Tibco Data Virtualization Open Source Assets

KPI Metrics Developer’s Guide

|  |  |
| --- | --- |
| **Project Name** | AS Assets KPI Metrics |
| **Release** | 2017Q4 |
| **Date** | December 5 2017 |
| **Primary Author** | Michael Tinius |
| **Document Owner** | TIBCO Software |
| **Client** | Open Source |
| **Document Location** | This document is only valid on the day it was printed. The source of the document will be found in the ASAssets\_KPI folder (https://github.com/TIBCOSoftware) |
| **Purpose** | Developer’s Guide |

This document presents a developer’s guide for the open source Data Virtualization KPI Metrics capability.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1.0 | 12/04/2017 | Mike Tinius | Initial revision |

References

This document has been distributed to:

|  |  |
| --- | --- |
| **Name** | **Version** |
| Tibco Data Virtualization Server (formerly Cisco Data Virtualization) | 7.0 or later |

Distribution

This document has been distributed to:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Title** | **Company** | **Date of Issue** | **Version** |
| Tibco Github Open Source community | N/A | N/A | Dec 2017 | 2017 Q4 |

Copyright Notice

COPYRIGHT© 2015 TIBCO Software Inc. This document is unpublished and the foregoing notice is affixed to protect TIBCO Software Inc. in the event of inadvertent publication. All rights reserved. No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of TIBCO Software Inc. The information contained in this document is confidential and proprietary to TIBCO Software Inc. and may not be used or disclosed except as expressly authorized in writing by TIBCO Software Inc. Copyright protection includes material generated from our software programs displayed on the screen, such as icons, screen displays, and the like.

Trademarks

All brand and product names are trademarks or registered trademarks of their respective holders and are hereby acknowledged. Technologies described herein are either covered by existing patents or patent applications are in progress.

Confidentiality

The information in this document is subject to change without notice. This document contains information that is confidential and proprietary to TIBCO Software Inc. and its affiliates and may not be copied, published, or disclosed to others, or used for any purposes other than review, without written authorization of an officer of TIBCO Software Inc. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

Content Warranty

The information in this document is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" AND TIBCO MAKES NO WARRANTY, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TIBCO Software Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Export

This document and related technical data, are subject to U.S. export control laws, including without limitation the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations of other countries. You agree not to export or re-export this document in any form in violation of the applicable export or import laws of the United States or any foreign jurisdiction.

For more information, please contact:

TIBCO Software Inc.  
3303 Hillview Avenue  
Palo Alto, CA 94304  
USA

**Table of Contents**

1 Introduction 5

Purpose 5

Audience 5

2 Github Repository Structure 6

Recommended Development Tools 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 7

JavaSource/lib 7

Release 7

Release/archive 7

3 Configure the AS KPI Metrics Development Environment 8

Checking Out the AS KPI Metrics DV Source Code 8

Checking Out the AS KPI Metrics CJP (Java) Source Code 8

Clone the AS KPI Metrics Git repository to your local machine 8

Create a Java project from the Git repository 12

Debugging AS KPI CJP (Java) Source Code 15

Copy CpuAndMemChecker to a separate Eclipse Java project 15

Build the AS KPI jar files 15

1. Introduction

## Purpose

The purpose of this document is to provide guidance on how develop the AS Assets KPI metrics.

## Audience

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

* Tibco Data Virtualization Server Developers
* a.k.a. Composite Information Server (CIS) Developers

1. 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 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.

## 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_KPI>.

## 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. An update to the PDToolStudio distribution to support Git is on the road map.

### 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.

1. Configure the AS KPI Metrics Development Environment

## Checking Out the AS KPI Metrics 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.

## Checking Out the AS KPI Metrics CJP (Java) Source Code

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 KPI Metrics Git repository to your local machine

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

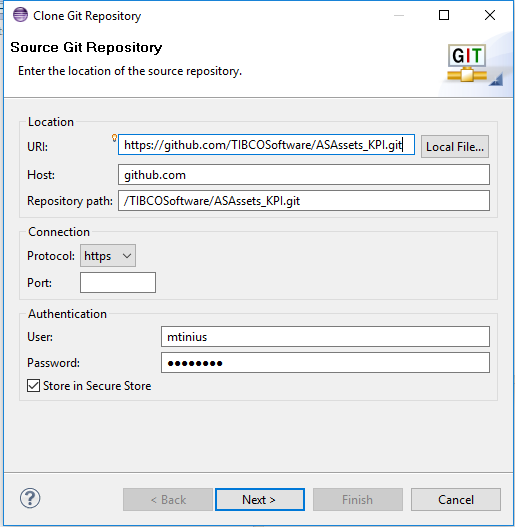
1. Open the Git perspective using the "Open Perspective" panel in the upper right of the Eclipse screen:



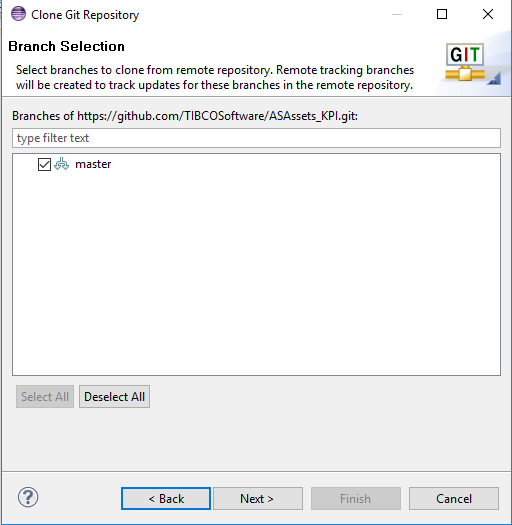
1. Choose the Git perspective:



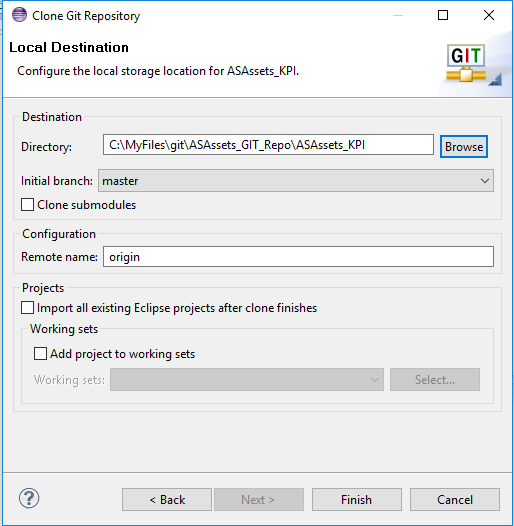
1. In the "Git Repositories" panel, click the "Clone a Git repository" link. Alternatively, go to the "File" menu and select "New"->"Other…" Choose "Git"->"Git Repository" from the resulting dialog.
2. In the resulting dialog, paste the Git repository URL ***https://github.com/cisco/ASAssets\_KPI.git***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



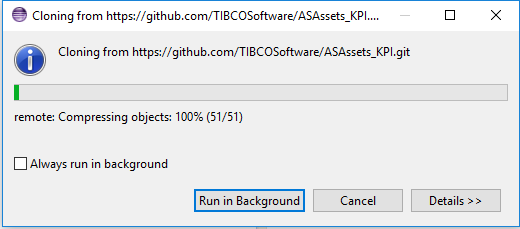
1. The "Branch Selection" screen should show one branch called "master". This should already be selected so click "Next >".



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



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



1. 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 Java project from the Git repository

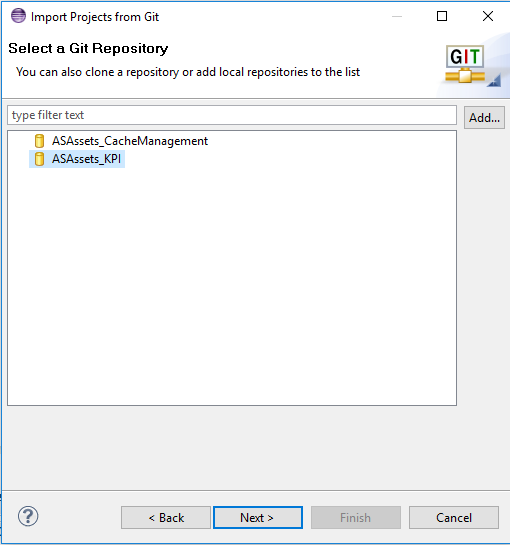
1. Open the Java perspective clicking the "Java" perspective button in the upper right of the Eclipse screen.
2. Right-click in the "Package Explorer" panel on the left side of Eclipse. Select "Import …"
3. Drill into "Git" and select "Projects from Git". Click "Next >".



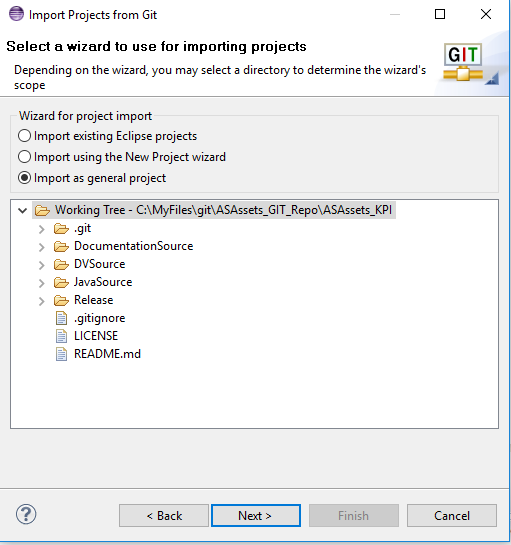
1. In the next panel choose "Existing local repository". Click "Next >".



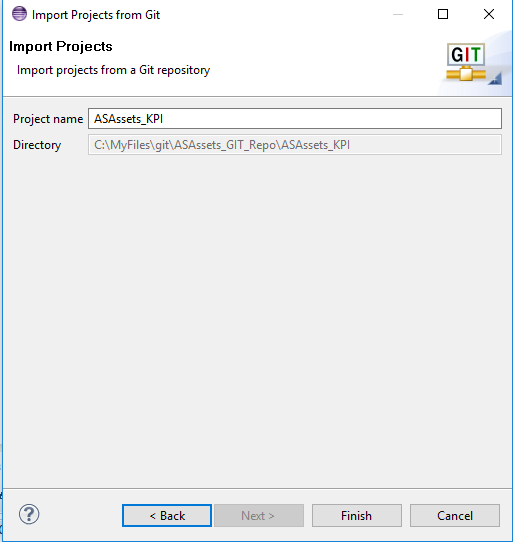
1. Choose the "ASAssets\_KPI" repository. Click "Next >".



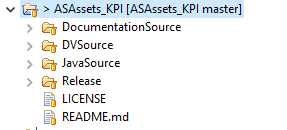
1. Select "Import as general project" as the root folder to import. Click "Next >".



1. There should only be one project to import. Click "Finish".



1. The imported project:



## Debugging AS KPI CJP (Java) Source Code

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

### Copy CpuAndMemChecker to a separate Eclipse Java project

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.

### Build the AS KPI 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.

1. Stop the CIS monitor from Control Panel:
2. Open a command line window to start CIS manually in debug mode
   1. C:\CiscoSystems\CIS7.0.5\bin>**composite\_server.bat debug**
3. Open your Eclipse workspace which contains **CpuAndMemChecker** java project
   1. Open the Debug Configurations and create a “Remote Java Application” configuration for you AS Utilities Java Project
   2. Set the parameters as shown below for Host and Port.
   3. Use this when debugging and setting breakpoints.

