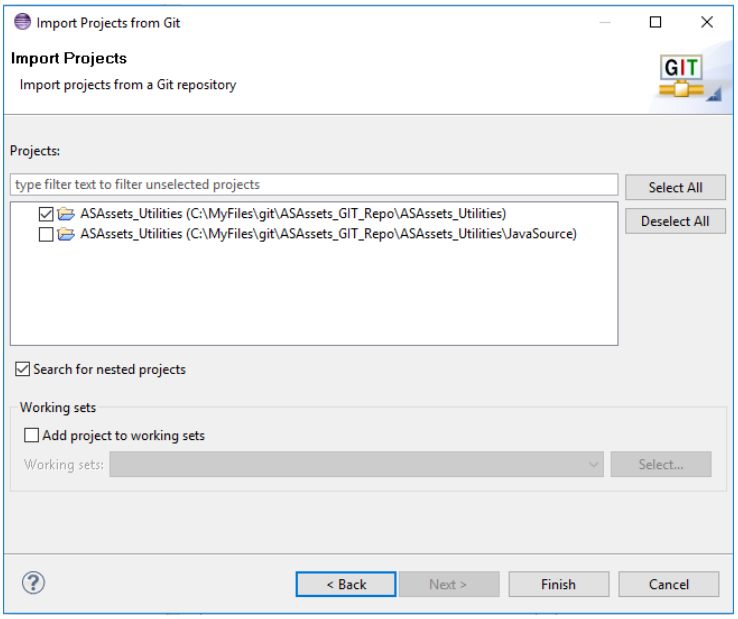
5. Choose the "ASAssets_Utilities" repository. Click "Next >".



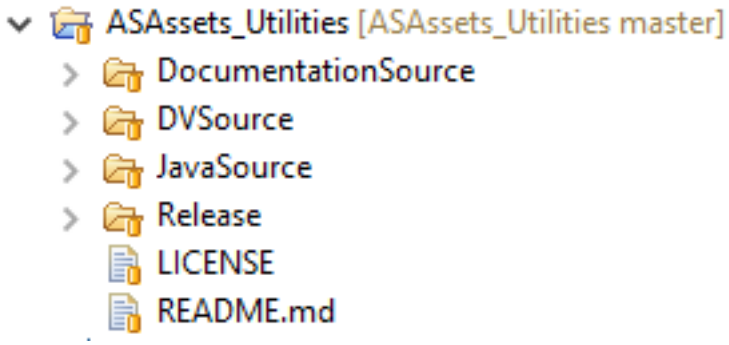
6. Select "Import existing Eclipse projects" as the root folder to import. Click "Next >".



7. Select only the “ASAssets_Utilities” project and click "Finish".

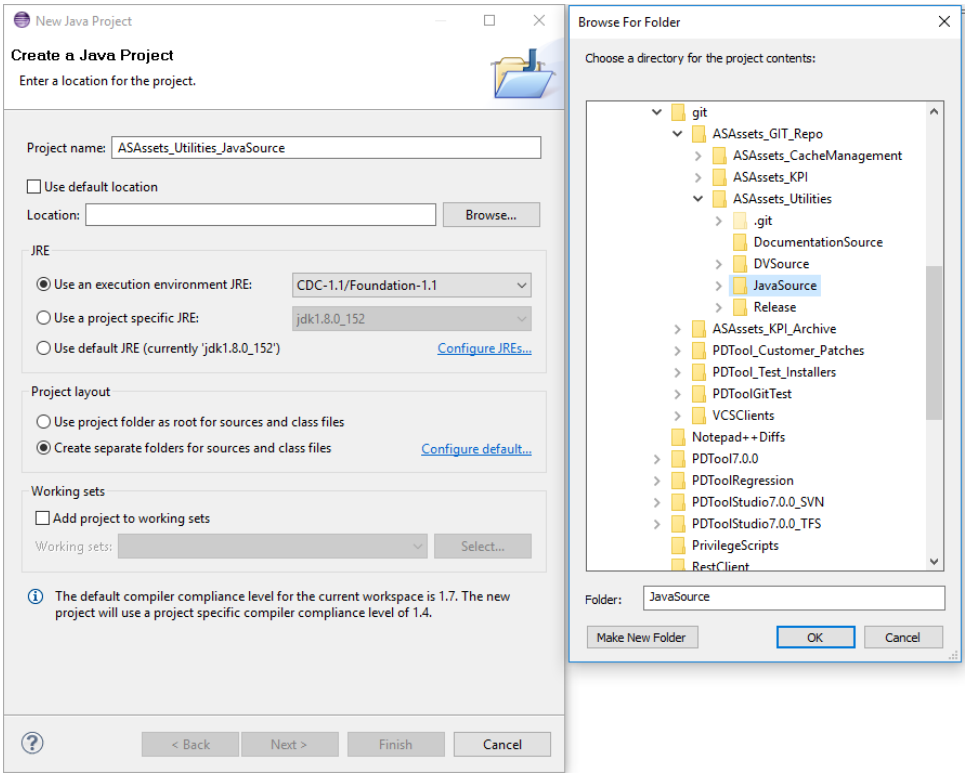


8. The imported project:

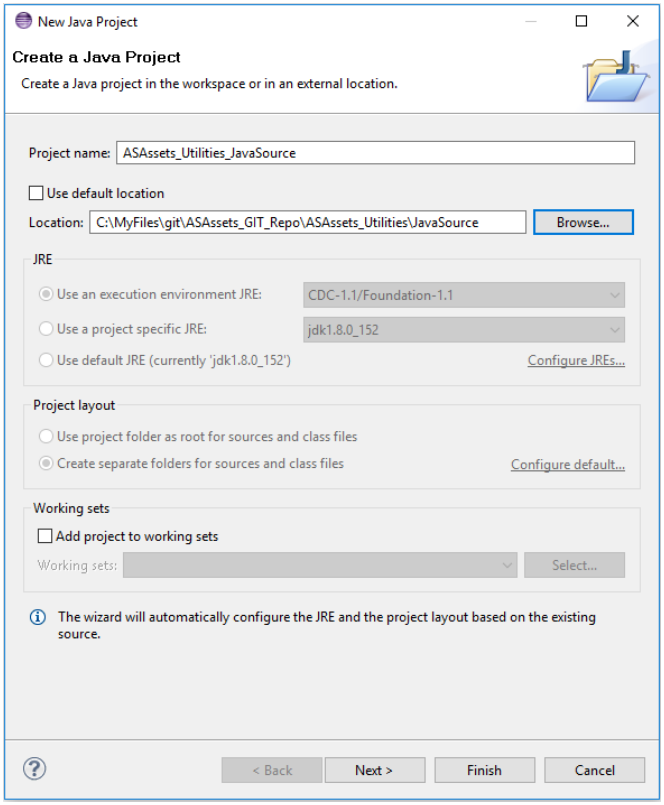


Create a Java project from the Git repository JavaSource folder

1. Open the Java perspective clicking the "Java" perspective button in the upper right of the Eclipse screen.
2. Select File > New > Java Project
    a. Enter ASAssets_Utilities_JavaSource
    b. Unselect "Use default location"
    c. Browse to the Git repository location and select ASAssets_Utilities/JavaSource



d. Click "Finish"



Build the AS Utilities jar files

Building the jar files for the CJP data sources is very straightforward. Each CJP folder contains a "build.xml" file that can be used to build jar files individually. Alternatively, the "build.xml" file in the project's root folder can be used to build all the jar files in one build. When one or more jar files are built, they will appear in the "dist" folder in the project's root folder.

Debugging AS Utilities CJP (Java) Source Code

These steps discuss how you can debug the AS Utilities Java code using a remote CIS instance to attach to.

1. Stop the CIS monitor from Control Panel:
2. Open a command line window to start CIS manually in debug mode
    a. C:\CIS7.0.5\bin>**composite_server.bat debug**
3. Open your Eclipse workspace for ASAssets_Utitlities_JavaSource
    a. Open the Debug Configurations and create a "Remote Java Application" configuration for the ASAssets Utilities Java Project
    b. Set the parameters as shown below for Host and Port.
    c. Use this when debugging and setting breakpoints.