



TIBCO® Data Science Team Studio Installation and Administration

Version 6.6.0

June 2021



Contents

System Requirements	7
Supported Database Platforms	8
Supported Hadoop Platforms	9
R Execute Prerequisites	12
Jupyter Notebooks Requirements	13
System Administration	14
Team Studio Licensing	14
Installation	14
Prerequisite Checklists	14
Server Prerequisites	14
Installation on the Team Studio Server	14
Installation for the R Server	16
Database Connection Prerequisites	16
Hadoop Connection Prerequisites	17
R Execute Prerequisites	19
Installing Quick Start	21
Planning the Installation or Upgrade Tasks	23
Preparing to Install Team Studio	23
Installer Configuration Options	25
Installer Help Options	25
Interactive Configuration Options	25
Automatic Installation	26
Manual Installation	28
Team Studio Default Ports	30
Default Ports for alpine.conf	30
Default Ports for server.xml	31
Default Ports for Jupyter Notebooks	32
Default Ports for Other Functions	32
Installing the R Connector	33
Port 6311	35
Team Studio Web Server	36
Configuring R Server	36
R Execute Prerequisites	38
R Execute Package Dependencies	39
Install and Configure Jupyter Notebooks	40
Jupyter Notebooks for Team Studio Spawners	41

Starting and Stopping the Docker Spawner	41
Starting and Stopping the Local Process Spawner	41
Python Packages Required for Jupyter Notebooks in Team Studio	42
Server Configuration	42
Server Ports	42
Setting the Port Number for Team Studio (Chorus) Server	42
Setting the Port Number for Team Studio (Alpine) Configuration	43
Post-installation Configuration Options	43
Configuring an External Server to Import Data with gpfdist	44
Installing a Database Server Certificate in the Team Studio JVM	45
Configuring JVM Command-Line Options	45
Configuring the HDFS Directory and Permissions for Results File Storage	46
Team Studio Related HDFS Configuration	46
Deleting Temporary Files	47
Database Stored Procedures	47
Installing Stored Procedures on Greenplum	47
Uninstalling Stored Procedures on Greenplum	48
Installing Stored Procedures on HAWQ	49
Uninstalling Stored Procedures on HAWQ	50
Installing Stored Procedures on PostgreSQL	51
Installing Team Studio DLLs in a PostgreSQL database on Windows	52
Uninstalling Stored Procedures on PostgreSQL	53
Security	53
Configure a Kerberos-Enabled Hadoop Data Source	53
Kerberos Authentication Integration Steps	54
Generate the User Account	54
Generating the User Account: LDAP	54
Generating the User Account: Non-LDAP	54
Generate the Keytab and Principal	55
Generating the Keytab and Principal on a Linux Server	55
Generating the Keytab and Principal on a Windows Server	55
Copying the Keytab to the Team Studio Server	56
Hadoop Cluster Configuration	56
Configuring HDFS and YARN	56
Setting HDFS Permissions	58
Setting HDFS Permissions When Upgrading	58
Configuring the Team Studio Server	59
TGT Generation	59
Adding the Data Source to Team Studio	59

Viewing Logging Information	60
Installing an SSL Certificate for a Database Connection	61
Configuring and Installing an SSL Certificate for the Team Studio Server	61
Configuring and Installing an SSL Certificate for the Team Studio Server (Manual)	62
Enabling LDAP Authentication	64
Configuring LDAP	64
LDAP Configuration Properties	65
Adding LDAP Users	66
Removing LDAP Users	67
Troubleshooting LDAP Configuration	67
LDAP Use Case Scenarios	68
Scenario 1: LDAP Authentication with Group Membership	68
Scenario 2: LDAP Authentication without Group Membership	69
Scenario 3: Import Users from an LDAP Group to Team Studio	69
Enabling Single Sign-On - SAML and Configuring SSO Options	70
Configuring the IDP	73
Change the SAML Log In Configuration for the Chorus Administrator	73
Configuring Kerberos in the Team Studio Client	74
Prerequisites for Configuring Kerberos in the Team Studio Client	74
Step-by-Step Guide to Configuring Kerberos in the Team Studio Client	75
Storing a Keytab for Jupyter Notebooks Running Python	77
Command-line Utilities for Managing the Services	77
Starting Team Studio	77
Stopping the Team Studio	78
Restarting the Team Studio	78
Backing up Team Studio	79
Restoring Team Studio	80
Start, Stop, or Restart Individual Services	80
Configuring Team Studio to Run as a Service	81
Team Studio Configuration Files	82
Team Studio Deploy Properties	82
The Properties File	83
Configuring Indexing Frequency for Database Instances	83
Team Studio Configuration Properties	83
Team Studio Log Files	89
Download Logs	91
Administering Team Studio	92
Connecting Team Studio to Data Sources	92
Database Data Sources	93

Connect to a JDBC Data Source	93
Connect to a Hive JDBC Data Source	94
Hive JDBC on CDH, HDP, or PHD	94
Connect to an Oracle Database	96
Enable Oracle Databases	97
Connect to a Greenplum Database	97
Connect to a Pivotal HAWQ Database	98
Connect to an Amazon RedShift Data Source	100
Connect to a BigQuery Data Source	101
BigQuery Data Source Connection Tests and Troubleshooting	104
Hadoop Data Sources	104
Adding a Hadoop Data Source from the Command Line	104
Adding a Hadoop Data Source from the User Interface	105
Connecting to a Hive Data Source on Hadoop	109
Connect to a MapR 4.x Data Source	110
Connect to a Pivotal Hadoop (PHD) Data Source	114
Connect to a YARN-Enabled Data Source	115
Hadoop Data Source Connection Tests and Troubleshooting	116
Workflow Editor Preferences	116
Algorithm Preferences	117
System Preferences	118
Data Source Preferences	119
UI Preferences	120
Work Flow Preferences	121
Datetime Formats Preferences	122
Administrator Options in Team Studio	124
Email Configuration	127
Process Control	128
Usage Statistics	129
Data Visibility	130
Browsing Datasets In Your Workspace	130
Browsing Datasets In the Entire Application	131
Controlling Data Source Visibility	132
Controlling Data Source Permissions	133
Adding Data to a Workspace	134
Data Source Associations	134
Associating a Data Source	135
Data Source Credentials	135
Data Administrators	136

- Data Source States 136
- Team Studio Licensing 138
- Manage Team Studio Users 138
 - Managing User Profiles 139
 - Add a New Person 140
 - Establish Your Identity 141
- Deployment Targets 141
- Upgrading 142
 - Preparing to Upgrade Team Studio 142
 - Running the Upgrade 143
- TIBCO Documentation and Support Services 145**
- Legal and Third-Party Notices 146**

System Requirements

The server on which you install Team Studio must meet these minimum requirements.

The following hardware, software, operating systems, third-party tools, and browsers have been tested with this version of Team Studio. For information about supported data sources, see "Operator Compatibility" in *TIBCO® Data Science Team Studio User's Guide*.



Some data sources have been deprecated and will no longer be supported after version 6.5. For a list of supported operators, see [Supported Database Platforms](#) and [Supported Hadoop Platforms](#). For a list of deprecated operators, see *TIBCO® Data Science Team Studio Release Notes*.

- For more information about using Jupyter Notebooks for Team Studio, see [Jupyter Notebook Requirements](#).
- To use the R Execute operator, you must have R Server installed on a separate computer. See [R Execute Prerequisites](#) for more information.


Hardware requirements

Item	Requirements
Dedicated server	
Redundant power	
Hard disk space	Disk space storage requirements (approximately 500 GB total, using RAID 1 mirroring) <ul style="list-style-type: none"> • 200GB storage file system for /usr/local/chorus • 200GB storage file system for /data/chorus • 50GB storage file system for /tmp • 50GB storage file system for /home/chorus
Processor	<ul style="list-style-type: none"> • Quad core or higher
RAM	<ul style="list-style-type: none"> • Minimum: 48 GB. • Recommended: 48+ GB (more recommended for improved performance).

Tested operating systems

Tested operating system	Tested versions ¹
Red Hat Enterprise	6.2 - 7.9 (64-bit)
CentOS	6.2, 6.5, 7.0, 7.6 (64-bit)

Required software

Installed on the server	Notes
Oracle Java 1.8 (JVM 64-bit) - See the Oracle download page .	This JDK is used by chorus user, not all users.  OpenJDK Java is not supported.
Bash Unix Shell	If you want to use another shell, contact technical support.

Tested third-party and TIBCO tools

Tool	Tested version
TIBCO Spotfire Analyst, TIBCO Spotfire Desktop (using the Export to SBDF operator)	10.3.x, 10.10.x, 11.2
Tableau Server	10.3
MADlib	1.15.1
PMML	1.3
open-source R	3.6

Supported browsers

Browser	Version
Microsoft® Internet Explorer	11
Google® Chrome	90.0.4430.212

Supported Database Platforms

You can use any of the following database platforms in Team Studio. Deprecated versions of databases will be removed from support in a future version of Team Studio.

Platform	Version	Data source	Analytic data source	Sandbox
Amazon RedShift	JDBC 4.2 compatible driver	Yes		
Apache Impala	2.2	Yes		
Azure SQL Data Warehouse		Yes		
Google BigQuery	BigQuery JDBC 4.2	Yes		

Platform	Version	Data source	Analytic data source	Sandbox
Greenplum database	5	Yes	Yes	Yes
Hive JDBC	13	Yes		
MS SQL Server	Recommended: 12.0, 13.0 (2014, 2016) or higher Deprecated in 6.5.0: 11.0 (2012)	Yes		
Oracle database	10g, 11g, 18c, Exadata	Yes	Yes	
Pivotal HAWQ	Deprecated in 6.5.0: 1.2	Yes	Yes	Yes
PostgreSQL	9.4 or higher. Deprecated in 6.5.0: <ul style="list-style-type: none"> 8.4 9.3 	Yes	Yes	Yes
SAP Hana	1.0 SPS 11 or higher	Yes		
Vertica	7.1 or higher	Yes		

Supported Hadoop Platforms

You can use any of these Hadoop platforms with Team Studio.

TIBCO® Data Science Team Studio version 6.6.0

Amazon EMR

- EMR 4.8.0
- EMR 5.0

Cloudera

CDH 6.3.4

Dataproc

Dataproc 1.3

TIBCO® Data Science Team Studio version 6.5.0

Amazon EMR

- EMR 4.8.0
- EMR 5.0

Cloudera

CDH 5.3 - CDH 6.2

Dataproc

Dataproc 1.3

HAWQ

- HAWQ 1.3 on PHD3.0
- HAWQ 1.3 on HDP 2.2

Hive

Default Hive for supported CDH and HDP versions

Hortonworks

- HDP 2.2
- HDP 2.3

IBM Big Insights

- Big Insights 4.0
- Big Insights 4.1

MapR

²

- MapR 4.0.1
- MapR 4.0.2
- MapR 4.1

PivotalHD

³

PHD 3.0

TIBCO® Spotfire Data Science version 6.4.x and Alpine® 6.3.x**Amazon EMR**

- EMR 4.8.0
- EMR 5.0

Cloudera

CDH 5.3 - CDH 6

HAWQ

- HAWQ 1.3 on PHD3.0

² Requires JDK7

³ Requires JDK7

- HAWQ 1.3 on HDP 2.2

Hive

Default Hive for supported CDH and HDP versions

Hortonworks

- HDP 2.2
- HDP 2.3

IBM Big Insights

- Big Insights 4.0
- Big Insights 4.1

MapR

⁴

- MapR 4.0.1
- MapR 4.0.2
- MapR 4.1

PivotalHD

⁵

PHD 3.0

⁴ Requires JDK7

⁵ Requires JDK7

R Execute Prerequisites

The R Execute operator was tested on 64-bit CentOS Linux with open-source R version 3.6.

The R Connector for TIBCO® Data Science Team Studio might work with newer versions of open-source R, but has been tested only with the specified versions.



You can configure the R Connector to use TIBCO® Enterprise Runtime for R (TERR™). For detailed instructions, see the article on the TIBCO Community site. Using TERR with Team Studio is currently untested and unsupported.

The R Connector for Team Studio should work with any open-source R server that is fully compatible with open-source R.



Team Studio can be extended for use with the R language and environment for statistical computing and graphics (see <https://www.r-project.org/>) through use of the R Connector for Team Studio, which is subject to free open source software license terms and available for download from a publicly available Github repository. The R Connector is not part of the Team Studio product and therefore not within the scope of your license for the product. Accordingly, the R Connector is not covered by the terms of your agreement with TIBCO for the Team Studio product, including any terms concerning support, maintenance, or warranties. Download and use of the R Connector is solely at your own discretion and subject to the free open source license terms applicable to the R Connector.

Requirement	Description
Hardware Requirements	<p>The R Connector for Team Studio requires a CentOS or RedHat operating system running on a 50GB server.</p> <div> <p>Windows servers are not supported. For details, see http://rforge.net/Rserve/rserve-win.html</p> <p>The Team Studio Web Server does not run on the same computer as the R Connector for Team Studio. Please see below for details.</p> </div>
Supported Java Versions	<p>The R Connector for Team Studio requires Oracle 64-bit Java 1.8 or higher to be installed.</p> <div> <p>The JRE is adequate, the JDK is not required.</p> </div>

In addition to system requirements, R Execute has R package requirements. See the following for more information about packages, and installing the R Connector.

Jupyter Notebooks Requirements

Ensure your environment meets the following requirements for using Jupyter Notebooks for Team Studio.

These recommendations are based on user count. For example, if you have 10 users, then you get (510MB * 10) + 2GB RAM, 10GB + 10GB disk space, and (10*.5) + .5 CPU cores. This projected use computes out to about 8GB of RAM, 20GB of free disk space, and 6 CPU cores.

With PySpark (Team Studio version 6.2 and later)

- Memory and disk space required per user: 1GB RAM + 1GB of disk + .5 CPU core.
- Server overhead: 2-4GB or 10% system overhead (whatever is larger), .5 CPU cores, 10GB disk space.
- Port requirements: Port 8000 plus 5 unique, random ports per notebook.
- Oracle JDK 1.8 is required on the Hadoop Data Source.

Without PySpark (Team Studio version 6.0 or 6.1)

- Memory and disk space required per user: 512MB RAM + 1GB of disk + .5 CPU core.
- Server overhead: 2-4GB or 10% system overhead (whatever is larger), .5 CPU cores, 10GB disk space.
- Port requirements: Port 8000.

To install Jupyter Notebooks for Team Studio in the Team Studio system, see the instructions in *TIBCO® Data Science Team Studio Installation and Administration* .

System Administration

As a system administrator for TIBCO® Data Science Team Studio, you might need to know how to work with log files, how to free up memory, and how to configure custom properties for the application from the command line.

You can find articles about these tasks in this documentation.

Team Studio Licensing

Team Studio has a licensing model that allows granularity and access control for users of the application. This affects all levels of the business organization, from data scientists to the front-line business user.

This topic explains the roles in the application and the permissions granted with each role. For more information on your license terms, contact your Team Studio Account Manager.

To view license information from within Team Studio, in the top right corner, click your user name, and then click **About**. This displays information about features that are enabled on your Team Studio instance. For more information on the licensed roles and user counts, from the sidebar menu, click **Administration** > **License Information**, or see [Administrator Options in Team Studio](#).

Installation

Before you install Team Studio, plan the installation, and then prepare your system for the installation. You can perform a quick installation, or you can specify options for the installation.

Prerequisite Checklists

The following checklists are intended to help you check your system's readiness for Team Studio.

Server Prerequisites

These checklists are provided to help ensure that all Team Studio components of a typical database installation are accounted for and completed.

Installation on the Team Studio Server

Installation requires unzipping the Team Studio installation bundle and running a startup script.

Team Studio must have access to the network and the computer specified as the dedicated Team Studio server for the duration of installation and configuration.

The startup script installs the application into two directories. After Team Studio gains access to the server, no further assistance is required during the installation step.

It takes approximately one hour to get Team Studio running and test the installation. After you complete the installation, you must configure the connection to the relevant data sources (database or Hadoop).

Question	Response	For Reference
Who will be assisting Team Studio and what will his/her availability be during installation?		
Is Oracle Java Version Java 1.8 or later installed? (Open Java is not supported.)		

Question	Response	For Reference
How much memory and disk space are available on the server?		
Are the following ports available on the Team Studio server?		
8080		
8543		
8983		
8443		
8000		
8001		
389		
587		
9090		
9091		
9092		
9093		
9094		
9095		
2570		
2571		
2572		
2573		
2574		
2575		
8005		
8009		

Question	Response	For Reference
3000		
3553		
3554		
3555		
3556		
3557		

Installation for the R Server

This checklist helps you prepare for a connection between Team Studio and a dedicated server for R-execute.

The R-execute feature requires a dedicated server with 50 GB hard drive space and additional configuration specified in the system requirements, dependencies, and installation information under [R Execute Prerequisites](#).

Question	Response	For Reference
Is Oracle Java 1.8 installed? (Open Java is not supported.)		
How much memory and disk space are available on the server?		
Are the following ports available on this server?		
2570		
2574		
6311		
2553		

Database Connection Prerequisites

This checklist is provided to help ensure that all Team Studio components of a typical database installation are accounted for and completed.

For connections to Greenplum and Oracle, installation requires SSH access to the database servers. The database (Greenplum and Oracle) connection configuration requires installing a set of functions in the database. These are available after the main Team Studio installation is complete in `$CHORUS_HOME/alpine-current/database_setup.zip`. This .zip file must be placed on a database server for installation.

Question	Response	For reference
Which database is installed?		
What are the IP address and port of the database host?		
Is there SSH access to the database?		
Are the database permissions set to allow connections from the Team Studio server?		
What is the name of the database to connect to?		
What user name and password do we use to connect to the database?		

Hadoop Connection Prerequisites

This checklist is provided to help ensure that all Team Studio components of a typical Hadoop-based installation are accounted for and completed.

The Hadoop connection configuration requires the HDFS host, HDFS port, Jobtracker host, and Jobtracker port. All Hadoop node hostnames must resolve to the proper computers from the Team Studio server. Team Studio needs access to a Hadoop administrator or anyone with access to the Hadoop configuration files (`*-site.xml`) if the inputs provided in the form below are not valid. Team Studio also might have to make changes to the host file of the Team Studio server if the Hadoop hostnames do not resolve.

The connection takes approximately two hours to configure and test if the Hadoop cluster is not configured for Kerberos. If it is, the user running Team Studio on the Team Studio server must have a keytab to authenticate in Kerberos. Team Studio requires that keytabs for the NameNode and Jobtracker are located on the Team Studio server. If any of these three elements is missing or invalid, Team Studio requires that a Hadoop administrator is available to contact during installation. Configuring the initial connection to a cluster configured for Kerberos takes approximately four hours.

Hadoop Cluster

Question	Response	For Reference
Which version of Hadoop is installed?		
Will a Hadoop administrator be available during installation?		
Is the NameNode of the resource manager enabled for high availability?		
Is the cluster configured for Kerberos?		

Question	Response	For Reference
Is the cluster running MapReduce (MRv1) or YARN (MRv2)?		
Do the HDFS and JobTracker/resource manager hostnames resolve to the correct computers from the Team Studio server?		If they do not, configure the hosts file so that these Hadoop hosts resolve properly.

Hadoop Cluster without High Availability

Question	Response	For Reference
What are the HDFS host and port?		Can be found in <code>core-site.xml</code> as <code>fs.default.name: hdfs://HDFSHOST:HDFSPO</code>

Hadoop Cluster with High Availability

Question	Response	For Reference
What is the name of the name service?		Can be found in <code>hdfs-site.xml</code> as <code>dfs.nameservices: hdfs://nameservice1</code>
What is the value for <code>dfs.ha.namenodes.<nameservice></code> ?		Can be found in <code>hdfs-site.xml</code> using the name of the name service.
What are the values for <code>dfs.namenode.rpc-address.<nameservice>.<namenode></code> ?		Can be found in <code>hdfs-site.xml</code> using the name of the name service, and each NameNode specified in the previous row.
What is the value for <code>dfs.client.failover.proxy.provider.<nameservice></code> ?		Can be found in <code>hdfs-site.xml</code> using the name of the name service.

MapReduce (MRv1)

Question	Response	For Reference
What are the Job host and port?		Can be found in <code>mapred-site.xml</code> as <code>mapred.job.tracker: hdfs://JOBHOST:JOBPORT</code>

YARN (MRv2)

Question	Response	For Reference
What is the YARN resource manager address?		Can be found in <code>yarn-site.xml</code> as <code>yarn.resourcemanager.address</code>

Kerberos (Ignore if Kerberos is not enabled)

Question	Response	For Reference
Is there a keytab that authenticates the Team Studio server?		
Are these required keytab files (merged or unmerged) located on the Team Studio server?		

R Execute Prerequisites

Before your users can use the R Execute operator, confirm that your Team Studio environment meets all requirements, including having installations of the supported version of R and Java on tested hardware configurations. Additionally, R Execute requires packages available from the Comprehensive R Archive Network (CRAN).

Supported R Versions

The R Execute operator was tested on 64-bit CentOS Linux with [open-source R](#) version 3.6.

The R Connector for Team Studio might work with newer versions of open-source R, but it has been tested only with version 3.6.

The R Connector for Team Studio should work with any open-source R server that is fully compatible with the open-source version of R.



The Team Studio product can be extended for use with the R language and environment for statistical computing and graphics (see <https://www.r-project.org/>) through use of the R Connector for Team Studio, which is subject to free open-source software license terms and available for download from a publicly available GitHub repository. The R Connector is not part of the Team Studio product and therefore is not within the scope of your license for the product. Accordingly, the R Connector is not covered by the terms of your agreement with TIBCO for the Team Studio product, including any terms that concern support, maintenance, or warranties. Download and use of the R Connector is solely at your own discretion and subject to the free open-source license terms applicable to the R Connector.

Required Server Hardware

The R Connector for Team Studio requires a CentOS or RedHat operating system running on a 50GB server.



Windows servers are not supported.



The Team Studio Web Server does not run on the same computer as the R Connector for Team Studio. Please see below for details.

Supported Java Versions

The R Connector for Team Studio requires Oracle 64-bit Java 1.8 or higher to be installed. (The JRE is adequate; the entire JDK is not required.)

These requirements are consistent with version 6 requirements.

R Packages

- [Rserve](#) (1.7-3),
- [data.table](#) (1.9.4 or higher).

`data.table` depends (directly or transitively) on the following:

- [Rcpp](#) (0.11.3 or higher),
- [plyr](#) (1.8.1 or higher),
- [stringr](#) (0.6.2 or higher),
- [chron](#) (2.3-45 or higher)
- [reshape2](#) (1.4.1 or higher).

These packages should be installed automatically upon service start-up (using the `start_services.sh` shell script) if they are not in the R local repository, because the R connector ships with the `.tar` files that contain the package sources. However, if there are file permission conflicts, the installation can fail. In such a case, first try fixing file permissions. If that does not help, try installing the packages manually.

```
install.packages(pkgs = c('/full/path/Rserve_1.7-3.tar.gz', '/full/path/
data.table_1.9.4.tar.gz'), repos = NULL, type='source')
```

You must supply the full path to the `.tar.gz` files, which is the directory in which you unzipped the R server `.tar` file.

Manual installation should never be necessary due to the automation built into the `start_r_component.R` script, which is called by the `start_services.sh` shell script. However, if for any reason you need to perform manual installation from the `.tar` files, note that the package installation order matters, due to the dependency graph. The recommended package order is as follows.

Rcpp, plyr, stringr, chron, reshape2, Rserve, data.table.

If you must install the packages manually, and you have an internet connection, the dependencies are downloaded automatically. In that case, you can install just the packages Rserve and data.table.

```
install.packages(pkgs = c('Rserve', 'data.table'))
```

You might need to select the CRAN repository programmatically; otherwise, you might be prompted to supply the name of the CRAN repository to use. A programmatic selection looks like the following.

```
install.packages(pkgs = c('Rserve', 'data.table'), repos='http://
cran.cnr.berkeley.edu/')
```

You can find the list of CRAN mirrors [here](#).

Port 6311

Port 6311 must be accessible at least on the localhost of the computer running the R Connector. This is the port on which the native component in R (Rserve) is listening to accept connections from Java (see [Rserve faq](#) for more details). This port need not be exposed to remote computers, just by using localhost/loopback.

To determine whether the port is open, try the following on the computer running the R Connector.

```
$netstat -anp | grep 6311
```

Not all processes could be identified, non-owned process info will not be shown, you would have to be root to see it all.)

```
tcp00127.0.0.1:63110.0.0.0:* LISTEN8627/Rserve
```

If netstat is not available, you can try nmap, telnet, or other tools to test whether this port is available at least on localhost.

Team Studio Web Server

The Team Studio Web Server does not run on the same computer as the R Connector for Team Studio.

R jobs can be computationally intensive, and long-running jobs should not starve the web application. For this reason, a port must be opened between the computer running the Team Studio Web Server and the one running the R Connector for Team Studio. The default port on which the R Connector for Team Studio is listening for remote connections is 2553. This setting can be changed in the `application.conf` file before starting the R Connector for Team Studio services using `start_services.sh`.

To check the firewall rules, check the `iptables` settings. (This might not be enough if there are hardware firewalls in place, or other issues.) The `iptables` command is high-priority; therefore, it usually requires superuser privileges. If you can use `sudo` try running `iptables` with `sudo`, as follows.

```
$sudo iptables -L
```

```
Chain INPUT (policy ACCEPT)
target prot opt source destination
```

```
Chain FORWARD (policy ACCEPT)
target prot opt source destination
```

```
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

Installing the R Connector

See [Installing the R Connector](#) for detailed information.

Installing Quick Start

For a basic installation, follow these steps.

Perform this task on a dedicated Linux server.

Prerequisites

- You must install on a Linux platform. See [System Requirements](#) for more information.
- You must have root access.

Procedure

1. Unzip the Team Studio download file into the home folder, for example `/home/dsts`.

2. Verify the integrity of the installer file by running the following commands. (Compare the output to make sure it matches.)

```
cat CHECKSUM
md5sum TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

3. Make the installer executable by running the following command.

```
chmod +x TIB_sfire-dsc-6.6.0_linux_x86_64.sh.
```

4. As the root user, run `./TIB_sfire-dsc-6.6.0_linux_x86_64.sh`.



If the installer fails with the following message

```
-bash: ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh: /bin/sh: bad interpreter: File too large
```

then as a workaround, run the following instead.

```
bash ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

This error is caused by the installer's large file size on some systems.

The license agreement is displayed.

5. Type `y` to continue.
6. Specify the user that should run Team Studio processes.
By default, this user is named `chorus`.
7. Enter the full path to the Team Studio installation directory and Team Studio data directory. By default, they are set to `/usr/local/chorus` and `/data/chorus`. Keep in mind that any directory you choose must be writable by the user. Verify the amount of free space on your volume by running `df -h`.
This key is stored as `server.key` in the shared data directory. By default this path is `/usr/local/chorus/shared/server.key`.
8. Enter your passphrase.
Your passphrase can be any combination of alphanumeric characters. This passphrase is used to generate a secret key to use to recover passwords from the database. Write it down and keep it in a safe place.
9. Choose whether to install the Team Studio workflow editor.
The installer performs a series of checks for a supported operating system version, the running user, and the Java version. If these requirements are not met, the installer quits.

When the installer completes, it displays a message indicating a successful installation.

```
Team Studio successfully installed:
Install Directory: /usr/local/chorus
Data Directory: /data/chorus
Version: <version>
Would you like to modify the system settings for Team Studio?
System settings can be modified later by executing chorus_control.sh configure. [y/N]:
Further information on these configuration options can be found at Team Studio
Configuration Properties.
```

10. Run the command `source chorus_path.sh`.
This command sets necessary environment variables
11. Switch to the user that was created to run Team Studio processes.
By default this user is `chorus`.
12. Put your `chorus.license` in the `$CHORUS_HOME/shared` directory.
13. Ensure the temporary file location is set to `alpine.java.io.tmpdir=$CHORUS_DATA/tmp` in `deploy.properties`.

What to do next

In your `$CHORUS_HOME` directory, start Team Studio, change to the chorus user by running `su - chorus`, and then type `./chorus_control.sh start`.

Planning the Installation or Upgrade Tasks

Use this overview task list to plan for installing or upgrading Team Studio.

Procedure

1. Review the [system requirements](#).
2. Prepare to install the software.
3. Determine if you are following the installation or the upgrading path.
 - Install the software.
 - Upgrade the software.
4. Ensure that you can start and stop the services.
5. If you are using stored procedures for your database, follow those instructions, found [here](#).
6. Open a browser (Chrome, Firefox, or Internet Explorer), and then navigate to `http://dsts_server_hostname:8080`.
7. Log in as the default admin user (`chorusadmin/secret`) to get started.

Preparing to Install Team Studio

Before you install Team Studio, you configure the server to prepare for the installation.

Perform this task on a Linux server.



If you are installing on a DCA, use PuTTY to establish an ssh connection to the GPDB Standby Master or DIA Module.

Prerequisites

Verify that your environment meets the prerequisites specified in [System Requirements](#).

Procedure

1. Create the user chorus at the shell prompt.

```
# useradd chorus
# groupadd chorus
# passwd chorus
```

When you enter the command `passwd chorus`, you are prompted for the password. For a DCA installation, specify `choruschorus`. For a non-DCA installation, you can specify anything. If you specify `chorus` as the password, the installation displays a message (as shown in the following example) about not using a dictionary word. However `chorus` is accepted after you enter it a second time.



```
[root@smdw/]# passwd chorus
Changing password for user chorus.
New UNIX password:<Enter chorus here>
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:<Enter chorus again>
passwd: all authentication tokens updated successfully.
[root@smdw/]#
```

Although the system does not display a confirmation, `chorus` is accepted as the password.

2. Switch user to `chorus`.

```
# su - chorus
```

3. Update the user's `.bash_profile`.

- a) Open a text editor.

```
$ vi ~/.bash_profile
```

- b) Add the following line to the `.bash_profile` to set the `JAVA_HOME` variable.

```
export JAVA_HOME=/usr/java/latest
```

- c) Close `.bash_profile` and source it to activate the changes.

```
$ source ~/.bash_profile
```

- d) Return to root.

```
$ exit
```

4. Create the installation and data directories as the root user.

- a) Create the path for the installation binaries.

```
# mkdir -p /usr/local/chorus
# chown -R chorus:chorus /usr/local/chorus
```

- b) Create the path for shared data.

```
# mkdir -p /data/chorus
# chown -R chorus:chorus /data/chorus
```

`/usr/local/chorus` and `/data/chorus` are the directories that are suggested to you when you run the installation script. You can substitute any directory names as long as they are owned by the `chorus` user.



Greenplum recommends 500GB of free disk space for production level usage. You can run the `df -h` command as root to see the free space you have on your mounted file systems. 1GB is sufficient for trial usage.



If you are installing on a DCA, skip the remaining steps.

5. Verify kernel settings as the `chorus` user.

Maximum user processes must be ≥ 131072 . Number of open files must be ≥ 65536 .

```
$ ulimit -a
```

- a) As root user, modify the lines below in two locations.

- In the file `/etc/security/limits.conf`.
- In the file `/etc/security/limits.d/90-nproc.conf`.

```
* soft nfile 65536
* hard nfile 65536
* soft nproc 131072
* hard nproc 131072
```


- Set the following parameters in `/etc/sysctl.conf`.

```
kernel.shmmax = 500000000
kernel.shmall = 4000000000
```

- If you made the changes to the configuration parameters in steps 5 and 6, restart the server.

Installer Configuration Options

Use the Team Studio installer configuration options to get information and customize the installation.

Installer Help Options

Use the following command-line commands to get help on the installer for Team Studio.

Installer help and information

Option	Description
<code>TIB_sfire-dsc-6.6.0_linux_x86_64.sh --help</code>	Print a help message that contains the available configuration options before you run the installer.
<code>TIB_sfire-dsc-6.6.0_linux_x86_64.sh info</code>	Print embedded information (the title, the default target directory, and the embedded script).
<code>TIB_sfire-dsc-6.6.0_linux_x86_64.sh --lsm</code>	Print the embedded LSM entry, or indicate if there is no LSM. (LSM files describe a software package in an easily-parseable way.)
<code>TIB_sfire-dsc-6.6.0_linux_x86_64.sh --list</code>	Print a list of the files in the archive.
<code>TIB_sfire-dsc-6.6.0_linux_x86_64.sh --check</code>	Check the integrity of the archive.

Interactive Configuration Options

The Team Studio installer command is `TIB_sfire-dsc-6.6.0_linux_x86_64.sh`. By appending the following options to this command, you can customize the configuration as described.

Installer options

Example

```
TIB_sfire-dsc-6.6.0_linux_x86_64.sh --silent --chorus_user=JDoe
```

Option	Description
<code>--disable_spec</code>	Skip the system requirement checks. If you do not have a supported system, you can still attempt to run the installer using this option, but it is not recommended.
<code>--chorus_user=CHORUS_USER</code>	Specify the Team Studio user. (The default is <code>chorus</code>)
<code>--chorus_path=CHORUS_PATH</code>	Specify the Team Studio installation path. (The default is <code>/usr/local/chorus</code> .)
<code>--data_path=DATA_PATH</code>	Specify the Team Studio data directory (The default is <code>/data/chorus</code> .)

Option	Description
<code>--passphrase=PASSPHRASE</code>	Specify the passphrase for encrypting and recovering passwords (the default is ' '.)
<code>--chorus_only</code>	Set up only the Team Studio collaboration framework; do not install the Team Studio workflow editor.
<code>-s, --silent</code>	Run script silently. If you choose this option, you must specify the necessary parameters listed above.

Using these advanced options is not recommended, except in a debug or recovery procedure. In the following example, a user needs to recover a file from the installer package, the user can extract the contents without running the install script to a specified directory.

Example

```
./TIB_sfiredsc-6.6.0_linux_x86_64.sh --noexec --keep --target /tmp/
```

Advanced installer configuration options

Option	Description
<code>--confirm</code>	Ask before running embedded script. This refers to the script that configures and installs Team Studio after the files are extracted.
<code>--noexec</code>	Do not run the embedded script.
<code>--keep</code>	Do not erase the target directory after running the embedded script.
<code>--nox11</code>	Do not spawn an xterm instance. Use this option if you are installing Team Studio on a headless server.
<code>--nochown</code>	Do not change the owner of the extracted files.
<code>--target NewDirectory</code>	Extract the installer files in the specified new directory.
<code>--tar arg1 [arg2 ...]</code>	Access the contents of the archive through the tar command.

Automatic Installation

The automatic installation is the preferred installation method. Perform this task on the server where Team Studio is installed.

Prerequisites

- You must have root access. If you do not have root access, see [Manual installation](#).
- You must be installing on a supported Linux platform. See [System requirements](#).

Procedure

- Copy the Team Studio download file into the home directory (for example, `/home/dsts`).

2. Run the following commands and compare the output to ensure it matches.

```
cat CHECKSUM
md5sum TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

This comparison ensures the integrity of the installer file.

3. If you have Team Studio installed and are running an upgrade, stop Team Studio by running the following command.

```
chorus_control.sh stop
```

4. Make the installer executable by running the following code.

```
chmod +x TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

5. As the root user, run the following code.

```
./TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```



On some systems, the installer large file size can cause the installer to fail with the following message.

```
-bash: ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh: /bin/sh: bad interpreter: File too large
```

If this happens, run the following workaround code.

```
bash ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh instead.
```

The license agreement is displayed.

6. Type `y` to continue.
7. Specify the user to run the Team Studio processes.

By default, this user is `chorus`.

8. Enter the full path to the installation directory and the data directory.

By default they are set to `/usr/local/chorus` and `/data/chorus`. Remember that any directory you specify must be writable by the user.

a) Verify the amount of free space on your volume by running `df -h`.

9. Enter your passphrase.

Your passphrase can be any combination of alphanumeric characters. This phrase is used to generate a secret key to be used for recovering passwords from the database. Write it down and keep it in a safe place.



This key is stored as `server.key` in the shared data directory. By default this path is `/usr/local/chorus/shared/server.key`.

10. Choose whether to install Team Studio.

The installer performs a series of checks for a supported OS version, the running user, and the Java version. If these requirements are not met, the installer quits.

If the installer completes successfully, it displays a success screen similar to the following:

```
*****
TIBCO Data Science Team Studio successfully installed:
Install Directory: /usr/local/chorus
Data Directory: /data/chorus
Chorus Version: <version>
Alpine Version: <version>
*****
Would you like to modify the system settings
for Alpine Chorus?
System settings can be modified later by
executing chorus_control.sh configure. [y/N]:
```

You can find more information on these configuration options at Team Studio [configuration properties](#).

11. Run the following command to set necessary environment variables.

```
source chorus_path.sh
```

12. Switch to the user that was created to run Team Studio processes.

By default this user is chorus.

13. Put your chorus.license in the \$CHORUS_HOME/shared directory.
14. Open the file deploy.properties and make sure the temporary file location is set to alpine.java.io.tmpdir=\$CHORUS_DATA/tmp.
15. Change to the chorus user by running the following command.

```
su - chorus
```
16. Change to the \$CHORUS_HOME directory.
17. Type the following command.

```
./chorus_control.sh start
```


 After about 15 seconds, Team Studio starts.
18. Open a Chrome or Firefox browser, and navigate to <tsdshost>:8080.
19. Log in as siteadmin using the default password.
20. Update the password immediately.

Manual Installation

Follow these steps to perform a manual installation of Team Studio.

Install on a support Linux platform. See [System Requirements](#) for more information.

Prerequisites

- Perform all steps in [Preparing to Install Team Studio](#).
- An administrator has configured a Team Studio user or group and assigned the required permissions for the user to create directories in the installation and data locations.

Procedure

1. Copy the Team Studio download file into the home directory (for example, /home/dsts).
2. Run the following commands and compare the output to ensure it matches.

```
cat CHECKSUM
md5sum TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

This comparison ensures the integrity of the installer file.

3. Make the installer executable by running the following code.

```
chmod +x TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

4. As the chorus (or other non-root user), run the following code.

```
./TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```



On some systems, the installer large file size can cause the installer to fail with the following message.

```
-bash: ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh: /bin/sh: bad interpreter: File too large
```

If this happens, run the following workaround code.

```
bash ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh instead.
```

The license agreement is displayed.

5. Type y to continue.
6. specify the user to run the Team Studio processes.

By default, this user is chorus.

7. Enter the full path to the installation directory and the data directory.

By default they are set to `/usr/local/chorus` and `/data/chorus`. Remember that any directory you specify must be writable by the user.

- a) Verify the amount of free space on your volume by running `df -h`.

8. Enter your passphrase.

Your passphrase can be any combination of alphanumeric characters. This phrase is used to generate a secret key to be used for recovering passwords from the database. Write it down and keep it in a safe place.



This key is stored as `server.key` in the shared data directory. By default this path is `/usr/local/chorus/shared/server.key`.

9. Choose whether to install Team Studio.

The installer performs a series of checks for a supported OS version, the running user, and the Java version. If these requirements are not met, the installer quits.

If the installer completes successfully, it displays a success screen similar to the following:

```
*****
TIBCO Data Science Team Studio successfully installed:
Install Directory: /usr/local/chorus
Data Directory: /data/chorus
Chorus Version: <version>
Alpine Version: <version>
*****
Would you like to modify the system settings
for Team Studio?
System settings can be modified later by
executing chorus_control.sh configure. [y/N]:
```

You can find more information on these configuration options at Team Studio [configuration properties](#).

10. Run the following command to set necessary environment variables.

```
source chorus_path.sh
```

11. Switch to the user that was created to run Team Studio processes.

By default this user is `chorus`.

12. Put your `chorus.license` in the `$CHORUS_HOME/shared` directory.

13. Open the file `deploy.properties` and make sure the temporary file location is set to `alpine.java.io.tmpdir=$CHORUS_DATA/tmp`.

14. Change to the `chorus` user by running the following command.

```
su - chorus
```

15. Change to the `$CHORUS_HOME` directory.

16. type the following command.

```
./chorus_control.sh start
```

After about 15 seconds, Team Studio starts.

17. Open a Chrome or Firefox browser, and navigate to `<dstshost>:8080`.

18. Log in as `siteadmin` using the default password.

19. Update the password immediately.

What to do next

After the installer has completed, you can perform additional configuration. Otherwise, you can indicate that you intend to finish the configuration later by typing `chorus_control.sh configure`.

Team Studio Default Ports

The installation of Team Studio sets ports for the configuration, XML, and other files and properties.

Default Ports for alpine.conf

The following default ports are assigned for specific functions for the configuration `alpine.conf`.

Ports	Function	File (relative to \$CHORUS_HOME)	Parameter	Computer accessing this port on the Team Studio server
2553	Akka Actor System	shared/ ALPINE_DATA_ REPOSITORY/ configuration/ alpine.conf	akka.remote. netty.tcp.port	Team Studio server
2570-2580	Akka ports	shared/ ALPINE_DATA_ REPOSITORY/ configuration/ alpine.conf	alpine.agent. baseAkkaPort (base port for all agents)	Team Studio server
3553-3563	Spark Runner Akka Ports (must be accessible from each cluster node)	shared/ ALPINE_DATA_ REPOSITORY/ configuration/ alpine.conf	alpine.spark. sparkAkka.akkare. remote.netty.tcp. port (configures all runner ports)	Team Studio server, Cluster
8080	Team Studio user interface (must be accessible from each cluster node)	shared/ chorus.properties shared/ ALPINE_DATA_ REPOSITORY/ configuration/ alpine.conf	server_port alpine.chorus.port	Team Studio server, End user

Ports	Function	File (relative to \$CHORUS_HOME)	Parameter	Computer accessing this port on the Team Studio server
9090 ⁶ -9095	Hadoop Agents	shared/ ALPINE_DATA_ REPOSITORY/ configuration/ alpine.conf shared/ ALPINE_DATA_ REPOSITORY/ configuration/ deploy.properties shared/ chorus.properties	alpine.agent .baseHttpPort (base port for all agents) alpine.port workflow.url = http:// localhost:9090	Team Studio server
8891-8901	defined by alpine.agent.baseAkkaPort in alpine.conf			Team Studio server

Default Ports for server.xml

The following default ports are assigned for specific functions for the configuration server.xml.

Ports	Function	File (relative to \$CHORUS_HOME)	Parameter	Computer accessing this port on the Team Studio server
8005	Jetty shutdown	alpine-current/jetty-distribution-9.2.12.v20150709/conf/server.xml	<Server port="8005" shutdown="SHUTDOWN">	Team Studio server
8009	Jetty AJP 1.3 Connector	alpine-current/jetty-distribution-9.2.12.v20150709/conf/server.xml	<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />	Team Studio server

⁶ If you change port 9090, then also set alpine.chorus.port=8080 in alpine.conf.

Ports	Function	File (relative to \$CHORUS_HOME)	Parameter	Computer accessing this port on the Team Studio server
8443	Jetty SSL Redirect	alpine-current/jetty-distribution-9.2.12.v20150709/conf/server.xml	<pre><Connector port="\$ {port.http}" protocol="HTTP/ 1.1"connectionT imeout="20000"r edirectPort="84 43" /> <Connector port="8009" protocol="AJP/ 1.3" redirectPort="8 443" /></pre>	Team Studio server

Default Ports for Jupyter Notebooks

Team Studio is configured with certain port requirements.

Jupyter Notebooks for Team Studio with PySpark requires full communication between the server where Jupyter Notebooks for Team Studio is installed and all cluster nodes.

Spark-based operators used in Team Studio require full communication between the Team Studio server and all cluster nodes.

Between Team Studio and the server where Jupyter Notebooks for Team Studio is installed, the default communication port is set to 8000. You can change this port in the file `$CHORUS_HOME/shared/chorus.properties` if necessary.

Default Ports for Other Functions

The following default ports are assigned for additional functions in the specified files.

Ports	Function	File (relative to \$CHORUS_HOME)	Parameter	Computer accessing this port on the
3000	Team Studio user interface access port	current/vendor/jetty/jetty.xml current/vendor/nginx/nginx.conf.erb	<pre><Set name='port'>300 0</Set>, proxy_pass http:// 127.0.0.1:3000/</pre>	Team Studio server
8543	Team Studio Internal Postgres Port	shared/chorus.properties	postgres_port	Team Studio server
8983	Solr Port	shared/chorus.properties	solr_port	Team Studio server

Ports	Function	File (relative to \$CHORUS_HOME)	Parameter	Computer accessing this port on the
user-defined	<p>Range of ports that the MapReduce AM can use when binding.</p> <p>Leave blank if you want all possible ports. This option is more useful for users with a firewall who need to specify exact port ranges.</p> <p>Example port range: 50000-50050, 50100-50200</p>	cluster configuration	yarn.app.mapreduce.am.job.client.port-range	Cluster

Installing the R Connector

The R Server/R Connector is an open-source library that provides Team Studio access to open-source R.

The R-Server Connector for Team Studio is an optional package available for download and designed to run on a separate server to provide interoperability with R. Contact Team Studio Support for the latest version of the R Connector for Team Studio package.

Perform this task on a server in your Team Studio environment.



You can configure the R Connector to use TIBCO® Enterprise Runtime for R (TERR™). For detailed instructions, see [the article on the TIBCO Community site](#). Using TERR with Team Studio is currently untested and unsupported.

Prerequisites

- Oracle 64-bit Java 1.8 or higher must be installed. (The JRE is adequate; the entire JDK is not required.)
- You must be the root user to run the installation commands.
- The R Connector for Team Studio consists of an R-side and a Java-side component, and the two components must reside on the same computer on which R is running.
- The R Connector for Team Studio requires a CentOS or RedHat operating system running on a 50GB server.



Windows servers are not supported.



The Team Studio Web Server does not run on the same computer as the R Connector for Team Studio.

- Port 6311 must be accessible at least on the localhost of the machine running the R Connector. See [Port 6311](#) for more information.
- Dedicate a separate directory to the unpacked components.
- Additional prerequisites are described in [R Execute Prerequisites](#).

Procedure

1. Download the R Connector binary from the [SDS-R-Connector Github repository](#).
2. Unpack the R Connector archive in the dedicated directory you specified using the following code.

```
$tar xvf TIB_sfire-dscr_ver-#_linux_x86_64.tar
```

where `ver-#` is the product version you are running (for example, 6.6.0).

The unpacked components are in the specified directory, along with the original archive.



Do not delete any unpacked components; otherwise, the services will fail to start.

3. Review the unpacked components in the current directory, along with the original archive.

```
$ls
```

```
prepare_services.sh alpine-r-connector.jar start_services.sh
prepare_r_component.R Rserve_1.7-3.tar.gz stop_services.sh
application.conf start_r_component.R
```

4. Run the `prepare_services` script.

The prepare script installs the R-side native component (if it is not already installed).

```
./prepare_services.sh
```

5. Run the `start_services` script.

The start script will start the R-side service (if it is not already started) and start the Java-side service (if it is not already started).

```
./start_services.sh
```

If the start-up succeeds, you see something like the following.

```
Welcome to the R Server for Data Science Team Studio
Checking if R is installed.
Starting the R-side Rserve component of the service.
The log of the R-side component will can be
found in start_R_component.Rout
R-side service started.
Starting Java-side service. Its log will be
found in AlpineRConnector.log
Done - check start_R_component.Rout and AlpineRConnector.log
if you are experiencing problems.
```

6. Verify that the R-side component is running by checking for the Rserve processes.

There are $N + 1$ such processes, where N is the number of R workers selected.

You can modify this choice before starting the services.

a) Open the file `application.conf`.

b) Modify the parameter `numActors`.

This setting should be consistent with the number of cores (real or in the case of a VM, virtual) on the computer. The default is 10, for a total of 11 processes.

```
ps -ef | grep Rserve
rstudio 3911 1 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3968 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3969 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3970 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3971 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3972 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3973 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio 3974 3911 0 15:22 ? 00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
```

```

rstudio  3975 3911 0 15:22 ?  00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio  3976 3911 0 15:22 ?  00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio  3977 3911 0 15:22 ?  00:00:00 /usr/lib64/R/library/Rserve/libs//Rserve --no-
save
rstudio  3979 2149 0 15:22 pts/0 00:00:00 grep Rserve

```

On most systems, you can simply check the number of the Rserve processes to get a more terse output.

```
$pidof Rserve | wc -w
```

```
11
```

7. If you have only the JRE (and not the JDK), check the Java service with the following code from the command line.

```

$ps -ef | grep alpine-r-connector.jar
rstudio 3913  1 1 15:22 pts/0 00:00:05 java -Xmx4096M -Xms1024M -XX:MaxPermSize=512M
-XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -Dconfig.file=./
application.conf -jar ./alpine-r-connector.jar

```

8. Optional: To troubleshoot problems, check `start_r_component.Rout` and `AlpineRConnector.log` by printing them to the screen. See the following for an example.

```
$cat AlpineRConnector.log
```

```

Java version 1.8 is OK
Checking if chosen port 2553 is free
Port is free - starting actor system
[INFO] [10/31/2014 12:37:05.497] [main] [Remoting] Starting remoting
[INFO] [10/31/2014 12:37:05.846] [main] [Remoting] Remoting started; listening on
addresses :[akka.tcp://rServeActorSystem@10.0.0.180:2553]
[INFO] [10/31/2014 12:37:05.880] [rServeActorSystem-akka.actor.default-dispatcher-2]
[akka://rServeActorSystem/user/master]
Starting RServeMaster
[INFO] [10/31/2014 12:37:05.883] [rServeActorSystem-akka.actor.default-dispatcher-2]
[akka://rServeActorSystem/user/master]
Number of expected R workers = 4
...

```



The R Connector start-up script (`start_services.sh`) is idempotent; that is, running the script a second time does not cause anything else to fail. The services are already started, so both logs simply say that the ports are already taken, and no additional R-side (Rserve) or Java processes are started if the script gets run a second time.

9. To stop the services (Rserve native processes and the Java process), at the command line, type the following code.

```
./stop_services.sh
```

Port 6311

Port 6311 must be accessible at least on the localhost of the machine running the R Connector.

Port 6311 is the port on which the native component in R (Rserve) is listening to accept connections from Java (see <http://rforge.net/Rserve/faq.html> for more details). This port need not to be exposed to remote machines, just by way of localhost/loopback.

To check if the port is open, run the following command from the command line on the computer running the R Connector.

```
$netstat -anp | grep 6311
```

```
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
```

```
tcp  0  0 127.0.0.1:6311    0.0.0.0:*        LISTEN  8627/Rserve
```

If netstat is not available, you can try running nmap, telnet, or another port checker to test if this port is available at least on localhost.

Team Studio Web Server

The Team Studio Web Server does not run on the same computer as the R Connector for Team Studio.

R jobs can be computationally intensive, and long-running jobs should not starve the web application. For this reason, a port must be opened between the computer running the Team Studio Web Server and the one running the R Connector for Team Studio. The default port on which the R Connector for Team Studio is listening for remote connections is 2553.

You can change This setting in the file `application.conf`. You must change the setting before starting the R Connector for Team Studio services using `start_services.sh`.

To check the firewall rules, check the `iptables` settings (this might not be enough if there are hardware firewalls in place, or other issues). The `iptables` command is high-priority; therefore, it usually requires superuser privileges. If you are a "sudoer," try running `iptables` with `sudo`.

```
$sudo iptables -L

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

Configuring R Server

To use the R Execute operator, you must first configure the R server.

For more information, see "R Execute" in the *TIBCO® Data Science Team Studio User's Guide*.

Prerequisites

You must be an administrator to edit preferences.

Procedure

1. From the menu, click **Actions > Preferences**.
2. Select **R Server**.

The screenshot shows the 'Edit Preferences' window. On the left, a sidebar lists various preference categories: Algorithm, System, Data Sources, UI, Work Flow, Datetime Formats, and R Server. The 'R Server' category is currently selected. The main area displays two settings: 'R Server Host' with the value '10.0.0.180' and 'R Server Port' with the value '2553'. Below these fields are 'Save' and 'Restore' buttons. A 'Done' button is located at the bottom right of the dialog.

3. Provide the following information.

Option	Description
R Server Host	IP address for the R server host computer.
R Server Port	<p>Port number for the R server host computer. The default port is 2553.</p> <p>The port setting must match the entry for the R Server Port in the file <code>application.conf</code> located in the directory where the R Connector is installed.</p> <ul style="list-style-type: none"> • Default port: 2553. • To change this port while the R Connector java process is already running, you must find the process ID and kill it, change the port number in <code>application.conf</code>, and run all services again using <code>./start_services.sh</code>. • Restarting the R-side process (Rserve) is not necessary in this case. • Only then can you change the setting in the R Server Port dialog box in the Team Studio user interface. <p>Do not change the port unless there are specific reasons to change it.</p>

Changing the host and port settings results in an update within Team Studio without the need to restart the Team Studio web server.

However, because there may be open connections to a previously selected R server with a different host and port, it's important to make sure that all the important workflows that contain R Execute operators are not currently running, or if they are, that it is OK for them to fail and for the user to restart them.



- The moment the user starts running a new R Execute Operator, Team Studio checks to determine whether the R Server host and port changed.
- If the host and port changed, the currently-running flows that are pointing to a different server are terminated, because Team Studio is set up to handle one R Server at a time.

You might not need to restart Team Studio, but make sure that users are not currently running R Execute operators while you are changing the R Server host and port.

4. Click **Save** to save changes.

To return to default values, click **Restore**.



If the R server is behind NAT, make sure to use the R server's hostname instead of the IP address, and configure the hostname in the `application.conf` file of the `alpine-r-connector`.

R Execute Prerequisites

The R Execute operator was tested on 64-bit CentOS Linux with open-source R version 3.6.

The R Connector for TIBCO® Data Science Team Studio might work with newer versions of open-source R, but has been tested only with the specified versions.





You can configure the R Connector to use TIBCO® Enterprise Runtime for R (TERR™). For detailed instructions, see [the article on the TIBCO Community site](#). Using TERR with Team Studio is currently untested and unsupported.

The R Connector for Team Studio should work with any open-source R server that is fully compatible with open-source R.



Team Studio can be extended for use with the R language and environment for statistical computing and graphics (see <https://www.r-project.org/>) through use of the R Connector for Team Studio, which is subject to free open source software license terms and available for download from a publicly available Github repository. The R Connector is not part of the Team Studio product and therefore not within the scope of your license for the product. Accordingly, the R Connector is not covered by the terms of your agreement with TIBCO for the Team Studio product, including any terms concerning support, maintenance, or warranties. Download and use of the R Connector is solely at your own discretion and subject to the free open source license terms applicable to the R Connector.

Requirement	Description
Hardware Requirements	<p>The R Connector for Team Studio requires a CentOS or RedHat operating system running on a 50GB server.</p> <div>  <p>Windows servers are not supported. For details, see http://rforge.net/Rserve/rserve-win.html</p> <p>The Team Studio Web Server does not run on the same computer as the R Connector for Team Studio. Please see below for details.</p> </div>

Requirement	Description
Supported Java Versions	<p>The R Connector for Team Studio requires Oracle 64-bit Java 1.8 or higher to be installed.</p> <div>  <div>The JRE is adequate, the JDK is not required.</div> </div>

In addition to system requirements, R Execute has R package requirements. See the following for more information about packages, and installing the R Connector.

R Execute Package Dependencies

After you have met the hardware and Java requirements, and you have installed open-source R, install the R packages required to run R Execute.

- Rserve 1.7-3
- data.table 1.9.4 or higher.

Note that data.table also depends (directly or transitively) on the following:

- Rcpp (0.11.3 or higher)
- plyr (1.8.1 or higher)
- stringr (0.6.2 or higher)
- chron (2.3-45 or higher)
- reshape2 (1.4.1 or higher)

These packages should be installed automatically upon service start-up (start_services.sh shell script) if they are not in the R local repository, because the R connector ships with the tar files that contain the package sources. However, if there are file permission conflicts, the installation can fail. In such a case, first try fixing file permissions. If that doesn't help, try installing the packages manually. The following example shows installing packages from the R or TERR console command line (or from your installation of RStudio).

```
install.packages(pkgs = c('/full/path/Rserve_1.7-3.tar.gz', '/full/path/
data.table_1.9.4.tar.gz'), repos = NULL, type='source')
```



You must supply the full path to the tar.gz files, which is the directory in which you unzipped the R server .tar file.

Manual installation should never be necessary due to the automation built into the start_r_component.R script, which is called by the start_services.sh shell script. However, if you must perform manual installation from the .tar files, note that the package installation order matters, due to the dependency graph. The recommended package order is as follows.

1. Rcpp
2. plyr
3. stringr
4. chron
5. reshape2
6. Rserve
7. data.table

If you must perform manual installation, and you have an internet connection, the dependencies are downloaded automatically. In that case, you can install just Rserve and data.table.

```
install.packages(pkgs = c('Rserve', 'data.table'))
```



You might need to select the CRAN repository programmatically; otherwise, you might be prompted to supply the name of the CRAN repository to use. A programmatic selection looks like the following:

```
install.packages(pkgs = c('Rserve', 'data.table'), repos='http://
cran.cnr.berkeley.edu/')

```

You can find the list of CRAN mirrors at <https://cran.r-project.org/mirrors.html>.

Install and Configure Jupyter Notebooks

Using Team Studio, you can integrate Jupyter Notebooks for Team Studio to run Python code.



As of version 6.6.0, Team Studio requires Python 3.4 (or later) on the cluster.

Installation Preparation

For a non-Docker installation, before running the installer for Jupyter Notebooks for Team Studio, install the package `bunzip2`.

Jupyter Notebooks for Team Studio is installed using a shell script installer. You must install Jupyter Notebooks for Team Studio on a separate server from your Team Studio installation. The shell script installer requires the following information.

- The location of the installation.
The default path is `/opt/notebooks`.
- The Spawner type.
You can run Jupyter Notebooks for Team Studio in a Docker container on that server, or you can run uncontainerized as a local managed process. The default is **Docker**.
- Whether the installation SSL-enabled. If so, provide the following.
 - SSL certificate path.
 - SSL key path.

The default is **No**.

After the installer completes, review the output for warnings or errors, or for further instructions on configuration changes to the `chorus.properties` file, and for information on starting or stopping the service.

Port Configuration

By default, communication between the Team Studio server and the server where Jupyter Notebooks is installed occurs over port 8000. You can change this option in the file `$CHORUS_HOME/shared/chorus.properties`.

Configure Jupyter Notebooks to use the PySpark package. Make sure you have five random ports available to maintain full communication between the server where Jupyter Notebooks for Team Studio is installed and all cluster nodes.



For Spark to work, you must make sure that full communication is open between the Team Studio server and all cluster nodes.

Python Packages

For Jupyter Notebooks for Team Studio to work correctly, certain Python packages must be installed by the user. For a list of packages that Jupyter Notebooks for Team Studio requires for integration, see [Python Packages Required for Jupyter Notebooks in Team Studio](#).

For more information about Jupyter Notebooks, see jupyter.org.

Jupyter Notebooks for Team Studio Spawners

Jupyter Notebooks for Team Studio uses JupyterHub to orchestrate user notebook processes. JupyterHub uses a feature called "spawners".

Spawners create the notebook process on a given platform. See [JupyterHub](#) and [Spawners](#) for more information.

Currently, Team Studio supports Docker and local process spawners.

Starting and Stopping the Docker Spawner

The Docker spawner is the preferred spawner because it supports user isolation, which can help improve security.

Perform these tasks from the directory where Jupyter Notebooks is installed.

Procedure

1. To start the Docker spawner, run the following command.

```
docker-compose up -d
```

The Docker spawner is started.

2. To stop the Docker spawner, run the following command.

```
docker-compose down
```

The Docker spawner is stopped.

Starting and Stopping the Local Process Spawner

You can use a local process spawner if you do not want to use Docker as part of your Jupyter Notebooks for Team Studio configuration.

Perform these tasks from the directory where Jupyter Notebooks is installed.

Prerequisites

The Local Spawner requires creating a user account for each user in Team Studio who uses the Jupyter Notebooks Server. The user name is in the form *chorus-username*, where *username* is the user's User ID in Team Studio.

Procedure

1. To start the local process spawner, run the following command.

```
systemctl start tsnotebooks
```

The local process spawner is started.

2. To stop the local process spawner, run the following command.

```
systemctl stop tsnotebooks
```

The local process spawner is stopped.

Python Packages Required for Jupyter Notebooks in Team Studio

Team Studio requires specific Python packages for data analysis. Download these package versions to run Jupyter Notebooks for Team Studio. (Dependent packages download by default.)

Package name	Version number
jupyterhub	1.0.0
notebook	5.7.8
numpy	1.15.0
pandas	0.24.0
scikit-learn	0.19.2
scipy	1.1.0
seaborn	0.9.0
tensorflow	1.12.0

Server Configuration

After you install Team Studio, you can configure the workspace environment using settings, files, and properties described in this section.

Server Ports

The Team Studio application is a server running on a host machine at a specified port.

Team Studio chorus server accesses several other servers running on the same or different hosts, including the embedded Team Studio alpine server, Solr server, database servers, and other data sources, mail services, authentication servers (LDAP), and so on.



Each server on a host machine must be listening on a unique port.

Your installation of Team Studio chorus server can include an embedded third-party Team Studio alpine server to manage workflows if the Team Studio license your organization purchased provides for it. If this is the case, then you must consider the following issues.

- The Team Studio chorus and alpine servers must be configured to listen at different ports and they each must be configured to connect to each other at the correct port.
- This configuration is handled automatically during installation, but if you reconfigure either server's port, you must change the port in the other's configuration to match the new port.

The following tasks show examples of changing the properties and configurations for the chorus and embedded alpine services.

Setting the Port Number for Team Studio (Chorus) Server

The Team Studio chorus application is a server running on a host machine at a specified port. In this procedure, the port for `chorus.properties` is set to 8081.

Perform this task from a command line on the computer where Team Studio chorus server is installed. Stop the service before changing the configuration.

Prerequisites

You must have administrative privileges to change the properties and configuration files.

Procedure

1. Using a text editor, open the file `chorus.properties` for editing.

```
public_url = SDS.example.com
server_port
```

3. Set the port number to 8081.



8081 is an example value. You can use any available port number.

```
public_url = SDS.example.com
server_port = 8081
```

4. Using a text editor, open the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf`.
5. Set the chorus port property to the new port number in the alpine configuration.

```
alpine {
  chorus {
    scheme = HTTP
    port = 8081
  }
}
```

6. Restart the service.

Setting the Port Number for Team Studio (Alpine) Configuration

You can set the alpine port number for the alpine chorus configuration of Team Studio. In the procedure, the port number for alpine is 8090.

Perform this task from a command line on the computer where Team Studio chorus server is installed. Stop the service before changing the configuration.

Prerequisites

You must have administrative privileges to change the properties and configuration files.

Procedure

1. Using a text editor, open the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/deploy.properties`.
2. Set the property `alpine.port` to the new port number.

```
alpine.port = 8090
```

3. Using a text editor, open the file `$CHORUS_HOME/shared/chorus.properties`.
4. Change the port number in the `workflow.url` property to the new Team Studio server port number.
5. Restart the service.

```
workflow.url=http://localhost:8090
```

Post-installation Configuration Options

After the installer is finished, you can perform additional configuration options.

Optionally, you can set any of these configuration options at any time by opening a shell and typing `chorus_control.sh configure`.

Configuring an External Server to Import Data with gpfdist

To enable data movement between Greenplum databases, you must install and start gpfdist on the Team Studio host.

Prerequisites

You must start two processes: Start one process for writing and one process for reading, each with different ports but pointing to the same directory. See the Greenplum Database Administrator Guide on how to configure gpfdist.

Procedure

1. Download the gpfdist package and install it.
2. Copy the gpfdist files from `<greenplum installation>/bin/gpfdist` to `$CHORUS_HOME/vendor/gpfdist-rhel5/bin/A`.
3. Copy the following files from `<greenplum installation>/lib` to `$CHORUS_HOME/vendor/gpfdist-rhel5/lib`.

```
libcom_err.so
libcom_err.so.3
libcom_err.so.3.0
libcrypto.so
libcrypto.so.0.9.8
libk5crypto.so
libk5crypto.so.3
libk5crypto.so.3.1
libkrb5.so
libkrb5.so.3
libkrb5.so.3.3
libkrb5support.so
libkrb5support.so.0
libkrb5support.so.0.1
liblber-2.3.so.0
liblber-2.3.so.0.2.26
liblber.so
libldap_r-2.3.so.0
libldap_r-2.3.so.0.2.26
libpq.so
libpq.so.5
libpq.so.5.3
libssl.so.0.9.8
libz.so
libz.so.1
libz.so.1.2.3
```



Make sure these files are present in the vendor directory each time you upgrade Team Studio.

4. Examine the gpfdist entry in `<installation directory>/shared/chorus.properties`.

```
gpfdist.ssl.enabled= false
```

5. Set `gpfdist.ssl.enabled` to true if gpfdist is configured with SSL certificates.



SSL certificates must be installed on all segment servers.

```
gpfdist.url= sample-gpfdist-server
```

This URL must be the externally accessible URL that can be resolved from the source and destination servers.

```
gpfdist.write_port= 8000
gpfdist.read_port= 8001
gpfdist.data_dir= /tmp
```

6. Start gpfdist with the `write_port` value and the `data_dir` value.

7. Start gpfdist with the `read_port` value and the `data_dir` value.
8. Restart Team Studio to activate the changes.

What to do next

For more complete information about gpfdist, refer to the Greenplum Database Administrator Guide.

Installing a Database Server Certificate in the Team Studio JVM

When Team Studio connects to a data source with SSL enabled, the Team Studio JVM authenticates the (database) server's X.509 certificate to establish trust. If the certificate is not signed by a known certificate authority (CA), an error occurs when you try to add the data source.

Java is distributed with public keys from well-known certificate authorities, such as Verisign or Thawte. If the certificate installed on the database server was not signed by one of these CAs, the solution is to install the server's certificate into the Java installation running Team Studio.

Follow these steps to install a server certificate into the Team Studio JVM:

Prerequisites

Procedure

1. Copy the certificate file, `server.crt`, from the database server to the host where Team Studio is running.
2. On the Team Studio host, open a terminal window and change to the folder where the `server.crt` was copied.
3. Run the following command to convert the certificate to a DER-encoded X.509 certificate:

```
openssl x509 -in server.crt -out server.crt.der -outform der
```
4. Run the following command to import the DER-encoded certificate into the Java keystore. You might need to run the command as root.

```
keytool -keystore $JAVA_HOME/lib/security/cacerts -alias postgresql -import -file server.crt.der
```
5. Restart Team Studio. You should be able to create a data source for the database without error.

Configuring JVM Command-Line Options

The `java_options` property configures command-line options for the Java Virtual Machine (JVM). The value of the `java_options` property is a single string that can contain several command-line options. The following `java_options` setting allows the JVM to find the Hadoop libraries, sets the JVM to server mode, and configures the amount of JVM heap memory to a minimum 512 MB and a maximum 2048 GB.



The `-XX:MaxPermSize=128m` option sets the size of the area where Java stores permanent objects to 128 MB.

See the Oracle Java documentation for available JVM options.

Prerequisites

Procedure

- Add the following code:

```
java_options= -Djava.library.path=$CHORUS_HOME/vendor/hadoop/lib/ -server -Xmx2048m -Xms512m -XX:MaxPermSize=128m
```

Configuring the HDFS Directory and Permissions for Results File Storage

This procedure describes how to create an HDFS directory for a user, set Active Directory user permissions, and set permissions to temp directories.

Prerequisites

Procedure

1. Create an HDFS Directory for user "chorus." This directory is used to cache the uploaded .jar files such as `spark-assembly.jar`.
2. Provide the user with read, write, and execute permissions for the `/user/chorus` directory.
3. The staging directory is typically set as `/user`. If it is not, create a directory using the modified `<staging directory>/chorus`.
4. To run Pig jobs, the Team Studio application attempts to create a folder `/user/<username>` as the Active Directory user. By default, the permissions are set to `hdfs:supergroup:drwxr-xr-x`, which prevents Team Studio from creating that folder. To grant write access to that folder to the Active Directory users who are running the Team Studio application, change permissions to `drwxrwxr-x` or `drwxrwxrwx`.
5. Set permissions to temporary directories.

To run Yarn, Pig, and similar jobs, each individual user might need to write temp files to the temporary directories. There are many Hadoop temp directories such as `hadoop.tmp.dir` and `pig.tmp.dir`, all of which are based on the `/tmp` directory by default. Therefore, the `/tmp` directory must be writable by everyone to enable them to run different jobs. Additionally, it must be executable by everyone to enable them to recurse the directory tree. Set the `/tmp` permissions using the following command:

```
hadoop fs -chmod +wx /tmp
```

Team Studio Related HDFS Configuration

Team Studio uses several temp directories in HDFS. These directories and files are created with HDFS, Yarn, MapRed, and other users.

The temp directories must be made accessible to user `chorus` and other relevant users at the base level. Only individual directories for the corresponding user can be viewed by the specified user. Those directories are:

- Standard output for operators: `@default_tmpdir/dsts_out/<user_name>/<workflow_name>/`
- Team Studio temporary output: `@default_tmpdir/dsts_runtime/<user_name>/<workflow_name>/`
- Team Studio model location: `@default_tmpdir/dsts_model/<user_name>/<workflow_name>/`

Set or change the permissions and ownership as follows:

- The `/tmp` directory should be readable and writable.
- The `/tmp/hadoop-yarn` directory should be readable and writable for Spark jobs.

The upgrade options are as follows (choose one):

- Change `/tmp/dsts_*` directories with full permissions, so everyone can read/write/execute.
- Delete the `/tmp/dsts_*` directories and let the upgraded Team Studio application recreate them. If you are using LDAP, the recreated directories will have the default structure `/tmp/dsts_*/<LDAP_username>/workflowname/operator/`, and permissions at this directory level can be limited to the `LDAP_username` as desired.

By default, `@default_tmpdir` is set to `/tmp`. For more information, see "Workflow Variables" in *TIBCO® Data Science Team Studio User's Guide*. Alternatively, for all newly created workflows, see [Workflow Editor Preferences](#).

Team Studio overwrites `@default_tmpdir/dsts*` files as users re-run workflows. Team Studio users can clear selected `@default_tmpdir/dsts_out` files using Clear Temporary Data. For more information, see "Clear Temporary Data" in *TIBCO® Data Science Team Studio User's Guide*. Hadoop administrators can safely clear `@default_tmpdir/dsts_runtime` from HDFS, because this directory is used to store information for which Team Studio users have chosen the option **Store Results = False**.



Handle `@default_tmpdir/dsts_model` with caution, because Team Studio users might need to export models from this directory.

Deleting Temporary Files

You can clean up temporary files periodically using the following settings.

Through settings in the `alpine.config` file located in the `ALPINE_DATA_REPOSITORY/configuration` directory, administrators can set the period for storing temporary files and how often to run the clean-up task.

The following are the configuration settings for the `sys.properties`:

Setting	Default Value	Description
<code>temporary_file_lifetime</code>	86400000	Specifies the length of time, in milliseconds, to store temporary files. The default: 86400000 ms (24 hours).
<code>temporary_file_scan_frequency</code>	86400000	Specifies the length of time, in milliseconds, of the interval between cleaning tasks. The default: 86400000 ms (24 hours).

Updated values take effect when Team Studio is restarted.

Database Stored Procedures

Team Studio supports installed stored procedures for Greenplum, HAWQ, and PostgreSQL.

Select the database you use for instructions on installing stored procedures.

Installing Stored Procedures on Greenplum

Follow these steps to install Team Studio stored procedures within a Greenplum data base (GPDB). Perform this task on the server side for the Greenplum data base.

Prerequisites

- GPDB must be installed and running.
- The Greenplum database administrator must own the `database_setup` directory and all the content in it.

Procedure

1. Verify that the master node can communicate with the Team Studio server.
 - a) Modify the file `/etc/hosts` on the Team Studio server to include the GPDB master node IP address and hostname.
 - b) Modify the file `/etc/hosts` on the GPDB master node to include the Team Studio IP address and hostname.
2. On the GPDB master host, create a folder called `/home/gpadmin/database_setup`.

3. Use secure copy (SCP) to transfer `$CHORUS_HOME/alpine-current/database_setup.zip` from the Team Studio server to the GPDB master node.

Place it in the folder `/home/gpadmin/database_setup`.

4. Unzip the `database_setup.zip` file.

This generates several folders.

```
unzip /home/gpadmin/database_setup/database_setup.zip
```

- a) If the Greenplum database administrator does not possess the ownership of this directory, then issue the `chown` command to reassign the ownership.

```
# chown -R gpadmin:gpadmin /home/gpadmin/database_setup/
```

5. Log in to the system as the GPDB administrator (for example, `gpadmin`) on the Greenplum master host.

```
# su - gpadmin
```

6. Set the search path to include the public schema.

7. Navigate to the `database_setup/Greenplum` directory.

```
$ cd /home/gpadmin/database_setup/Greenplum
```

8. Run the Team Studio installer (the `.bin` file).

```
$ sh alpine_miner_installer_Greenplum.bin
```

- a) Read and accept the license agreement.
- b) Specify the Greenplum installation path. The setup places the required shared library in the directory `$GPDBHOME/lib/postgres/`.
- c) Specify if the installer should copy the shared library to the segment hosts. Enter `y` for multi-node clusters.
- d) If you entered `y`, then enter the full path to the file containing the segment host names.

You can create your own host file. For example, create a file `/tmp/hostfile` and add all the segment host names or IP addresses one after the other in the file, similar to the following.

```
segmenthost1
segmenthost2
segmenthost3
/path/to/hostfile_gpinitssystem
(or /tmp/hostfile)
```

- e) Specify the default database to install the Team Studio functions.
 - f) Specify the port on which the Greenplum database is running.
 - g) Specify if you would like to create the Team Studio demo database.
 - h) Verify the shared library exists on each segment node.
9. Modify the file `/var/lib/pgsql/data/pg_hba.conf` to allow users access to the appropriate databases.

Using the `miner_demo` database as an example, add the following lines to the end of `pg_hba.conf`.

```
local miner_demo miner_demo trust
host miner_demo miner_demo 192.168.1.0/24 password
```

10. Reload the Greenplum database to activate the changes made in the configuration file.

```
$ gpstop -u
```

Uninstalling Stored Procedures on Greenplum

In the case where a Team Studio stored procedure installation is incomplete or fails in some way, you can uninstall stored procedures within a Greenplum database (GPDB).

Perform this task in an environment where the Greenplum database is installed.

Prerequisites

- GPDB must be installed and running.

- The Greenplum database administrator must own the `database_setup` directory and all of the content in it.

Procedure

1. Extract the installer `alpine_miner_installer_Greenplum.bin`.

```
./alpine_miner_installer_Greenplum.bin --keep --noexec
```
2. Change to the Greenplum folder, where the SQL file `alpine_miner_uninstall_functions.sql` is located.

```
cd Greenplum
```
3. Run the following code.

```
psql -d [your database name] -f alpine_miner_uninstall_functions.sql
```
4. Remove the `alpine_miner.so` file from your Greenplum binaries folder.

```
rm ${GPHOME}/lib/postgresql/alpine_miner.so
```

Result

The stored procedure installation is removed.

Installing Stored Procedures on HAWQ

Follow these steps to install Team Studio stored procedures within a HAWQ database. Perform this task on the server side withing HAWQ.

Prerequisites

- HAWQ must be installed and running.
- The database administrator must own the `database_setup` directory and all the content in it.

Procedure

1. Verify that the master node can communicate with the Team Studio server.
 - a) Modify the file `/etc/hosts` on the Team Studio server to include the HAWQ master node IP address and hostname.
 - b) Modify the file `/etc/hosts` on the HAWQ master node to include the Team Studio IP address and hostname.
2. On the HAWQ master host, create a folder called `/home/gpadmin/database_setup`.
3. Use secure copy (SCP) to transfer `$CHORUS_HOME/alpine-current/database_setup.zip` from the Team Studio server to the HAWQ master node.
 Place it in the folder `/home/gpadmin/database_setup`.
4. Unzip the `database_setup.zip` file.
 This generates several folders.

```
unzip /home/gpadmin/database_setup/database_setup.zip
```

 - a) If the HAWQ database administrator does not possess the ownership of this directory, then issue the `chown` command to reassign the ownership.

```
# chown -R gpadmin:gpadmin /home/gpadmin/database_setup/
```
5. Log in to the system as the HAWQ administrator (for example, `gpadmin`) on the HAWQ master host.

```
# su - gpadmin
```
6. Set the search path to include the public schema.
7. Navigate to the `database_setup/HAWQ` directory.

```
$ cd /home/gpadmin/database_setup/HAWQ
```

8. Run the Team Studio installer (the .bin file).

```
$ sh alpine_miner_installer_HAWQ.bin
```

- a) Read and accept the license agreement.
- b) Specify the HAWQ installation path. The setup places the required shared library in the directory `$HAWQHOME/lib/postgres/`.
- c) Specify if the installer should copy the shared library to the segment hosts. Enter `y` for multi-node clusters.
- d) If you entered `y`, then enter the full path to the file containing the segment host names.

You can create your own host file. For example, create a file `/tmp/hostfile` and add all the segment host names or IP addresses one after the other in the file, similar to the following.

```
segmenthost1
segmenthost2
segmenthost3
/path/to/hostfile_gpinitssystem
(or /tmp/hostfile)
```

- e) Specify the default database to install the Team Studio functions.
 - f) Specify the port on which the HAWQ database is running.
 - g) Specify if you would like to create the Team Studio demo database.
 - h) Verify the shared library exists on each segment node.
9. Modify the file `/var/lib/pgsql/data/pg_hba.conf` to allow users access to the appropriate databases.

Using the `miner_demo` database as an example, add the following lines to the end of `pg_hba.conf`.

```
local miner_demo miner_demo trust
host miner_demo miner_demo 192.168.1.0/24 password
```

10. Reload the HAWQ database to activate the changes made in the configuration file.

```
$ gpstop -u
```

Uninstalling Stored Procedures on HAWQ

In the case where a Team Studio stored procedure installation is incomplete or fails in some way, you can uninstall stored procedures within a HAWQ database.

Perform this task in an environment where the HAWQ database is installed.

Prerequisites

- HAWQ must be installed and running.
- The HAWQ database administrator must own the `database_setup` directory and all of the content in it.

Procedure

1. Extract the installer `alpine_miner_installer_hawq.bin`.

```
./alpine_miner_installer_hawq.bin --keep --noexec
```
2. Change to the HAWQ folder, where the SQL file `alpine_miner_uninstall_functions.sql` is located.

```
cd HAWQ
```
3. Run the following code.

```
psql -d [your database name] -f alpine_miner_uninstall_functions.sql
```
4. Remove the `alpine_miner.so` file from your HAWQ binaries folder.

```
rm ${GPHOME}/lib/hawq/alpine_miner.so
```

Result

The stored procedure installation is removed.

Installing Stored Procedures on PostgreSQL

Follow these steps to install Team Studio stored procedures within a PostgreSQL database on Linux. Perform this task on the server side for the PostgreSQL data base on a Linux computer.

Prerequisites

- PostgreSQL must be installed and running.
- The PostgreSQL database administrator must own the database_setup directory and all the content in it.

Procedure

1. In the PostgreSQL data directory, find the file postgresql.conf (for example, /data/pgsql/9.3/data/postgresql.conf).

2. Change date style as follows.

```
datestyle = 'iso, mdy'
```

3. Reload the configuration change (without being forced to restart Postgres).

```
# su - postgres
# /usr/bin/pg_ctl reload
```

4. Log in to Postgres using the psql command, and then run the following query.

```
SELECT pg_reload_conf();
```

5. Verify that the master node can communicate with the Team Studio server.

- a) Modify the file /etc/hosts on the Team Studio server to include the Postgres master node IP address and hostname.
- b) Modify the file /etc/hosts on the Postgres master node to include the Team Studio IP address and hostname.

6. On the Postgres master host, create a folder called /home/postgres/database_setup.

7. Use secure copy (SCP) to transfer \$CHORUS_HOME/alpine-current/database_setup.zip from the Team Studio server to the Postgres master node.

Place it in the folder /home/gpadmin/database_setup.

8. Unzip the database_setup.zip file.

This generates several folders.

```
unzip /home/postgres/database_setup/database_setup.zip
```

- a) If the Postgres database administrator does not possess the ownership of this directory, then issue the chown command to reassign the ownership.

```
# chown -R postgres:postgres /home/postgres/database_setup/
```

9. Log in to the system as the Postgres administrator (for example, postgres) on the Postgres master host.

```
# su - postgres
```

10. Set the search path to include the public schema.

11. Navigate to the database_setup/Postgres directory.

```
$ cd /home/postgres/database_setup/Postgres
```

12. Run the Team Studio installer (the .bin file).

```
$ sh alpine_miner_installer_Postgres.bin
```

- a) Read and accept the license agreement.
- b) Specify the Postgres installation path. The setup places the required shared library in the directory \$PGHOME/lib/postgres/.
- c) Specify if the installer should copy the shared library to the segment hosts. Enter y for multi-node clusters.

- d) If you entered `y`, then enter the full path to the file containing the segment host names.

You can create your own host file. For example, create a file `/tmp/hostfile` and add all the segment host names or IP addresses one after the other in the file, similar to the following.

```
segmenthost1
segmenthost2
segmenthost3
/path/to/hostfile_pginitssystem
(or /tmp/hostfile)
```

- e) Specify the default database to install the Team Studio functions.
 f) Specify the port on which the Postgres database is running.
 g) Specify if you would like to create the Team Studio demo database.
 h) Verify the shared library exists on each segment node.

The installer can fail with the following error.

```
.....
./install.sh: line 162: lsb_release: command not found
*****
Path /usr/pgsql-9.3/lib is the Postgres lib path to copy alpine_miner.so to?
(y/n)
Provide the path of Postgres lib to copy alpine_miner.so to
or press ENTER to accept the default
*****
Postgres lib path:
Copying sharedLib/9.3/alpine_miner..so to /usr/pgsql-9.3/lib/alpine_miner.so
Shared library file sharedLib/9.3/alpine_miner..so does not exist.
```

If you see this error, then you must install the following packages and rerun the installer.

```
yum install redhat-lsb redhat-lsb-core
```

13. Locate and open for editing the file `/var/lib/pgsql/data/pg_hba.conf`.
 14. Modify the file to allow users access to the appropriate databases.
 For example, using the `miner_demo` database, add the following lines to the end of `pg_hba.conf`.

```
local miner_demo miner_demo trust
host miner_demo miner_demo 192.168.1.0/24 password
```

15. Reload the Postgres database to activate the changes made in the configuration file.

```
# su - postgres
# /usr/bin/pg_ctl reload
```

Installing Team Studio DLLs in a PostgreSQL database on Windows

Follow these steps to install Team Studio stored procedures within a PostgreSQL database on Windows. Perform this task on the server side for the PostgreSQL data base on a Windows computer.

Prerequisites

- PostgreSQL must be installed and running.
- The PostgreSQL database administrator must own the `database_setup` directory and all the content in it.
- Determine whether you need to the 64-bit or the 32-bit DLL file.

Procedure

1. From your Team Studio installation, install the appropriate DLL file.
 - Download `alpine_miner.64bit.dll` for 64-bit.
 - Download `alpine_miner.dll` for 32-bit.
2. From the Team Studio database setup, set up the file `Postgres.zip`.
 This file contains two SQL files.

- `create_demo_db.sql`
 - `alpine_miner_setup.sql`
3. Copy the files downloaded Step 3 to the `PG_HOME\bin` directory.
 4. Verify the installation is correct by installing the `miner_demo` database.
From the command prompt, use the following command. (In this example, we use the `template1` database and the `postgres` user.)

```
PG_HOME\bin\psql.exe -f create_demo_db.sql -d template1 -U postgres
```
 5. Install the functions to the database that Team Studio workflow operators work with).
In this example, we use the `miner_demo` database, and the `postgres` user.

```
PG_HOME\bin\psql.exe -f alpine_miner_setup.sql -d miner_demo -U postgres
```

Uninstalling Stored Procedures on PostgreSQL

If a Team Studio stored procedure installation is incomplete or fails in some way, you can uninstall stored procedures within a PostgreSQL database.

Perform this task in an environment where the PostgreSQL database is installed.

Prerequisites

- PostgreSQL must be installed and running.
- The PostgreSQL database administrator must own the `database_setup` directory and all of the content in it.

Procedure

1. Extract the installer `alpine_miner_installer_postgres.bin`.

```
./alpine_miner_installer_postgres.bin --keep --noexec
```
2. Change to the Postgres folder, where the SQL file `alpine_miner_uninstall_functions.sql` is located.

```
cd Postgres
```
3. Run the following code.

```
psql -d [your database name] -f alpine_miner_uninstall_functions.sql
```
4. Remove the `alpine_miner.so` file from your Postgres binaries folder.

```
rm ${PGHOME}/lib/postgres/alpine_miner.so
```

Result

The stored procedure installation is removed.

Security

The following topics describe how to complete various security-related tasks in Team Studio.

Configure a Kerberos-Enabled Hadoop Data Source

Team Studio can support a Hadoop cluster set up for Kerberos authentication.

Kerberos is a network authentication protocol that provides two-way mutual authentication: Both the user and the server verify each other's identity. If an existing Hadoop cluster has Kerberos authentication set up, Team Studio must be configured to authenticate with the Kerberos service.

Prerequisites

Your Hadoop cluster should be properly configured with Kerberos.

- The machine should be running a Linux operating system with Kerberos installed.
- The Linux user running Team Studio should be able to authenticate with Kerberos using a local key tab file.
- JCE should be installed on each node. To get a copy of the extension, see <https://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html> for JDK 6 or <https://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html> for JDK 7.
- The files should be copied to `$JRE_HOME/lib/security`.



Configuring a Kerberos-enabled Hadoop datasource prevents the Team Studio application from connecting to non-kerberized Hadoop data sources (for versions earlier than 5.2).

Kerberos Authentication Integration Steps

The procedures for integrating with Kerberos authentication depend on the Team Studio version installed and the upgrade plan.

The topics in this section walk you through the steps necessary to integrate Team Studio with the Kerberos network authentication protocol.

Generate the User Account

Follow these steps for the access protocol you use (LDAP or non-LDAP) to generate a user account as part of integrating Team Studio with Kerberos.

Generating the User Account: LDAP

Follow these steps to generate a user account as part of integrating Team Studio with Kerberos under LDAP (either open or Microsoft Active Directory).

Procedure

1. Create the following users in LDAP/AD.

- `mapred`
- `hdfs`
- `yarn`
- `serviceuser`

All of these users are within the supergroup `group`.



The users `mapred`, `hdfs`, and `yarn` are present when the Hadoop cluster has been set up properly.

2. In the KDC server, create the principal and keytab for the Team Studio host.
3. Copy the keytab to the host

Generating the User Account: Non-LDAP

Follow these steps to generate a user account as part of integrating Team Studio with Kerberos in a non-LDAP environment.

Procedure

- Create the user on your server to run MapReduce jobs.

For example, if the user johndoe is the login name for the Team Studio application, this user name also must be defined within the Linux system on the Hadoop cluster.

Generate the Keytab and Principal

The examples in this section walk you through generating the keytab and principal on either a Windows or Linux system.

Generating the Keytab and Principal on a Linux Server

If you are generating the keytab and principal for a Linux server, follow these steps. Perform this task on the Kerberos Linux server.

In the examples below, MYREALM is the realm and myhost.myparentdomain.local is the fully qualified domain name of the host specified to generate the principal.

Prerequisites

When you create the keytab and principal on the Kerberos server, make sure that the hostname is in lowercase.

For example, if your machine's hostname is myhost.myparentdomain.local, when you create the principal and keytab in KDC, use myhost.myparentdomain.local.

Procedure

1. Access the kadmin shell using either the command `kadmin` or `sudo kadmin.local`.
2. From the kadmin shell, run the following command to create the principal.

```
addprinc -randkey serviceuser/myhost.myparentdomain.local@MYREALM
```
3. From the kadmin shell, run the following command to create the corresponding keytab file.

```
xst -norandkey -k chorus.keytab serviceuser/myhost.myparentdomain.local
```

Generating the Keytab and Principal on a Windows Server

If you are generating the keytab and principal for a Windows server, follow these steps. Perform this task on the Kerberos Linux server.

In the examples below, MYREALM is the realm and myhost.myparentdomain.local is the fully qualified domain name of the host specified to generate the principal.

Prerequisites

When you create the keytab and principal on the Kerberos server, make sure that the hostname is in lowercase.

For example, if your machine's hostname is myhost.myparentdomain.local, when you create the principal and keytab in KDC, use myhost.myparentdomain.local.

Procedure

1. Create the Team Studio principal.

```
setspn.exe -A serviceuser/myhost.myparentdomain.local@MYREALM serviceuser  
setspn.exe -L serviceuser
```
2. Create the Team Studio keytab.

```
ktpass.exe -princ serviceuser/myhost.myparentdomain.local@MYREALM -out chorus.keytab  
-crypto all -ptype KRB5_NT_PRINCIPAL -desonly -pass bERGucm!mr -mapuser MYREALM  
\serviceuser
```

For example, to create a principal for user johndoe, the Active Directory hostname is `ad.tds.local`, the realm is `TSDS.LOCAL`, and the principal creation command looks like the following example.

```
setspn.exe -A chorus/ad.dsts.local@TSDS.LOCAL johndoe
```

The keytab creation command looks like the following example

```
ktpass.exe -princ chorus/ad.dsts.local@DSTS.LOCAL -out chorus.keytab -crypto all -  
ptype KRB5_NT_PRINCIPAL -desonly -pass DSTSIsCool!!2 -mapuser DSTS\johndoe
```

where `DSTSIsCool!!2` is the johndoe user's password. The `chorus.keytab` file gets created in the current directory from where the `ktpass.exe` command was run.

Copying the Keytab to the Team Studio Server

After generating the keytab, copy it to the correct directory on the host. Perform this task on the computer where Team Studio is installed.

Prerequisites

- You must have generated the keytab.
- You must have write access to the server where Team Studio is installed.

If you are using LDAP/AD, you must set up user account synchronization on your Hadoop cluster using System Security Service Daemon (SSSD) or some other tool. You can also perform user synchronization manually or using a custom script.

Procedure

- Copy the generated keytab to the Team Studio host.
For example, copy the keytab into the `/home/chorus/keytab` directory of `alpinehost.myparentdomain.local` host.

Hadoop Cluster Configuration

This section demonstrates how to configure several settings for Kerberos on the Hadoop cluster. It includes examples of the files you must change, customizing the values of each key as necessary.

According to Cloudera documentation ([Configure Secure YARN](#)), you must set the YARN configuration that is detailed in [Configuring HDFS and YARN](#). Similarly, according to the book *Hadoop Security*, by O'Reilly, Spivey, Joey Echeverria, the following advice is given.

"In addition to configuring the NodeManager to use Kerberos for authentication, we need to configure the NodeManager to use the `LinuxContainerExecutor`. The `LinuxContainerExecutor` uses a `setuid` binary to launch YARN containers. This allows each NodeManager to run the containers using the UID of the user that submitted the job. This is required in a secure configuration to ensure that Alice can't access files created by a container Bob launched. Without the `LinuxContainerExecutor`, all of the containers would run as the `yarn` user and containers could access each other's local files. First set the following parameters in the `yarn-site.xml` file" (p. 57).

In Team Studio testing, this setting is not required to enable Kerberos authentication. We added the above configuration due to the Cloudera and Hadoop Security recommendations. The System Administrator must determine whether to set this configuration.

Configuring HDFS and YARN

To configure HDFS and YARN for your Kerberos integration of Team Studio, follow these steps. Perform this task in the files `core-site.xml`, `hadoop-policy.xml`, and `yarn-site.xml`.

In the following tasks, When `*` is used as a value, it represents the most broad settings for each key-value pair. Whenever you see `SERVICEUSER`, replace that with the proper user name for your account (for example, johndoe).

Prerequisites

You must have write access to the configuration files on the computer where Team Studio server is installed.

Procedure

1. Find and open for editing the file `core-site.xml`.

2. Add the following properties to the `core-site.xml`.

Remember to customize your *SERVICEUSER* accordingly.

```
<property>
<name>hadoop.proxyuser.SERVICEUSER.groups</name>
<value>*</value>
<description>* Allows the superuser <SERVICEUSER> to impersonate any members of any
groups. Limit groups as desired by specifying a different value.</description>
</property>
<property>
<name>hadoop.proxyuser.SERVICEUSER.hosts</name>
<value>*</value>
<description>* Allows the superuser to connect from any hosts to impersonate a user.
Limit hosts as desired by specifying a different value.</description>
</property>
<property>
<name>hadoop.security.authorization</name>
<value>true</value>
</property>
```

3. Save your changes and close the file.

4. Find and open for editing the file `hadoop-policy.xml`.

5. Add the following properties to `hadoop-policy.xml`.

Remember to customize your *SERVICEUSER* accordingly.

```
<property>
<name>security.job.submission.protocol.acl</name>
<value>yarn, mapreduce, SERVICEUSER</value>
<description>yarn, mapreduce, and SERVICEUSER are allowed to submit jobs. Specifying
the "*" value would allow any user to submit jobs.</description>
</property>
<property>
<name>security.datanode.protocol.acl</name>
<value>*</value>
</property>
<property>
<name>security.client.protocol.acl</name>
<value>*</value>
</property>
```

6. Save your changes and close the file.

7. Find and open for editing the file `yarn-site.xml`.

8. Add the following properties to `yarn-site.xml`.

```
<property>
<name>yarn.nodemanager.container-executor.class</name>
<value>org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor</value>
</property>
<property>
<name>yarn.nodemanager.linux-container-executor.group</name>
<value>yarn</value>
</property>
```

9. Save your changes and close the file.

Setting HDFS Permissions

Setting HDFS permissions includes creating a `serviceuser` directory and then setting permissions for Active Directory and various temporary directories.

Perform this task on the computer where Team Studio is installed.

Prerequisites

You must have write access to the configuration files.

Procedure

1. Create the HDFS directory `/user/serviceuser/` with the owner:group as `serviceuser:supergroup`. This directory is used to cache the uploaded JAR files such as `spark-assembly.jar`.



This staging directory is typically set as `/user`. If not, create the directory using `/<staging directory>/serviceuser`.

2. Give the `/user/serviceuser` directory read, write, and execute permissions for the `serviceuser`.
3. Set the Active Directory (AD) permissions.

To run Pig jobs, the Team Studio application attempts to create a folder `/user/<username>` as the AD user. By default, the permissions are set to `hdfs:supergroup:drwxr-xr-x`, which prevents Team Studio from creating that folder.

- a) Change permissions to grant write access to that folder to the AD users running the Team Studio application (`drwxrwxr-x` or `drwxrwxrwx`).

4. Set permissions for the temporary directory HDFS `/tmp`.

To run YARN, Pig, and similar jobs, each individual user might need to write temp files to the temporary directories. There are many Hadoop temp directories such as `hadoop.tmp.dir`, `pig.tmp.dir`, and so on. By default, all of them are based off of the `/tmp` directory.

- a) Make the `/tmp` directory writable by everyone so that everyone can run different jobs.
- b) Make the `/tmp` directory executable by everyone so that everyone can recurse the directory tree. Set the `/tmp` permissions using the following command:

```
hadoop fs -chmod +wx /tmp
```

Setting this option allows all users to recurse the directory tree.

5. Set the permissions for the temporary directories HDFS `/tmp/tsds_*`.

The Team Studio application generates these directories in HDFS:

- `/tmp/tsds_out/<username>`
- `/tmp/tsds_model/<username>`
- `/tmp/tsds_runtime/<username>`

- a) Set or change the permissions and ownership appropriately, as follows.

- The `/tmp` directory should be readable, writable, and executable.
- The `/tmp/tsds_*/<username>` directories for the corresponding user can be viewed by that user.
- The `/tmp/hadoop-yarn` directory should be readable and writable for Spark jobs.

Setting HDFS Permissions When Upgrading

When you upgrade Team Studio, remember to check the HDFS directories for any permissions updates. Perform this task after upgrading the computer where Team Studio is installed.

Prerequisites

You must have write access to the computer where Team Studio is installed.

Procedure

- Either change `/tmp/tsds_*` directories with full permissions, so everyone can read, write, and execute, or delete the directories of `/tmp/tsds_*` and let the upgraded Team Studio application recreate them. The recreated directories have this default structure if using `LDAP/tmp/tsds_*/<LDAP_username>/workflowname/operator/`, and permissions at this directory level can be limited to the `LDAP_username` as desired.

Configuring the Team Studio Server

After you configure the Hadoop cluster, the next step is to configure the Team Studio server.

Procedure

1. Find the following configuration files:
 - `$CHORUS_HOME/shared/chorus.properties`
 - `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/deploy.properties`
2. Add the following Kerberos properties to the `java_options` variable:


```
-Djava.security.krb5.realm=<kdc_realm> -Djava.security.krb5.kdc=<kdc_server>
```
3. Replace `<kdc_realm>` and `<kdc_server>` with the name of the realm and the address for the Kerberos Key Distribution Center (KDC) host. You can find these values in your Kerberos configuration file (usually `/etc/krb5.conf`).
4. Restart Team Studio.

TGT Generation

There is no need to re-run `kinit` to generate a TGT (Ticket Granting Ticket) on the Team Studio host.

Keep in mind that the TGT is destroyed by `kdestroy`, so if you run `kdestroy` you must run `kinit` again to generate the TGT. After doing so, you must restart the Team Studio application for the changes to take effect.

Adding the Data Source to Team Studio

After configuring the Team Studio server, add the data source to Team Studio. Perform this task on a computer where Team Studio is installed.

Prerequisites

To perform this task, you must be either a data administrator or an application administrator. If you do not have administrator permissions, talk to your administrator to obtain credentials before continuing.

Procedure

1. From the Team Studio user interface, from the menu, click **Data**.
2. Click **Add Data Source**.
The Add Data Source dialog box is displayed.
3. From the **Data Source Type** list box, select **Hadoop Cluster**.

4. Specify the required properties.
 - **Data Source Name**
 - **Name Node Host**
 - **Name Node Port**
 - **Resource Manager Host**
 - **Resource Manager Port**
5. Specify the **Data Source User** as `serviceuser`.
You can specify another name as needed.
6. Specify the **Group** to which `serviceuser` belongs.
7. Click **Configuration Connection Parameters**, and then specify additional parameters, as needed.
These parameters are specified as key-value pairs, and according to your cluster.



In the following examples, MYREALM is the Kerberos realm, and `/home/chorus/thisismy.keytab` refers to the location of your keytab file on the Team Studio host.

Key	Value (to modify)
<code>yarn.app.mapreduce.am.staging-dir</code>	<code>/user</code>
<code>yarn.resourcemanager.scheduler.address</code>	<code>123.45.6.7:8030</code>
<code>mapreduce.jobhistory.principal</code>	<code>mapred/_HOST@MYREALM</code>
<code>hadoop.security.authentication</code>	<code>kerberos</code>
<code>dfs.datanode.kerberos.principal</code>	<code>hdfs/_HOST@MYREALM</code>
<code>dfs.namenode.kerberos.principal</code>	<code>hdfs/_HOST@MYREALM</code>
<code>yarn.resourcemanager.principal</code>	<code>yarn/_HOST@MYREALM</code>
<code>alpine.principal</code>	<code>chorus/[fully qualified domain name of the host where the keytab was generated]@MYREALM</code>
<code>alpine.keytab</code>	<code>/home/chorus/thisismy.keytab</code>

Viewing Logging Information

After you add the data source to Team Studio, you can view logging information. Connection attempts are logged on the Team Studio host in `jetty.log`. This file can be found at `$CHORUS_HOME/shared/log/jetty.log`.

To assist in debugging Kerberos issues, additional Kerberos logging can be generated by editing the `chorus.properties` and `deploy.properties` files.

Procedure

1. Browse to the following locations

```
$CHORUS_HOME/shared/chorus.properties
$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/deploy.properties
```

2. In the `java_options` variable, add the following.

```
-Dsun.security.krb5.debug=true
```

What to do next

When you are finished testing, you can remove this property.

Installing an SSL Certificate for a Database Connection

If you connect to the database using SSL, you must install an SSL certificate.

Perform this task on the computer where Team Studio server is installed. You might need to run as either administrator or root user.

Prerequisites

- You must have created and saved to the server computer a certificate `server.crt` to install.
- You must ensure that the server is configured to support SSL.
- You must have write access to the computer where Team Studio server is installed.

Procedure

1. Copy the certificate (`server.crt`) from the database server to the web-server that hosts Team Studio.
2. Open the command line on the computer, and then change the directory to the folder where the file `server.crt` is copied.
3. Run the command `openssl x509 -in server.crt -out server.crt.der -outform der`.
4. Run the `keytool` command.

```
keytool -keystore $JAVA_HOME/lib/security/cacerts -alias postgresql -import -file server.crt.der
```

5. Restart Team Studio.
You should be able to create a connection to an SSL-enabled PostgreSQL, Pivotal HAWQ, or Greenplum database.

What to do next

See documentation [PostgreSQL JDBC/SSL Connections](#) for more details.

For more information, see [Installing Stored Procedures on Greenplum](#).

Configuring and Installing an SSL Certificate for the Team Studio Server

Configuring Team Studio with an SSL certificate is a good practice. The Team Studio installer provides a step-by-step interface to help you do this.



If you use a self-signed certificate, users receive an untrusted SSL certificate warning in their browser.

Procedure

1. After installing Team Studio, run `chorus_control.sh configure` and select option 5.
2. Run through the prompts provided to set up the necessary credentials for the SSL certificate.



If the system prompts you for a passphrase and you never set one during the install process, you can set one now. Type in the passphrase twice to set it, and then once more to authenticate it.

```
Enter pass phrase for /usr/local/chorus/shared/server.key:
Verifying - Enter pass phrase for /usr/local/chorus/shared/server.key:
```

```

Enter pass phrase for /usr/local/chorus/shared/server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Enter pass phrase for /usr/local/chorus/shared/server.key.org:
Which port you want to use for https? [default='8443']:
```

After this is completed, a message should indicate that the update was successful.

To see the changes, run `chorus_control.sh restart`.

Configuring and Installing an SSL Certificate for the Team Studio Server (Manual)

Configuring Team Studio with an SSL certificate is a good practice. This procedure shows you how to generate an SSL certificate with OpenSSL.



If you use a self-signed certificate, users receive an untrusted SSL certificate warning in their browser.

Procedure

1. Generate an RSA private key.

```
openssl genrsa -des3 -out server.key 1024
```

2. Generate a certificate signing request (CSR).

```
openssl req -new -key server.key -out server.csr
```

3. Respond to the questions as shown in this example:

```

What is your first and last name?
[Unknown]: chorus-ga.greenplum.com
Note: Enter the URL for Chorus.
What is the name of your organizational unit?
[Unknown]: Data and Insights
What is the name of your organization?
[Unknown]: Greenplum
What is the name of your City or Locality?
[Unknown]: San Mateo
What is the name of your State or Province?
[Unknown]: California
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=chorus-ga.greenplum.com, OU=Data and Insights,
O=Greenplum, L=San Mateo, ST=California, C=US correct?
[no]: yes
Enter key password for <chorus>
(RETURN if same as keystore password.)
```

4. Enter a common name that is the fully qualified domain name (FQDN) of your server, or the value localhost.
5. Remove the passphrase from the key.

```
cp server.key server.key.org
openssl rsa -in server.key.org -out server.key
```

Without this step, you must type the password you created in step 1 each time you start Team Studio.

6. Generate a self-signed certificate from the CSR.

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

If you want an official SSL certificate (recommended), submit this CSR to a signing authority such as Thawte or Verisign and continue to the next step when you have the certificate (.crt) file.

7. Install the private key and certificate into Team Studio.

Set the following properties in `chorus.properties` to point to the locations of your private key and certificate files:

```
ssl.enabled= true
ssl_server_port= 8443
ssl_certificate= /usr/local/chorus/current/config/test.crt
ssl_certificate_key= /usr/local/chorus/current/config/test.key
public_url = nate.alpinedata.com
```

8. Verify that the `public_url` matches the FQDN you specified for the certificate in step 2.
9. Restart Team Studio to apply the configuration.



To run Team Studio on port 443 (the default SSL port, for example, `https://hostname:443`), set up a web server proxy to Team Studio.

10. Create a Java TrustStore.

```
$JAVA_HOME/bin/keytool -import -file server.crt -alias localhost
```

If this is the first time you are running the keytool utility, you are prompted to create a password.

11. Open `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/deploy.properties`.

12. Locate the line starting with `alpine.catalina.opts`, and append the following text to the end of the line:

```
-Djavax.net.ssl.trustStore=/home/chorus/.keystore -
Djavax.net.ssl.trustStorePassword=changethis
```

The TrustStore location is in the user's home directory by default. In this case, the user is `chorus`. The `trustStorePassword` should match the password you set in step 7.

13. Open `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/agent-jetty.ini` and add the following lines:

```
-Djavax.net.ssl.trustStore=/home/chorus/.keystore
-Djavax.net.ssl.trustStorePassword=changethis
```

with the same store location and password as the `deploy.properties` file.

14. Save and close the `agent-jetty.ini` file.

15. Open the `alpine.conf` file for editing, and then locate and modify the following lines:

```
chorus {
    active = true
    scheme = HTTPS
    host = nate.alpinedata.com
    port = 8443
}
```

The scheme should be HTTPS to indicate the use of HTTPS instead of HTTP. The host should match the common name you specified when generating the certificate. This is typically the hostname of the server or `localhost`. The port is the value specified as the SSL port in `chorus.properties` - by default, this is 8443. These values must match.

16. Start or restart all services using `chorus_control.sh restart`.

Enabling LDAP Authentication

By default, Team Studio manages users through the PostgreSQL database. However, it can be configured to authenticate against an external LDAP server.

The Team Studio collaboration framework uses the LDAPv3 server, including Active Directory support, to manage and authenticate users. For more information about the LDAP server, see <http://www.ietf.org/rfc/rfc2251.txt>.

LDAP provides the following benefits:

- Adding users to Team Studio: When a user is added, Team Studio maintains a read-only copy of common user information, such as the user's name and department.
- Authenticating users with LDAP.

Configuring LDAP

Follow these steps to configure LDAP authentication.

Prerequisites

Procedure

1. Try connecting to your AD or LDAP installation with a separate LDAP exploration tool to ensure that all configuration properties are correct before you attempt to configure these in Team Studio.
2. Install Team Studio.
3. Edit the <installation directory>/shared/ldap.properties file to configure LDAP in Team Studio.
4. Change the default entries of the ldap.properties to match your LDAP installation. See the ldap.properties.active_directory or ldap.properties.opensource_ldap files for examples. Here is an example:

```
LDAP Settings for Active Directory
# Set this property to true to enable LDAP authentication. Default is false.
ldap.enable = false
# Host and port for accessing LDAP server.
ldap.host = localhost
ldap.port = 389
# Set this property to use Transport Level Security (TLS) for accessing LDAP server.
Default is false.
ldap.start_tls = false
#LDAP root for search and query
ldap.base = DC=www,DC=example,DC=com
# username and password used for binding to LDAP server
ldap.bind.username = uid=admin,ou=system
ldap.bind.password = q2W#e4R%

#----- Uncomment following properties to enable group membership authentication
-----#
# Note that all three entries must either be commented or uncommented
# List of LDAP group names that are used for verifying group membership.
# NOTE: For release 5.3, only one group is supported.
#ldap.group.names = OtherGroup
# Search base for looking up members in the groups above.
#ldap.group.search_base = DC=www,DC=example,DC=com
#Group Filter for Active Directory. This will work only for Active Directory
#ldap.group.filter = (memberOf={0})
#-----#
# Search base for user authentication
ldap.user.search_base = OU=CorpUsers,DC=www,DC=example,DC=com
#Search filter for user authentication. This will work only for Active Directory
ldap.user.filter = (sAMAccountName={0})
```



```
# Mappings of Chorus user properties to LDAP user attributes.
ldap.attribute.uid = sAMAccountName
ldap.attribute.ou = department
ldap.attribute.gn = givenName
ldap.attribute.sn = sn
ldap.attribute.mail = mail
ldap.attribute.title = title
```



If you want to add users from two different groups (for example, Marketing and Sales) but Team Studio supports only one LDAP group, you have two options:

- Add a new LDAP group (MarketingSales) to include users from Marketing and Sales. Then bulk import using the rake command below from the MarketingSales group.
- Disable group search by commenting the lines below from `ldap.properties`. Then, as an Admin, manually add each user to Team Studio.
 - `ldap.group.search_base`
 - `ldap.group.filter`
 - `ldap.group.names`

5. Restart Team Studio as follows after making changes to `ldap.properties`:

```
$ chorus_control.sh restart
```

6. Bulk import LDAP users with a rake command. This rake task reads the LDAP configuration from the `ldap.properties` file and imports users from the LDAP group specified in the `ldap.group.names` property into the Team Studio database.



Release 5.3 supports just one group.

```
cd $CHORUS_HOME
export RAILS_ENV=production
export PATH=$PATH:$CHORUS_HOME/current/bin
cd $CHORUS_HOME/current
rake ldap:import_users
```

LDAP Configuration Properties

The following table lists configuration properties related to LDAP.

For more information, see [Enabling LDAP Authentication](#).

LDAP Parameter	Description
<code>ldap.enable</code>	Boolean value to enable or disable LDAP (<code>false</code> by default).
<code>ldap.host</code>	LDAP server IP or host name.
<code>ldap.port</code>	LDAP server port.
<code>ldap.base</code>	LDAP base DN.
<code>ldap.start_tls</code>	Upgrades LDAP connection a secure connection using TLS (<code>false</code> by default).
<code>ldap.bind.username</code>	User name for LDAP server.
<code>ldap.bind.password</code>	Password for LDAP server.

LDAP Parameter	Description
ldap.group.names	Name of group(s) to search for users. (Optional. LDAP users outside of specified groups cannot be added or authenticated as Team Studio users.)
ldap.group.search_base	Group DN to search for users. (Optional. LDAP users outside of specified groups cannot be added or authenticated as Team Studio users.)
ldap.group.filter	Group filter used to search for users. (Optional. LDAP users outside of specified groups cannot be added or authenticated as Team Studio users.)
ldap.user.search_base	Base DN to search for users when authenticating.
ldap.user.search_filter	Search filter for users when authenticating.
ldap.attribute.uid	LDAP username attribute (uid by default). Required. For Active Directory, this is often sAMAccountName. Another common value is cn.
ldap.attribute.ou	LDAP attribute name for Organizational Unit or Department (ou by default). Maps to Department in Team Studio.
ldap.attribute.gn	LDAP attribute name for First name (gn or givenName by default). Maps to First Name in Team Studio.
ldap.attribute.sn	LDAP attribute name for Last name. (sn by default). Maps to Last Name in Team Studio.
ldap.attribute.mail	LDAP attribute name for e-mail address. (mail by default). Maps to Email in Team Studio.
ldap.attribute.title	LDAP attribute name for User's title. (title by default). Maps to Title in Team Studio.

Adding LDAP Users

The following procedures demonstrate how to add and remove LDAP users through Team Studio.

Procedure

1. As an admin user, select **People** from the navigation menu.
2. On the **People** page, click **Add Person**.
3. To search for an LDAP user, type the user name, and then click **Check for Account**. The user's information is retrieved from the LDAP server and automatically filled in.
4. Click **Add Person**.

Removing LDAP Users

Users removed from the authorized group on the LDAP server are not automatically removed from Team Studio. They can no longer log into Team Studio, but the user account still exists in the application. To remove the account, manually delete it from Team Studio.

Prerequisites

Procedure

1. Select **People** from the navigation menu.
2. Browse to find the account.
3. Select the person to delete, and then click **Delete**.

Troubleshooting LDAP Configuration

Here are solutions for some common issues you might encounter while enabling LDAP authentication.

LDAP user does not appear in Chorus

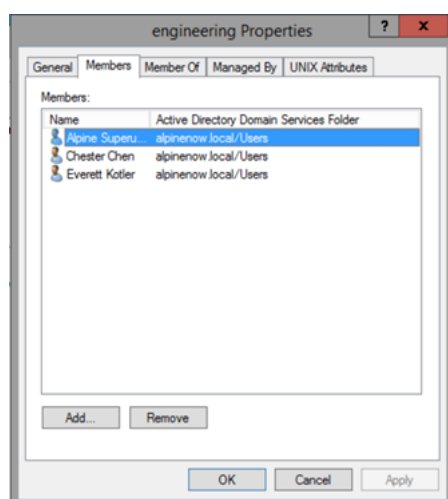
Issue: After running `rake ldap:import_users`, you do not see all of the users from your group imported into Team Studio.

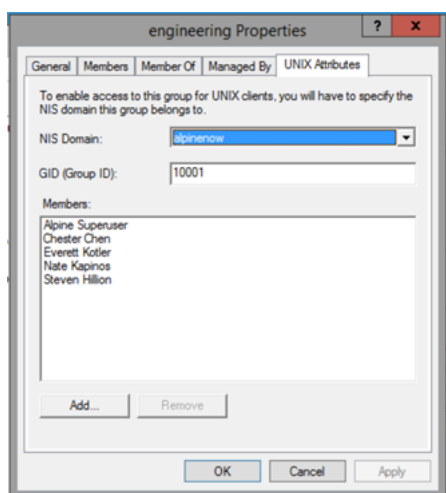
Potential cause: The Team Studio license allows fewer users than exist in the LDAP group.

Potential resolution: Identify the number of allowed users by checking the Team Studio license, and then limit the imported users to the number that the license allows. For more information, see the License Agreement for your Team Studio installation.

Potential cause: The Windows server running LDAP needs to communicate with the Unix servers, but some LDAP users are not listed on the Network Information System ([NIS](#)) domain.

Potential resolution: Add the users to the NIS global user list using the LDAP server tool.





No entry found

Error message: No entry found for user ___ in LDAP group Team Studio Users. Contact your system administrator.

Potential cause:

1. Create a group on the LDAP server.
2. Set `ldap.properties` to search that group.
3. Attempt to log in with a user who is not a member of the group.

Potential resolution: Verify that the user is a member of the desired group.

LDAP settings are mis-configured

Error message: LDAP settings are mis-configured. Please contact your System Administrator.

Potential cause: `ldap.properties` file contains parameters that are invalid, such as `ldap.bind.username` or `ldap.bind.password`.

Potential resolution: Validate parameters outside of Team Studio, and then modify as needed in `ldap.properties`.

LDAP Use Case Scenarios

The following LDAP use case scenarios describe steps for authenticating users who have or do not have group memberships, and how to import users from an LDAP group to Team Studio.

Scenario 1: LDAP Authentication with Group Membership

This workflow describes the steps for authenticating a user with a group membership. In this scenario, the first user's membership in a specified group is verified. After successful membership verification, the user is authenticated against the LDAP server with a qualified distinguished name (DN) and a user-supplied password.

Procedure

1. The user provides a user name and password to log in.
2. The login request is received by the Team Studio back end.

3. The Team Studio back end verifies that the user name is for a registered/licensed user. Note that the user's password is not being authenticated at this time; the only thing being verified is that the user is a valid Team Studio user.
4. If the user is a valid Team Studio user, the Team Studio back end sends a query message to the LDAP server to verify the user's group membership. The query parameters are read from the `ldap.properties` file.
5. The query request returns a result that verifies whether the current user is a member of the specified group.
6. If the current user is a member of the specified group, the Team Studio back end sends an authentication request to the LDAP server. The request parameters consist of a DN and a user-supplied password. The DN is constructed using the parameters from the `ldap.properties` file.
7. The LDAP server authenticates the user and returns.
8. If the user is authenticated successfully, the Team Studio back end navigates the user to the dashboard page. If the user is not authenticated, an error message is displayed.

Scenario 2: LDAP Authentication without Group Membership

This workflow describes the steps for authenticating a user without group membership. In this scenario, the user is authenticated against the LDAP server with a user ID and a user-supplied password.

Procedure

1. User provides user name and password to log in.
2. Login request is received by the Team Studio back end.
3. The Team Studio back end verifies that the user name is a registered/licensed user. Note that we are not authenticating the user's password at this time. We are only verifying that the user is a valid Team Studio user.
4. The Team Studio back end sends an authentication request to LDAP server. The request parameters consists of a Distinguished Name (DN) and user supplied password. The DN is obtained by querying the LDAP server.
5. LDAP Server authenticates the user and returns.
6. If the user is authenticated successfully by LDAP server, the Team Studio back end sends the user to the dashboard page. If LDAP server fails to authenticate the user, appropriate error message is displayed to the user.

Scenario 3: Import Users from an LDAP Group to Team Studio

After installing, customers must run a rake task to import users from an LDAP server to Team Studio if they authenticate users through a group membership.

Prerequisites

Before running the rake command, customers must edit the `ldap.properties` file to fill in values for LDAP server parameters. For detailed instructions, see [Enabling LDAP Authentication](#).

Procedure

1. The admin user starts the **rake ldap:import_users** command from the `$CHORUS_HOME/current` directory.

2. The Team Studio back end starts processing the rake task.
3. The Team Studio back end reads the LDAP server configuration from the `ldap.properties` file and sends a query to the LDAP server to fetch a list of members for a specified group.
4. The LDAP server returns with a list of members for the specified group.
For each user in the list:
5. The Team Studio back end sends a query to the LDAP server to fetch the properties (first name, last name, email address, user ID, and so on) for a specified user.
6. The LDAP server returns a list of properties for the specified user.
7. The Team Studio back end creates or updates a user in the Team Studio database.
8. The user is created successfully in the Team Studio database using the Collaborator role. The admin user can modify the role for each user, for example, by adding the Developer or Admin role.
9. The Team Studio back end completes the rake task and returns.

Enabling Single Sign-On - SAML and Configuring SSO Options

You can now use single sign-on with Team Studio to streamline your user provisioning and security.



SAML authentication has been tested with specific configurations of Shibboleth. While SAML is a standard, there is a great deal of variability in configuration between different IDP products, so it is possible that other IDP implementations, or other configurations of Shibboleth, might not interoperate correctly with Team Studio.

Prerequisites

Before configuring SAML, you must configure your `server_url` key in the `chorus.properties` file. It should look something like this:

```
server_url = http://mytsdsinstallation.mydomain.com:8080
```

Procedure

1. Log in as siteadmin, and from the sidebar menu, click **Administration**.
2. Click **Authentication**. The Authentication Configuration panel is displayed. By default, it shows **Internal Chorus Authentication** enabled and others disabled. Only one authentication system can be enabled at a time.

Authentication Configuration

Internal Chorus Authentication	✓
Enabled	
LDAP Enabled	✗
SAML Enabled	✗

Edit

3. Click **Edit** to change the settings. If you do not see an **Edit** option, ensure that you are logged in as siteadmin. Only a siteadmin can edit authentication settings.
4. Now you have the ability to enable a different authentication system. Click to enable SAML. The page changes to display a list of configuration options. We will go through them one by one.
5. Review the **Identity Provider (IdP)** section.

SAML Authentication Configuration

Identity Provider (IdP)

Identity provider metadata source

- ☒ Parse from IdP metadata url
☐ Copy and paste metadata

IdP Metadata Url *

- ☐ Require signed assertions from IdP

For the Identity Provider metadata source, you have two options:

- **Parse from IdP metadata:** This option attempts to fetch the IdP metadata from a URL that you enter. Additionally, whenever you start Team Studio, it fetches the metadata again. This design is useful for when you are first configuring the system.



If your IdP is momentarily unavailable when Team Studio is starting, Team Studio can become unresponsive. Try using the copy-and-paste option after you have configured the system correctly.

- **Copy and paste metadata:** Access the IdP metadata and paste it into the box provided. This stores the IdP metadata in the Team Studio database.

For extra security, you can enable **Require signed assertions from IdP**. With this enabled, Team Studio looks for a signature on incoming assertions from your IdP and displays an error if the signature is missing or invalid.

6. Configure the **Service Provider** section. The service provider is the application that you are using SAML to log into (in this case, Team Studio).

Service Provider

Custom SP Entity ID

- ☐ SAML sign own sp assertions

Usually, you can leave **Custom SP Entity ID** blank. The default entity ID is used, which is the URL where the Team Studio SAML SP metadata is located. This URL is of the form: `http://<tsds domain name>/auth/saml/metadata`.

If you want Team Studio to digitally sign outgoing SAML assertions, select the **SAML sign own sp assertions** box. If you enable this setting, you must also configure an SSL certificate/private key pair.

7. Configure the **Source of Username in SAML response** - set whether the user ID is obtained from the **NameID** or an attribute within the SAML assertion.

Source of Username in SAML response

- ☒ NameID
☐ Attribute

8. Configure the **Attribute map** section.

Attribute map

'email' attribute identifier *

'first_name' attribute identifier *

'last_name' attribute identifier *

'title' attribute identifier

'dept' attribute identifier

SAML authentication includes user provisioning and user updating. In other words, if a new user attempts to log in to Team Studio using SAML and is not already present in the Team Studio system, Team Studio creates a new user for them and then logs them in. This means that the SAML assertion must contain all of the required information for creating a new user in Team Studio. Some of the information must be parsed out of the SAML assertion. The User Roles can either be parsed from the SAML assertion, or retrieved using an external script.

The attribute map allows you to adapt the key format within your own system's SAML assertion, to the Team Studio key format. The following example demonstrates these settings.

For the **Administration Role** and **Application Role** settings, you either configure a slightly more complicated mapping setup, or you can specify the path to an external script installed in your system's path. The external scripts can be written in any language, but must be executable from the command prompt. They return an exit code that indicates which role to assign the newly created user.

Attribute map: "Administration Role"

Mapping method

- ☐ Parse from SAML authentication assertion
- ☐ Provide a custom script to retrieve the value

Attribute map: "Application Role"

Mapping method

- ☐ Parse from SAML authentication assertion
- ☐ Provide a custom script to retrieve the value

To review an example, download the sample scripts.

- `saml_administration_role_role_mapping.sh`
- `saml_application_role_role_mapping.sh`

9. Complete the **Other** section.

Other

☐ Send "Single Logout Request" to IdP on logout

If this is set, upon logout, Chorus will first redirect the browser to the IdP with a "Single Logout Request", and only when the IdP responds will the user will be logged out of Chorus.

Where to redirect after successful logout

If this is not set, upon logout, the user will be redirected to the internal Chorus login page. However, when SAML is enabled, only the superuser 'chorusadmin' can login using this method. So, most people will want to enter a URL from their IdP into this field.

Allowed clock drift (in seconds)

0

- **Send Single Logout Request:** When this option is selected, before terminating the session, Team Studio sends a SLO request to the IdP. Only when the IdP responds does Team Studio terminate the local session.
- **Where to direct after successful logout:** This option is usually your IdP's landing page. If left blank, users are redirected to the internal Team Studio login page. However, when SAML is enabled, only the siteadmin user can log in using that authentication method.
- **Allowed clock drift (in seconds):** This option allows some lock skew between the Team Studio server and your IdP server.

10. Click **Update** to save the authentication settings.

11. Log out of Team Studio. Reload the page. When it is reloaded, Team Studio attempts to redirect to your IdP.

What to do next

If you have trouble with your SAML configuration and need to return to the administration panel, you can skip authentication by appending `?skip_saml=true` to your Team Studio URL. (For example, `http://mytsdsinstallation.mydomain.com/?skip_saml=true`.)

For information about configuring the IDP, see [Configuring the IDP](#).

Configuring the IDP

After you enable single sign-on in Team Studio, when you visit the Team Studio URL and are not logged in, it attempts to redirect to the IDP login page instead of showing the normal Team Studio login page.

This might or might not succeed, depending on how the IDP is configured. In either case, the IDP must be configured to recognize Team Studio before authentication succeeds. The details of this process are specific to each IDP implementation; see your IDP documentation for details.

Below is a list of common steps for IDP configuration:

Prerequisites

Procedure

1. Download the Team Studio SAML metadata XML file from `http://mytsdsinstallation.mydomain.com/auth/saml/metadata`.
2. Inspect the metadata file and ensure that any URLs in it can be resolved by users' web browsers. The IDP redirects web browsers to these URLs at various points in the process. If the browser cannot resolve them, authentication fails. If the URLs are incorrect, you can manually fix the XML file or set the entity ID in the Team Studio authentication configuration to the correct value, and then re-download the file.
3. Provide this metadata file to your IDP using whatever mechanism your IDP provides.
4. Ensure that the IDP has access to whatever public certificates are necessary to validate the private key that was uploaded to the Team Studio authentication configuration panel earlier.
5. Ensure that the IDP is configured to provide the User ID and Role using the attribute names that Team Studio was configured to expect.
6. Ensure the changes to the IDP configuration have taken effect (a restart may be necessary).

Change the SAML Log In Configuration for the Chorus Administrator

After you have completely configured SAML, for additional security, you might want to disable As administrator, if you have trouble with your SAML configuration and need to return to the administration panel, you can skip authentication by appending `?skip_saml=true` to your Team Studio URL. (For example, `http://mytsdsinstallation.mydomain.com/?skip_saml=true`.) When SAML is fully configured, you can change the configuration to disable the ability to append the URL to skip authentication.

Edit this setting in the file `chorus.properties` on the computer where Team Studio is installed.

Prerequisites

You must have write access to the Team Studio server.

Procedure

1. Stop the Chorus service.
2. Open the file `chorus.properties`.

3. Add the key `saml_skip_saml_login_disabled=true`.
4. Restart the Chorus service.

What to do next

If you need to adjust your Authentication configuration, you must edit this setting again.

1. Stop the Chorus service.
2. Set the key `saml_skip_saml_login_disabled=false`.
3. Restart the Chorus service.
4. Log in as `chorusadmin` and make the configuration adjustments.
5. Set `saml_skip_saml_login_disabled=true`.
6. Restart the Chorus service.

Configuring Kerberos in the Team Studio Client

To connect to a Hadoop cluster with Kerberos authentication, configure your Team Studio client first to make sure the client is authorized by the Kerberos server.

Prerequisites for Configuring Kerberos in the Team Studio Client

To configure Kerberos in the Team Studio client, start with these prerequisites.

Prerequisites

Procedure

1. Make sure the Team Studio server can connect to the hosts with the host name (FQDN). For DNS lookup, choose option A or B:
 - a) Option A: Modify the `/etc/hosts` file of the Team Studio server and cluster nodes to include the host names and `ipaddress` of each server.
 - b) Option B: DNS lookup for all client and Hadoop nodes, as follows:

1. On the DNS server, add

```
alpinechorusserver IN A ipaddress
clusternode1 IN A ipaddress
clusternode2 IN A ipaddress
```

to these files:

```
/var/named/alpinenow.local.zone
/var/named/alpinenow.local.rr.zone
```

2. Restart the `named` service and verify using `telnet`, as follows:

```
service named restart
telnet hostname port
```

2. Install JCE on each node. Use the relevant `jce` file:

This is for using AES-256.

1. Remove the expired `local_policy.jar` and `US_export_policy.jar` files from `$JAVA_HOME/jre/lib/security`.
2. Unzip the JCE file for JRE6 (`jce_policy-6.zip`) or JRE7 (`jce_policy-7.zip`).
3. Copy the new `local_policy.jar` and `US_export_policy.jar` files to `$JAVA_HOME/jre/lib/security`.

Step-by-Step Guide to Configuring Kerberos in the Team Studio Client

After you complete the prerequisites, follow these steps to configure Kerberos in the Team Studio client. Perform this task on the server where Team Studio is installed.

Prerequisites

Make sure you have met all of the [Prerequisites for Configuring Kerberos in the Team Studio Client](#).

Procedure

1. On the Team Studio server, install the Kerberos application and run the following command as root. Note that this step is completed by Cloudera Manager if the Team Studio machine is running on a cluster node.

```
yum install krb5-devel.x86_64
yum install krb5-libs and krb5-workstation
```

2. Copy the `krb5.conf` file from the Kerberos server (not the Kerberos client). Save as `/etc/krb5.conf` on the Team Studio server (for example, `/etc/krb5.conf` for MIT KDC or AD KDC). The following `krb5.conf` file example uses AD KDC:

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = ALPINENOW.LOCAL
dns_lookup_kdc = true
dns_lookup_realm = false
ticket_lifetime = 86400
renew_lifetime = 604800
forwardable = true
default_tgs_enctypes = RC4-HMAC
default_tkt_enctypes = RC4-HMAC
permitted_enctypes = RC4-HMAC
udp_preference_limit = 1

[realms]
ALPINENOW.LOCAL = {
  kdc = 10.0.0.109
  admin_server = 10.0.0.109
}
```

3. Ensure that the user has total permissions on the HDFS directories `/tmp`, `/tmp/tsds_out`, `/user`, `/user/chorus`, and any other desired HDFS directory.

4. View the current ticket status using:

```
klist -e
```

5. Generate the principal for your client using option A, B, or C:

- Option A: Run `kadmin.local` from the Kerberos server to generate principal for your client:

```
kadmin.local
#addprinc -randkey [username]/[servername]@ALPINE
#xst -norandkey -k client.keytab [username]/[servername]@ALPINE
```

The user who runs `kinit` should be the user who runs jobs. For example, if we are running Hadoop jobs with the user `jenkins` user on `host.local`, we should run the following:

```
kadmin.local
#addprinc -randkey jenkins/host.local@ALPINE
#xst -norandkey -k /root/keytab/client/myjenkinskeytab.keytab jenkins/
host.local@ALPINE
```



If `kadmin.local` command is run from any machine other than the kdc server, repeat these steps:

3 (`kinit` as root user to generate a ticket)

4 (`klist` to view the ticket status)

5 (`kadmin.local` from kdc server to generate the principal)

- Option B: Use the existing keytab file if it exists. This file is for you to connect to the Kerberos server for authorization.
- Option C: Use `kadmin` in place of `kadmin.local`. Keep in mind that for some kdc versions, `kadmin` does not support `-norandkey`, so the keytab files are `/etc/krb5.keytab` and the password is changed every time we run `xst` within `kadmin`.

6. Copy the keytab files to your client server.

```
scp myjenkinskeytab.keytab root@host.local:/home/jenkins/keytab/
```

7. View the principal of your keytab file. You can do this with any user who has permission to access the keytab file.

```
klist -e -k -t /home/jenkins/keytab/myjenkinskeytab.keytab
```

8. Run `kinit` with your created credential:

```
kinit -k -t /home/jenkins/keytab/myjenkinskeytab.keytab jenkins/host.local@ALPINE
```

9. Add your `kinit` in `crontab` to renew every day with the user who is running the jobs:

```
crontab -e
0 6 * * * kinit -k -t /home/jenkins/keytab/myjenkinskeytab.keytab jenkins/
host.local@ALPINE
```

10. Verify user permissions:

- Hadoop account users should have read/write permissions to the `/tmp` directory in HDF:

```
hadoop fs -ls /
drwxrwxrwx - jenkins      superuser      0 2014-08-09 17:53 /tmp
```

- Take note of the user running the `kinit` command. In this example, the user `jenkins` has permission to access the keytab file.
- `jenkins` should have read/write permissions to the `/user` and `/tmp/tsds_out` directories in HDFS.
- `jenkins` is configured in the later steps for job tracker in the Hadoop data source.

11. Consider special instructions for specific Hadoop distributions. To run Team Studio on MapR, the Alpine host must have the MapR client installed.

12. Verify the connection between the Team Studio server and each node of the cluster.

telnet:

```
telnet namenode 8020
telnet: connect to address 10.0.0.xx: Connection refused # namenode is down or
firewall is up
telnet namenode 8020 # after turning on namenode
Connected to namenode. # Team Studio server can communicate with namenode
```

If the connections are not accessible, consider removing firewall restrictions using the `iptables` service.

13. Troubleshoot the following:

- Log locations:

```
[root@node3 hadoop]# pwd # this node contains both datanode and secondary
namenode
/var/log/gphd/hadoop
[root@node3 hadoop]# ls
hadoop-hdfs-datanode-node3.host.local.log
hadoop-hdfs-secondarynamenode-node3.host.local.log
```

.log is the latest version from the node.

- Time sync: Verify that the time on the Team Studio server is the same as that for the kdc server and Kerberos clients.
- Kerberos with HA (high availability): For kerberized clusters with HA enabled, try connecting to the active namenode before configuring both namenodes.

14. [Configure](#) a Kerberos-Enabled Hadoop data source.

Storing a Keytab for Jupyter Notebooks Running Python

You can manually configure Team Studio to store a keytab locally for Jupyter Notebooks for Team Studio running Python.

Perform this task on the computers where Team Studio is installed, and then on the computer where Jupyter Notebooks for Team Studio running Python is installed.

Prerequisites

- You must have generated the keytab.
- You must have write access to the server where Team Studio is installed.
- You must have write access to the server where Jupyter Notebooks for Team Studio running Python is installed.

Procedure

1. Open the file `chorus.properties`.
2. Add the line `notebook.kerberos.transfer_keytab_from_chorus = false`.
By default, this option is set to `true`.
3. Shut down the Chorus instance.
4. Log into the server on which Jupyter Notebooks for Team Studio running Python is installed.
5. Shut down all containers (you can use the Docker command `rm -f`).
6. Browse to the Notebooks installation location.
7. Find the folder named `hdfs_configs`.
8. In the `hdfs_configs` folder, replace the file `alpine_keytab.keytab` with your keytab file, using the name `alpine_keytab.keytab`.
9. In the `hdfs_configs` folder, create a file named `set_principal.sh` containing the line `export KERBEROS_PRINCIPAL="PRINCIPLE@EXAMPLE.COM"` with the principal to use with the kinit.
10. Restart Chorus.

Command-line Utilities for Managing the Services

We provide several utilities for system administrators to start and stop Team Studio from the command line, as well as from the application.

Do not perform the tasks in this section as root.

Starting Team Studio

As system administrator, you can use this command-line utility to start all services running on Team Studio.

Perform this task from the command-line interface on the computer on which Team Studio is installed.

To start one or more individual services, provide the service name after the `start` command. For more information, see [Start, stop, or restart individual services](#).

Prerequisites

Do not perform this task as root.

Procedure

1. Log in as user chorus.
2. Change to the directory where Team Studio is installed.

```
$ cd <Team Studio install path>
```

3. Run the following command.

```
$ source chorus_path.sh
$ chorus_control.sh start
```

Result

The Team Studio starts.

What to do next

After starting the services, access the application at *<hostname>:<port>*. The default port is 8080.

Provide the following credentials: *siteadmin/<password>*.

Stopping the Team Studio

As system administrator, you can use this command-line utility to stop all services running on Team Studio. Perform this task from the command-line interface on the computer on which Team Studio is installed.

To stop one or more individual services, provide the service name after the `stop` command. For more information, see [Start, stop, or restart individual services](#).

Prerequisites

Do not perform this task as root.

Procedure

1. Log in as user chorus.
2. Change to the directory where Team Studio is installed.

```
$ cd <Team Studio install path>
```

3. Run the following command.

```
$ source chorus_path.sh
$ chorus_control.sh stop
```

Result

The Team Studio stops.

Restarting the Team Studio

As system administrator, you can use this command-line utility to restart all services running on Team Studio.

Perform this task from command-line interface on the computer on which Team Studio is installed.

To restart one or more individual services, provide the service name after the `restart` command. For more information, see [Start, stop, or restart individual services](#).

Prerequisites

Do not perform this task as root.

Procedure

1. Log in as user chorus.
2. Change to the directory where Team Studio is installed.

```
$ cd <Team Studio install path>
```

3. Run the following command.

```
$ source chorus_path.sh
$ chorus_control.sh restart
```

Result

The Team Studio restarts.



Any currently-running jobs are aborted if the Team Studio service is restarted.

What to do next

After restarting the services, access the application at `<hostname>:<port>`. The default port is 8080.

Provide the following credentials: `siteadmin/<password>`.

Backing up Team Studio

To protect against data loss and to plan for disaster recovery, add a regular backup procedure for your installation of Team Studio to your other disaster recovery plans.

Perform this task on the server where Team Studio is installed. During the backup process, the following backup file is dumped into your backup directory.

```
chorus_backup_YYYYMMDD_HHMMSS.tar
```

We recommend running a cron job to backup Team Studio at least daily.



Team Studio logs and indexes are not stored in the backup file. After you restore the database, trigger index building.

Prerequisites

- Be sure that Team Studio is shut down before you run the backup process.
- Do not run this task as root.

Procedure

1. Change to the directory where Team Studio is installed.

```
$ cd <Team Studio install path>
```

2. Run the following code.

```
$ source chorus_path.sh
$ chorus_control.sh backup [-d dir] [-r days]
```

Option	Description
-d	Specifies the directory where the backup files are written. If you do not specify a backup directory, the backup utility creates the default backup directory <code>/data/chorus/bak</code>

Option	Description
-r	Specifies the number of days of backup files should be kept in the backup directory. Files more than -r days old are removed. If -r is not specified, no files are removed.

For example, the following command backs up the Team Studio files to /data/tsds/daily_bu and deletes backup files that are more than 10 days old.

```
chorus_control.sh backup -d /data/tsds/daily_bu -r 10
```

Restoring Team Studio

You can restore Team Studio manually.

Perform this task on the server where Team Studio needs to be restored.

Prerequisites

You must have a backup that contains your configuration and data files.

Procedure

1. Reinstall Team Studio following the general instructions in [Planning the Installation or Upgrade Tasks](#), but do not start the server.
2. Restore the configuration and data files from the most recent backup.
Example

```
$ cd <tsds installation path>
$ source chorus_path.sh
$ chorus_control.sh restore /data/chorus/daily_bu/chorus_backup_20121108_012809.tar
```

3. Start Team Studio.

Start, Stop, or Restart Individual Services

Team Studio consists of six services. You can start, stop, or restart all of the services, or you can start, stop, or restart the individual services.

The individual services that comprise Team Studio are as follows.

- postgres
- workers
- scheduler
- solr
- webserver
- alpine

Command	Action
chorus_control.sh start <service_name>	Starts the specified service.
chorus_control.sh stop <service_name>	Stops the specified service.
chorus_control.sh restart <service_name>	Restarts the specified service.

If you type one of the above commands but do not specify *<service name>*, then the command applies to all services.

Configuring Team Studio to Run as a Service

Follow these steps to configure the application to start when the system boots up and control with the **service** command:

Perform this task on the server where Team Studio is installed.

Prerequisites

You must have write access to the server.

Procedure

1. Create a file called `chorus.server` in the `/etc/init.d` directory and copy the following content into that file (if you need to start the application as a different user than the default `chorus`, change that in the content below):

```
#!/bin/bash
#set -x
#
# Starts a Server
# chkconfig: 345 90 10
# description: Chorus Server
export CHORUS_HOME=/usr/local/chorus
export CHORUS_USER=chorus
export PID_PATH=/var/lock/subsys
export PATH=$PATH:$CHORUS_HOME
export PGPASSFILE=$CHORUS_HOME/.pgpass
#Source Function Library
. /etc/rc.d/init.d/functions
RETVAL=0
#PIDFILE="/var/lock/subsys/chorus.pid"
PIDFILE=$PID_PATH"/chorus.pid"
desc="Chorus Server Daemon"
#####
start() {
    echo -n "Starting $desc (CSD): "
    # source $CHORUS_HOME/chorus_path.sh /dev/null2>&1
    daemon --user=$CHORUS_USER $CHORUS_HOME/chorus_control.sh start
    # daemon --user=$CHORUS_USER $CHORUS_HOME/chorus_control.sh main /dev/null2>&1
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch $PIDFILE
    return $RETVAL
}
stop() {
    echo -n "Stopping $desc (CSD): "
    # source $CHORUS_HOME/chorus_path.sh /dev/null2>&1
    daemon --user=$CHORUS_USER $CHORUS_HOME/chorus_control.sh stop
    RETVAL=$?
    sleep 5
    echo
    [ $RETVAL -eq 0 ] && rm -f $PIDFILE
    # MONITOR=`ps aux | grep chorus_control | grep -v grep | awk '{print $2}'`
    # kill -9 $MONITOR
}
checkstatus(){
    ps -ef | grep chorus
}
restart() {
    stop
    start
}
case "$1" in
    start)
```

```

    start
    ;;
stop)
    stop
    ;;
status)
    checkstatus
    ;;
restart)
    restart
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart}"
    exit 1
esac
exit $RETVAL

```

2. Run this command to make the newly created script start when the system boots up:

```
chkconfig --add chorus.server
```

3. Now you can start and stop the processes by running the following commands. Also, the services start automatically when the system boots up.

```
service chorus.server start
service chorus.server stop
```

Team Studio Configuration Files

Use the property files in this section to configure your deployment of Team Studio.

Team Studio Deploy Properties

Many of the Team Studio workflow engine configurations are defined in the file `deploy.properties`.

The `deploy.properties` file is located in the directory `<installation_directory>/shared/ALPINE_DATA_REPOSITORY/configuration/`. For example, `/usr/local/chorus/shared/ALPINE_DATA_REPOSITORY/configuration/deploy.properties`.

Tomcat settings

Use the key `alpine.catalina.opts` to set the JVM settings for the Team Studio services and other tomcat settings.

```
alpine.catalina.opts=-server -Xms4096M -Xmx8192M -XX:PermSize=1024M -
XX:MaxPermSize=1024M -DREST_ENABLED=true -Djava.security.egd=file:/dev/./urandom
```

Kerberos settings



Configuring a kerberos-enabled Hadoop data source prevents the Team Studio application from connecting to non-kerberized Hadoop data sources.

To use a Kerberos-enabled Hadoop data source within Team Studio third-party applications (such as the Team Studio workflows), include the following configurations in the file `deploy.properties`.

```

alpine.kerberos.enabled=true
alpine.kerberos.kdc_server=<kdc_server>
alpine.kerberos.kdc_realm=<kdc_realm>
alpine.kerberos.keytab_filepath=<my_keytab_filepath>
alpine.kerberos.principal=<MAPRED principal>

```

Kerberos property for Team Studio	Setting
<code>alpine.kerberos.enabled</code>	true or false. Specifies if Kerberos is enabled for Team Studio.
<code>alpine.kerberos.kdc_server</code>	The name of the Key Distribution Center (KDC) server.

Kerberos property for Team Studio	Setting
<code>alpine.kerberos.kdc_realm</code>	The name of the KDC realm. Usually the same as the DNS domain name.
<code>alpine.kerberos.keytab_filepath</code>	The path to the Kerberos file containing the keytab entry for the specified principal.
<code>alpine.kerberos.principal</code>	The mapreduce principal name used by the server.



Team Studio must be configured. For more information, see >>Configure a Kerberos-Enabled Hadoop data source<<.

The Properties File

Much of the Team Studio configuration is defined by properties in the `chorus.properties` file, described below.

The file is located in the `<installation directory>/shared/` directory; for example, `/usr/local/chorus/shared/chorus.properties`.

In the same directory as `chorus.properties` is a file named `chorus.properties.example`. This file contains all of the properties that can be set in the `chorus.properties` file. Review this file to learn about the configuration options for Team Studio.

The properties in the actual configuration file, `chorus.properties`, might be a subset of the properties in `chorus.properties.example`. You can include any attribute you find in `chorus.properties.example` in your `chorus.properties` configuration file.

To see a detailed breakdown of options you can configure in `chorus.properties`, see [Team Studio Configuration Properties](#).

You must restart Team Studio for changes you make in the `chorus.properties` file to take effect.

Configuring Indexing Frequency for Database Instances

The `reindex_datasets_interval_hours` configuration property in the `chorus.properties` file is used to set how frequently datasets are reindexed.

The following example sets datasets to be reindexed once a day (every 24 hours).

```
reindex_datasets_interval_hours= 24
```

Team Studio Configuration Properties

Use the file `chorus.properties` to configure much of Team Studio.

The configuration file and its companion example file are located in the directory `<installation directory>/shared/`. The example file, `chorus.properties.example`, contains examples of all of the properties that you can set in `chorus.properties`. You can use this example file to learn about the possible properties you can set in `chorus.properties`. You can include any attribute from the example file in the configuration file.




You must restart Team Studio to the changes you make in `chorus.properties` to take effect.


chorus.properties options

Property	Default setting	Description
public_url	<i>mycomputer.example.com</i>	Network location of the Team Studio application. Do not include HTTP: or the port number.
server_port	8080	The Team Studio application default port.
postgres_port	8543	The PostgreSQL port.
solr_port	8983	The Apache Solr port.
java_options	- Djava.library.path= \$CHORUS_HOME /vendor/ hadoop/lib/ -server - Xmx2048m - Xms512m - XX:MaxPermSize=128	Command-line options to include on the Java command line when running the Team Studio application.
ssl.enabled	FALSE	Use to configure SSL for Team Studio. Set <code>ssl.enabled=true</code> to enable SSL.
ssl_server_port	8443	Use to configure SSL for Team Studio. Set the port number with <code>ssl_server_port</code> .
ssl_certificate	n/a	Use to configure SSL for Team Studio. Set to the path of the server's SSL certificate.
ssl_certificate_key	n/a	Use to configure SSL for Team Studio. Set to the location of the private key.
workflow.enabled	true	Enables the workflow feature in Team Studio. The Team Studio server port is set during installation.

Property	Default setting	Description
<code>workflow.url</code>	<code>http://localhost:8070</code>	<p>The URL of the Team Studio server, which is installed with Team Studio. The URL can be an IP address or fully-qualified machine name. Whichever is used, it should be reachable from a browser.</p> <p>If you must change the port number after installing Team Studio, be sure to also change the port number in the Team Studio Tomcat server configuration file, <code>\$CHORUS_HOME/alpine/apache-tomcat-7.x.x/conf/server.xml</code>. Look for the <code><Connector></code> element with attribute <code>protocol="HTTP/1.1"</code> under the <code><Service name="Catalina"></code> element.</p>
<code>smtp.address</code>	<code>localhost</code>	Configures the SMTP connection that Team Studio uses to deliver email notifications to users. Sets the network address of the SMTP service.
<code>smtp.port</code>	<code>587</code>	Configures the SMTP connection that Team Studio uses to deliver email notifications to users. Sets the port for the SMTP service.
<code>smtp.user_name</code>	<code>USER_NAME</code>	Configures the SMTP connection that Team Studio uses to deliver email notifications to users.
<code>smtp.password</code>	<code>PASSWORD</code>	Configures the SMTP connection that Team Studio uses to deliver email notifications to users.
<code>smtp.authentication</code>	<code>login</code>	Configures the SMTP connection that Team StudioTeam Studio uses to deliver email notifications to users.
<code>smtp.enable_starttls_auto</code>	<code>false</code>	Configures the SMTP connection that Team Studio uses to deliver email notifications to users.
<code>mail.enabled</code>	<code>FALSE</code>	If true, Team Studio delivers job completion and failure notifications to users by email.
<code>mail.from</code>	<code>FROMNAME <noreply@chorus.com></code>	Sets the <code>from</code> header in the email message.
<code>mail.reply_to</code>	<code>REPLY NAME <noreply@chorus.com></code>	Sets the <code>reply_to</code> header in the email message.

Property	Default setting	Description
sandbox_recommended_size_in_gb	5	<p>The sandbox-related setting. The default unit is in GB.</p> <div>  <p>This value provides a visual indicator that indicates when a workspace's sandbox exceeds the recommended size.</p> </div>
worker_threads	1	Configures the thread pool size of webserver and worker processes.
webserver_threads	20	The number of webserver threads determines the maximum number of simultaneous web requests.
database_threads	25	<p>The number of worker threads determines the maximum number of asynchronous jobs, such as table copying or importing, that can be run simultaneously.</p> <p>Each web or worker thread can use its own connection to the local PostgreSQL database. Therefore, the sum of <code>worker_threads</code> and <code>webserver_threads</code> must be less than the <code>max_connections</code> configured in <code>postgresql.conf</code>. The <code>max_connections</code> parameter can be based on the operating system's kernel shared memory size. For example, on OS X this parameter defaults to 20.</p>
session_timeout_minutes	480	The default session timeout time. The number of minutes you can be inactive before you are logged out.
clean_expired_api_tokens_interval_hours	24	renamed in 6.2 from <code>clean_expired_sessions_interval_hours</code> .
delete_unimported_csv_files_interval_hours	1	
delete_unimported_csv_files_after_hours	12	
instance_poll_interval_minutes	5	
reindex_search_data_interval_hours	24	
reindex_datasets_interval_hours	24	Sets the frequency for dataset reindexing.

Property	Default setting	Description
<code>reset_counter_cache_interval_hours</code>	24	
<code>file_download.name_prefix</code>	n/a	This optional string is prefixed on all generated file names. For example, if a user downloads a dataset, the name of the file downloaded is the specified prefix, followed by the dataset name and then the <code>.csv</code> extension. Only the first 20 characters of the prefix is used.
<code>file_sizes_mb.workfiles</code>	10	Maximum upload work file size.
<code>file_sizes_mb.csv_imports</code>	100	Maximum size for imported files.
<code>file_sizes_mb.user_icon</code>	5	Maximum size for the user icon.
<code>file_sizes_mb.workspace_icon</code>	5	Maximum size for the workspace icon.
<code>file_sizes_mb.attachment</code>	10	Maximum size for file attachments.
<code>logging.syslog.enabled</code>	false	If true, logs are written to syslog rather than to files.
<code>logging.loglevel</code>	info	The minimum severity of messages to log. Can be debug, info, warn, error, or fatal.
<code>oracle.enabled</code>	TRUE	Enables use of Oracle databases.
<code>gpfdist.ssl.enabled</code>	false	To enable data movement between databases, gpfdist must be installed and running on the Team Studio host. Two gpfdist processes must be started with different ports pointing to the same directory. An SSL certificate must be installed on all segment servers.
<code>gpfdist.url</code>	sample-gpfdist-server	
<code>gpfdist.write_port</code>	8000	
<code>gpfdist.read_port</code>	8001	
<code>gpfdist.data_dir</code>	/tmp	
<code>tableau.enabled</code>	TRUE	If false, Tableau is disabled even if other Tableau parameters are specified.

Property	Default setting	Description
<code>tableau.url</code>	>ip address>	The URL of the Tableau server. The URL can be an IP address or a fully-qualified computer name. Whichever is used, it should be reachable from a browser.
<code>tableau.port</code>	8000	<p>The Tableau server port.</p>  This port must be opened on your Tableau server.
<code>tableau.sites</code>	marketing, sales	The list of Tableau sites. Team Studio supports this parameter starting in version 5.3. If this option is not present, Team Studio publishes to the default Tableau site.
<code>newrelic.enabled</code>	false	Enables New Relic application performance monitoring. See http://newrelic.com for more information.
<code>newrelic.license_key</code>	NEWRELIC_LICENSE_KEY	
<code>default_preview_row_limit</code>	500	The maximum number of preview rows.
<code>execution_timeout_in_minutes</code>	300	The workfile execution timeout in minutes.
<code>visualization.overlay_string</code>	n/a	<p>This optional string is displayed on all visualizations, both when displaying and when saving.</p> <p>Only the first 40 characters of the prefix are used.</p>
<code>database_login_timeout</code>	10	Database connection timeout, in seconds. If you are using Google BigQuery as a data source and you are copying large amounts of data between databases, consider increasing this value so the operation does not fail unexpectedly.
<code>jdbc_schema_blacklist.postgresql</code>	[information_schema, pg_catalog]	Specifies a list of PostgreSQL schemas that are excluded from display, index, and search. (That is, they are effectively excluded from Team Studio).

Property	Default setting	Description
<code>jdbc_schema_blacklist.sqlserver</code>	[db_accessadmin, db_backupoperator, db_datareader, db_datawriter, db_ddladmin, db_denydatareader, db_denydatawriter, db_owner, db_securityadmin, dbo, INFORMATION_SCHEMA, sys]	Specifies a list of SQL Server schemas that are excluded from display, index, and search. (That is, they are effectively excluded from Team Studio).
<code>jdbc_schema_blacklist.teradata</code>	[All, Crashdumps, DBC, dbcmngr, Default, EXTUSER, LockLogShredder, PUBLIC, SQLJ, SysAdmin, SYSBAR, SYSLIB, SYSSPATIAL, SystemFe, SYSUDTLIB, Sys_Calendar, TDPUSER, TDQCD, TDStats, tdwm, TD_SYSFNLIB, TD_SYSEXML]	Specifies a list of Teradata schemas that are excluded from display, index, and search. (That is, they are effectively excluded from Team Studio).

Team Studio Log Files

Depending on the log level set in `chorus.properties`, the volume of log files can vary substantially.

Supported log levels:

- debug
- info
- warn

- error
- fatal

Log name	File path	Description
production.log	<chorus-root>/shared/log/ production.log	Contains information about requests sent to the Team Studio web server, and various debugging information such as server errors, file not found errors, and permission denied messages.
worker.production.log	<chorus-root>/shared/log/ worker.production.log	Contains logs for the background worker threads that Team Studio uses to perform various asynchronous tasks such as database imports and checking instance statuses.
scheduler.production.log	<chorus-root>/shared/log/ scheduler.production.log	Contains information about jobs that the scheduler issues to different background workers. This mainly shows that a task was scheduled. See worker.production.log for more detailed information about what happened during execution of a task.
solr-production.log	<chorus-root>/shared/log/ solr-production.log	Contains information about solr search queries issued against Team Studio.

nginx

nginx maintains `access.log` and `error.log` files in `<chorus-root>/shared/log/nginx`.

syslog

As an alternative to the log files listed above, all logs can be combined in one file using syslog as the logger. To turn on syslog as the logger, put `logging.syslog = true` in `<chorus>/shared/chorus.properties`.

logrotate

You can use the Linux command `logrotate` to rotate your log files and prevent accumulation. By running `logrotate your_logrotate.conf` from a cron job, you can ensure that the logs get rotated at preset intervals.

Here is an example of a `your_logrotate.conf` configuration file that rotates all the important Team Studio log files:

```
daily
rotate 4
copytruncate
size 10M
<chorus>/shared/log/production.log {
}
```

```
<chorus>/shared/log/nginx/access.log {
}
<chorus>/shared/log/nginx/error.log {
}
<chorus>/shared/log/solr-production.log {
}
<chorus>/shared/log/worker.production.log {
}
<chorus>/shared/log/scheduler.production.log {
}
```

See the [logrotate manual page](#) for more information about the features of logrotate.



If you use syslog, you do not need to rotate your logs manually; syslog rotates the log files for you.

Download Logs

Team Studio provides a download of the application logs. When contacting Support, please provide the logs. Having the logs makes it much easier for Support to assist you. Perform this task on the server where Team Studio is installed.

Prerequisites



You must have administrator privileges to download logs. If you do not have the permissions, contact your administrator.

Procedure

1. In the upper right corner, click your user name, and then click **Support**.

2. Click **Download Logs**. The logs download in a .zip file.

There are several types of logs, depending what type of agents you have installed on your Team Studio instance.

Team Studio workflow editor installation logs

A log of the installation process is stored here. Send this file along if you have had problems with your installation.

Team Studio workflow editor logs

Team Studio workflow editor install logs

This log contains information about the installation or upgrade process.

Team Studio agent logs

For each Team Studio agent you have enabled, the log downloader outputs one file per agent. If you know what agents you are using, you can inspect these logs to determine whether there are underlying problems with your data source. These are also very useful in helping Support understand the situation.

Alpine.log

This log contains debug logs and information about your Team Studio installation. Error information about operators also can be found here.

Team Studio collaboration framework logs

Information about Team Studio processes and errors can be found here. This folder includes most information about the web instance, including logs for the job scheduler in the Team Studio collaboration framework, Jetty, Solr, and the production web logs. In addition, you can find information about the [supervisord](#) processes.

Config files

These files have information about your preferences and runtime settings for Team Studio. This includes information such as Spark defaults and operator preferences.

License file

This folder contains the license information for your Team Studio installation.

Postgres logs

These logs are for the Postgres database that supports the Team Studio web application. This is different than any Postgres databases you might be using as a data source.

Properties files

These files are useful for Support to see how your instance is configured. These files contain information about your enabled agents, ports that are in use, Java options, and other configuration information.

Registered HDFS data sources

If you have data sources on HDFS, you can see configuration information about them here.

Administering Team Studio

As administrator, you can configure the Team Studio experience, manage user permissions, set preferences, and other tasks. This help provides information on those tasks.

Connecting Team Studio to Data Sources

Review and follow these steps to connect your installation of Team Studio to your data sources. Perform this task on the computer where you have installed Team Studio.

Prerequisites

Test network connectivity and configure the Team Studio server.

Procedure

1. Enable web sockets.
Verify that web sockets are correctly enabled by using a web socket test.
2. Access the cluster nodes, including the NameNode and DataNodes for Hadoop.
Verify that you can connect to them by using the command `$ telnet hostname port`.
3. Enable read and write permissions for the appropriate directories, including `/tmp` for Hadoop.
Verify this step by writing to a file in one of those directories and running a MapReduce job, if applicable.
4. Ensure that the appropriate agent is enabled for your data source.
5. Configure the necessary ports in `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf`.
6. If you are using Spark, ensure the following.
 - The Spark host is added in `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf`.
`alpine.spark.sparkAkka.akka.remote.netty.tcp.hostname = IP address for Team Studio Server`
 - Full communication is open between the Team Studio server and all cluster nodes.
7. Ensure the Team Studio server can access the LDAP server if applicable.
Verify that you can connect by using `$ telnet hostname port`.

What to do next

Connect to either a database data source or a Hadoop data source.

Database Data Sources

You can add a database as a data source in Team Studio from the sidebar menu by selecting **Data** and then selecting **Add Data Source**.

For each database or JDBC data source, provide the following information about that data source.

Connect to a JDBC Data Source

You can connect your Team Studio installation to a JDBC data source. Perform this task on the computer where Team Studio is installed.

Prerequisites

- Check [System Requirements](#) to ensure you are using a supported version of the JDBC data source.
- You must have write access to the computer where Team Studio server is installed.
- Before you add a JDBC data source to Team Studio, place the associated JDBC driver for the data source in the `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc_driver/Public` and `$CHORUS_HOME/shared/libraries` folders

Procedure

1. Open the Add Data Source dialog box.

2. For **Data Source Name**, provide a user-facing name.
You can provide any useful text.
3. Optional: Provide a useful **Description**.
4. For **JDBC URL**, provide the JDBC URL used to connect to the data source.
5. Provide the **Database Account** and **Database Password**.

These values are your database credentials.

6. Optional: Select **Set database credentials as a shared account** if you intend to allow all users to access the data source without using their own credentials.

Users access the database with your credentials as the data source owner. If you do not select this check box, each user must provide credentials for that data source to access it. You can check the box later if you change your mind.

Connect to a Hive JDBC Data Source

This topic describes how to make a Hive data source available as a JDBC connection to Team Studio. For information about which Hadoop distributions support Hive as a JDBC data source, see [System Requirements](#). For information about adding Hive as a Hadoop data source, see [Connect to a Hive Data Source on Hadoop](#).

Prerequisites

Procedure

- Place the appropriate Hive JAR files in the `~/ALPINE_DATA_REPOSITORY/jdbc_driver/Public` and `$CHORUS_HOME/shared/libraries` folders. The list of necessary JARS is as follows:

- `commons-logging-*.jar`
- `hive-common*.jar`
- `hive-exec*.jar`
- `hive-jdbc*.jar`
- `hive-metastore*.jar`
- `hive-service*.jar`
- `httpclient*.jar`
- `httpcore*.jar`
- `libfb303*.jar`
- `libthrift*.jar`
- `log4j*.jar`
- `slf4j-api*.jar`

The * indicates that the version might be different, depending upon vendor. These JARs should all be available from the vendor installation.

Hive JDBC on CDH, HDP, or PHD

Procedure

- Fill in the required fields, marked with an asterisk *.
 - **Data Source Name:** Set a user-facing name for data source. You can choose anything you like.
 - **Description:** Enter some optional text with information about this data source.
 - **Hadoop Version:** The distribution of Hadoop that is running your Hive server. CDH5, HDP and PHD are supported Hadoop distributions. Note: the JAR files copied into `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc_driver/Public/` and `$CHORUS_HOME/shared/libraries` must match the Hadoop distribution you select here.
 - **JDBC URL:** the JDBC URL used to connect to the data source.
 - **CDH:** The URL should be in the format `jdbc:hive2://SERVER_HOSTNAME:PORT`
 - **HDP:** The URL should be in the format `jdbc:hive2://SERVER_HOSTNAME:PORT/default?stack.name=hdp;stack.version=<hdpversion>`
 - **PHD:** The URL should be in the format `jdbc:hive2://SERVER_HOSTNAME:PORT/default?stack.name=phd;stack.version=<phdversion>`
 - **Authentication:** Team Studio supports standard password authentication and Kerberos authentication. Select the type of authentication that is configured on your Hive server.
 - **Database Account and Database Password:** If **Account/Password** authentication type is selected, enter the Hive metastore account and password. By default, Hive uses an account of **hive** with password of **hive**.
 - **Kerberos:** If the Kerberos authentication type is selected, enter the Kerberos Principal and Kerberos Keytab Location. The Kerberos principal must have permission to access the Hive server, and is typically `hive/myHadoopcluster.com@mycompany.com`.
 - **Set database credentials as a shared account:** If your authentication type is set to **Account/Password**, the option to share the database credentials is available. If you check **Set database credentials as a shared account**, all users can access the data source without providing their own credentials - they are accessing the database with your credentials as the data source owner. If you

do not check this box, each user must enter his or her own credentials for that data source in order to access it. You can change this setting later if you change your mind.

Connect to an Oracle Database

You can add Oracle as a data source, but to do so you must first enable it in the `chorus.properties` file. You might need to add `oracle.enabled=true`. You must also copy the `ojdbc8.jar` file to your `/shared/libraries` folder.



Team Studio can exclude schemas from displaying. The list of blacklisted schemas is in `chorus.properties` and is editable.

Prerequisites

Procedure

1. Follow the steps in [Enable Oracle Data Sources](#). Supported Oracle versions can be found in [System Requirements](#).

ADD DATA SOURCE [X]

Data Source Type

Adding an Oracle database instance requires read permissions on the database. Currently supporting Oracle 10g and 11g.

Data Source Name *

Description

Host * Port *

Database Name *

Database Account * Database Password *

☐ Set database credentials as a shared account ⓘ

* Required field

Cancel Add Data Source

2. Fill in the required fields and, if you like, the optional **Description**.
 - **Data Source Name:** Set a user-facing name for data source. You can choose anything you like.
 - **Database Name:** Enter the actual name of your Oracle database.
 - **Database Account** and **Database Password:** Provide your credentials for that database.
 - If you check the **Set database credentials as a shared account** box, all users can access the data source without providing their own credentials - they are accessing the database with your credentials as the data source owner. If you do not check this box, each user must enter his or her

own credentials for that data source in order to access it. You can change this setting later if you change your mind.

Enable Oracle Databases

This topic describes how to enable an Oracle database prior to adding Oracle as a data source.

Prerequisites

Procedure

1. Place the Oracle client driver JAR, `ojdbc8.jar`, in the following locations:

```
<installation directory>/shared/ALPINE_DATA_REPOSITORY/jdbc_driver/  
<installation directory>/shared/ALPINE_DATA_REPOSITORY/jdbc_driver/Public  
<installation directory>/shared/libraries
```



If the JAR file is not in the correct place, the error "JDBC Driver class not found" appears.

You can find the Oracle client driver at <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>.

2. Set the following permissions on the file:

```
chmod 644 ojdbc8.jar  
chown chorus:chorus ojdbc8.jar
```

3. Set `oracle.enabled` to `true` in the `chorus.properties` file and restart Team Studio.

Connect to a Greenplum Database

You can connect your Team Studio installation to a Greenplum database. Perform this task on the computer where Team Studio server is installed.

Prerequisites

- Check [System Requirements](#) to ensure you are using a supported version of Greenplum.
- You must have write access to the computer where Team Studio server is installed.
- If another Team Studio user has greater privileges for a shared account database than you, the owner, that user must provide credentials to see the parts of the database denied to you.

Procedure

1. Open the Add Data Source dialog box.

2. For **Data Source Name**, provide a user-facing name.
You can provide any useful text.
3. For **Database Name**, set the actual database name.
Postgres is the default because many Greenplum users have a database with that name. If you are adding a database with a different name, provide the name here.
4. Provide the **Database Account** and **Database Password**.
These values are your database credentials.
5. Optional: Select **Set database credentials as a shared account** if you intend to allow all users to access the data source without using their own credentials.
Users access the database with your credentials as the data source owner. If you do not select this check box, each user must provide credentials for that data source to access it. You can check the box later if you change your mind.
6. Select **Use SSL** if you are using SSL-enabled PostgreSQL and Greenplum
If you choose this option, communications between Team Studio and the database are secured using SSL.

If you specify SSL, you must have [installed an SSL certificate](#).
7. Optional: Provide an entry for the **Description**.

Connect to a Pivotal HAWQ Database

You can connect your Team Studio installation to a Pivotal HAWQ database. Perform this task on the computer where Team Studio server is installed. Supported HAWQ versions can be found in [System Requirements](#).

Prerequisites

- Check [System Requirements](#) to ensure you are using a supported version of Pivotal HAWQ.
- You must have write access to the computer where Team Studio server is installed.
- If another Team Studio user has greater privileges for a shared account database than you, the owner, that user must provide credentials to see the parts of the database denied to you.

Procedure

1. Open the Add Data Source dialog box.

2. For **Data Source Name**, provide a user-facing name.
You can provide any useful text.
3. For **Database Name**, set the actual database name.
Postgres is the default because many Greenplum users have a database with that name. If you are adding a database with a different name, provide the name here.
4. Provide the **Database Account** and **Database Password**.
These values are your database credentials.
5. Optional: Select **Set database credentials as a shared account** if you intend to allow all users to access the data source without using their own credentials.
Users access the database with your credentials as the data source owner. If you do not select this check box, each user must provide credentials for that data source to access it. You can check the box later if you change your mind.
6. Select **Use SSL** if you are using SSL-enabled PostgreSQL and Greenplum
If you choose this option, communications between Team Studio and the database are secured using SSL.

If you specify SSL, you must have [installed an SSL certificate](#). See documentation about [PostgreSQL JDBC/SSL Connections](#) for more information.
7. Optional: Provide an entry for the **Description**.

Connect to an Amazon RedShift Data Source

You can connect Team Studio to an Amazon RedShift data source. Perform this task on the computer where Team Studio server is installed.

Prerequisites

You must have write access to the Team Studio server. You must also have access to the Amazon RedShift configuration files.

Procedure

1. Copy the RedShift driver to the following directories, and then change the ownership of these copies to the user who runs Team Studio (usually user 'chorus').

- `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc_driver/Public`
- `$CHORUS_HOME/shared/libraries`

2. Change the ownership of these copies to the user who runs Team Studio.

Usually, that user name is 'chorus'.

3. Create a new redshift directory named `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc/redshift/`.
4. Copy the file `driver.properties` from the directory `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc/default` to the newly created redshift directory.
5. Edit the contents of the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc/redshift/driver.properties` as follows.

```
# Specify the JDBC class driver for the desired database type.
# Examples:
# Oracle = oracle.jdbc.driver.OracleDriver
# Greenplum = org.postgresql.Driver
# DB2 = com.ibm.db2.jcc.DB2Driver
# Netezza = org.netezza.Driver
# PostgreSQL = org.postgresql.Driver
# SQLServer = com.microsoft.sqlserver.jdbc.SQLServerDriver
# MySQL = com.mysql.jdbc.Driver
# Teradata = com.teradata.jdbc.TeraDriver
# Vertica = com.vertica.jdbc.Driver
# Sybase = com.sybase.jdbc2.jdbc.SybDriver
# Informix = com.informix.jdbc.IfxDriver
# SAPDB = com.sap.dbtech.jdbc.DriverSapDB
# InterBase = interbase.interclient.Driver
# HSqlDB = org.hsqldb.jdbcDriver
# MariaDB = org.mariadb.jdbc.Driver
# MySQL = com.mysql.jdbc.Driver
driverClass=com.amazon.redshift.jdbc41.Driver
```

6. Locate and open for editing the file `additional_jdbc_drivers.rb`.

The path is similar to path similar to `/data/chorus/install/releases/5.9.1.0.3973-5d95f7c97/components/core/app/mixins/sequel/extensions/additional_jdbc_drivers.rb`

7. Add a line for the redshift class so that the content resembles the following.

```
module Sequel
  module AdditionalJdbcDrivers
    MAP = {
      mariadb: ->(db) { org.mariadb.jdbc.Driver },
      teradata: ->(db) { com.teradata.jdbc.TeraDriver },
      vertica: ->(db) { com.vertica.jdbc.Driver },
      hive2: ->(db) { org.apache.hive.jdbc.HiveDriver },
      hive: ->(db) { org.apache.hadoop.hive.jdbc.HiveDriver },
      redshift: ->(db) { com.amazon.redshift.jdbc41.Driver }
    }
  end
end
```

```

MAP.each do |key, driver|
  ::Sequel::JDBC::DATABASE_SETUP[key] = driver
end
end
end
end

```



You must apply this change to the file `additional_jdbc_drivers.rb` again after upgrading Team Studio.

8. Restart Team Studio.
9. Open the Add Data Source dialog box.

10. Provide the **Data Source Type**, the **Data Source Name**, and (optionally), the **Description**.
11. Set the data connection (**JDBC URL**) using a URL similar to the following.



You can copy your RedShift URL from your AWS RedShift page

```

jdbc:redshift://armen-jjredshift.csyb6t8bifc8.us-west-1.redshift.amazonaws.com:5439/armenjjdb

```

12. Optional: Select **Set database credentials as a shared account** if you intend to allow all users to access the data source without using their own credentials.

Users access the database with your credentials as the data source owner. If you do not select this check box, each user must provide credentials for that data source to access it. You can check the box later if you change your mind.

Connect to a BigQuery Data Source

You can connect Team Studio to a GCP BigQuery data source. Perform this task on the computer where Team Studio server is installed.

Prerequisites

You must have write access to the Team Studio server. You must also have access to the GCP BigQuery configuration files.

Procedure

1. Copy the BigQuery driver to the following directories, and then change the ownership of these copies to the user who runs Team Studio (usually user 'chorus').

- `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc_driver/Public`
- `$CHORUS_HOME/shared/libraries`

2. Change the ownership of these copies to the user who runs Team Studio.

Usually, that user name is 'chorus'.

3. Create a new bigquery directory named `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc/bigquery/`.
4. Copy the file `driver.properties` from the directory `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc/default` to the newly created bigquery directory.
5. Edit the contents of the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/jdbc/bigquery/driver.properties` as follows.

```
# Specify the JDBC class driver for the desired database type.
# Examples:
# Oracle = oracle.jdbc.driver.OracleDriver
# Greenplum = org.postgres#ql.Driver
# DB2 = com.ibm.db2.jcc.DB2Driver
# Netezza = org.netezza.Driver
# PostgreSQL = org.postgresql.Driver
# SQLServer = com.microsoft.sqlserver.jdbc.SQLServerDriver
# MySQL = com.mysql.jdbc.Driver
# Teradata = com.teradata.jdbc.TeraDriver
# Vertica = com.vertica.jdbc.Driver
# Sybase = com.sybase.jdbc2.jdbc.SybDriver
# Informix = com.informix.jdbc.IfxDriver
# SAPDB = com.sap.dbtech.jdbc.DriverSapDB
# InterBase = interbase.interclient.Driver
# HSQLDB = org.hsqldb.jdbcDriver
# MariaDB = org.mariadb.jdbc.Driver
# MySQL = com.mysql.jdbc.Driver
driverClass=com.simba.googlebigquery.jdbc42.Driver

# BigQuery (like Hive) does not support "schema"."tablename".
# For BigQuery, this entry must be empty string: identifierQuotation=
# with no whitespace (except newline) or characters after the equals sign
identifierQuotation=
```

6. Locate and open for editing the file `additional_jdbc_drivers.rb`.

The path is similar to path similar to `/usr/chorus/install/releases/6.3.2.0.8068-7ac910ae3/components/core/app/mixins/sequel/extensions/additional_jdbc_drivers.rb`

7. Add a line for the bigquery class so that the content resembles the following.

```
module Sequel
  module AdditionalJdbcDrivers
    MAP = {
      mariadb: ->(db) { org.mariadb.jdbc.Driver },
      teradata: ->(db) { com.teradata.jdbc.TeraDriver },
      vertica: ->(db) { com.vertica.jdbc.Driver },
      hive2: ->(db) { org.apache.hive.jdbc.HiveDriver },
      hive: ->(db) { org.apache.hadoop.hive.jdbc.HiveDriver },
      bigquery: ->(db) { com.simba.googlebigquery.jdbc42.Driver }
    }

    MAP.each do |key, driver|
      ::Sequel::JDBC::DATABASE_SETUP[key] = driver
    end
  end
end
```



You must apply this change to the file `additional_jdbc_drivers.rb` again after upgrading Team Studio.

8. Open the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf` and add the following configuration parameter.

```
database.bigquery.batchSize (default = 1000)
```



Due to idiosyncrasies in the way BigQuery handles batch updates, uploading a narrow table can result in an error. This configuration parameter addresses possible batch size issues.

9. Restart Team Studio.
10. Open the Add Data Source dialog box.

11. Provide the **Data Source Type**, the **Data Source Name**, and (optionally), the **Description**.
12. Set the data connection (**JDBC URL**) using a URL similar to the following.



You can copy your BigQuery URL from your GCP BigQuery page

```
jdbc:bigquery://https://www.googleapis.com/bigquery/v2:443;ProjectId=teamstudio-user@teamstudio-alpine.iam.gserviceaccount.com
```

13. Specify **Workspace Visibility**.
A data source with **Limited** visibility must be manually associated with a workspace for members of that workspace to use the data source. By default, this option is set to **Public**.
14. Optional: Select **Set database credentials as a shared account** if you intend to allow all users to access the data source without using their own credentials.

Users access the database with your credentials as the data source owner. If you do not select this check box, each user must provide credentials for that data source to access it. You can check the box later if you change your mind.

BigQuery Data Source Connection Tests and Troubleshooting

If you encounter error messages or other problems with your data source connection, check this topic for suggestions for troubleshooting.

BigQuery data source: "Too many queries"

This error results when the batch size is too small. To correct this error, open the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf` and add the configuration parameter `database.bigquery.batchSize` (default = 1000).

BigQuery data source: "Cannot parse query"

This error results when the batch size is too big. To correct this error, open the file `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf` and add the configuration parameter `database.bigquery.batchSize` (default = 1000).

Google recommends that BigQuery data source users insert data using the Google API. Using DML issued through JDBC can cause problems because of statement size limitations; therefore, you should avoid using Team Studio operators to copy data into BigQuery. For more information on BigQuery limitations, see <https://cloud.google.com/bigquery/quotas>.

Hadoop Data Sources

These topics show you how to add a Hadoop data source from the command line or through the Team Studio user interface, and how to connect to various data sources.

Adding a Hadoop Data Source from the Command Line

To add an HDFS data source, first make sure the Team Studio server can connect to the hosts, then use the Add Data Source dialog box to add it to Team Studio. Supported Hadoop distributions are listed in Team Studio [System Requirements](#).

Prerequisites

Procedure

1. Ensure the Team Studio user has read/write permissions on the HDFS directories. Any HDFS directory that will be used within the application must be readable and writable. In addition, these directories must be readable and writable:
 - `/tmp`
 - `/tmp/tsds_out`
 - `/user`
 - `/user/chorus`
2. Make sure the Team Studio server can connect to the hosts with the fully qualified domain name (FQDN).

Option A: Modify the `/etc/hosts` file of the Team Studio server and cluster nodes to include host names and IP address of each server.

Option B: DNS lookup for all client and Hadoop nodes.

On the DNS server, add these lines:

```
alpinechorusserver IN A ipaddress
clusternode1 IN A ipaddress
clusternode2 IN A ipaddress
```

They should be added to these files:

```
/var/named/alpinenow.local.zone
/var/named/alpinenow.local.rr.zone
```

Now restart the named service and verify that you can connect by using telnet.

```
service named restart
telnet hostname port
```

Adding a Hadoop Data Source from the User Interface

To add an HDFS data source, first make sure the Team Studio server can connect to the hosts, and then use the Add Data Source dialog box to add it to Team Studio.

Supported Hadoop distributions are listed in Team Studio [System Requirements](#).

Prerequisites

You must have data administrator or higher privileges to add a data source. Ensure that you have the correct permissions before continuing.

Procedure

1. From the menu, select **Data**.
2. Select **Add Data Source**.
3. Choose **Hadoop Cluster** as the data source type.

ADD DATA SOURCE ✕

Data Source Type
Hadoop Cluster

Adding a Hadoop data source requires read permissions on the cluster. Resource Manager information and write access are required for workflows.

Data Source Name *

Description

Hadoop Version *

Select...

☐ Use High Availability

☐ Disable Kerberos impersonation

Name Node Host * Port *

Resource Manager Host Port

Workspace Visibility ⓘ

Public

Hadoop Credentials * Group List *

[Configure Connection Parameters](#)

[Test Configuration...](#)

[Load Configuration from File...](#)

Cancel Add Data Source

4. Specify the following data source attributes:

Data Source Name	Set a user-facing name for the data source. This should be something meaningful for your team (for example, "Dev_CDH5_cluster").
Description	Enter a description for your data source.
Hadoop Version	Select the Hadoop distribution that matches your data source.
Use High Availability	Check this box to enable High Availability for the Hadoop cluster.
Disable Kerberos Impersonation	<p>If this box is selected and you have Kerberos enabled on your data source, then the workflow uses the user account configured as the Hadoop Credentials here.</p> <p>If this box is cleared, the workflow uses the user account of the person running the workflow.</p> <p>If you do not have Kerberos enabled on your data source, you do not need to select this box. All workflows run using the account configured as the Hadoop Credentials.</p>
NameNode Host	<p>Enter a single active NameNode to start. Instructions for enabling High Availability are in Step 10.</p> <p>To verify the NameNode is active, check the web interface. (The default is <code>http://namenodehost.localhost:50070/</code>)</p>
NameNode Port	Enter the port that your NameNode uses. The default port is 8020.
Job Tracker/Resource Manager Host	For MapReduce v1, specify the job tracker. For YARN, specify the resource manager host.
Job Tracker/Resource Manager Port	Common ports are 8021, 9001, 8012, or 8032.
Workspace Visibility	<p>There are two options here:</p> <ul style="list-style-type: none"> • Public - Visible and available to all workspaces. • Limited - Visible and available only to workspaces they are associated with. <p>To learn more about associating a data source to a workspace, see Data Visibility.</p>
Hadoop Credentials	Specify the user or service to use to run MapReduce jobs. This user must be able to run MapReduce jobs from the command line.

Group List

Enter the group to which the Hadoop account belongs.

5. For further configuration, choose **Configure Connection Parameters**.
6. Specify key-value pairs for YARN on the Team Studio server. Selecting **Load Configuration** from Resource Manager attempts to populate configuration values automatically.

- `yarn.resourcemanager.scheduler.address`
- `yarn.app.mapreduce.am.staging-dir`



Be sure the directory specified above in the `staging-dir` variable is writable by the Team Studio user. Spark jobs produce errors if the user cannot write to this directory.

Required if different from default:

- `yarn.application.classpath`
 - The `yarn.application.classpath` does not need to be updated if the Hadoop cluster is installed in a default location.
 - If the Hadoop cluster is installed in a non-default location, and the `yarn.application.classpath` has a value different from the default, the YARN job might fail with a "cannot find the class AppMaster" error. In this case, check the `yarn-site.xml` file in the cluster configuration folder. Configure these key:value pairs in the UI using the **Configure Connection Parameters** option.
- `yarn.app.mapreduce.job.client.port-range`
 - This describes a range of ports to which the application can bind. This is useful if operating under a restrictive firewall that needs to allow specific ports.

Recommended:

- `mapreduce.jobhistory.address = FQDN:10020`



Operators that use Pig for processing do not show the correct row count in output if `mapreduce.jobhistory.address` is not configured correctly. For more information, see Pig operators do not show row count output correctly. .

- `yarn.resourcemanager.hostname = FQDN`
- `yarn.resourcemanager.address = FQDN`
- `yarn.resourcemanager.scheduler.address = FQDN:8030`
- `yarn.resourcemanager.resource-tracker.address = FQDN:8031`
- `yarn.resourcemanager.admin.address = FQDN:8033`
- `yarn.resourcemanager.webapp.address = FQDN:8088`

- `mapreduce.jobhistory.webapp.address = FQDN:19888`

7. Save the configuration.

Additional configuration might be needed for different Hadoop distributions.

To connect to a Cloudera (CDH) cluster, follow the instructions above.

To connect to MIT KDC Kerberized clusters:

- [Configure Kerberos in the Team Studio client](#)
- [Configure a Kerberos-Enabled Hadoop Data Source](#)

To connect to a MapR cluster:



- [Connect to a MapR4.x Data Source](#)

To connect to a PHD cluster:

- [Connect to a Pivotal Hadoop \(PHD\) Data Source](#)

To connect to a YARN Enabled cluster:

- [Connect to a YARN Enabled Data Source](#)

If you do not have all of these parameters yet, you can save your data source as "Incomplete" while working on it. For more information, see [Data Source States](#).

8. To perform a series of automated tests on the data source, click **Test Connection**.

TEST DATA SOURCE CONFIGURATION SETTINGS	
Run All Tests	
Ping Name Node	Test network reachability of the name node using ping.
Ping Resource Manager	Test network reachability of the Resource Manager using ping.
DNS Resolve Name Node	Test DNS resolution of the name node host's IP address.
DNS Resolve Resource Manager	Test DNS resolution of the resource manager host's IP address.
Connect to Name Node	Check if the name node and port is accessible.
Connect to Resource Manager	Check if the Resource Manager host and port is accessible.
HDFS Accessibility	Test that the HDFS root directory contents can be listed.
Done	

9. Click **Save Configuration** to confirm the changes.

10. When the connectivity to the active NameNode is established above, set up NameNode High Availability (HA) if enabled.

Required:

- `dfs.ha.namenodes.nameservice1`
- `dfs.namenode.rpc-address.nameservice1.namenode<id>` (required for each namenode id)
- `dfs.nameservices`
- `dfs.client.failover.proxy.provider.nameservice1`

Recommended:

- `ha.zookeeper.quorum`



Support for Resource Manager HA is available.

To configure this, add `failover_resource_manager_hosts` to the advanced connection parameters and list the available Resource Managers.

If one of the active Resource Managers fails during a job running, you must re-run the job, but you no longer must reconfigure the data source that failed. If one of the active Resource Managers fails while a job is not running, you do not need to do anything. Team Studio uses another available Resource Manager instead.

Connecting to a Hive Data Source on Hadoop

You can create a Hive data source natively on Hadoop, without using JDBC. It is much faster than connecting to Hive over JDBC, and it has support for running HiveQL queries on the HQL Execute operator.

Adding a Hive data source is very similar to adding a Hadoop data source. No extra JARs are needed to get Hive functionality.

Configuring Kerberos on Hive is a similar process to configuring Kerberos on Hadoop, with the following additions.

- You must add `hive.metastore.kerberos.principal`.
- When you create the data source, you must add the following two parameters.
 - `stack.version=2.3.4.0-3485` - This parameter is essential for running jobs on the server. (If the value 2.3.4.0-3485 does not work, try 2.3.6.0-3796).
 - `hive.additional.parameter.disabled=true` - This parameter is necessary for the `stack.version` parameter to be accepted. If this parameter is not set, the connection returns the error "stack.version cannot be changed at runtime".

Prerequisites

You must have write access on the server where Team Studio is installed.

Procedure

1. From the menu, select **Data**.
2. Select **Add Data Source**.
3. In the Add Data Source dialog box, choose **Hadoop Hive** from the **Data Source Type** drop-down list.

The only thing different from the section about Hadoop in [Hadoop Data Sources](#) is that the user needs to specify the Hive Metastore location. This location can be found in the file `hive-site.xml` on the cluster configuration and usually starts with `thrift://`.

Connect to a MapR 4.x Data Source

This topic describes how to configure Team Studio to connect to a MapR 4.x data source.

Prerequisites

Procedure

1. Edit the `$CHORUS_HOME/shared/chorus.properties` file and add the below listed option into the list of values for the `java_options` parameter for dynamically loading the MapR FS libraries:

```
java_options = -Djava.library.path=$CHORUS_HOME/vendor/hadoop/lib -
Djava.security.egd=file:/dev/./urandom -server -Xmx4096m -Xms2048m -Xmn1365m -
XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:ParallelGCThreads=3
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./ -XX:+CMSClassUnloadingEnabled
```

2. Make sure that Team Studio can resolve the DNS names of all cluster nodes. Configuring the `/etc/hosts` file might be necessary:

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
172.27.0.2  chorus.alpinenow.local  chorus
172.27.0.4  mapr4a.alpinenow.local  mapr4a
172.27.0.5  mapr4b.alpinenow.local  mapr4b
172.27.0.6  mapr4c.alpinenow.local  mapr4c
```

3. Either add `mapr` user/group with the uid 505 and gid 505 on the Team Studio computer, or Team Studio user/group with uid 507 and gid 507 on all MapR nodes. All mapR clients should use the same uid and gid to ignore permission issues.

```
groupadd mapr --gid 505
useradd mapr --gid 505 --uid 505

groupadd chorus --gid 507
useradd chorus --gid 507 --uid 507
```

4. The MapR 4.0.1 client must be installed and configured on the Team Studio computer so Team Studio can communicate with the MapR cluster (version 4.0.1 or 4.1.0). For more information about MapR 4.0.1 client installation, see [this page](#). After you install the MapR 4.0.1 client, copy the native libraries into the directory that we configured in the `chorus.properties` file:

```
cp /opt/mapr/hadoop/hadoop-2.4.1/lib/native/* $CHORUS_HOME/vendor/hadoop/lib/
```

Then edit the `yarn-site.xml` and `mapred-site.xml` files (from `/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop` directory) and be sure to use the correct host names (in the below examples, `mapr4x.alpinenow.local` host names are used with 2 failover resource managers).

`mapred-site.xml`:

```
<configuration>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>mapr4c.alpinenow.local:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>mapr4c.alpinenow.local:19888</value>
  </property>
</configuration>
```

`yarn-site.xml`:

```
<configuration>
  <!-- Resource Manager HA Configs -->
  <property>
    <name>yarn.resourcemanager.ha.enabled</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.resourcemanager.ha.automatic-failover.enabled</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.resourcemanager.ha.automatic-failover.embedded</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.resourcemanager.recovery.enabled</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.resourcemanager.cluster-id</name>
    <value>yarn-mapr41.alpinenow.local</value>
  </property>
  <property>
    <name>yarn.resourcemanager.ha.rm-ids</name>
    <value>rm1,rm2</value>
  </property>
  <property>
    <name>yarn.resourcemanager.ha.id</name>
    <value>rm1</value>
  </property>
  <property>
    <name>yarn.resourcemanager.zk-address</name>
    <value>mapr4a.alpinenow.local:5181,mapr4b.alpinenow.local:5181,mapr4c.alpinenow.local:5181</value>
  </property>

  <!-- Configuration for rm1 -->
  <property>
    <name>yarn.resourcemanager.scheduler.address.rm1</name>
    <value>mapr4a.alpinenow.local:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address.rm1</name>
    <value>mapr4a.alpinenow.local:8031</value>
  </property>
```

```

<property>
  <name>yarn.resourcemanager.address.rm1</name>
  <value>mapr4a.alpinenow.local:8032</value>
</property>
<property>
  <name>yarn.resourcemanager.admin.address.rm1</name>
  <value>mapr4a.alpinenow.local:8033</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address.rm1</name>
  <value>mapr4a.alpinenow.local:8088</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.https.address.rm1</name>
  <value>mapr4a.alpinenow.local:8090</value>
</property>
<!-- Configuration for rm2 -->
<property>
  <name>yarn.resourcemanager.scheduler.address.rm2</name>
  <value>mapr4b.alpinenow.local:8030</value>
</property>
<property>
  <name>yarn.resourcemanager.resource-tracker.address.rm2</name>
  <value>mapr4b.alpinenow.local:8031</value>
</property>
<property>
  <name>yarn.resourcemanager.address.rm2</name>
  <value>mapr4b.alpinenow.local:8032</value>
</property>
<property>
  <name>yarn.resourcemanager.admin.address.rm2</name>
  <value>mapr4b.alpinenow.local:8033</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address.rm2</name>
  <value>mapr4b.alpinenow.local:8088</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.https.address.rm2</name>
  <value>mapr4b.alpinenow.local:8090</value>
</property>
<!-- :::CAUTION::: DO NOT EDIT ANYTHING ON OR ABOVE THIS LINE -->
</configuration>

```



If you are not sure which values to use for the above listed parameters, navigate to your MapR cluster console from your web browser: https://your_mapr_cluster_host:8443.

5. Edit the `$CHORUS_HOME/shared/ALPINE_DATA_REPOSITORY/configuration/alpine.conf` file and make sure that the `mapr4` agent is enabled:

```

alpine {
  chorus {
    # scheme = HTTP
    # host = myhostname //change to other hostname
    # port = 9090 //change to other port
    # debug = true //change to true for debugging
  }
  hadoop.version.cdh4.agents.2.enabled=false
  hadoop.version.cdh5.agents.4.enabled=false
  hadoop.version.cdh53.agents.7.enabled=false
  hadoop.version.phd2.agents.1.enabled=false
  hadoop.version.phd3.agents.8.enabled=false
  hadoop.version.mapr4.agents.6.enabled=true
  hadoop.version.mapr3.agents.3.enabled=false
  hadoop.version.hdp2.agents.5.enabled=false
  hadoop.version.hdp22.agents.8.enabled=false #( same agent as phd3)
  hadoop.version.iop.agents.8.enabled=false #( same agent as phd3)
  hadoop.version.cdh54.agents.9.enabled=false
}

```


6. After the preceding steps are done, restart Team Studio and navigate to the Data Source configuration page from your web browser. Create a new Hadoop connection and select MapR4 from the **Hadoop Version** drop-down list. Then configure the connection with the correct parameters:

Also, configure the additional parameters as follows by clicking the **Additional Parameters** link:

mapreduce.jobhistory.address	mapr4c.alpinenow.local:10020
mapreduce.jobhistory.webapp.address	mapr4c.alpinenow.local:19888
yarn.app.mapreduce.am.staging-dir	/var/mapr/cluster/yarn/rm/staging
yarn.resourcemanager.admin.address	mapr4b.alpinenow.local:8033
yarn.resourcemanager.resource-tracker.address	mapr4b.alpinenow.local:8031
yarn.resourcemanager.scheduler.address	mapr4b.alpinenow.local:8030
mapreduce.job.map.output.collector.class	org.apache.hadoop.mapred.MapRFsOutputBuffer
mapreduce.job.reduce.shuffle.consumer.plugin.class	org.apache.hadoop.mapreduce.task.reduce.DirectShuffle

7. If the MapR4 cluster has HA enabled for Resource Manager, also add the following parameter in addition to the above parameters. The value should be a comma-separated list of available resource manager host names.

failover_resource_manager_hosts	mapr4b.alpinenow.local,mapr4a.alpinenow.local
---------------------------------	---

8. If zero configuration failover is enabled in the MapR4 cluster, add the following parameters in addition to the above parameters:

yarn.resourcemanager.ha.custom-ha-enabled	true
---	------

yarn.client.failover-proxy-provider	org.apache.hadoop.yarn.client.MapRZKBasedRMFailoverProxyProvider
yarn.resourcemanager.recovery.enabled	true

CONFIGURE CONNECTION PARAMETERS		
mapreduce.jobhistory.address	mapr4c.alpinenow.local:10020	Delete
mapreduce.jobhistory.webapp.address	mapr4c.alpinenow.local:19888	Delete
yarn.app.mapreduce.am.staging-dir	/var/mapr/cluster/yarn/rm/staging	Delete
yarn.resourcemanager.admin.address	mapr4b.alpinenow.local:8033	Delete
yarn.resourcemanager.resource-tracker.address	mapr4b.alpinenow.local:8031	Delete
yarn.resourcemanager.scheduler.address	mapr4b.alpinenow.local:8030	Delete
failover_resource_manager_hosts	mapr4b.alpinenow.local, mapr4a.alpinenow.local	Delete
yarn.resourcemanager.ha.custom-ha-enabled	true	Delete
yarn.client.failover-proxy-provider	org.apache.hadoop.yarn.client.MapRZKBasedRMFailoverProxyProvider	Delete
yarn.resourcemanager.recovery.enabled	true	Delete

[Load Configuration from Resource Manager](#)

Cancel Save

Connect to a Pivotal Hadoop (PHD) Data Source

You might be required to add an additional parameter to configure a Team Studio data source to connect to PHD 3.0.

Perform this task from the Pivotal Hadoop UI, and also in the Team Studio data source connection UI.

Prerequisites

- You must have access to the Pivotal Hadoop command line and the configuration UI.
- You must be able to add a the Team Studio data source connection configuration user interface.

Procedure

- Open the Pivotal Hadoop configuration UI.
- Locate the file `mapred-default.xml`.
- In the file `mapred-default.xml`, locate the following class path parameters.

`mapreduce.application.classpath`
`mapreduce.application.framework.path`
- Check if these parameters contain an environment variable named either `stack.name` or `stack.version`.
 - If the parameter does not exist, then it is not needed by Team Studio to configure the data source connection. You can continue to Step 8.
 - If the environment variable exists, then you must provide the version information when you configure the data source in Team Studio. Continue to step 5.
- Open a command-line prompt on the Pivotal Hadoop cluster.
- Run the following command.

`hadoop version`

The output should be `/usr/phd/<yourversion#>` where *yourversion#* is the version of Hadoop you are running (for example, `2.4.0.2.1.2.0-403`).

7. Make a note of the version number (both major and minor).
8. Open Team Studio Web UI.
9. From the menu, click **Data**.
The Data Sources window is displayed.
10. Click **Add Data Source**.
The Add Data Source dialog box is displayed.
11. From the **Data Source Type** drop-down menu, select **Hadoop Cluster**.
12. Provide all required information.
13. If you found `stack.version` or `stack.name` from Step 4, then click **Configure Connection Parameters**.

If you found neither `stack.version` nor `stack.name`, then skip the next steps. Test the configuration to confirm that it is correct, and then click **Add Data Source**.
14. In the resulting Configure Connection Parameters dialog box, provide the key (either `stack.version` or `stack.name` and the value (the version number from step 6).

CONFIGURE CONNECTION PARAMETERS	
stack.version	2.4.0.2.1.2.0-402



- Set the parameter `stack.version=<your cluster version>` to run jobs on the server. (For example, `stack.version=2.3.4.0-3485`)
- Set `hive.additional.parameter.disabled=true` to ensure that the `stack.version` parameter is accepted.


15. Save and test the configuration, and then click **Add Data Source**.

Connect to a YARN-Enabled Data Source

To connect to a YARN-enabled cluster, Team Studio must have access to the following ports on each node of the cluster:

Prerequisites

Port	Configuration Parameter
8020	<code>fs.default.name/dfs.namenode.rpc-address.<nameservice>.namenode<x></code>
8030	<code>yarn.resourcemanager.scheduler.address</code>
8031	<code>yarn.resourcemanager.resource-tracker.address</code>
8032	<code>yarn.resourcemanager.address</code>

Port	Configuration Parameter
8088	<code>yarn.resourcemanager.webapp.address</code>
10020	<code>mapreduce.jobhistory.address</code>
50010	<code>dfs.datanode.address</code>
50020	<code>dfs.datanode.ipc.address</code>
user-specified (or blank)	<div>  <p>This parameter is unspecified on many clusters, meaning that Team Studio chooses an arbitrary open port when it runs MapReduce jobs. Instead, you can set this parameter to a specific port range either in the Team Studio data source configuration or on the cluster.</p> </div>

For more options, see [Team Studio Default Ports](#).

Hadoop Data Source Connection Tests and Troubleshooting

You can test Hadoop connections to troubleshoot the connection to the datasource. Team Studio provides a variety of tests to verify connectivity and troubleshoot problems. Perform this task on the computer where Team Studio is installed.

Prerequisites

You must have access to the Hadoop data source.

Procedure

1. From the menu, click **Data**.
The data sources are displayed.
2. On the right side of the display, click **Edit Data Source**.
The Edit Data Source Configuration dialog box is displayed.
3. Click **Test Configuration**.
the Test Data Source Configuration Settings dialog box is displayed.
4. Run the tests.
You can run all tests at the same time.
The progress is displayed from this configuration.

What to do next

Click **Details** for any of the tests to review specific test details and error messages.

Workflow Editor Preferences

Use the Preferences dialog box to modify a variety of Workflow Editor preferences.

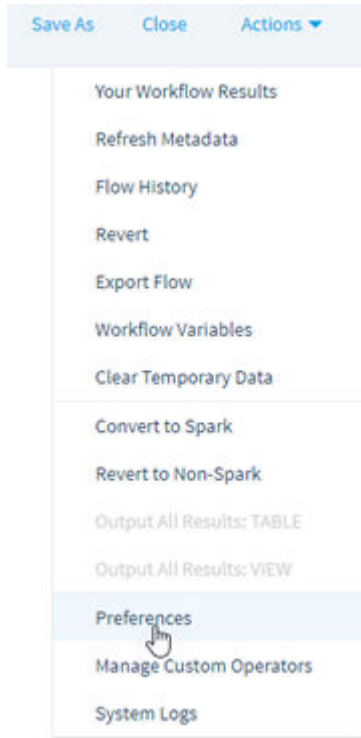


You must be an administrator to edit preferences.

Prerequisites

Procedure

- To open the Preferences dialog box, click the **Actions** menu, and then select **Preferences**.



Using the **Preferences** dialog box:

For each tab, save changes by clicking **Save**. If you do not click **Save**, you are prompted to save or revert the changes when you change preference sections.

You can configure the Editor preferences at the system level.

- Edit the `alpine.config` file located in `ALPINE_DATA_REPOSITORY/configuration`.
- Defaults are available in `alpine.config.defaults`.
- Values in `.defaults` are overwritten at system startup, so all customizations should be made in `alpine.config`.

Algorithm Preferences

Use algorithm preferences to control computation and certain behaviors when running a workflow.



You must be an administrator to edit preferences.

Prerequisites

Procedure

- To open the Preferences dialog box, click the **Actions** menu and then select **Preferences**.
- Select **Algorithm**.

The screenshot shows the 'Edit Preferences' window. On the left, a tree view under 'Preference' includes 'Algorithm', 'System', 'Data Sources', 'UI', 'Work Flow', 'Datetime Formats', and 'R Server'. The main area displays three settings: 'Distinct Value Count' with a text input '100000', 'Summary Statistics Distinct Value Count' with a text input '1000', and 'Decimal Precision Digits' with a spinner set to '7'. Below these are 'Save' and 'Restore' buttons. A blue 'Done' button is at the bottom right.

Distinct Value Count

A general setting for the way the application stores and analyzes the set of possible values that each variable can take on.

In many cases, Team Studio must be aware of all possible distinct values of a given variable or column. For example, when building a classification model, Team Studio might have to store the set of classes represented by the distinct values of the dependent variable. Storing these values often requires memory, and so we limit the maximum size of memory used to store and analyze these distinct values.



This setting is widely used throughout the application, so be careful of changing this value, but consider using it when you encounter memory issues.

Summary Statistic Distinct Value Count

A special case of the Distinct Value Count preference.

The Summary Statistics operator must store the distinct values of each column so that it can compute the number of distinct values and the most common values for each column. This can consume a lot of memory, so use this setting to control the amount of memory used by the Summary Statistics operator.

You can also limit the memory that the Summary Statistics operator uses by setting **Calculate Number of Distinct Values** to false in the operator properties.

Decimal Precision Digits

Controls the number of decimal places used to display results throughout the application.

Click **Save** to save changes. Click **Restore** to return to default values.

System Preferences

Use system preferences to change log levels and grant access to logs.

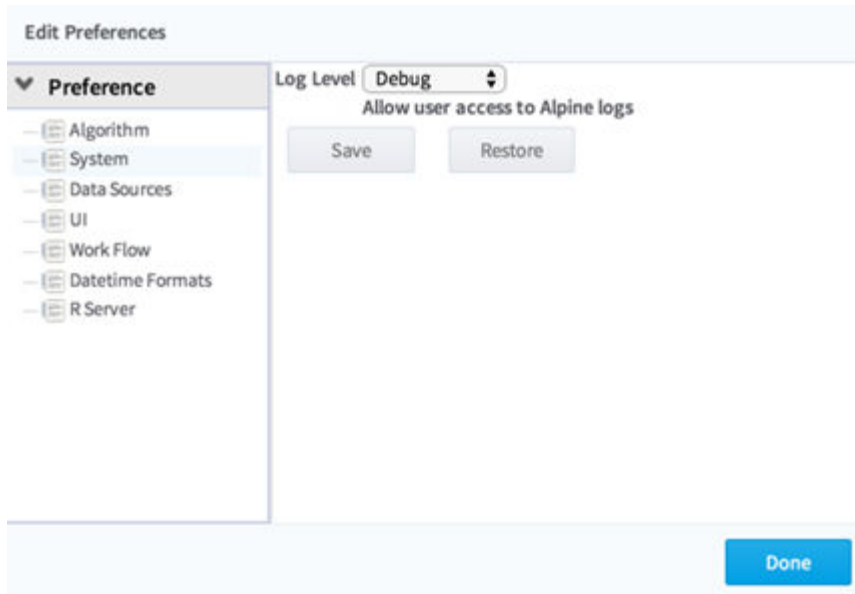


You must be an administrator to edit preferences.

Prerequisites

Procedure

1. To open the Preferences dialog box, click the **Actions** menu and then select **Preferences**.
2. Select **System**.



Log Level

Sets the log level of the `alpine.log` file to **Info** or **Debug** (for more detail) mode.

Allow user access to Team Studio logs

Grants non-admin users access to the system logs (from the UI).

Click **Save** to save changes. Click **Restore** to return to default values.

Data Source Preferences

Use data source preferences to set the amount of time allowed for testing database connections before system timeout. Data source preferences apply to all data source connections used in a workflow.

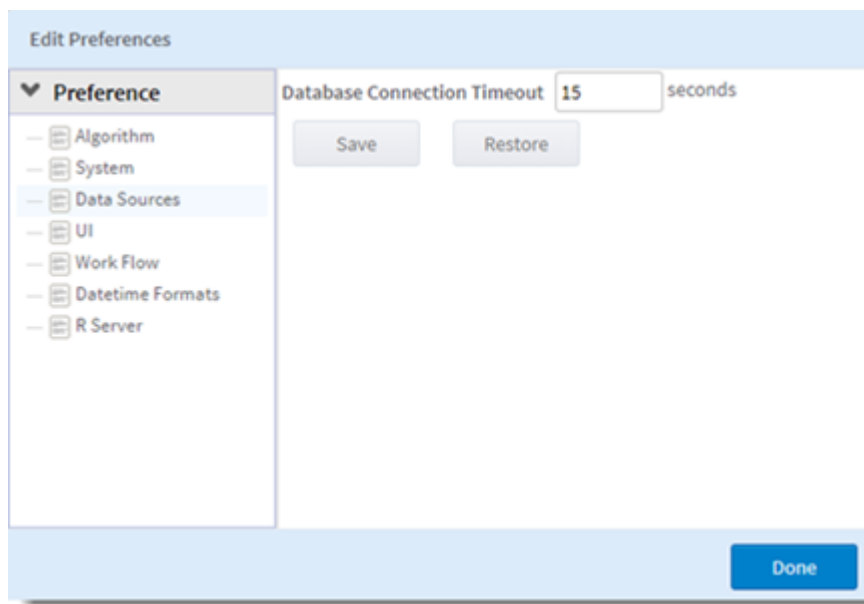


You must be an administrator to edit preferences.

Prerequisites

Procedure

1. To open the Preferences dialog box, click the **Actions** menu and then select **Preferences**.
2. Select **Data Sources**.



Database Connection Timeout (seconds)

The amount of time allowed for testing database connections before the system reaches timeout (in seconds).

Click **Save** to save changes. Click **Restore** to return to default values.

UI Preferences

Use UI preferences to control various display options.



You must be an administrator to edit preferences.

Prerequisites

Procedure

1. To open the Preferences dialog box, click the **Actions** menu, and then select **Preferences**.
2. Select **UI**.

Data Preview Max Rows

Number of rows to display by the **Data Preview** right-click option.

Max Points of Scatter Plot

Number of points to display by scatter plots.

Max Points of Cluster

Number of points to display in Cluster visualizations.

Standard notation threshold

Threshold number of digits a value can have before being displayed using Scientific Notation format.

Decimal Precision

Maximum number of decimals to display for values.

Click **Save** to save changes. Click **Restore** to return to default values.

Work Flow Preferences

Use Work Flow preferences to modify the global temp directory.

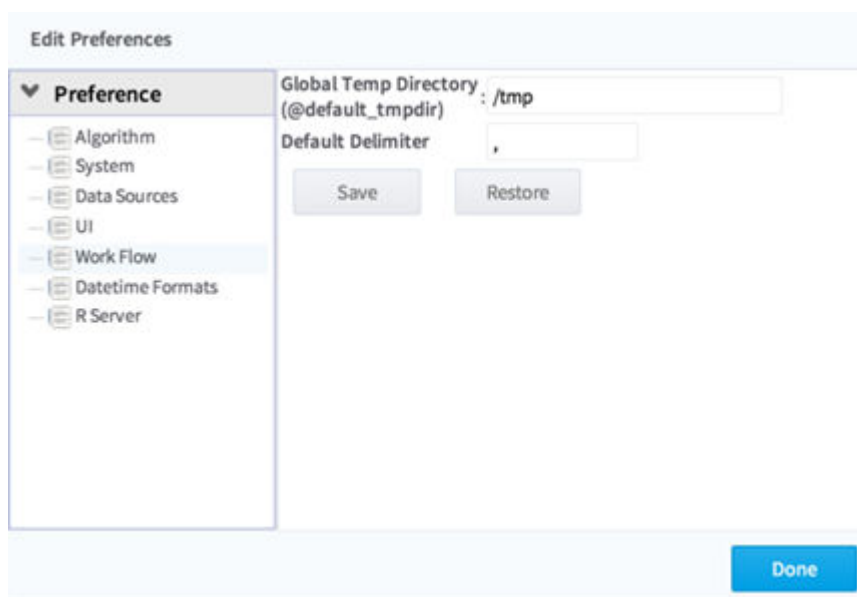


You must be an administrator to edit preferences.

Prerequisites

Procedure

1. To open the Preferences dialog box, click the **Actions** menu, and then select **Preferences**.
2. Select **Work Flow**.



Global Temp Directory

- Sets the `@default_tmpdirworkflow` variable for all flows created in Team Studio.
- Sets the `/tmp` directory used by the MapReduce code in Team Studio when running Hadoop operators.
- Ensures that UNIX users running the Team Studio application have read/write permissions for this `tmp` directory, as well as the staging directory specified in the `mapred-site.xml` or `conf.xml` files from the cluster.

Default Delimiter

- Sets the default delimiter of HDFS files created by Team Studio for storing intermediate results.



The Work Flow preferences setting sets the delimiter at a global level across all workflows. However, you can override this global setting at the workflow variable level for a specific workflow. For details, see "Workflow Variables" in *TIBCO® Data Science Team Studio User's Guide*.

Click **Save** to save changes. Click **Restore** to return to default values.

Datetime Formats Preferences

Use the Datetime Formats Preferences to modify the appearance of dates and times in the Workflow Editor.



You must be an administrator to edit preferences.

The Datetime formats are used by the Workflow Editor in the following cases:

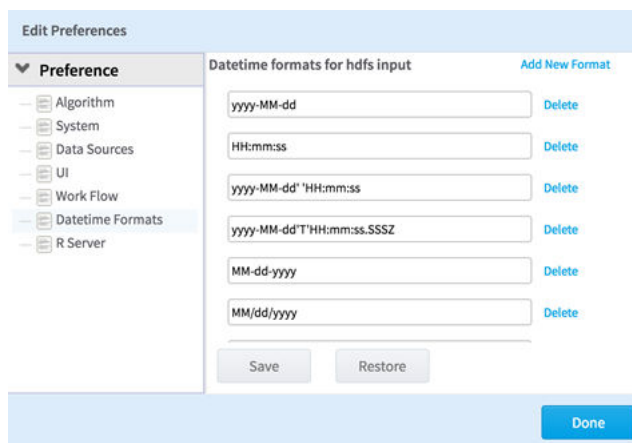
- Using the Hadoop File Structure configuration to import `datetime` data from Hadoop File System text files. See the "Hadoop File" operator help in *TIBCO® Data Science Team Studio User's Guide* for details.
- Using the Pig Execute operator to execute Hadoop Pig script against `datetime` data fields and leverage Pig Datetime functions such as `GetMonth(datetime)` or `GetDay(datetime)`. See the "Pig Execute" operator help in *TIBCO® Data Science Team Studio User's Guide* for details.
- Using the Variable operator to convert source `datetime` formats into new `datetime` formats or to convert `datetime` data into new data fields with a Pig `Date`Time function. See the "Variable" operator help in *TIBCO® Data Science Team Studio User's Guide* for details.

- Using the Set operator to combine two or more data sources that contain datetime data types. See the "Set Operations" in *TIBCO® Data Science Team Studio User's Guide* for details.
- Using the Null Value Replacement operator to replace null values with a default datetime format. See the "Null Value Replacement" operator help in *TIBCO® Data Science Team Studio User's Guide* for details.
- Using the Row Filter operator to filter data by datetime formats or by a value derived from an applied the Pig Datetime function. See the "Row Filter" operator help in *TIBCO® Data Science Team Studio User's Guide* for details.

Prerequisites

Procedure

1. To open the Preferences dialog box, click the **Actions** menu, and then select **Preferences**.
2. Select **Datetime Formats**.



Adding a New Datetime Format

To add a new input datetime format, click **Add New Format**.

Click **Save** to save changes. Click **Restore** to return to default values.

Custom Datetime Formats

Datetime data type formats must follow Joda-Time API formatting. For more information about Joda-Time formatting, see [Joda Time Formatting](#).

Some commonly used Joda-Time pattern letters include:

- Uppercase Y refers to the Year of the era (>0); lowercase y refers to the year.
- Uppercase M refers to the Month of the year; lowercase m refers to the minute of the hour.
- Uppercase D refers to the Day of the year (1-365); lowercase d refers to the day of the month (1-31).
- Uppercase E refers to the Day of the week in text (Tuesday); lowercase e refers to the numeric day of the week (1-7).
- Uppercase H refers to the Hour of the day (1-24); lowercase h refers to the clock hour of the half day (1-12).
- Uppercase S refers to the Fraction of a Second; lowercase s refers to the second of the minute (1-60).

The count of pattern letters determines the overall datetime format. For example, YYYY specifies a 4-digit year format, such as 2019.

Administrator Options in Team Studio

The Administrator Console provides a set of tools for administrators to view information about their users, licensing, and running workflows in one place. Administrators can also download logs from the console.



You must be an Administrator to see this option.

For information about downloading logs, see [Download Logs](#).

To navigate to the Administrator Console, select **Administration** from the sidebar menu.

The Administrator Console

Administration

Application Settings

[License Information](#)
[Email Configuration](#)
[Authentication](#)
[Deployment Targets](#)

Application Status

[Application Process Status](#)
[Log Files](#)
[Workflow Management](#)
[Event Log](#)
[Usage Statistics](#)

License Information

From this page, an administrator can get an idea of license features and usage at a glance.

License Information

Application Version

6.0

License Expiration

Dec 31, 2021

Licensed Features

Feature	Enabled?
Model Operations	✓
Custom Operators	✓
Jobs & Scheduling	✓
Milestones	✓
Touchpoints	✓
Application API	✓

Licensed MAC Address

AB:CD:EF:12:34:56

MAC Address Details

Licensed Address: **AB:CD:EF:12:34:56**

All Available MAC Addresses:

- AB:CD:EF:12:34:56
- 12:34:56:78:90:12
- 11:23:58:13:21:34

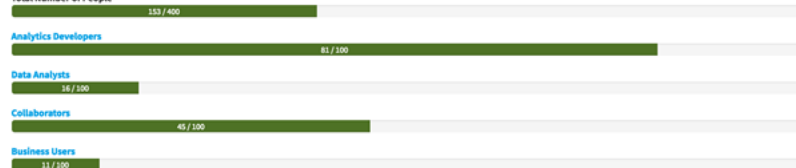
Any one of the available MAC addresses can be used for the application license.

Administration Roles

Administrators: 46

License Roles

Total Number of People



Features:

- **Application Version:** The running version of Team Studio.
- **License Expiration:** When the Team Studio license expires.
- **Licensed Features:** If the **Enabled?** column is checked, those features are available in the current Team Studio installation. If the **Enabled?** column is not checked, those features are not part of your current license plan.
 - **Model Operations:** The ability to export and import custom models using PMML Alpine Model (AM), and PFA format.
 - **Custom Operators:** The ability to use and create Custom Operators. For more information, see "Custom Operators" in *TIBCO® Data Science Team Studio Development Kit*.
 - **Jobs & Scheduling:** The ability to schedule jobs such as loading or analyzing data.
 - **Milestones:** The ability to set up milestones and track progress in Team Studio Connect.
 - **Touchpoints:** The ability to create and use Touchpoints. For more information, see "Touchpoints" in *TIBCO® Data Science Team Studio User's Guide*.
 - **Application API:** The ability to build access and use the Team Studio API. For more information, see "Team Studio API" in the *TIBCO® Data Science Team Studio Development Kit*.
- **Licensed MAC Address:** The MAC address that this version of Team Studio is registered to.
- **Administration Roles:** How many Administrators are registered in this Team Studio instance.
- **License Roles:** The allotted amount of space for each user type, and how many users are registered within each role. By clicking on any of the license roles, an administrator can manage users for each type. For example:

Business Users

10 people			Filter...	Search
	business user QA	bususer	bususer@alpinenow.com	
	James Bond Senior hitman	Bond	007@alpinenow.com	

Email Configuration

You can configure information to use for emailing people about their Team Studio notifications. All new users to Team Studio receive a welcome email, and you can configure who receives notifications for finished workflows or jobs. Use this dialog box to set up your SMTP server.

See [Email Configuration](#) for more information.

Authentication

From this page, an administrator can configure SAML or select an LDAP or Team Studio authentication methods.

See [Enable LDAP Authentication](#) and [Enable Single Sign-On SAML](#) for details.

Deployment Targets

You can configure the deployment targets engines have access to using this administration page. Deployment targets are servers that host real-time PFA engines. Team Studio provides the ScORE server application for deployment - contact your Team Studio representative for details.

For more information, see [Deployment Targets](#).

Application Process Status

You can see the Team Studio currently running processes from this section. Along with the IDs of these processes, you can also see their uptime and their status, and you can start and stop them from the application. This enables Team Studio administrators to troubleshoot processes without having to use the command line.

See [Process Control](#) for more information.

Log Files

Administrators can access log files for the application to help with troubleshooting and maintenance. In addition, these files are very helpful to provide when filing a support request. More information on the log files can be found at [Download Logs](#).

Workflow Management

You can monitor running workflows from the Administrator Console. This allows administrators to see all the running workflows at a glance, and be able to stop them if necessary. The table includes the workflow name, the workspace, the user running the flow, and when it started, allowing administrators to diagnose long-running flows easily.

Selecting **Refresh** polls for any new running workflows. Selecting **View Logs** shows the logs from the workflow so users can track progress. Selecting **Stop** stops the workflow remotely.

Workflow Management

Workflow Name (ID)	Workspace Name (ID)	User Name (ID)	Process ID	Start Time	
DocTests (7960)	Allison's Workspace (772)	allison (258)	0f4c7152-1bda-e474-d586-2804c476541f	2016-03-03 11:13:21 -08:00	<div>Refresh</div> <div>Stop</div>
View logs...					

Event Log

View and download a full event log for audit purposes. Information is tracked over time for action type (for example, upgrading a work file version), with details regarding the user who performed the action, the IP address, and the names of any work files, data sources, or jobs that were associated to the action.

Event Log

Download Full Event Log							
Event ID	Action	Actor ID	Actor User Name	Actor Sign In IP	Target	Target ID	Target Name
116976	NoteOnWorkfile	5	rkaur	::ffff:10.0.2.17	Workfile	17442	svm_pmml.pmml
116975	WorkfileUpgradedVersion	5	rkaur	::ffff:10.0.2.17	Workfile	17442	svm_pmml.pmml
116974	NoteOnWorkfile	5	rkaur	::ffff:10.0.2.17	Workfile	17443	lir_export.pmml

Usage Statistics

Administrators can view usage statistics for users in Team Studio. The login data includes user name, email, and creation date, as well as total logins over, activity in last 30 days, and latest login date.

More information is available at [Usage Statistics](#)

Email Configuration

You can configure email notifications from Team Studio using the Email Configuration dialog box, which is available only to administrators.

Prerequisites

Procedure

1. To navigate to the email configuration settings, from the sidebar menu, click **Administration**, and then click **Email Configuration**.

Email Configuration

Email Enabled ✓

Email Message Settings

From Address alpinenotif <alpinenotif@alpinenow.com>

Reply-to Address dont_need_no_stinkin_emails@alpinenow.com

SMTP server configuration

Address smtp.gmail.com

Port 587

HELO domain gmail.com

Authentication login

Username alpinenotif@alpinenow.com

Password *****

STARTTLS Auto ✓

OpenSSL Verify Mode none

2. To edit settings, click **Edit Configuration**.

From Address	Address from which emails are sent.
Reply-to Address	Address that is sent to if a user responds to an email notification.
Address	Address of the SMTP server.
Port	Port of the SMTP server.
HELO Domain	<p>HELO is an SMTP command sent by an email client when connecting to an email server. The command tells the server that the client wants to initiate an email transaction. It is followed by the client's domain name.</p> <p>For example, if you are using a Gmail SMTP server, the HELO Domain is gmail.com.</p>

Authentication	<p>Authentication method for sending passwords in the emails sent to new users. This setting can be one of the following.</p> <ul style="list-style-type: none"> • Plain: Send the password as nonsecure plain text. • Login: Send the password Base64-encoded. • Cram MD5: Combines a Challenge/Response mechanism to exchange information and a cryptographic Message Digest 5 algorithm to hash important information.
Username	User name for the email service.
Password	Password for the email service.
STARTTLS Auto	When enabled, automatically detects if STARTTLS is enabled in your SMTP server and, if so, uses it.
OpenSSL Verify Mode	When using TLS, you can set how OpenSSL checks the certificate. This setting is useful if you need to validate a self-signed certificate or a wildcard certificate, or both.



If **Username** and **Password** are not set, SMTP authentication is not enabled. An SMTP server that does not require authentication is vulnerable to malicious activity. Proper network security, including host whitelisting and firewall and routing rules, should be ensured.

3. Click **Update** to save the changes.
4. To test the new settings, click **Send Test Email**.

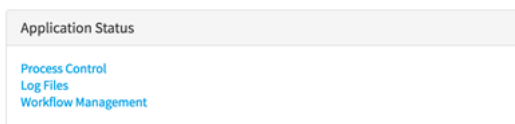
Process Control

System administrators no longer have to monitor Team Studio services manually from the command line. Instead, we have leveraged the supervisord library to make it easier for you to troubleshoot, start, and stop services from within the application. For more information, see supervisord.org.

Prerequisites

Procedure

1. From the sidebar menu, click **Administration**.
2. To see the running processes, from the **Application Status** list, select **Process Control**.



This screen contains information about five Team Studio processes: alpine, indexer, scheduler, solr, and workers.

State	Description	Name	Action
running	pid 23112, uptime 5 days, 23:36:31	alpine	Restart Stop
running	pid 2278, uptime 5 days, 16:03:36	indexer	Restart Stop
running	pid 22965, uptime 5 days, 23:36:59	scheduler	Restart Stop
running	pid 13194, uptime 1:43:51	solr	Restart Stop
running	pid 22944, uptime 5 days, 23:37:01	workers	Restart Stop

Supervisor 3.2.3

- **State:** The state of the application (usually 'running').
- **Description:** The pid of the process and the current uptime.
- **Name:** The name of the process.
- **Action:** Allows you to restart or stop the process.



These actions affect your entire system. Use them wisely.

If one of your processes fails or stops, this process control system attempts to bring it back up without any intervention from you. This design provides a layer of stability for Team Studio installations.

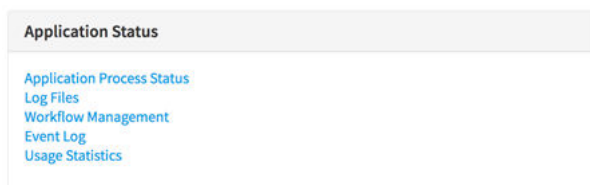
Usage Statistics

Team Studio provides login and activity statistics for users. This can help administrators better understand how their system is being used.

Prerequisites

Procedure

- 1.
2. From the sidebar menu, click **Administration**.
3. To see the latest activity aggregates for Team Studio users, from the **Application Status** list, select **Usage Statistics**.



The **Usage Statistics** screen appears. This screen displays information about Team Studio user activity. The top line, highlighted yellow, shows usage aggregations across all users in your instance of Team Studio. Each following line shows statistics specific to an individual user.

Usage Statistics

User Name	User Email	User Created Date	Total Logins	Actions in the Last 30 Days	Last Login
—	—	—	6897	2966	2017-08-21 11:37:09 UTC
avalanche_qa	robot_qa@alpinenow.com	2017-08-20 11:07:29 UTC	167	116	2017-08-21 11:37:09 UTC
chorusadmin	chorusadmin@example.com	2016-12-13 13:16:18 UTC	898	419	2017-08-21 11:29:38 UTC
rkaur	rkaur@alpinenow.com	2016-12-16 03:01:17 UTC	103	614	2017-08-19 06:10:49 UTC
jlee	jlee@alpinenow.com	2017-01-31 22:08:48 UTC	46	14	2017-08-18 18:55:59 UTC
josh	josh@alpinenow.com	2017-07-10 22:53:14 UTC	11	64	2017-08-18 17:10:58 UTC

- **User Name:** Team Studio user name.
- **User Email:** Email address associated with the user account.
- **User Created Date:** Creation date of the user.
- **Total Logins:** Total count of logins since the creation date of the user.
- **Actions in the Last 30 Days:** Rolling count of activities over the past 30 days.
- **Last Login:** Date of the latest login by the user.

Logins are tracked based on browser sessions when users go through the Team Studio authentication process, Single-Sign On or LDAP authentication.

Activities are counted for each action a user takes while logged in to Team Studio, including things like saving a workflow, creating a new workspace, changing a data source, and more. This is an aggregated form of the Event Log, highlighted in [Administrator Options in Team Studio](#).

Data Visibility

Data visibility in Team Studio is a system for managing what data sources users can see and access within workspaces. The goal is to provide more granular control and security by allowing users to only know about certain data in the application.

The Team Studio data visibility system has four tenets. All of them come together to form a robust and cohesive data visibility offering.

Browsing Datasets In Your Workspace

As an administrator you can select and view the contents of any data source that you have the permissions to view.

Perform this task from the **Data Sources** section.



Prerequisites

You must have administrative credentials, and you must have permissions to see the data source. These can be shared credentials, individual credentials, or workspace credentials.

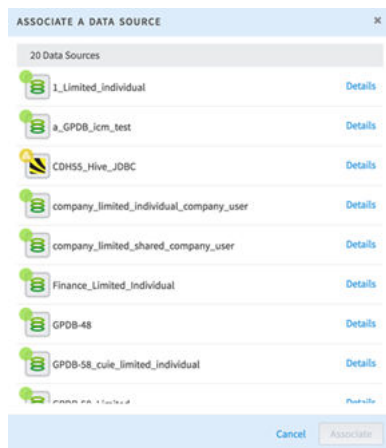
- **Shared credentials-** If the data source has shared credentials, then everyone can access it using the same set of credentials. You can see the tables and use them in Team Studio workflows.
- **Individual credentials-** If you have individual credentials for this data source, you are prompted to enter them. Everyone that accesses the data source must use his or her credentials. If you do not have credentials, or if you do not know your data source credentials, then contact the person in charge of your data source.
- **Workspace credentials-** As a data administrator, you can specify credentials for the entire workspace. Everyone in the workspace shares the same set of credentials, but those credentials apply only to the active workspace.



Workspace credentials take precedence over individual credentials if both exist in this workspace.

Procedure

1. To associate more data sources to this workspace, click **Associate a Data Source**.



A dialog box is displayed containing the data sources not yet associated. You can also view details about the data sources including the owner and the host address.

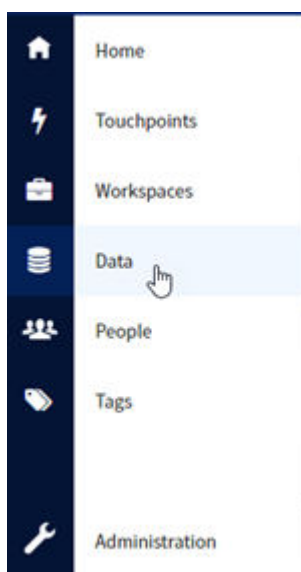
2. Select the data source you want to add, and then click **Associate**.

Browsing Datasets In the Entire Application

From the Data section, you can see a list of all data sources connected to the application, and you can add more data sources.

Procedure

- From the sidebar menu, click **Data**.



Result

From this section of the application, you can also control options such as data source visibility and permissions.

Controlling Data Source Visibility

Data sources can be global; that is, they can be designated as "public" or they can be scoped as "limited", restricting their visibility and available to only the workspaces with which they are associated.

Perform this task from the main Data Sources section of the application, accessible from the **Data** sidebar menu. You cannot edit data sources from a workspace.

Prerequisites

You must have Data Administrator credentials to control data source visibility.

Procedure

1. Select the data source.
2. In the contextual sidebar, click **Edit Data Source**.
3. In the **Workspace Visibility** drop-down list box, select either **Public** or **Limited**.



A data source set to **Limited** must be associated manually with a workspace for members of that workspace to use the data source.

4. Click **Save Configuration**.

Controlling Data Source Permissions

Besides changing data source visibility, you can also change the level of permissions on the data source. A user might be able to see a data source in the Data Sources section in their workspace, but they will not be able to access it unless they have permission.

Perform this task from the main Data Sources section of the application, accessible from the **Data** sidebar menu. You cannot edit data sources from a workspace.

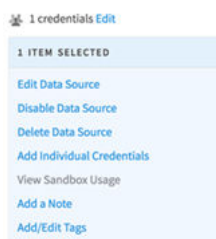
Prerequisites

The permission options range from shared by everyone to workspace-specific credentials to individual credentials for each user. Depending on your use case, you can change this permission scheme at any time.

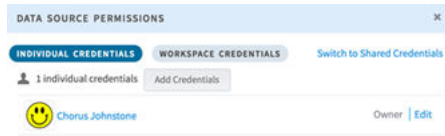
Procedure

- 1.
2. Select the source from the data primary navigation.
3. Click **Edit**.


In the following example, the data source has Individual credentials and only one user account has access.



4. In the Data Source Permissions dialog box, perform one of the following tasks.



- View the individual credentials, as well as credentials granted to any workspaces.
- Click **Add Credentials** to add new user accounts for individual credentials.



These credentials must also be present on the data source in order to validate properly.
- Switch to shared credentials (which means that one set of credentials will be used by everyone), by clicking **Switch to Shared Credentials**.

Adding Data to a Workspace

For limited-visibility data sources, you must associate them with each workspace you want them to be accessible to.


Prerequisites

You must be either a data administrator or an application administrator.

Procedure

1. From the sidebar menu, click **Data**.
2. Select the data source.
3. Click **Associate Data Source to a Workspace**.

If this option does not appear, ensure that the data source you have selected is Limited visibility. If it is Public, the data source is already available to all workspaces.

- 

Associating a data source to a workspace makes it visible, but users must still have proper credentials to access the data within. You can set this up in one of three ways.

 - **Shared credentials:** Everyone uses the same set of credentials-this is the most broad scope.
 - **Workspace credentials:** Everyone in the workspace uses the same set of credentials. See [Controlling Data Source Permissions](#).
 - **Individual credentials:** Each member that wants to access the data source must have credentials for that data source.

Data Source Associations

In Team Studio, you must associate data sources with workspaces.

This process is similar to (but not exactly like) how datasets are associated with workspaces. Associated data sources are displayed in a new workspace section labeled **Data Sources**.

- All public data sources are automatically associated to workspaces.

- Limited data sources must be manually associated to a workspace.

Workfiles (workflows, SQL, and notebooks) can use only the data sources available to the workspace. Datasets in the workspace must come from data sources available to that workspace.

For more information, see "Data Sources" in the *TIBCO® Data Science Team Studio User's Guide*.

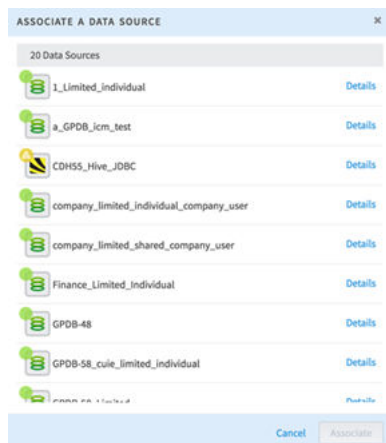
Associating a Data Source

Follow these steps to associate a data source to a workspace in Team Studio.

Prerequisites

Procedure

- 1.
2. Navigate to a workspace. You must be a member of the workspace and have the Data Administrator role.
3. Select the **Data Sources** section.
4. Click **Associate a Data Source**, and then from the dialog box, select a data source.



5. Optional: Click **Details** to see more information about its data source.
6. After choosing the data source, click **Associate** to save the changes.

Data Source Credentials

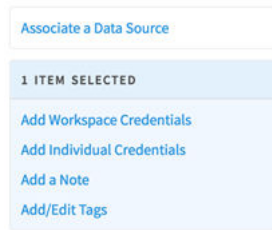
A data source credential option is available in Team Studio that provides limited visibility.

You can use this credential option when you need to specify a data source that is not just "individual" and is not "shared by all." You can provide authorization credentials that specify only that workspace.

Each workspace can have its own set of shared credentials. These credentials are not shared globally across the application, but are scoped to activity just within the workspace. This feature is available for database data sources only.



Workspace credentials override any existing credentials when accessing the data source in that workspace.

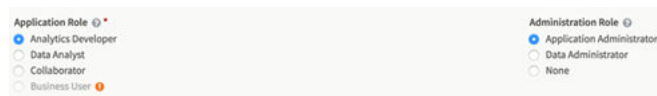


From the **Data Sources** section of the workspace, specify **Add Workspace Credentials** or **Add Individual Credentials**.

Data Administrators

The Data administrator has permissions to see all data sources and manage the data source associations for workspaces. However, this role does not have full application administration permissions.

You can find the Data Administrator role in the Roles section of your profile. You can have only one administration role at a time. Like the Application Administrator role, it is not controlled by licensing limits, so you can have as many administrators in Team Studio as you need.



To learn more about the Data Administrator role, see [Team Studio Licensing](#).

Data Source States

Data source states allow users to have more control over their data sources. Some of the features are:

- Users can save incomplete data source configurations and come back to them later.
- Users can disable a data source from the Team Studio web application.
- Users can view the status of a data source at a glance.

Stateful Data Sources

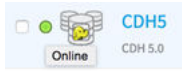
Data sources within Team Studio have states that reflect their status in the system. There are four different states:

- Online
- Offline
- Incomplete
- Disabled

These states have an impact on whether a user can browse them or use them in workflows or sandboxes.

Online State

A data source becomes active when a user enters correct connection information and a successful connection is made. An active data source can be used and Team Studio indexes it. In the list of data sources, this is displayed with a green status marker and the following tooltip:



Offline State

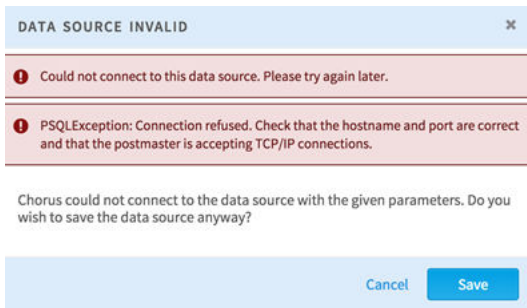
This data source was once normal, active, and correctly configured. However, for some reason the data source is no longer connecting correctly. The data source might still be indexed by Team Studio, but might not be up to date. Verify that the data source is connected before using it in a workflow.



Incomplete State

An incomplete data source is one that a user has begun to add, but has not yet filled out every parameter and has not yet connected to the underlying data source. For that reason, the data source is still in a draft state. Because users might not have all the information needed to configure a new data source at one time, this state allows users to save progress and return to it later. The data source is not indexed or available until it is properly configured.

To save a data source that is incomplete or has invalid information, click **Save**. The following dialog box is displayed.



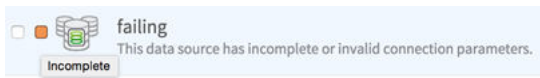
If you click **Save** in this dialog box, the data source is in the Incomplete state and is not usable until a complete connection occurs. If you click **Cancel**, you are redirected to the **Add Data Source** screen to edit the configuration.



For database data sources, the user must fill out all fields with red asterisks. The user can save the data source configuration with invalid data, rendering it Incomplete.

For Hadoop data sources, the user must fill out all fields with red asterisks, but can save the configuration while leaving the extra connection parameters section empty.

After all of the required fields are completed, the data source can be validated and used. In the list of data sources, this is displayed as plain, read-only text and a red square:

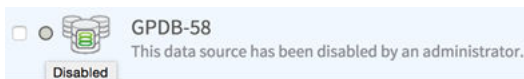


Disabled State

A disabled data source is one that has been active and correctly configured, but has been manually disabled by an application administrator. A disabled data source cannot be used and cannot be browsed by application users. Team Studio does not index or check the status of disabled data sources.

An administrator can reactivate a disabled data source by selecting **Enable Data Source** from the sidebar menu. When the data source is reactivated, Team Studio attempts to test it and validate the configuration. If this works normally, it becomes Active. If the data source does not validate, it remains disabled.

In the list of data sources, a disabled data source displays as plain read-only text, without a link. It displays a gray circle as a status marker:



Data Source States in Workflows

A data source's state affects how workflows appear to the user.

When selecting a workflow, users can now see the status of the underlying data source. For example:



The icon changes depending on the data source status.



Existing workflows with an underlying incomplete or disabled data source cannot be opened.

Team Studio Licensing

Team Studio has a licensing model that allows granularity and access control for users of the application. This affects all levels of the business organization, from data scientists to the front-line business user.

This topic explains the roles in the application and the permissions granted with each role. For more information on your license terms, contact your Team Studio Account Manager.

To view license information from within Team Studio, in the top right corner, click your user name, and then click **About**. This displays information about features that are enabled on your Team Studio instance. For more information on the licensed roles and user counts, from the sidebar menu, click **Administration > License Information**, or see [Administrator Options in Team Studio](#).

Manage Team Studio Users

The **People** page displays a complete list of Team Studio users, developers, and administrators.

On this page, you can do the following:

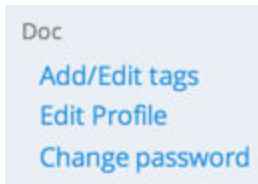
- See a list of all users, developers, and administrators, sorted by last name (change the default to sort by first name).
- Select a user name to go to that user's page for more information.



Only an administrator can add or delete a user.

Managing User Profiles

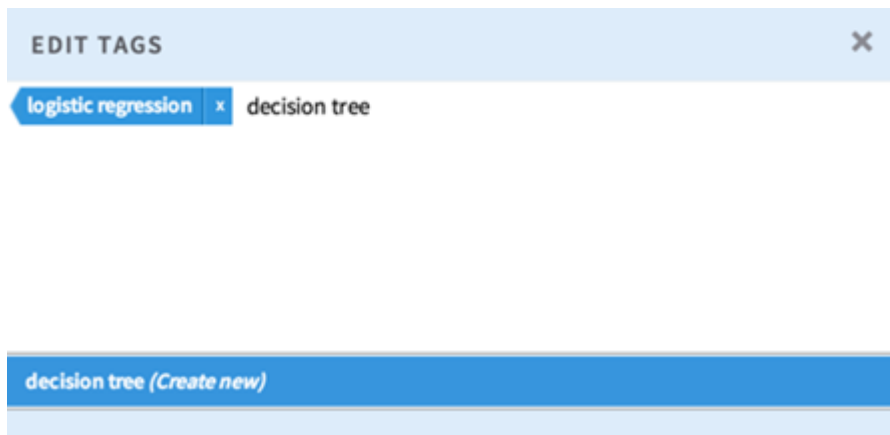
Users can manage their profile information details by clicking their names, and then clicking **Your Profile**. The resulting Team Studio User Management user interface provides the following options:



Prerequisites

Procedure

1. To associate specific topics that are discovered in a search, select **Add/Edit Tags**.



2. To change basic profile information, select **Edit Profile** and edit the following properties as necessary:

Account Information

- User Name: The unique name for each user.

Application Roles

- Administrators can give other users administrator and/or developer privileges.

Personal Information

- You can edit the fields **First Name**, **Last Name**, **Email**, **Title**, **Department**, and **Description**. **First Name**, **Last Name**, and **Email** are required fields.

You can subscribe to receive email notifications.

3. Click **Save Changes** to keep your changes, or click **Cancel** to revert to the previous version.
4. To reset your password, select **Change password**.

Add a New Person

This topic describes how to add a new user to the **People** page, which includes a complete list of Team Studio users, developers, and administrators.



You must be an administrator to carry out this task.

Prerequisites

Procedure

1. Open the **People** page and click **Add Person**.
The **New Person** page is displayed.

2. Fill in the required fields, including a **Username** and **Password** for the new person. The account owner can change the password later.



If Team Studio is configured with LDAP/AD (see the [LDAP Authentication Documentation](#)), there are no password fields.

Multiple administrators can exist within Team Studio, and an administrator can make any other user an administrator. To do this, select the **Administrator** check box.

A person can have one of four roles within the application. Each role has a different permissions level. Analytics Developers have the highest permissions level, and Business Users have the lowest. For more information on the application roles, see [Team Studio Licensing](#).

3. Click **Add Person**.
The new account is created.
4. To view the list of people, select **People** from the Quick menu on the left sidebar.
The Contextual Sidebar includes the following actions.
 - **Add Tags** or **Edit Tags** (any user)
 - **Edit Profile** (the selected user or an administrator only)
 - **Delete Person** (an administrator only)
 - **Change password** if not using LDAP (the selected user or an administrator only)

Establish Your Identity

Team Studio is intended to expose you to the knowledge and expertise of colleagues in your organization you do not work closely with. After a Team Studio administrator creates your credentials, open your User Profile and provide information about yourself to help that exposure and your interactions.

As a user, you can edit your profile in one of two ways:

- From the contextual sidebar of the **People** page, click **Edit Profile**.
- On the home page, on the global navigation bar, click your name, and then click **Your Profile**.

Prerequisites

Procedure

1. When your profile opens in the main panel, in the contextual sidebar, click **Edit Profile**. Your profile page appears. You can edit nearly every aspect of it, including changing your password (you cannot, however, change the user name assigned by the administrator who added you). For example, you might add a picture and fill out the relevant fields with information you believe could be useful to others in your organization.

People > User Profile > Edit Profile

collab someone

Account Information

Username
collab

Roles

Application Role(s) *

Analytics Developer
Data Analyst
Collaborator
Business User

Administration Role(s)

Administrator

Personal Information

First Name *
collab

Last Name *
someone

Email *
collab@apinew.com

Title
QA

Department

Description
5/1000 characters
☒ Subscribe to email notifications

[Save Changes](#) [Cancel](#)

2. When you are finished, click **Save Changes**.
Your updated profile is visible to you and other accounts.

Deployment Targets

You can configure the deployment targets to which the engines have access using this administration page.

Deployment targets are servers that host real-time PFA engines. Team Studio provides the ScORE server application for deployment with Team Studio 6.3 - contact your Team Studio representative for details.

Administration

Administration > Deployment Targets

Deployment Targets

URL	Name	Environment	Server Type
http://localhost:8982/	Prod	Production	ScORE
http://localhost:8981/	Dev	Development	ScORE

[Create New Target](#)

1 ITEM SELECTED

Development ScORE [Cancel](#) [Save](#) [Delete](#)

Each deployment target has the following properties:

URL	Location of the ScORE server.
Name	Name of this server (presented to end users when configuring engines).
Environment	Development or production. For production environments, engines can only be deployed if they have been explicitly approved.
Server Type	Currently, ScORE is the only server type that Team Studio supports.

You can select and delete deployment targets from this page. If you delete a deployment target, Team Studio undeploys all of its engines.

Upgrading

When you get ready to update to a new version of Team Studio, follow these steps.

Preparing to Upgrade Team Studio

Before you install a new version of Team Studio, follow these steps to prepare your server. Perform this task on the server where you plan to upgrade Team Studio.

Prerequisites

- Be sure your server meets the system requirements.
- Back up any previous installations. See [Backing up Team Studio](#) for more information.

Procedure

1. Open the file `chorus.properties`, and ensure that the property `workflow.url` is set to `http://localhost:9090`, rather than an external IP address.
2. Download the Team Studio installation package and validate it.
The package is in the form `chorus-6.0.0.0.<build number>-<sha>.sh`, where `<sha>` is a hash that maps to a specific code commit.
3. Put the installation package in a folder where the user `chorus` has write privileges.



For installation on a DCA standby master or DCA (UAP Edition) DIA module, this folder should be in the `/home/chorus` directory.

4. Run MD5 on the binary.

This tool generates a string that you can compare to the value provided with the installation package to verify that you downloaded the correct file. For example, run the following.

```
# md5sum ~/chorus-6.0.0.0.1549-9d20ac862.sh
MD5 chorus-6.0.0.0.1549-9d20ac862.sh=
935d82688591bf0e6f0ba05dc5837fd3
```

You can compare `935d82688591bf0e6f0ba05dc5837fd3` with the value provided with the installation package. If the values match, then you downloaded the correct file.

5. Log in as `chorus` if you are not already logged in as this user.

Because only the user who has privileges to start and stop the system can upgrade the software, you must log in as chorus.

6. Source `chorus_path.sh` according to the appropriate value for the environment variable `$CHORUS_HOME`.

```
$ echo $CHORUS_HOME
/usr/local/chorus
$ source $CHORUS_HOME/chorus_path.sh
```

7. Make sure Team Studio is stopped.

```
$ chorus_control.sh stop
```

8. Write down the Team Studio memory startup settings and Kerberos realm settings.

You must transfer these settings into the file `deploy.properties`. See [Team Studio Deploy Properties](#) for details.

What to do next

- If you plan to use `gpfdist`, then copy the `gpfdist` files from the Greenplum installation into the vendor directory of the `$CHORUS_HOME` folder. See [Configuring an External Server to Import Data with gpfdist](#) for more information.
- [Run the upgrade.](#)

Running the Upgrade

After preparing the computer for the upgrade, follow these steps to complete the task. Perform this task on the computer that you have prepared for the upgrade.

Prerequisites

- Review the [data source requirements](#) for your data source.
- You must have completed the [preparations for the upgrade](#).



If you are upgrading from an older version than the immediately previous version, you must upgrade each of the previous releases. For example, to upgrade from version 5.x, you must first upgrade to version 6.0 before upgrading to this release.

Procedure

1. As the chorus user, make the installer executable.

This instruction applies if you downloaded the installer as the chorus user. If you downloaded with another account, log in as root for this step.

```
$ chmod +x ~/TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

2. As the chorus user, run the installer with the following command.

```
$ ./TIB_sfire-dsc-6.6.0_linux_x86_64.sh
```

3. Type `y` to accept the license agreement.

If you type `n`, the installer exits.

The installer prompts you for the installation directory.

4. Provide the installation directory for the installation you are upgrading from.
5. When the installer completes the installation, and before you start the Team Studio server, perform the following steps.
 - a) Place the file `chorus.license` in the `$CHORUS_HOME/shared` directory
 - b) Review and customize the properties listed in the file `chorus.properties`.

Specifically, set the property `workflow.url = http://localhost:9090/`. See [Team Studio Configuration Properties](#) for more information.

6. Add the following items to the file `~/.bashrc`.

This step is required because the installer cannot write to this file.

```
# User specific environment and startup programs
PATH=$PATH:$HOME/bin
export JAVA_HOME=[/path/to/jdk1.7]
# default CHORUS_HOME=/usr/local/chorus
export CHORUS_HOME=[/path/to/installation/directory]
# default CHORUS_DATA=/data/chorus
export CHORUS_DATA=[/path/to/data/directory]
export PATH=$PATH:$JAVA_HOME/bin
export PATH
```

7. In the file `$CHORUS_HOME/shared/chorus.properties`, verify the following Java options.

```
java_options = -Djava.security.egd=file:/dev/./urandom -server -Xmx4096m -Xms2048m -
Xmn1365m -XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -
XX:ParallelGCThreads=3 -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./ -
XX:+CMSClassUnloadingEnabled
```

8. Ensure you specify enough Jetty threads.

- a) Check the number of cores specified in the file `/proc/cpuinfo`. In the following example, we have 8 processors, zero actual cores, and 8 processing units.

```
$ cat /proc/cpuinfo | grep processor
processor       : 0
processor       : 1
processor       : 2
processor       : 3
processor       : 4
processor       : 5
processor       : 6
processor       : 7
$ cat /proc/cpuinfo | grep 'core id'
$ cat /proc/cpuinfo | grep processor | wc -l
8
```

- b) If the server has more than 16 processors, in each of the following `jetty.xml` files, modify the `maxThreads` to 140.

- `$CHORUS_HOME/current/vendor/jetty/jetty.xml`
- `$CHORUS_HOME/current/vendor/jetty/etc/jetty.xml`

```
<Set name='maxThreads'>140</Set>
```

9. If you have a Team Studio installation from 2017 or earlier, when you upgrade to version 6.3.2 or later, in the file `alpine.conf`, modify the default output directories to the following values.

```
alpine.hdfs.output.dir=tsds_out/
alpine.hdfs.runtime.dir=tsds_runtime/
alpine.hdfs.model.dir=tsds_model/
```

What to do next

[Start Team Studio.](#)

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

Product-Specific Documentation

The following documents for this product can be found in the TIBCO Documentation Library.

- *TIBCO® Data Science Team Studio Release Notes*
- *TIBCO® Data Science Team Studio System Requirements*
- *TIBCO® Data Science Team Studio Version and Licensing*
- *TIBCO® Data Science Team Studio Installation and Administration*
- *TIBCO® Data Science Team Studio User's Guide*
- *TIBCO® Data Science Team Studio Development Kit*

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking Register on the website.

System Requirements for TIBCO® Data Science Team Studio

For information about the system requirements for Team Studio, see *TIBCO® Data Science Team Studio System Requirements*.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](https://community.tibco.com). For a free registration, go to <https://community.tibco.com>.

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO Data Science Team Studio, TIBCO Spotfire, Alpine, and Chorus are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme.txt file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2017-2021. TIBCO Software Inc. All Rights Reserved.

Index

A

administration 14
 alpine.catalina.opts 82
 alpine.conf 92
 alpine.keytab 59
 alpine.principal 59
 Amazon EMR 9
 Amazon RedShift 8
 Apache Impala 8
 Azure SQL Data Warehouse 8

B

back up 79
 browser version 7

C

CentOS 7
 check 25
 chorus_only 25
 chorus_path 25
 chorus_user 25
 chorus.server 81
 Cloudera 9
 configuration 92
 configuration parameters 59
 configure 43
 confirm 25
 cron back up 79

D

data 132, 133
 data source 92
 data_path 25
 database configuration 47, 49, 51, 52
 databases 7, 8
 deploy 82
 dfs.datanode.kerberos.principal 59
 dfs.namenode.kerberos.principal 59
 disable_spec 25
 Docker spawner 41

G

GPDB 47
 Greenplum 8, 47, 48

H

Hadoop 53, 92, 109
 Hadoop configuration 56
 Hadoop platforms 7, 9

hadoop.security.authentication 59
 HAWQ 9, 49, 50
 HDFS configuration 56
 help 25
 Hive 9, 109
 Hive JDBC 8

I

IBM Big Insights 9
 info 25
 installation 25, 26, 28
 installer help 25

J

Java version 7
 Jupyter 13
 Jupyter Notebook ports 32
 Jupyter Notebooks 40

K

keep 25
 Kerberos 55, 56, 109
 keytab 55

L

license 14, 138
 LinuxContainerExecutor 56
 list 25
 local process spawner 41
 lsm 25

M

MADlib 7
 mapreduce.jobhistory.principal 59
 MawpR 9
 MS SQL 8

N

NameNode host 59
 NameNode port 59
 nochown 25
 noexec 25
 notebooks 13
 nox11 25

O

open-source R 12, 35, 36, 38
 Oracle 8

P

- passphrase 25
- Pivotal 114
- Pivotal HAWQ 8
- PivotalHD 9
- PMML 7
- ports 31
- postgres service 80
- PostgreSQL 8, 51–53
- prerequisites 14
- principal 55
- properties 82
- PySpark 13, 40
- Python 13

Q

- quick install 21, 23

R

- R engine version 7
- R Server 12, 16, 35, 36, 38
- Red Hat Enterprise 7
- ResourceManager host 59
- ResourceManager port 59

S

- SAML 73
- SAP Hana 8
- scheduler service 80
- server.xml 31
- service restart 78
- service start 77
- service stop 78

- serviceuser 58
- setuid 56
- silent 25
- solr service 80
- Spark 40, 92
- Spark ports 40
- stored procedure 48, 50, 53
- stored procedures 47, 49, 51, 52

T

- Tableau server version 7
- tar 25
- target 25
- Team Studio service 80
- Tomcat 82

U

- upgrade 44, 142

V

- Vertica 8
- visibility 132, 133

W

- webserver service 80
- workers service 80

Y

- YARN configuration 56
- yarn.app.mapreduce.am.staging-dir 59
- yarn.resourcemanager.principal 59
- yarn.resourcemanager.scheduler.address 59