



ThoughtSpot Data Integration Guide

Release 6.0 December 13, 2019

© COPYRIGHT 2015, 2019 THOUGHTSPOT, INC. ALL RIGHTS RESERVED.

910 Hermosa Court, Sunnyvale, California 94085

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from ThoughtSpot, Inc.

All rights reserved. The ThoughtSpot products and related documentation are protected by U.S. and international copyright and intellectual property laws. ThoughtSpot and the ThoughtSpot logo are trademarks of ThoughtSpot, Inc. in the United States and certain other jurisdictions. ThoughtSpot, Inc. also uses numerous other registered and unregistered trademarks to identify its goods and services worldwide. All other marks used herein are the trademarks of their respective owners, and ThoughtSpot, Inc. claims no ownership in such marks.

Every effort was made to ensure the accuracy of this document. However, ThoughtSpot, Inc., makes no warranties with respect to this document and disclaims any implied warranties of merchantability and fitness for a particular purpose. ThoughtSpot, Inc. shall not be liable for any error or for incidental or consequential damages in connection with the furnishing, performance, or use of this document or examples herein. The information in this document is subject to change without notice.

Table of Contents

Introduction to Data Integration	3
Embrace	
Overview.....	5
JDBC and ODBC setup prerequisites	8
ODBC driver client	
ODBC driver overview.....	9
ODBC on Windows	
Install the ODBC driver on Windows.....	13
Configure multiple connections on Windows.....	23
Deploy SSL with ODBC on Windows.....	28
Set up the ODBC driver for SSIS	35
Install the ODBC driver on Linux.....	46
Best Practices for Using ODBC	57
JDBC driver client	
JDBC driver overview	58
Use the JDBC driver.....	60
Set up the JDBC driver for Pentaho	66
Troubleshooting	
Troubleshooting Data Integrations.....	80
Enable ODBC logs.....	81
Enable JDBC logs.....	87
Schema not found error with ODBC.....	88
How to improve throughput	90
ODBC tracing on Windows.....	91
Reference	
Supported SQL commands	93
Connection configuration.....	95
Keyword reference.....	98
TSQL reference	106
tsload flag reference	119

tscli command reference.....	122
Date and time formats reference.....	167
Row level security rules reference	171
Formula function reference.....	184
Alert codes reference	202
User action codes reference	211
Frequently asked questions	213

Introduction to Data Integration

This guide explains how to integrate ThoughtSpot with other data sources for loading data. It also includes information on installing and using the ThoughtSpot clients (ODBC, JDBC, and more).

ThoughtSpot Clients

ThoughtSpot provides certified clients to help you load data easily from your ETL tool or another database. These include ODBC and JDBC drivers.

You can obtain the ThoughtSpot client downloads from the Help Center. Always use the version of the ThoughtSpot clients that corresponds with the version of ThoughtSpot that you are running. When upgrading, make sure to upgrade your clients as well.

▲ Important: The ETL tool must add a data transformation step if the source column data type does not exactly match the target's, ThoughtSpot's, column data type. The driver does not do any implicit conversions.

Methods for loading data

There are several ways to load data into ThoughtSpot, depending on your goals and where the data is located. Always consider your requirements for recurring loads when planning how best to bring the data into ThoughtSpot.

Here are the options, with information on where to find the documentation for each method:

Method	Description
ThoughtSpot Loader (tsload)	ThoughtSpot Loader is a command line tool to load CSV files into an existing database schema in ThoughtSpot. This is the fastest way to load extremely large amounts of data, and it can be run in parallel. You can also use this method to script recurring loads. See the ThoughtSpot Administrator Guide for details.

Method	Description
User Data Import	Users can upload a spreadsheet through the web interface with User Data Import. This is useful for giving everyone easy access to loading small amounts of their own data. See the ThoughtSpot Administrator Guide for details.
ODBC	ThoughtSpot provides an ODBC (Open Database Connectivity) driver to enable transferring data from your ETL tool into ThoughtSpot.
JDBC	ThoughtSpot provides a JDBC (Java Database Connectivity) driver to enable transferring data from your ETL tool into ThoughtSpot.
Microsoft SSIS (SQL Server Integration Services)	You can use the ODBC driver to connect to SSIS and import data into ThoughtSpot. Basic instructions are included in this guide.
Connect to Pentaho	You can use the JDBC driver to connect to Pentaho and import data into ThoughtSpot. Basic instructions are included in this guide.

Where to go next

- [**Server-side prerequisites for using JDBC/ODBC to import data**](#)

You must follow setup prerequisites for importing data using JDBC/ODBC.

- [**About the ODBC Driver**](#)

You can use the ThoughtSpot ODBC driver to bring data into ThoughtSpot from your ETL tool or database.

- [**About the JDBC Driver**](#)

Java Database Connectivity (JDBC) is a Java standard API that allows applications to interact with databases in a standard manner. ThoughtSpot has JDBC support through a JDBC driver that we provide.

Embrace overview

Summary: Using Embrace, you can perform live query on external databases.

If your company stores source data externally in data warehouses, you can use ThoughtSpot Embrace to directly query that data and use ThoughtSpot's analysis and visualization features, without moving the data into ThoughtSpot. If you decide later you want to copy your data into ThoughtSpot, you can also do that with Embrace.

Embrace supports the following external databases:

- Snowflake
- Amazon Redshift (*in beta*)

To enable Embrace, contact ThoughtSpot support.

How it works

You create a connection to the external database, choosing the columns from each table that you want to explore in your live query. Primary key and foreign key relationships are imported along with the primary and foreign key tables. If there are any joins in the tables of your connection, they are also imported. After your connection is complete, it becomes a **linked** data source in ThoughtSpot that allows you to query the external database directly. It's easy to apply transformations and filter the data also.

Key benefits

- Set up and deploy ThoughtSpot faster by connecting directly to the external database.
- Eliminate the need to move data into ThoughtSpot for analysis.
- Centralize data management and governance in the external database.
- Save significant time and money by avoiding ETL pipelines.
- Set up and schedule sync of data into ThoughtSpot.
- Connect to multiple external databases.

Embrace modes

Embrace has two operating modes:

- **Linked:** ThoughtSpot queries your data in the external database.
- **Synced:** ThoughtSpot queries a copy of your data stored in ThoughtSpot.

When you create your connection to an external database, by default, it is a **Linked** connection. If you want to copy the external data into ThoughtSpot, you must sync the data. The features available with Linked and Synced tables are slightly different.

Features in Embrace modes

Feature	Linked Tables	Synced Tables
<i>Simple Search</i>	Yes	Yes
<i>Complex searches like Versus, Inline Subquerying, Growth</i>	Yes	Yes
<i>Search Suggestions for column names</i>	Yes	Yes
<i>Search Suggestions for column values</i>	Yes	Yes
<i>Headlines at the bottom that summarize tables</i>	Yes	Yes
<i>All Chart Types & Configurations</i>	Yes	Yes
<i>SpotIQ Instant Insights</i>	No	Yes
<i>SpotIQ pre-computed insights</i>	No	Yes
<i>Table and Column Remapping</i>	Yes	N/A
<i>Custom Calendar</i>	No	Yes
<i>Materialized Views</i>	No	Yes
<i>Indexing of table columns</i>	Yes	Yes

Next steps

- [Add a connection](#)

Create the connection between ThoughtSpot and tables in an external database.

- **Sync** Set your connection to copy tables from the external database into ThoughtSpot.

- **Modify a connection**

Edit, remap or delete a connection to tables in an external database.

- **Connectors reference**

Source cloud data connectors, and their connection credentials, supported by Embrace.

JDBC and ODBC setup prerequisites

Before you can use JDBC or ODBC to import data into ThoughtSpot, you must do the following server-side configuration:

1. Open up the ThoughtSpot firewall to allow incoming requests to Simba server.

```
tscli firewall open-ports --ports 12345
```

2. Confirm that the `simba_server` process is up. Output of the command below should contain exactly one line, as shown below.

```
ps -ef | grep simba_server | grep -v grep
admin      26679 25672  0 Jul13 ?          00:01:49 simba_se
rver_main --logbufsecs=0
```

For assistance, contact ThoughtSpot Support.

Overview of the ODBC Driver

Summary: Use the ODBC driver to bring data in from your ETL tool or database.

ThoughtSpot comes packaged with an ODBC (Open Database Connectivity) driver, so that you can transfer data between ThoughtSpot and other databases. Basic knowledge of ODBC data source administration is helpful when setting up ODBC.

Supported operating systems for the ODBC driver are:

- Microsoft Windows 32-bit
- Microsoft Windows 64-bit
- Linux 32-bit
- Linux 64-bit

Version compatibility and connection parameters

To ensure compatibility, always use the ODBC driver with the same version number as the ThoughtSpot instance to which you are connecting. You can make a secure ODBC connection to the ThoughtSpot database by configuring a user and password combination with the driver. For detailed information about connection parameters, see the [ODBC and JDBC configuration properties](#)

Supported Data Types

The ODBC driver supports these data types:

- INT
- BIGINT
- BOOLEAN
- DOUBLE
- FLOAT
- DATE
- TIME
- TIMESTAMP
- DATETIME

- CHAR
- VARCHAR

Source and target data compatibility

By default, ThoughtSpot takes a permissive approach to data type compatibility between source and target data in ODBC. In this mode, ThoughtSpot *assumes* that the incoming data matches exactly with the target data types and loads the table as is.

Alternatively, you can explicitly require that ThoughtSpot match the source data types exactly and, if it can't find a match, it returns an error and the data load fails. In this mode, for example, if the target ThoughtSpot data type for a column is INT, the source data type for that column must be INT in order for the data load to succeed.

By toggling ***strict*** and ***permissive*** `true` and `false` options, you can configure settings along a scale of behavior between the permissive, automatic approach and the strictness of the “must match” approach.

Strictness			
		true	false
Permissiveness	true	Data types are inferred and automatically converted. ThoughtSpot returns an error in cases where the data conversion is not possible. Data load fails in its entirety if any data contains mismatches. You must correct the problem in the source data and try the load again.	Data types are inferred and automatically converted. No error is thrown even if source and target data types don't match. Data load continues even when the source and target data types don't match. This means your data load may contain data types that you do not intend or that are not helpful. You are responsible for checking and validating the data in this case.
	false		

false	The source and target data types must match. If any data contains mismatches, ThoughtSpot returns an error to the client a data load fails in its entirety. You must correct the problem in the source data and try the load again.	No data types are inferred and conversion does not check for matches. This is the most permissive configuration.
	This is the strictest configuration.	

Your customer support engineer can assist you in configuring custom ODBC behavior. Regardless of the configuration you choose, you must validate that the results of data loading as *they appear* in ThoughtSpot are what you require.

Data type conversion matrix

The following table describes the conversion matrix between SQL data types and ThoughtSpot data types.

Source SQL Data Types	BOOL	INT32	INT64	DOUBLE	FLOAT	CHAR	DATE	TIME	DATETIME
SQL_BIT	Y	Y	Y	Y	Y	Y	-	-	-
SQL_TINYINT	Y	Y	Y	Y	Y	Y	-	-	-
SQL_SMALLINT	Y	Y	Y	Y	Y	Y	-	-	-
SQL_INTEGER	Y	Y	Y	Y	Y	Y	-	-	-
SQL_BIGINT	Y	Y	Y	Y	Y	Y	-	-	-
SQL_CHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL_VARCHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL_LONGVARCHAR	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL_BINARY	-	-	-	-	-	Y	-	-	-
SQL_VARBINARY	-	-	-	-	-	Y	-	-	-
SQL_LONGVARBINARY	-	-	-	-	-	Y	-	-	-

Source SQL Data Types	BOOL	INT32	INT64	DOUBLE	FLOAT	CHAR	DATE	TIME	DATETIME
SQL_DOUBLE	Y	Y	Y	Y	Y	Y	-	-	-
SQL_REAL	Y	Y	Y	Y	Y	Y	-	-	-
SQL_FLOAT	Y	Y	Y	Y	Y	Y	-	-	-
SQL_NUMERIC	Y	Y	Y	Y	Y	Y	-	-	-
SQL_GUID	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_MINUTE_TO_SECOND	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_HOUR_TO_SECOND	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_HOUR_TO_MINUTE	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_DAY_TO_SECOND	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_DAY_TO_MINUTE	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_DAY_TO_HOUR	-	-	-	-	-	Y	-	-	-
SQL_INTERVAL_YEAR	-	Y	Y	-	-	Y	-	-	-
SQL_INTERVAL_MONTH	-	Y	Y	-	-	Y	-	-	-
SQL_INTERVAL_DAY	-	Y	Y	-	-	Y	-	-	-
SQL_INTERVAL_HOUR	-	Y	Y	-	-	Y	-	-	-
SQL_INTERVAL_MINUTE	-	Y	Y	-	-	Y	-	-	-
SQL_INTERVAL_SECOND	-	Y	Y	-	-	Y	-	-	-
SQL_TYPE_TIME	-	-	-	-	-	Y	-	Y	Y
SQL_TYPE_DATE	-	-	-	-	-	Y	Y	-	Y
SQL_TYPE_TIMESTAMP	-	-	-	-	-	Y	Y	Y	Y

If a conversion is not possible, an error is returned to the client to indicate conversion failure. The ETL tool must add a data transformation step if the source column data type does not exactly match the target's ThoughtSpot column data type. The driver does not do any implicit conversions.

Install the ODBC driver on Windows

Summary: Use this procedure to obtain the Microsoft Windows ODBC driver and install it.

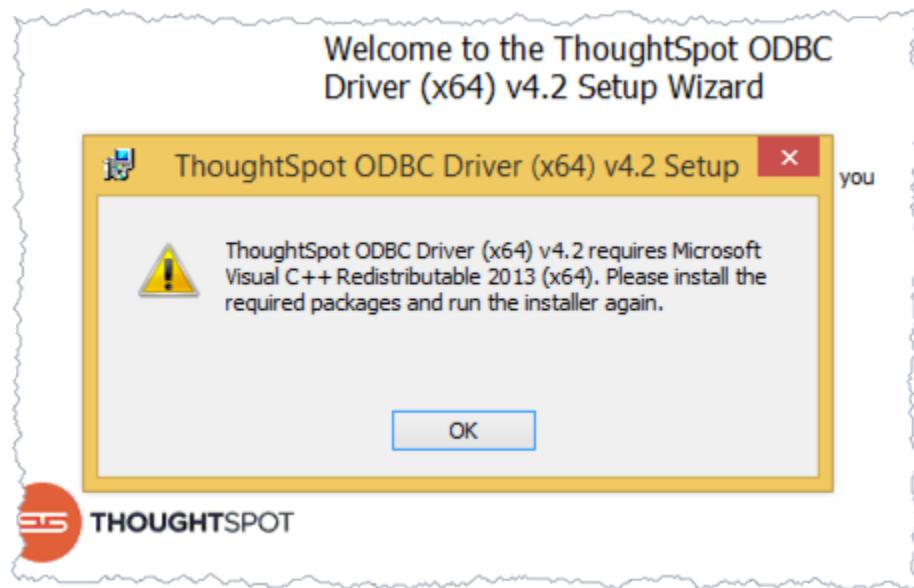
ThoughtSpot's ODBC connection relies on the [SimbaEngine X SDK](#) to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the Simba ODBC driver on a Windows workstation.

Make sure you have read the overview material in the [ODBC driver overview](#). This workstation is the same machine where you plan to run your ETL activities.

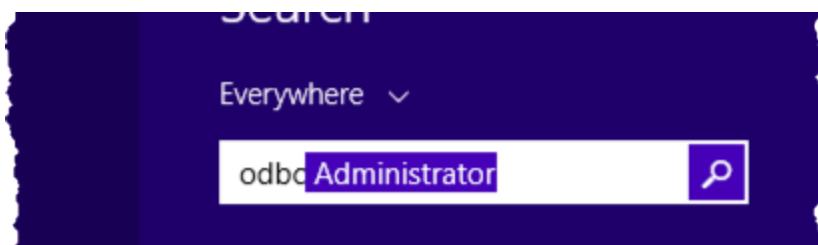
Prerequisites

These instructions include directions to use the `ssh` command. Make sure your Windows workstation is equipped with a tool [such as Putty](#) for making `ssh` connections to your ThoughtSpot server.

The ODBC driver for Windows requires Visual C++ Redistributable for Visual Studio 2013. You are prompted to install it during installation of the driver if it isn't already installed.



To check if this Microsoft tool is already installed, search for it on your workstation.



If it isn't installed, make sure you [download and install it](#) before continuing.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

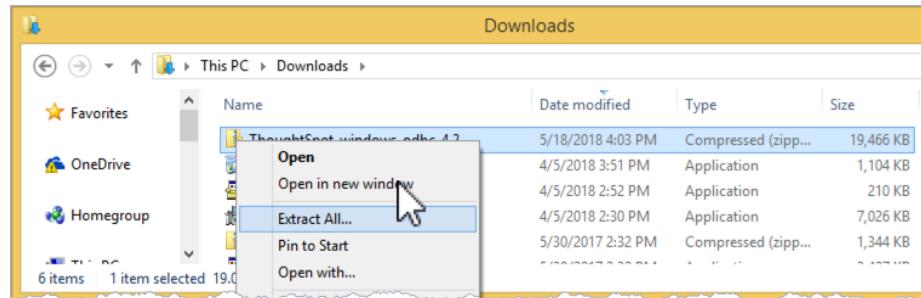
```
$ tscli node ls  
172.18.231.17  
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Download the driver

On the workstation where you want to connect from, do the following:

1. Navigate to the [Downloads](#) page.
2. Download the **ODBC Driver for Windows**.
3. Unzip the file you downloaded at a convenient location on your workstation.



4. Take a moment to examine the contents of the new directory.

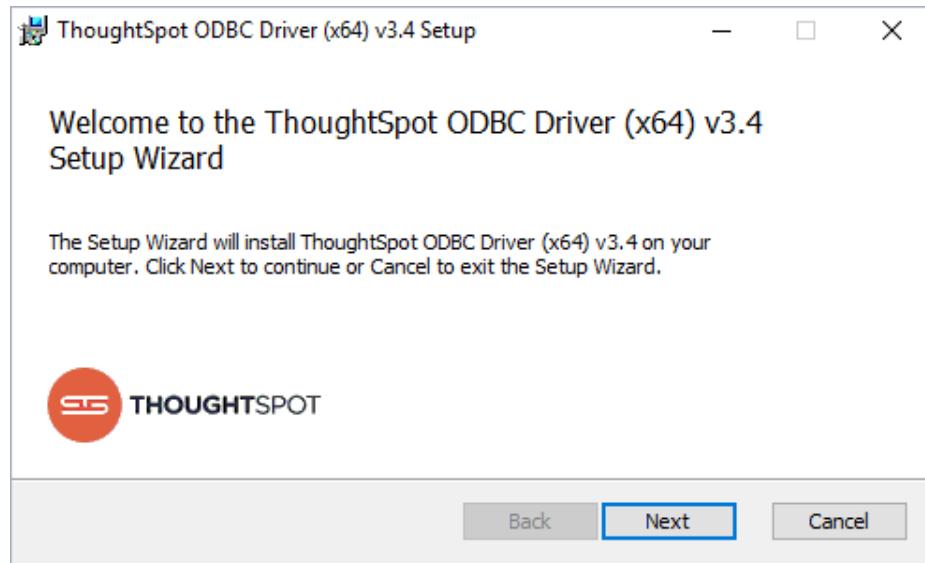
There are two different Windows ODBC installers included in the file you downloaded.

- ThoughtSpotODBC (x86).msi for Windows 32-bit
- ThoughtSpotODBC (x64).msi for Windows 64-bit

Install the driver and supporting software

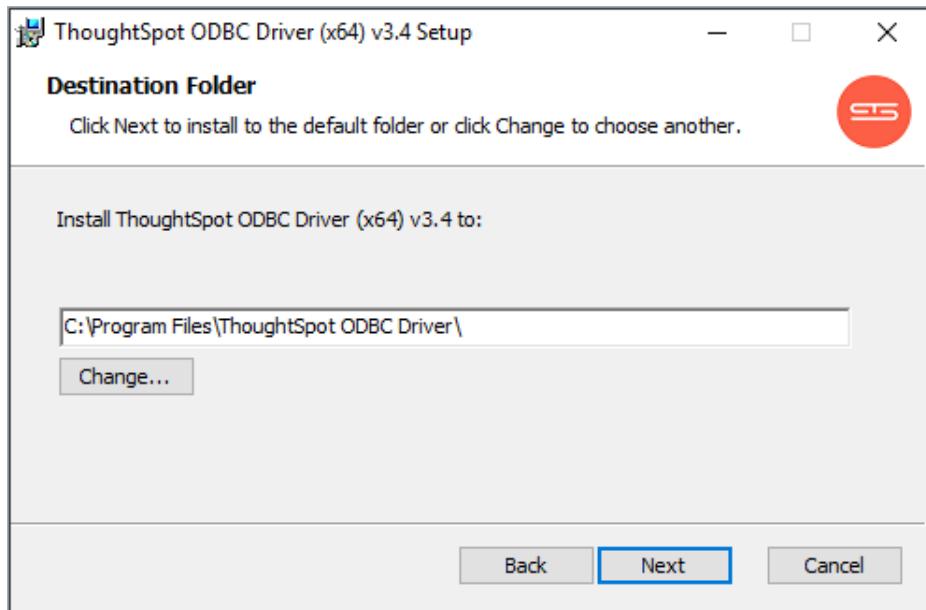
The installation process installs the Simba driver and adds the ODBC Administrator software to your workstation. You use this software to configure the driver.

1. Launch the installer for your version of Windows.
2. Click **Next** to continue.

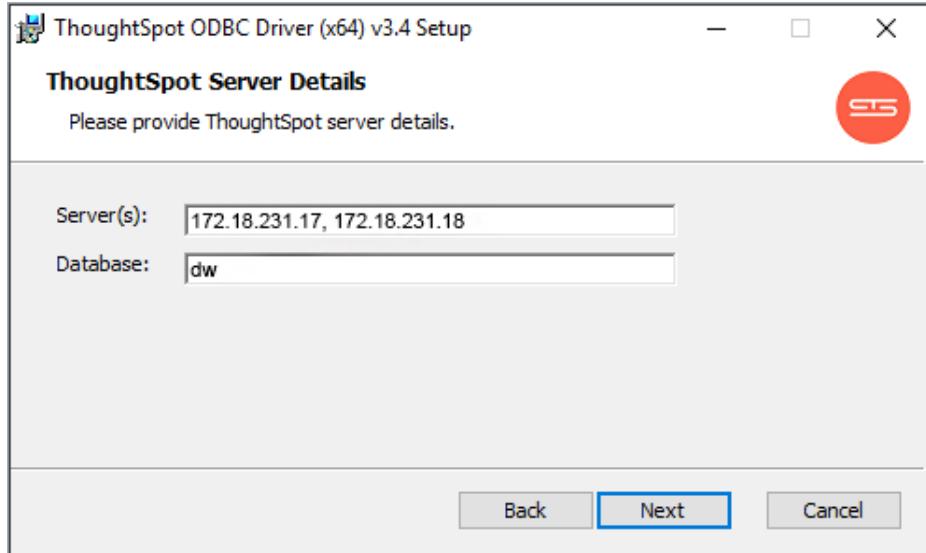


3. Accept the End User License Agreement (EULA), and click **Next**.

4. Specify the destination folder where the driver will be installed.



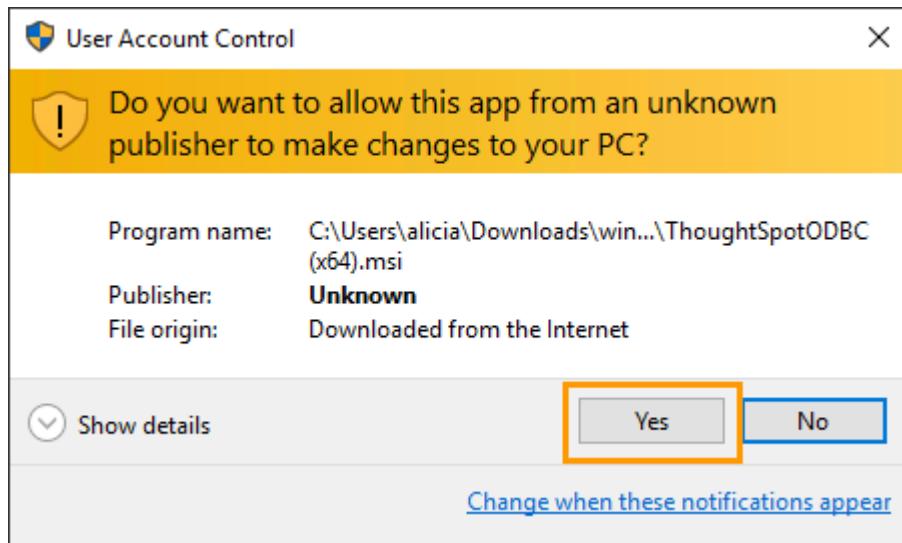
5. Enter the ThoughtSpot server details, and click **Next**.



- For **Server(s)**, provide a comma separated list of the IP addresses of each node on the ThoughtSpot instance.
- For **Database**, optionally specify the database to use. If you skip this entry, you must provide the database each time you connect using ODBC.

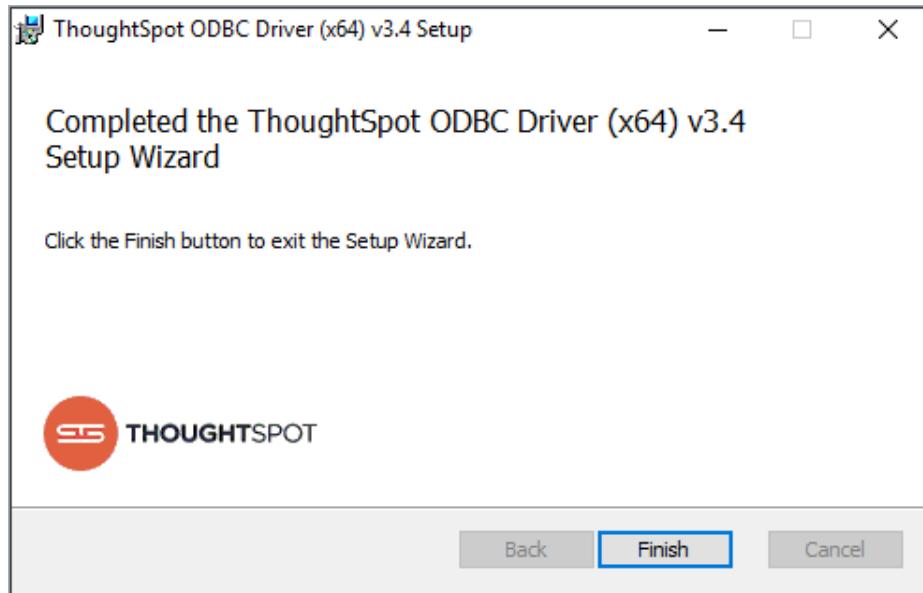
6. Confirm that the install can begin by clicking **Install**.

7. You may see a security warning.



8. Click **Yes** to continue.

A confirmation message appears when the installation is complete.



9. Click **Finish**.

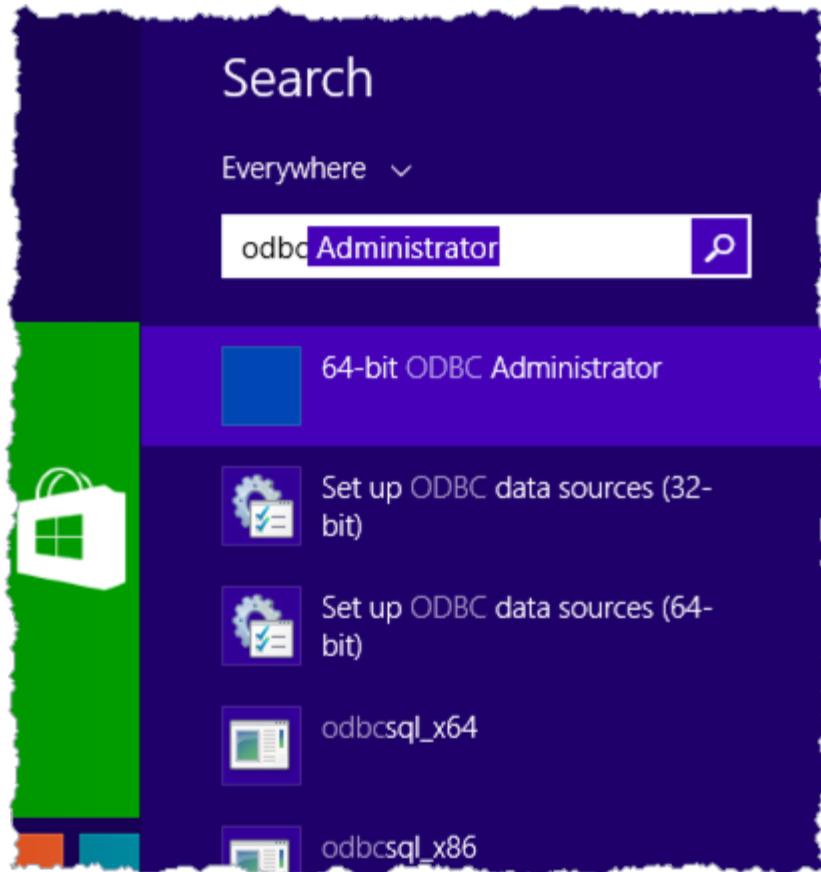
Configure the driver and test your connection

After installation completes, use the ODBC Administrator to configure the ODBC connection on your Windows workstation. For example, you may want to add a default schema or change the server IP address or the default database.

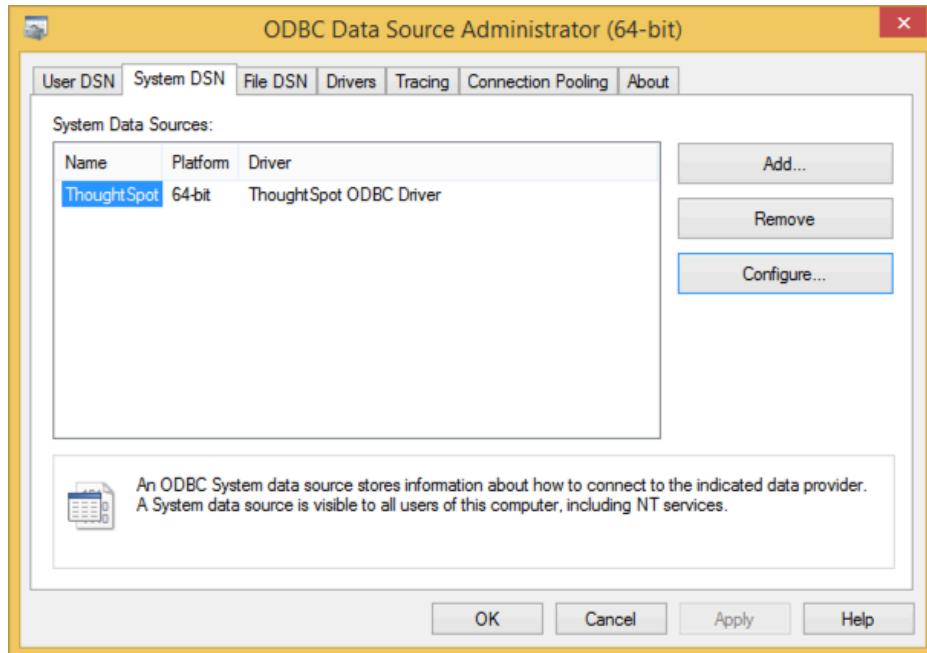
It is recommended to add a default schema. If you don't specify a default schema, you must supply it every time you use the ODBC driver.

At this point, you can test your ODBC connection to ThoughtSpot. It is important to recall that the username/password you use belongs to a ThoughtSpot application user. Typically, this user is a user with data management or administrative privileges on the application.

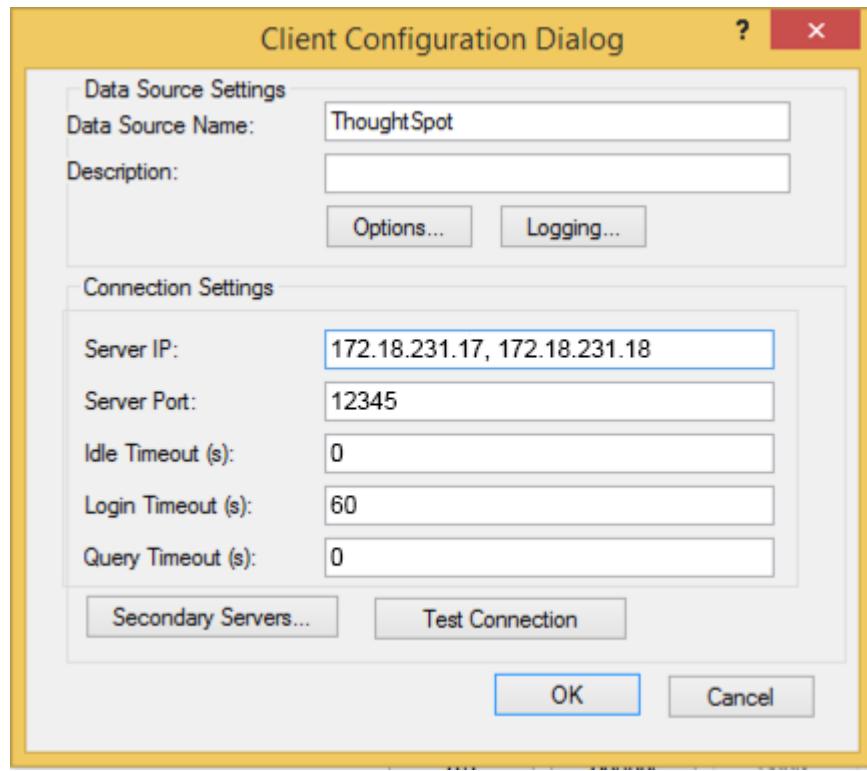
1. Before trying the ODBC connection, confirm a username/password that can log into the ThoughtSpot applications.
2. Click the **Data** tab, and confirm the user's privileges.
3. Return to your workstation.
4. Locate and open the **ODBC Data Source Administrator (64-bit)** application.



5. Click the **System DSN** tab.



6. Select **ThoughtSpot** and click **Configure...**



7. Click **Options...**

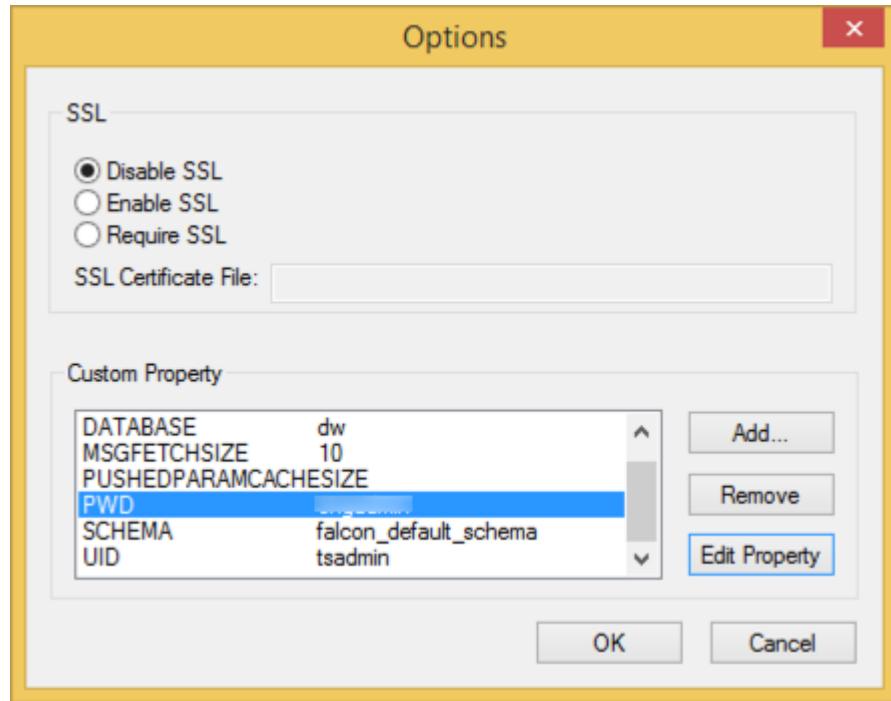
8. Ensure you have the following **Custom Property** values set:

Custom Property	Value
SCHEMA	falcon_default_schema is the default
UID	The username of a user with data management privilege.
PWD	The password for the username you specify.

You don't have to use the ThoughtSpot default schema. You can specify your own. We recommend that you define a default schema. Otherwise, you must supply a schema every time you use the ODBC driver. Moreover, without a schema (or if the schema is not present), the ODBC driver returns an error that states that the schema could not be found.

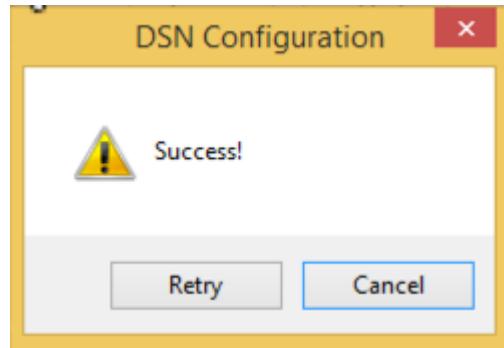
Similarly, adding the `UID` and `PWD` properties are not required. If you don't add them, you are prompted to supply them each time you connect.

When you are done, your options should look similar to the following:



9. When you are done, click **OK** to save your new properties.

10. Click **Test Connection** to test your database connection.



11. Click **Cancel** to close the **DSN Configuration** dialog.

12. Click **OK** to close the **Client Configuration Dialog** the dialog.

13. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application.

Now, you are ready to begin using the connection you've configured.

Related information

- [Enable ODBC logs.](#)
- [Configure multiple connections on Windows.](#)

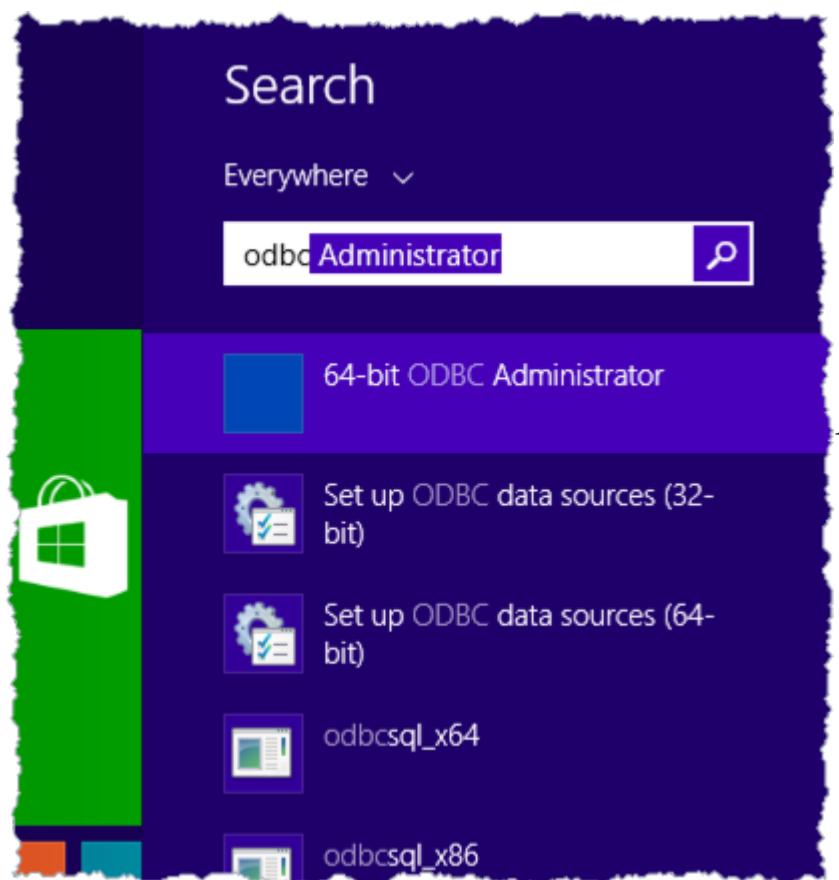
Configure multiple connections on Windows

Summary: You can add multiple ODBC data sources.

Use this procedure if you want to add an additional data source after creating a [single source succeeds](#).

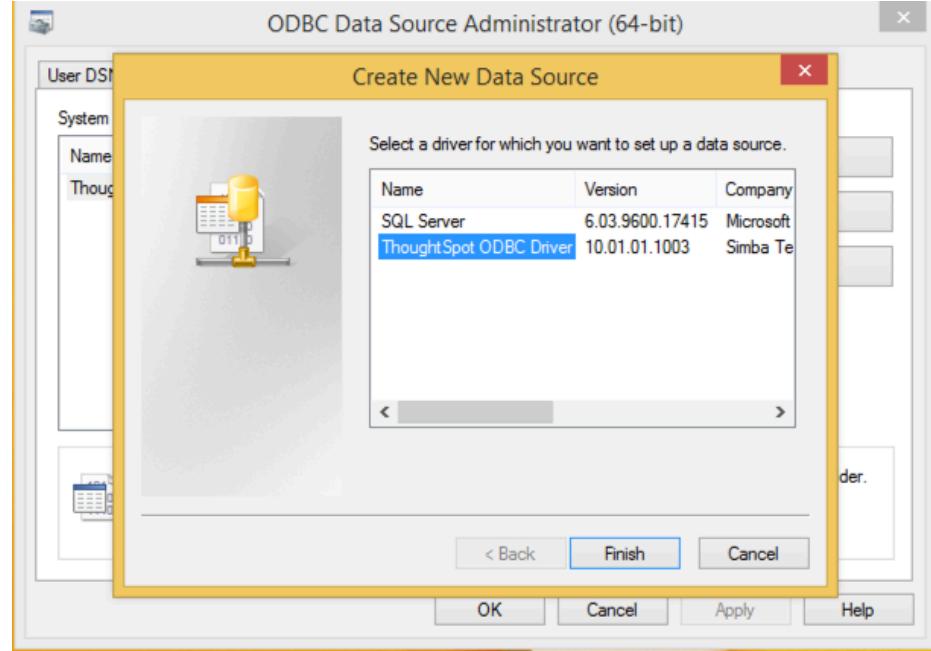
The main reason for needing to set up multiple ThoughtSpot ODBC data sources is that you have a production cluster and a test or development cluster.

1. Locate and open the **ODBC Data Source Administrator (64-bit)** application.



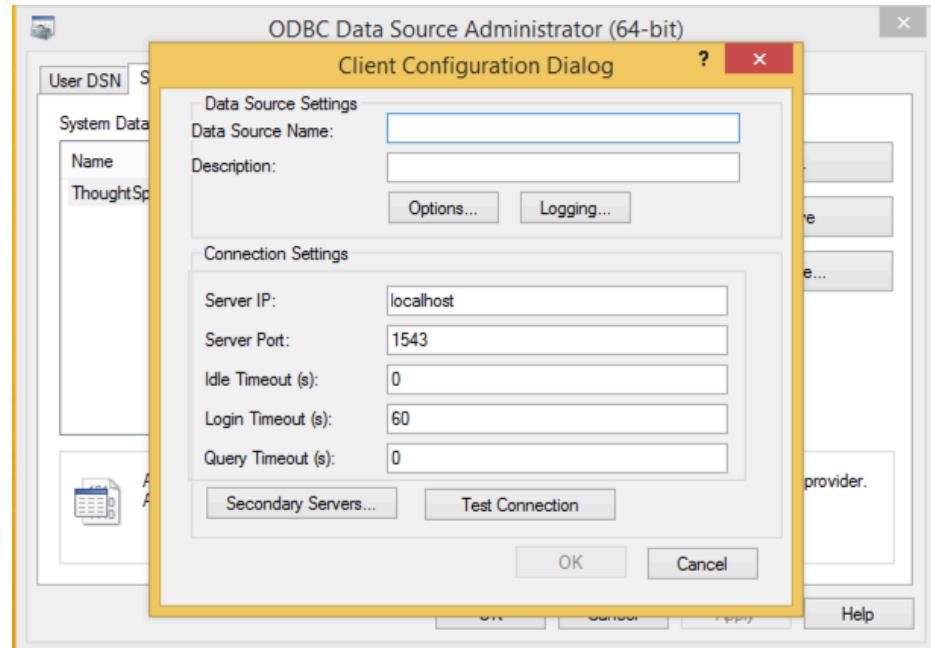
2. Click the **System DSN** tab.
3. Select **Add**.

The system lists the available drivers.



4. Choose the **ThoughtSpot ODBC Driver** and click **Finish**.

The system displays the **Client Configuration Dialog** dialog.

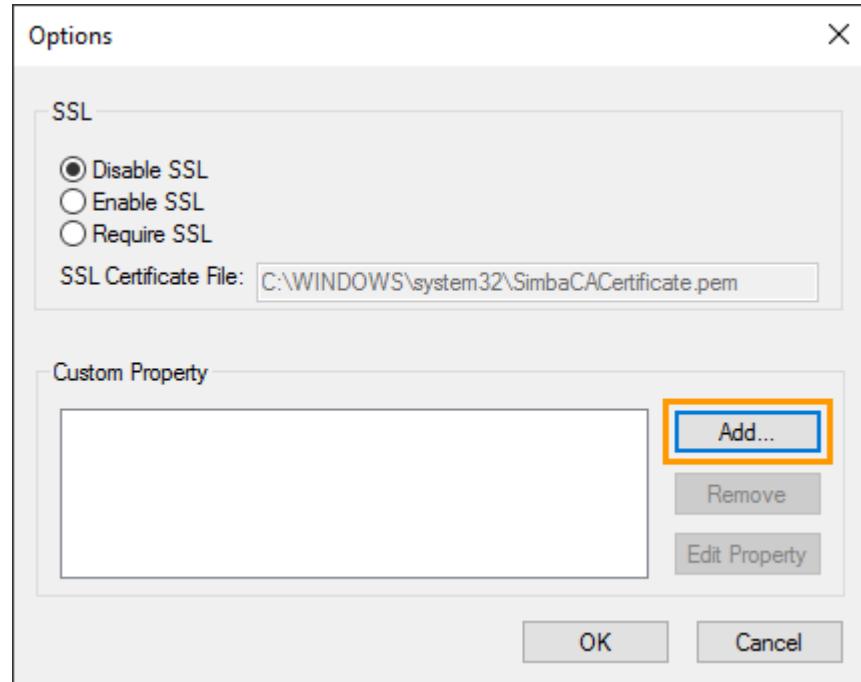


5. Enter your data source configuration.

Configuration Property	Value
Data Source Name	The name you want to call the data source.
Description	A description of the data source.
Server IP	A list of the IP addresses for each node, separated by commas.
Server Port	12345
Idle Timeout	Time in seconds after which an idle ODBC connection times out.
Login Timeout	Time in seconds after which a login request times out.
Query Timeout	Time in seconds after which a query times out.

6. Configure custom properties by clicking **Options**.

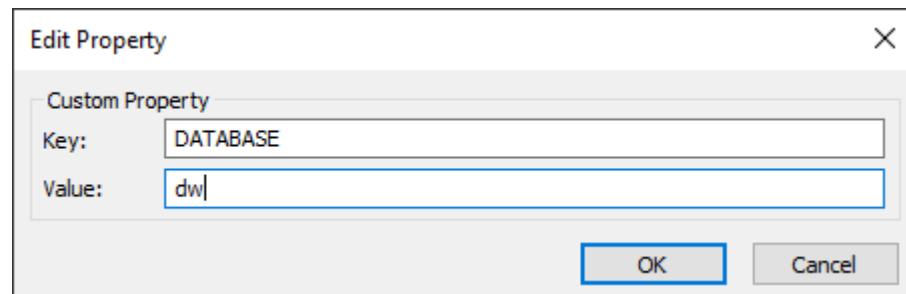
The system displays the **Options** dialog.



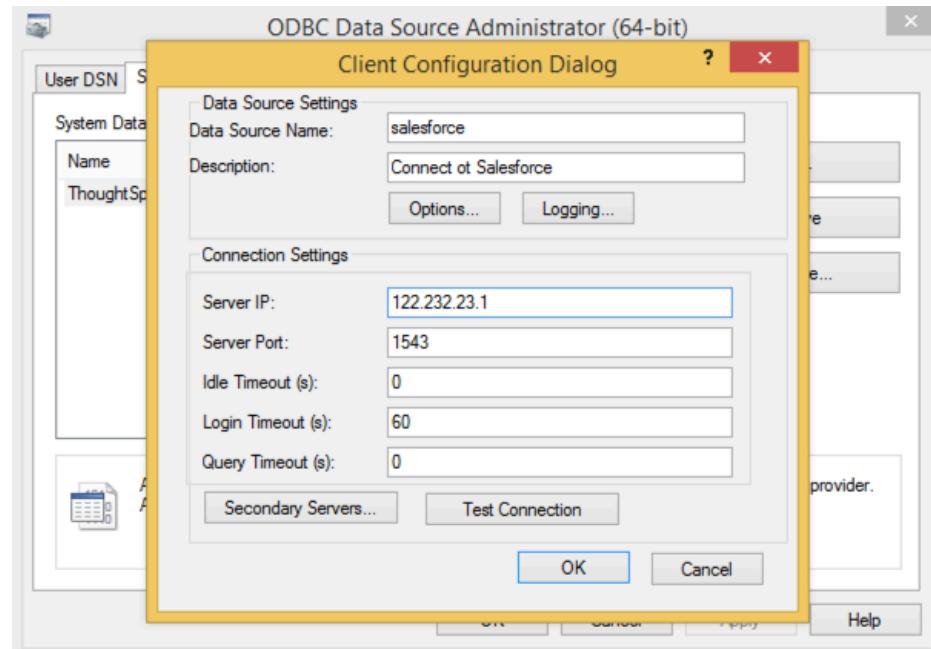
7. Add these properties using the **Add** to enter an option and click **OK** after to save an option.

Option	Value
DATABASE	The default database to connect to.
SCHEMA	The default schema to connect to. Use <code>falcon_default_schema</code> if you aren't sure.
CONNECTIONTIMEOUT	Optional. Seconds before an idle connection times out.

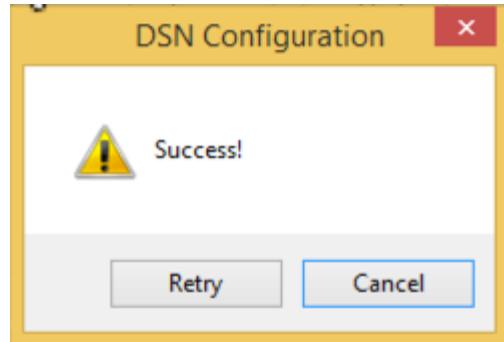
The key must be defined exactly as it appears here, using all capital letters. You can find other supported properties in [ODBC and JDBC configuration properties](#).



- When you are done, click **OK** to save your new configuration.



- Click **Test Connection** to test your database connection.



If your test connection fails, enable ODBC logging to troubleshoot.

10. Click **Cancel** to close the **DSN Configuration** dialog.
11. Click **OK** to close the **Client Configuration Dialog** the dialog.
12. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application

Deploy SSL with ODBC on Windows

You can configure a secure ODBC connection between your ThoughtSpot cluster and a remote Windows Machine. This article explains the SSL resources and ODBC configuration options you need to enable SSL for an ODBC connection.

Prerequisites

Before configuring SSL over the ThoughtSpot ODBC connection, make sure that your system administrator has created and configured your network's Certificate Authority. Additionally, the system administrator should have available both the proper Private Key and Server Certificate.

Configure the ThoughtSpot cluster nodes

⚠ Important: Portions of this procedure require that you work with your ThoughtSpot Customer Service or Support Engineer.

The [SimbaServer Configuration Properties reference](#) includes full details on [SSL Configuration Properties](#).

Before you change your ODBC configuration, decide on a path where you will store the Private Key and Server Certificate, for example, you could decide to use `/home/admin/Simba_SSL/` as the path.

Then, do the following on *every ThoughtSpot node* in your cluster.

1. Create the path on the node.
2. Copy the SSL certificate and private key to this path.
3. Edit the node's `/etc/thoughtspot/simba.ini` file (Simba server configuration) with your favorite editor.
4. Add the following lines:

```
SslCertfile=/home/admin/Simba_SSL/Server-Certificate.pem  
SslKeyfile=/home/admin/Simba_SSL/Private-Key.pem  
UseSsl=Required
```

5. Restart the Simba service.

You must work with your ThoughtSpot Customer Success or Support Engineer to do this.

Deploy the certificate on your windows workstation

Please note that the SSL settings on the server and client are interdependent.

The [SimbaClient for ODBC Configuration Properties](#) reference describes how to set parameters on the client to use SSL (scroll down to useSsl section at the end). The Simba documentation also provides a chart showing [configuration properties for SSL](#) where you can see how different combinations of SSL settings on client and server will behave. For example:

- Setting both server and client to `UseSsl=Enabled` provides the ability for clients to connect with or without SSL.
- Setting both server and client to `UseSsl=Required` requires that all clients use SSL.

Note: Note that the SSL and certificate parameters can be set through the pre-defined options on the options dialog, but customers have reported that these are not always reliable. In the following procedure, we recommend using custom properties to define these settings (either preemptively, or as a solution if the ODBC connection over SSL does not work with the pre-defined options). There is no harm in setting both. Example settings are: `UseSSL = Required` and `SslCACertfile = C:\ODBC-SSL\CA.pem`

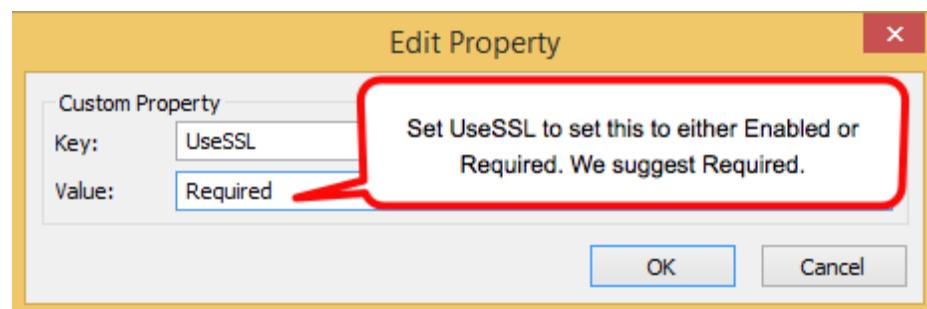
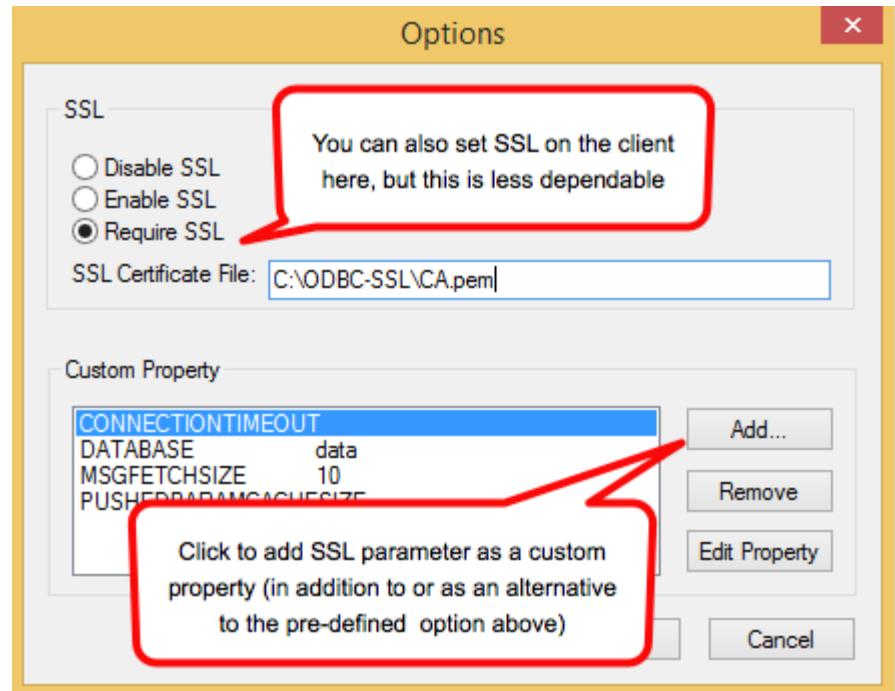
On the workstation you want to use for your ODBC connection, specify the level of SSL you want to use on the client along with the path to the CA certificate, and then test the connection.

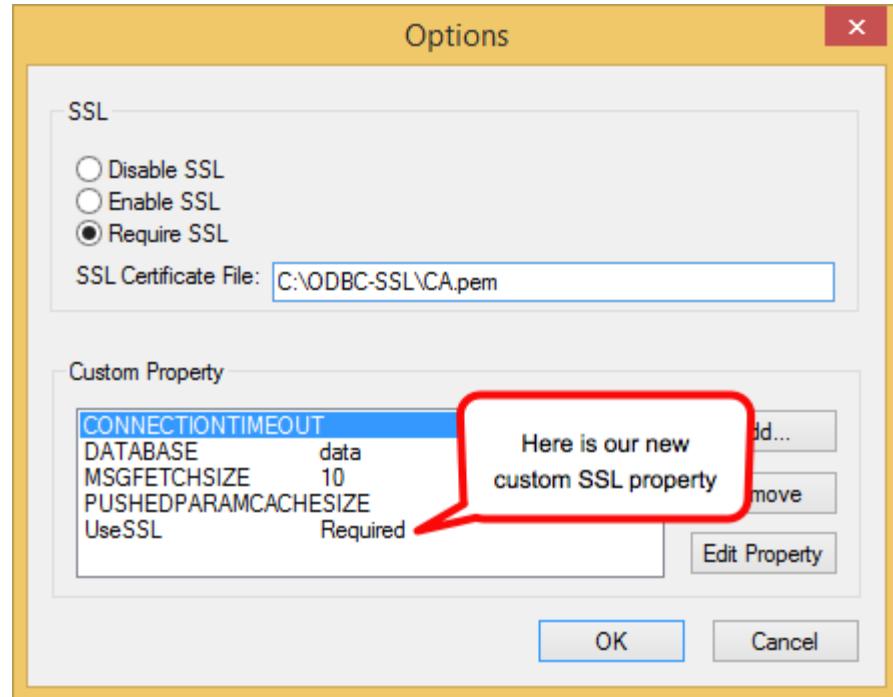
1. Save the CA certificate to a secure location on the workstation disk.

Choose a location where the certificate is unlikely to be deleted by mistake, for example,

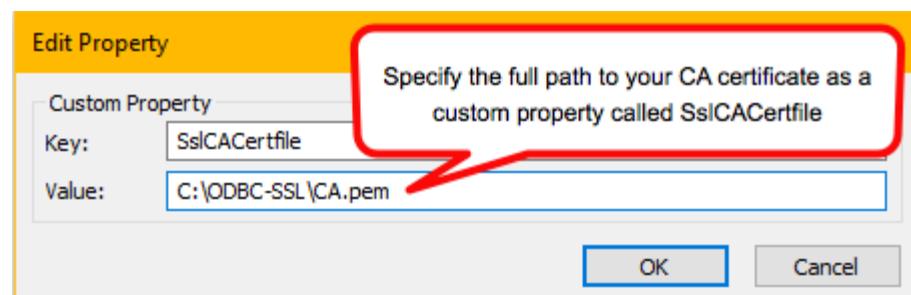
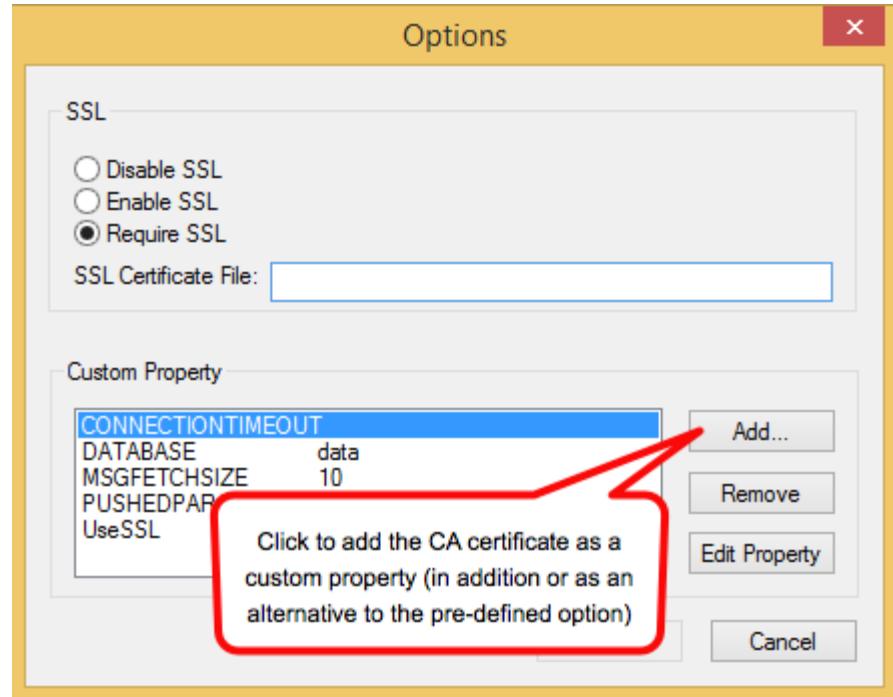
`C:\ODBC-SSL\CA.pem` is an example of a full path to such a location.

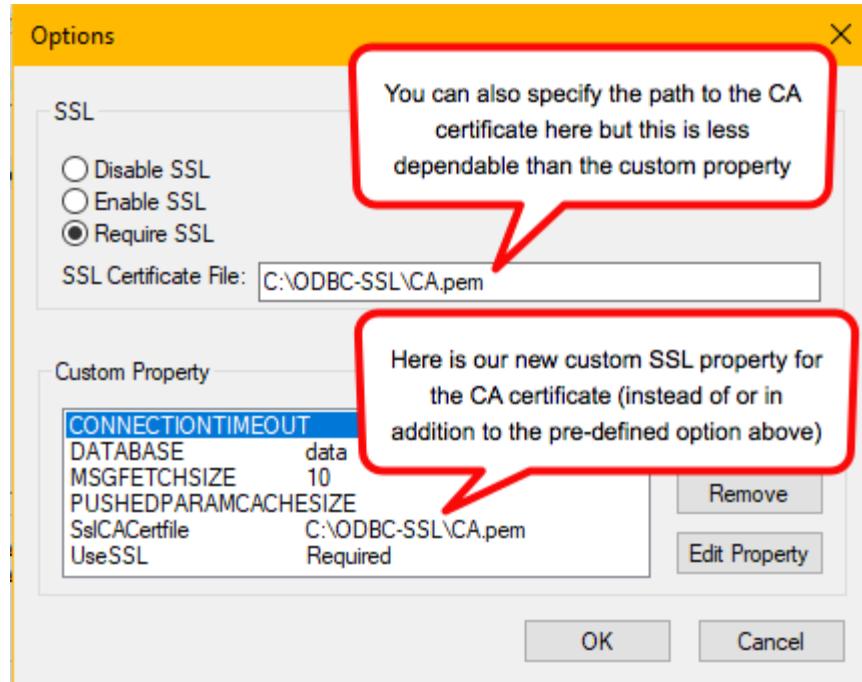
2. Open your ThoughtSpot ODBC connection configuration dialog.
3. Click **Options**.
4. Check the **Require SSL** option and/or add SSL as a custom property.



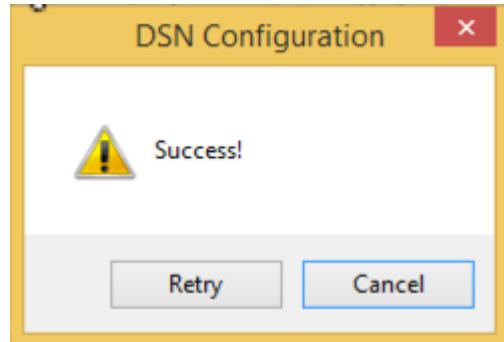


5. Enter the location of the CA certificate in the **SSL Certificate File** field and/or add the CA certificate as a custom property. Be sure to provide the full path to the certificate (`{certificate_directory}\{CA_certificate}.pem`).





6. When you are done, click **OK** to save your new properties.
7. Click **Test Connection** to test your database connection.



8. Click **Cancel** to close the configuration dialog.
9. Click **OK** to close the **Client Configuration Dialog** the dialog.
10. Click **OK** to close the **ODBC Data Source Administrator (64-bit)** application.

Set up the ODBC Driver for SSIS

Summary: Use SSIS to set up the ODBC Driver.

Microsoft SSIS (SQL Server Integration Services) is a data integration and workflow applications platform you can use to connect to ThoughtSpot. The platform is a component of the Microsoft SQL Server database software.

You can use a SSIS connection to perform data migration tasks. Its data warehousing tool is useful for data ETL (extraction, transformation, and loading). The SSIS Import/Export Wizard creates packages that transfers data with no transformations. It can move data from a variety of source types to a variety of destination types, including text files and other SQL Server instances.

Use SSIS to set up the ODBC Driver by creating a connection manager. This manager connects an OLE DB Source and the ODBC Destination.

Prerequisites

On Windows 64-bit, you have to install both the 32-bit and 64-bit ThoughtSpot ODBC drivers. In addition, they must be named the same, such as ThoughtSpot. By default they are named ThoughtSpot-32 and ThoughtSpot-64. This is required because the 64-bit SSIS shows a list of 32-bit ODBC drivers when you configure an ODBC target. However, it executes the 64-bit driver. If the drivers aren't named the same, then you can get an error stating the driver doesn't exist.

Set up the driver

To set up the ODBC driver using SSIS:

1. Open your SQL Server visual development tool that is based on Microsoft Visual Studio.
2. Select **OLE DB Source**, and click **New**.

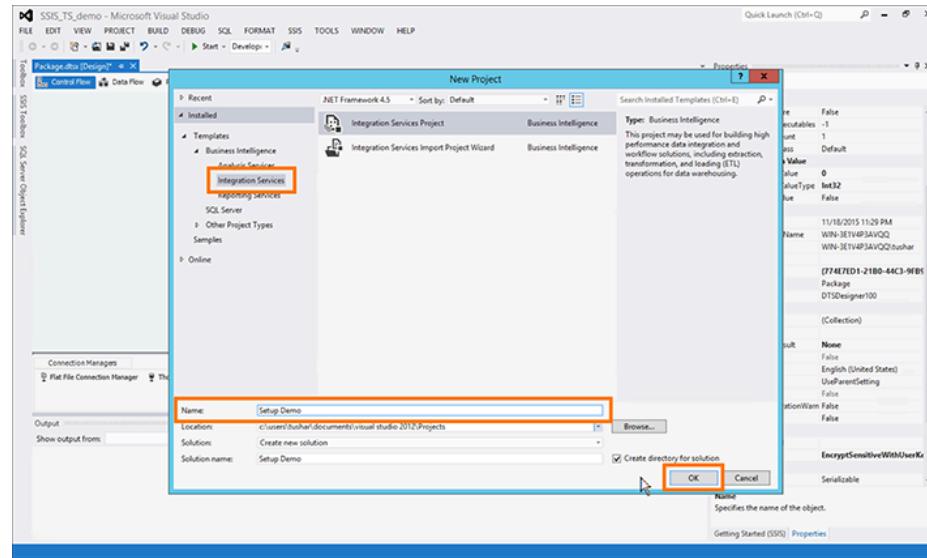
Where ODBC provides access only to relational databases, OLE DB provides access to data regardless of its format or location.

3. Add the server by name from the machine accessible list.

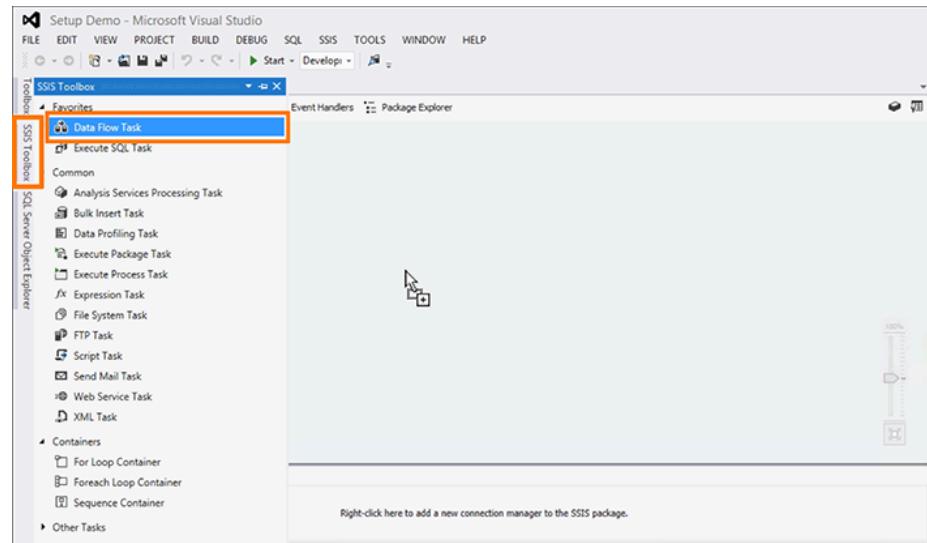
- Enter the authentication information: db name, user name, password, and test connection.

You can add the UID and password by clicking on **Options**.

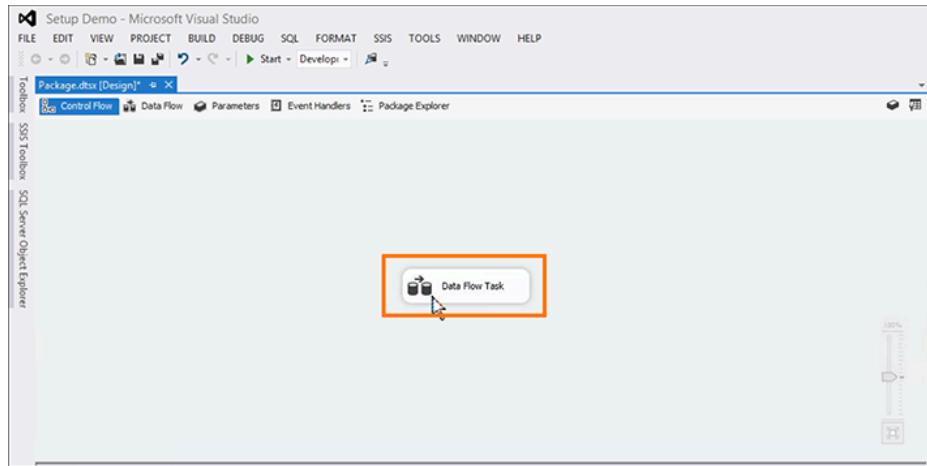
- Click **File** and select **New**, then **Project**.
- Select the **Integration Services** tab under **Installed > Templates > Business Intelligence**.
- Enter a name in the **Name** field and click **OK**.



- Select the **SSIS Toolbox** tab on the left hand side of the platform, and drag and drop **Data Flow Task** to the main window.



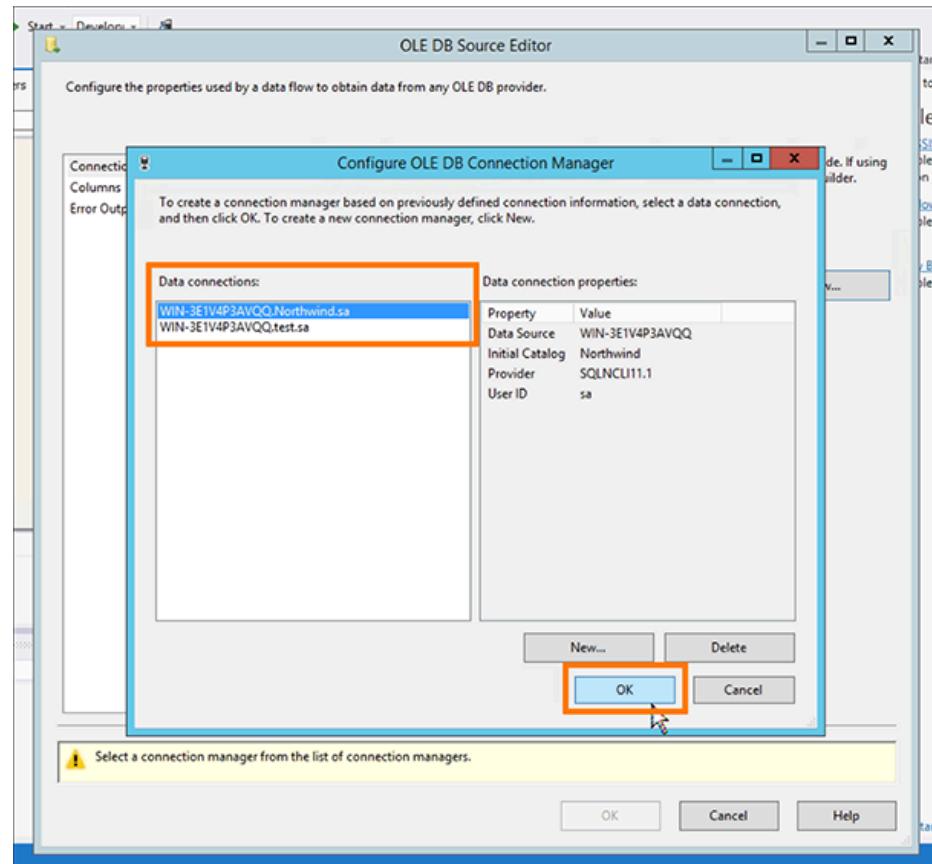
9. Double click the **Data Flow Task** icon when it appears in the center of the page.



10. Navigate back to the **SSIS Toolbox** tab. You now want to create sources and destinations.

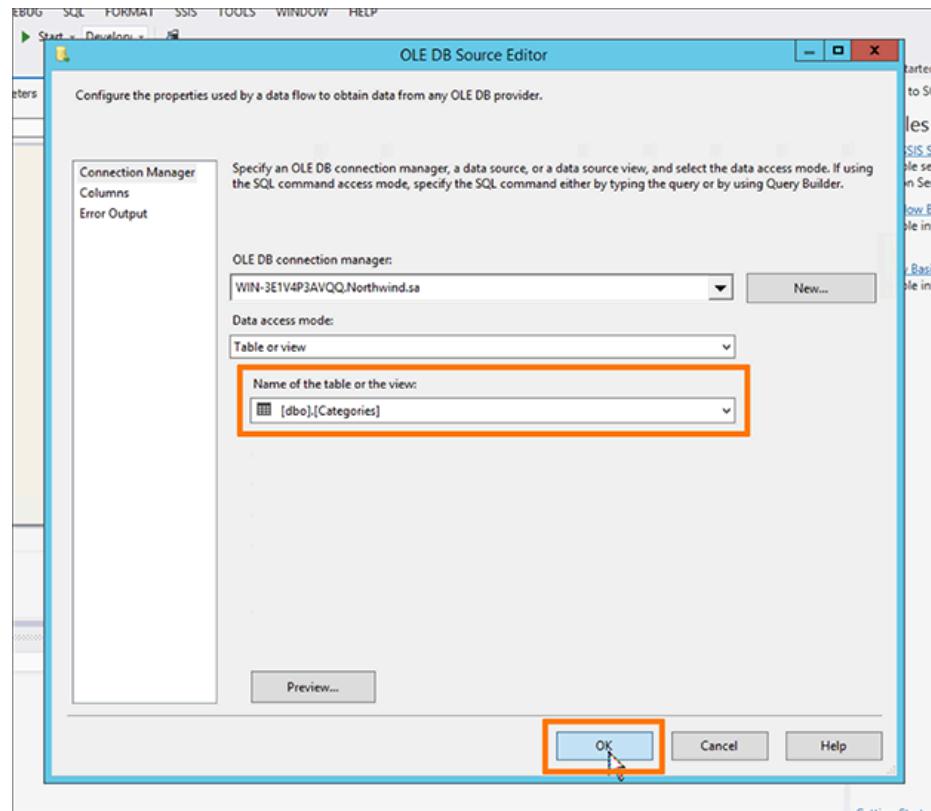
Create sources and destinations

1. Under **Other Sources**, find **OLE DB Source** and drag and drop it to the main window.
2. Double click the **OLE DB Source** icon when it appears in the center of the page to open the OLE DB Source Editor.
3. Select a new OLE DB connection manager by clicking **New**.
4. In the Configure OLE DB Connection Manager window, select your **Data connection** and click **OK**.



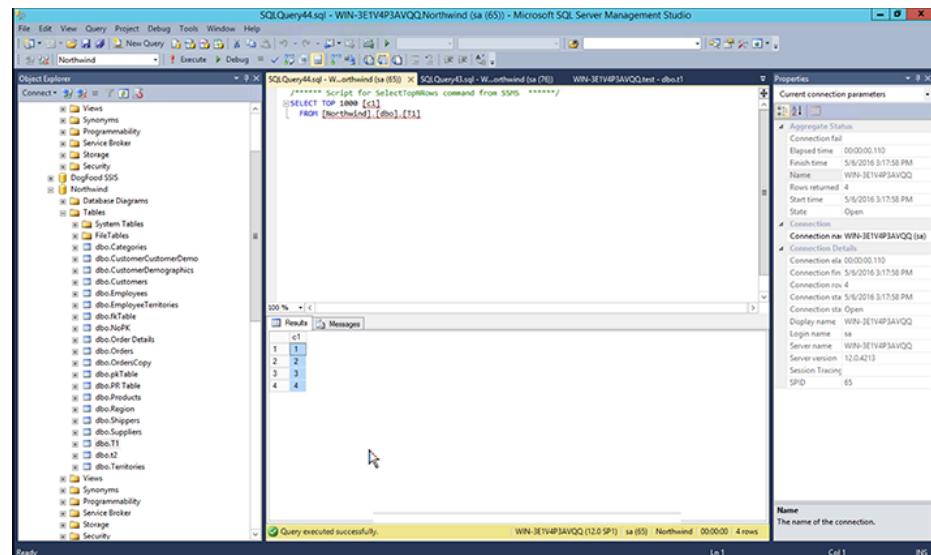
If you do not see your data connection, you will have to create a new one in the Connection Manager by clicking **New**.

5. Back in the OLE DB Source Editor, select the **Name of the table or the view**, and click **OK**.



6. Select the table, and see what columns are in it.

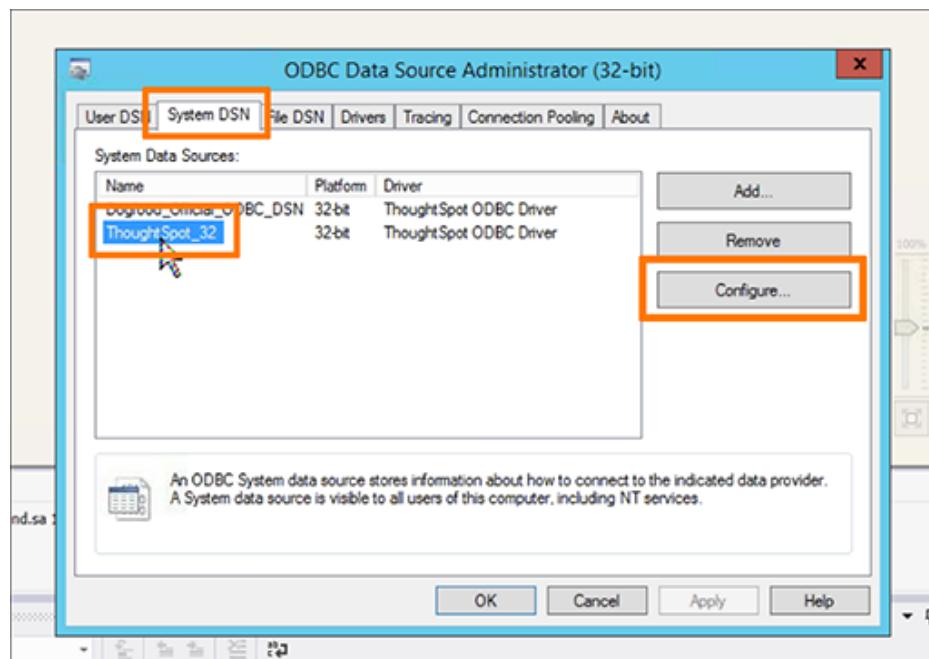
In this example, a single column, `c1`, is selected.



Configure the ODBC Data Source Administrator

The ODBC Data Source Administrator has to be configured to connect to ThoughtSpot and bring the table in.

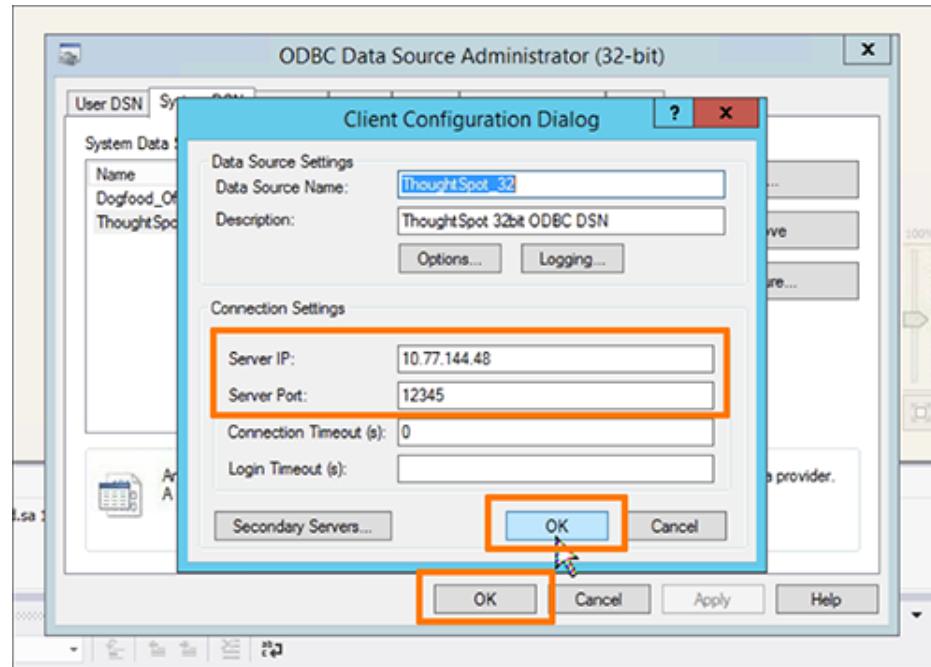
1. Search for and open your **ODBC Data Sources (32-bit)** program.
2. Click the **System DSN** tab and select **ThoughtSpot_32**.
3. Click **Configure**.



4. In the Client Configuration Dialog, enter the **Server IP** and **Server Port**.

Enter any node IP that has Simba server running on it. In **Secondary Servers**, you must specify all node IPs, because ThoughtSpot must resolve to the server Simba runs on, and that server can change after an upgrade. Enter one server IP per line. The line return serves as a separator. Comma separated values are not supported.

5. Click **OK** twice to close the Client Configuration Dialog and the ODBC Data Source Administrator.

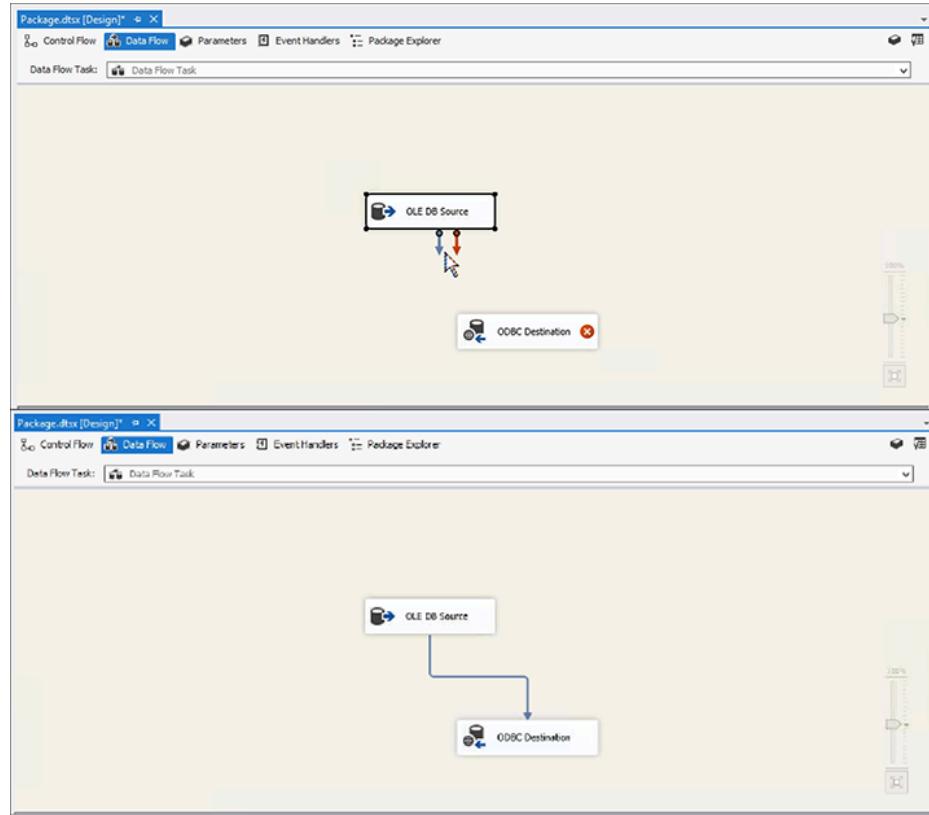


Create a file to take the feed

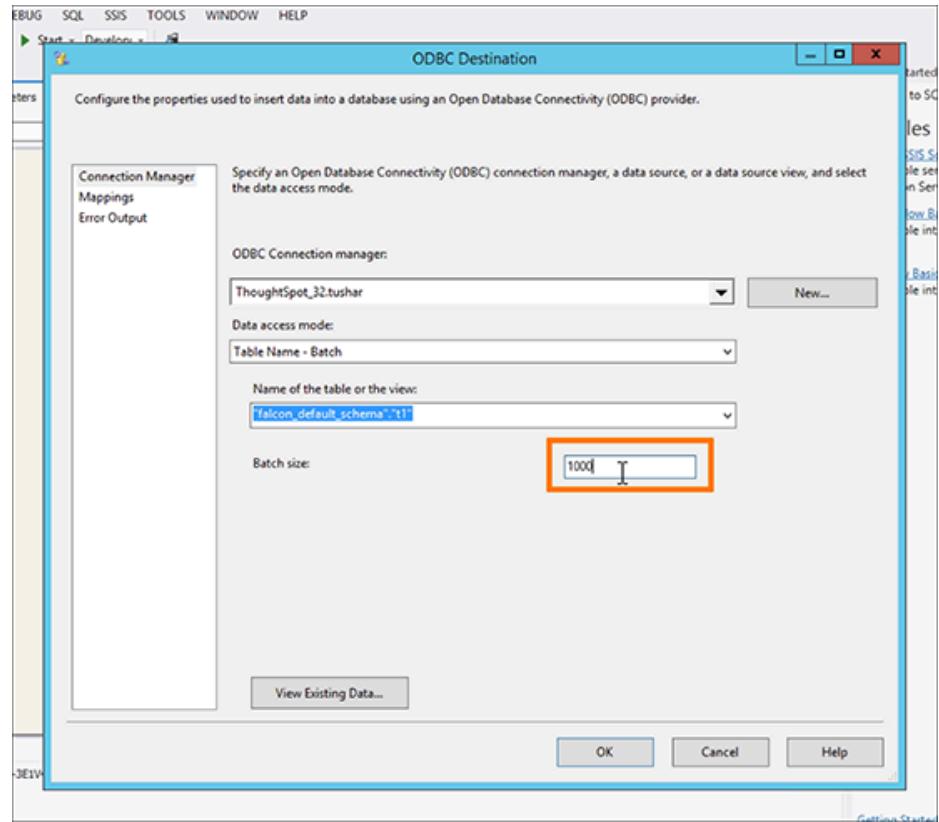
Now that you have set up your source, create the empty table in ThoughtSpot to take this feed. SSIS does not allow you to create the table in ThoughtSpot. You have to do this first in TSQL. In Pentaho, it will create the table in ThoughtSpot, but not in SSIS.

Create the ODBC Destination. Use the one you created and named in the ODBC Data Source Administrator.

1. In the **SSIS Toolbox** tab, under **Other Destinations**, drag and drop **ODBC Destination** to the main window.
2. Drag the **blue arrow** to connect the OLE DB Source icon to the ODBC Destination icon.
3. Double click the **ODBC Destination** icon.



4. Use ODBC Destination to set the **Batch size** for the connection in the Connection Manager tab. You can set the size to be up to 10,000.



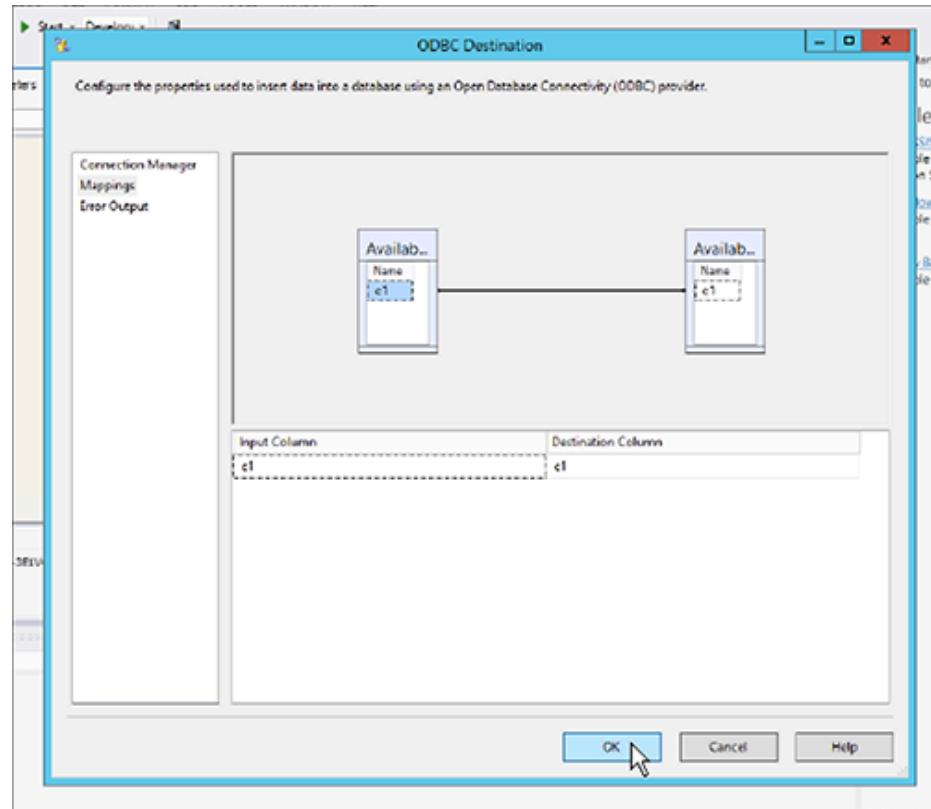
If the load fails, the entire batch will be lost, and you will have to start that load over again.

5. Set the **Transaction Size** to match the total number of rows that are expected to be loaded in the load cycle.

Your transaction size can be quite large—even spanning a million rows. However, too many small batches can leave the cluster in a rough state. This is because each batch acts as a separate transaction and creates a separate commit. Too many of these will slow down our system since each transaction creates a “data version” in our system. In Pentaho, the transaction size setting is called Commit Size.

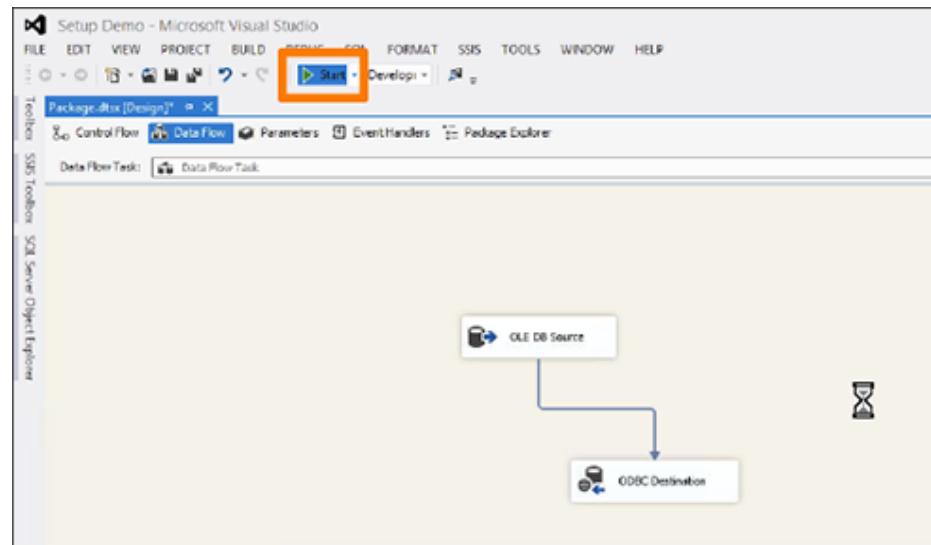
6. Set the **Transaction Option** attribute of the Data Flow Task to **Supported**.
7. In the **Mappings** tab, validate the mapping or change it.

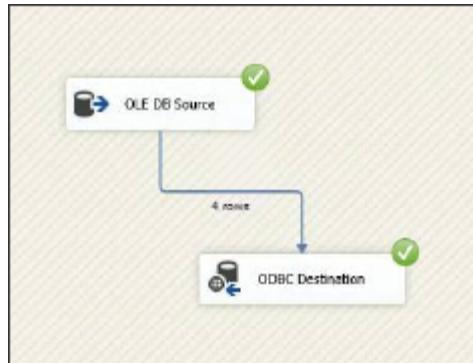
You can have different column names in each database if you map them. Of course, they must be of the same or compatible datatype.



8. Start the import job by clicking the **Start** button.

You should see an animation indicating that the data is transferring over. When the import is complete, the number of successfully transferred rows is displayed.





You can validate the import using TSQL or from the **Data** screen.

Install the ODBC Driver on Linux

Summary: Use this procedure to obtain the Linux ODBC driver and install it.

ThoughtSpot's ODBC connection relies on the [SimbaEngine X SDK](#) to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the Simba ODBC driver on a Linux workstation.

Make sure you have read the overview material in the [ODBC driver overview](#). This workstation is the same machine where you plan to run your ETL activities.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls  
172.18.231.17  
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Install the Simba client

On your workstation, where you want to connect from, do the following to get the ODBC driver:

1. Open a browser on your workstation.

2. Navigate to the [Downloads](#) page.
3. Click **ODBC Driver for Linux** to download the driver.
4. Open a terminal on your workstation.
5. Change directory to the location where you downloaded the file.
6. Optionally, move the file to a permanent location on your machine.

When you expand the downloaded file it will create a directory in the location.

7. Unzip the zip file:

```
gunzip ThoughtSpot_linux_odbc_<version>.tar.gz
```

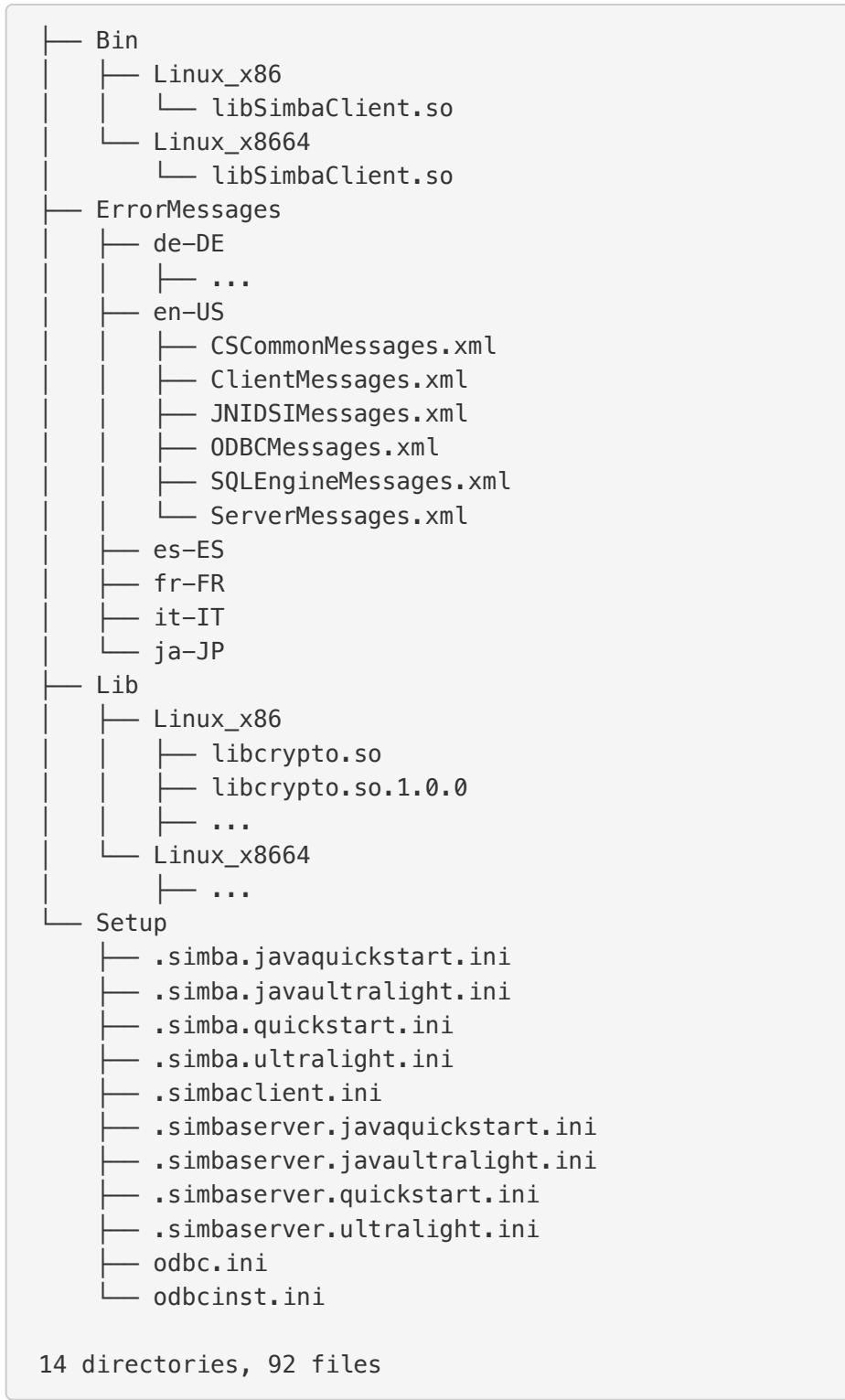
8. Extract the contents of the `tar` file.

```
tar -xvf ThoughtSpot_linux_odbc_<version>.tar
```

This extracts a subdirectory called `linux` into the current directory.

9. Take a moment to examine the contents of the new directory.

The structure contains a Simba client library, supporting libraries and setup files for two different architectures. It also continues error messages for multiple languages.



The `linux/Setup` directory contains the key ODBC configuration files and sample Simba client configurations you can use later in this procedure.

10. You must know your workstation architecture to continue, confirm your workstation's architecture.

You can use the `arch` or the `uname` command or both.

```
$ arch
x86_64
$ uname -a
Linux nebuladocs-production-4vfnv 4.4.108-1.el7.elre
po.x86_64 #1 SMP Mon Dec 25 09:55:39 EST 2017 x86_64 x8
6_64 x86_64 GNU/Linux
```

In previous examples, the workstation is a 64 bit workstation. Your workstation may be 32-bit.

You can use this architecture information in the procedures that follow.

(Optional) Install unixODBC tools for testing

The procedures on this page rely on the unixODBC tools to test your configuration and connection. If you are experienced with ODBC and want to skip this, you can. Simply substitute your preferred mechanism in the subsequent procedures where references are made to the unixODBC tools.

⚠ Warning: Your ThoughtSpot installation contains a version of the unixODBC tools. These tools are incompatible with CentOS. Do not use these tools if you are performing this procedure on your ThoughtSpot server.

1. Search for the unixODBC tools on your system.

The `yum` package manager searches for software already installed or available on your system or from the configured repositories. Depending on your workstation configuration, you may need to use the `sudo` command with your workstation.

```
$ yum search unixODBC
...
* updates: repos-lax.psychz.net
=====
N/S matched: unixODBC
=====
opensips-unixodbc.x86_64 : OpenSIPS unixODBC Storage support
unixODBC-devel.i686 : Development files for programs which will use the unixODBC library
unixODBC-devel.x86_64 : Development files for programs which will use the unixODBC library
erlang-odbc.x86_64 : A library for unixODBC support in Erlang
freeradius-unixODBC.x86_64 : Unix ODBC support for freeradius
unixODBC.i686 : A complete ODBC driver manager for Linux
unixODBC.x86_64 : A complete ODBC driver manager for Linux
```

Make note of the correct package to install for your architecture.

2. Install the appropriate package for your architecture.

In this case the command installs the tools for a 64-bit architecture. A 32-bit package needs the `unixODBC.i686` package.

```
[admin@nebula-docs-odbc-test-cxmrn ~]$ yum install unixODBC.x86_64
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: mirror.linuxfix.com
 * elrepo: repos.lax-noc.com
 * epel: mirror.hmc.edu
 * extras: centos-distro.cavecreek.net
 * rpmforge: mirror.lstn.net
 * updates: repos-lax.psychz.net
Resolving Dependencies
--> Running transaction check
--> Package unixODBC.x86_64 0:2.3.1-11.el7 will be installed
...
Complete!
```

3. Verify the files were installed.

```
$ ls /usr/bin/isql  
/usr/bin/isql  
$ ls /usr/bin/odbcinst  
/usr/bin/odbcinst
```

Set up your environment

In this section, you set parameters in your workstation to support your ODBC connection.

1. Copy the library for your architecture from the `Lib` directory on your Linux machine.

Library	Architecture
/linux/Lib/Linux_x86	32-bit
/linux/Lib/Linux_x8664	64-bit

2. Add the location's path to the `LD_LIBRARY_PATH` environment variable.

For example if your architecture is 64-bit and you keep the library in your `home` directory:

```
export LD_LIBRARY_PATH=~/linux/Lib/Linux_x8664/
```

3. Use the `echo` command to verify the path was added correctly.

```
echo $LD_LIBRARY_PATH
```

4. Copy the `odbc.ini` file to the `/etc` directory.

```
$ cp ~/linux/Setup/odbc.ini /etc
```

If you have trouble making the copy, use the `sudo` command to make the move.

5. Copy the `odbcinst.ini` file to the `/etc` directory.

```
$ cp ~/linux/Setup/odbcinst.ini /etc
```

6. Copy the hidden `.simba.quickstart.ini` file to the `/etc` directory, renaming it in the process to `simbaclient.ini`.

```
$ cp ~/linux/Setup/.simba.quickstart.ini /etc/simbaclient.ini
```

7. Update your environment with the `ODBCSYSINI` and `ODBCINI` variables.

```
$ export ODBCSYSINI=/etc/  
$ export ODBCINI=/etc/odbc.ini
```

8. Use the `/usr/bin/odbcinst` command to confirm your settings:

```
$ /usr/bin/odbcinst -j  
unixODBC 2.3.1  
DRIVERS.....: /etc/odbcinst.ini  
SYSTEM DATA SOURCES: /etc/odbc.ini  
FILE DATA SOURCES..: /etc/ODBCDataSources  
USER DATA SOURCES..: /etc/odbc.ini  
SQLULEN Size.....: 8  
SQLLEN Size.....: 8  
SQLSETPOSIROW Size.: 8
```

Edit the `/etc/simbaclient.ini` file

When you are ready, follow this procedure to configure the driver.

1. Edit the `/etc/simbaclient.ini` file with your favorite editor.

2. Change the `ErrorMessagesPath` property to point to the location where you unzipped the client.

```
[Driver]
ErrorMessagesPath=<path_to_error_messages_directory>
```

3. Comment out the `# Generic ODBCInstLib` value.
4. Uncomment the `ODBCInstLib` property.

When you are done, your file looks like the following:

```
# Generic ODBCInstLib
#   iODBC
#ODBCInstLib=libiodbcinst.so

#   SimbaDM / unixODBC
ODBCInstLib=libodbcinst.so
```

5. Save and close the `/etc/simbaclient.ini` file.

Edit the odbcinst.ini file

The `odbcinst.ini` file is a registry and configuration file for ODBC drivers. Depending on your workstation architecture, you configure the 32-bit or 64-bit driver.

1. Open the file `/etc/odbcinst.ini` in your favorite editor.
2. Comment out the driver that you don't need.

For example, if you are using 64-bit, comment out 32-bit.

3. Edit the `Driver` line so that it contains the path to the file `libSimbaClient.so`

Use the path where you copied the library files. For example, for the 64-bit ODBC driver:

```
[ThoughtSpot(x64)]  
APILevel = 1  
ConnectFunctions = YYY  
Description = ThoughtSpot 64bit ODBC driver  
Driver = /home/admin/linux/Bin/Linux_x866  
4/libSimbaClient.so  
DriverODBCVer = 03.52  
SQLLevel = 1
```

4. Make sure the remaining driver is named `ThoughtSpot` without any special characters.

When you are done, you should see something similar to the following:

```
# [ThoughtSpot]  
#APILevel = 1  
#ConnectFunctions = YYY  
#Description = ThoughtSpot 32bit ODBC driver  
#Driver = /usr/local/scaligent/toolchain/l  
ocal/simba/odbc/linux/Bin/Linux_x86/libSimbaClient.so  
#DriverODBCVer = 03.80  
#SQLLevel = 1  
  
[ThoughtSpot]  
APILevel = 1  
ConnectFunctions = YYY  
Description = ThoughtSpot 64bit ODBC driver  
Driver = /home/admin/linux/Bin/Linux_x866  
4/libSimbaClient.so  
DriverODBCVer = 03.80  
SQLLevel = 1
```

5. Save and close the `/etc/odbcinst.ini` file.

Edit the odbc.ini file

The `odbc.ini` file is a registry and configuration file for ODBC DSNs (Data Source Names). This file relies on the drivers registered in the `/etc/odbcinst.ini` file. Depending on your workstation architecture, you configure the 32-bit or 64-bit driver.

1. Open the file `/etc/odbc.ini` in the editor of your choice.

2. Comment out the configuration that you don't need.

For example, if you are using 64-bit, comment out 32-bit.

3. Locate the `Description` section for the type of Linux you are using (32-bit or 64-bit).
4. Locate the line that begins with `ServerList`.
5. Replace `127.0.0.1` with a comma separated list of the IP addresses of each node on the ThoughtSpot instance.

The syntax for the `ServerList` is:

```
ServerList = <node1_IP> 12345, <node2_IP> 12345 [, <node3_IP> 12345, ...]
```

If you need to obtain the IP addresses of the ThoughtSpot cluster nodes, run the command `tscli node ls` from a Linux shell on a ThoughtSpot appliance.

6. Do not edit the port number, leave it as `12345`.

When you are done, your entry will look similar to the following (this example is for the 64-bit ODBC driver):

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList = 172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The certificate file can be obtained by extracting the SDK tarball
#LogLevel = 0 # Set log level to enable debug logging
#LogPath = # Set the debug log files path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

7. Save and close the `odbc.ini` file.

Test your ODBC connection

At this point, you can test your ODBC connection to ThoughtSpot. It is important to recall that the username/password you use belongs to a ThoughtSpot application user. Typically, this user is a user with data management or administrative privileges on the application.

1. Before trying the ODBC connection, make sure you can use this username/password to login into the ThoughtSpot application.
2. Confirm the user's privileges by going to the **Data** tab.
3. Go back to your workstation's terminal shell.
4. Use the `/usr/bin/isql` and confirm you can connect.

Specify the `ThoughtSpot` DSN:

```
/usr/bin/isql -v ThoughtSpot tsadmin adminpwd
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
SQL>
```

Now, you are ready to begin using the connection you've configured.

Best Practices for Using ODBC

Summary: To successfully use ODBC, following these best practices is recommended.

When developing tools that use the ODBC driver, use these best practices:

- When setting up ODBC for the first time, begin by using the ThoughtSpot `tsload` for the initial data loads. This allows you to do more in-depth troubleshooting on any initial loading issues. After initial loads work properly, switch to ODBC to perform incremental loads.
- You should create the parameterized SQL statement outside of ODBC. Using this method, the SQL statement can be sent to ThoughtSpot in batches by the ODBC driver, so you only have to update the memory itself. ETL tools have this implemented already (end users shouldn't have to actually write the `INSERT` statement). But as a developer, you may be writing code that leverages the ODBC driver, so this tip can help you write your SQL for the best performance with the driver.
- Data can be loaded into a table through multiple parallel connections. You can achieve this by splitting the input data into multiple parts. Then, load those individual parts through multiple parallel connections. You can use parallel loading even while loading to a single table or multiple tables at the same time.
- When doing an incremental data load, note that the same `UPSERT` behavior that occurs in TSQL also occurs. This means that if you import a row whose primary key matches an existing row, the existing row will be updated with the new values.

Related information

- [Enable ODBC logs](#)
- [Introduction to loading and managing data](#)
- [Loading and constraints](#)

JDBC Driver Overview

Summary: Use JDBC to interact with databases in a standard manner.

Java Database Connectivity (JDBC) is a Java standard API that allows applications to interact with databases in a standard manner. ThoughtSpot has JDBC support through a JDBC driver that we provide.

Connector type

There are different types of JDBC connectors. Driver types categorize the technology used to connect to the database. The ThoughtSpot JDBC driver is a type 4 connector. It uses Java to implement a networking protocol for communicating with ThoughtSpot.

This driver is Java driver. There is no client installation or configuration.

When to use JDBC

JDBC can be used whenever you want to connect to ThoughtSpot to insert data programmatically from a Java program or application. You should begin by using the ThoughtSpot Loader for initial data loads and then use JDBC for incremental loads. This is because the ThoughtSpot Loader is generally faster than JDBC. Information on using the ThoughtSpot Loader is available in the ThoughtSpot Administrator Guide.

Version Compatibility

To ensure compatibility, always use the JDBC driver with the same version number as the ThoughtSpot instance to which you are connecting.

Performance Considerations

These are some general recommendations for maximizing the performance of JDBC:

- Insert in batches rather than doing single inserts at a time using the

`PreparedStatement::addBatch()` and `PreparedStatement::executeBatch` commands.

- If you need to upload a lot of data, consider running multiple connections with batch inserts in parallel.

ⓘ Note: The ETL tool must add a data transformation step if the source column data type does not exactly match the target's, ThoughtSpot's, column data type. The driver does not do any implicit conversions.

Use the JDBC Driver

Summary: How to configure the JDBC driver.

ThoughtSpot's ODBC connection relies on the [SimbaEngine X SDK](#) to connect through ODBC or JDBC to ThoughtSpot's remote data stores. The instructions on this page explain how to configure the JDBC driver.

The ThoughtSpot JDBC driver is supplied by a `.jar` file you install on a workstation. This workstation is the same machine where you plan to run your ETL activities.

JDBC configuration parameters

Information	Description
Driver name	<code>com.simba.client.core.jdbc4.SCJDBC4Driver</code>
Server IP address	The ThoughtSpot appliance URL or IP address.
Simba port	The simba port, which is <code>12345</code> by default.
Database name	This is not the machine login username. The ThoughtSpot Database name to connect to.
username	The name of a ThoughtSpot user with administrator permissions.
password	The password of a ThoughtSpot application user. This is not the machine or SSH userpassword.

For more JDBC configuration options, see also:

- [JDBC properties reference in this ThoughtSpot documentation](#)
- [SimbaClient for JDBC Configuration Properties reference](#)

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls  
172.18.231.17  
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports 12345`
5. Exit or close the shell.

Install the driver

The JDBC driver is a `.jar` packaged application. To use the package, you download it, install it

1. Log in to the local machine where you want to install the JDBC driver.
2. Click [Here](#) to download the JDBC driver.
3. Click **JDBC Driver** to download the file `thoughtspot_jdbc<version>.jar`.
4. Move the driver to the desired directory on your local machine.
5. Add the downloaded JDBC driver to your Java class path on the local machine.

Write your application

Using JDBC with ThoughtSpot is the same as using any other JDBC driver with any other database. You must provide the connection information, create a connection, execute statements, and close the connection.

Specify each of the nodes in the cluster in the connection string, as shown. This enables high availability for JDBC connections. To find out the nodes in the cluster, you can run the command `tscli node ls` from the Linux shell on the ThoughtSpot instance.

The format for the connection is:

```
jdbc:simba://<node1>:12345,<node2>:12345,<node3>:12345[,...];  
LoginTimeout=<seconds>;DATABASE=<db>;SCHEMA=<schema>
```

For example:

```
jdbc:simba://192.168.2.248:12345,192.168.2.249:12345,192.16  
8.2.247:12345;  
LoginTimeout=5;DATABASE=test;SCHEMA=falcon_default_s  
chema
```

As shown, the `DATABASE` and `SCHEMA` parameters need to be in all caps. For the `simba` JDBC driver to work with Spark, the `DATABASE` and `SCHEMA` must be specified in the URL. They cannot be specified as a name/value pair as a map or property. For example:

```
val tssqldf1 = sparkSession.read.format("jdbc").options(Map("ur  
l" ->  
"jdbc:simba://10.84.78.181:12345;DATABASE=movieratings;SCHEMA=f  
alcon_default_schema", "driver" ->  
"com.simba.client.core.jdbc4.SCJDBC4Driver", "dbtable" -> "Movi  
es", "user" ->  
"tsadmin", "password" -> "admin")).load()
```

This `InsertData.java` example shows how to use ThoughtSpot with JDBC. This is an example of a reference JDBC application:

```
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class InsertData {

    // JDBC class to use.
    private static final String DB_DRIVER = "com.simba.client.cor
e.jdbc4.SCJDBC4Driver";
    // jdbc_example should be an existing database.

    private static final String DB_CONNECTION = "jdbc:simba://19
2.168.2.129:12345;
    192.168.2.249:12345,192.168.2.247:12345;
    LoginTimeout=5;DATABASE=jdbc_example;SCHEMA=falcon_defaul
t_schema";

    private static final String TABLE_NAME = "jdbc_example";
    private static final String DB_USER = "<username>";
    private static final String DB_PASSWORD = "<password>";

    // Assuming everything in local directory use:
    // javac InsertData.java
    // java -cp .:thoughtspot_jdbc4.jar InsertData
    public static void main(String[] argv) {

        try {
            insertRecordsIntoTable();
        }
        catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    /**
     * Insert some records using batch updates.
     * Assumes a table exists: CREATE TABLE "jdbc_example" ( "t
ext" varchar(10) );
     */
    private static void insertRecordsIntoTable() throws SQLException {

        System.out.println("Inserting records.");
        Connection dbConnection = getDBConnection();
```

```
PreparedStatement preparedStatement = null;
String insertTableSQL = "INSERT INTO falcon_default_schem
a.jdbc_example (text) VALUES (?)";

try {
    preparedStatement = dbConnection.prepareStatement(insertT
ableSQL);

    // Create multiple statements and add to a batch update.
    for (int cnt = 1; cnt <= 10; cnt++) {
        preparedStatement.setString(1, "some string " + cnt);
        preparedStatement.addBatch();
        System.out.println("Record " + cnt + " was added to th
e batch!");
    }
    preparedStatement.executeBatch(); // For large numbers o
f records, recommend doing sets of executeBatch commands.
    System.out.println("Records committed");

}
catch (SQLException sqle) {
    sqle.printStackTrace();
}
finally {

    if (preparedStatement != null) {
        preparedStatement.close();
    }
    if (dbConnection != null) {
        dbConnection.close();
    }
}
}

/** Create a connection to the database. */
private static Connection getDBConnection() {
    Connection dbConnection = null;
    try {
        Class.forName(DB_DRIVER);
    }
    catch (ClassNotFoundException e) {
        System.out.println(e.getMessage());
    }
    try {
        dbConnection = DriverManager.getConnection(DB_CONNECTIO
```

```
N, DB_USER,DB_PASSWORD);
    return dbConnection;
}
catch (SQLException sqle) {
    System.out.println(sqle.getMessage());
}

return dbConnection;
}

}
```

Related Information

- [Enable JDBC logs](#)
- [Connection configuration](#)
- [Supported SQL commands](#)

Set up the JDBC driver for Pentaho

Summary: JDBC to connect to the ThoughtSpot Simba server from Pentaho.

You can use the Pentaho Data Integration (PDI) to create a JDBC connection. The Pentaho Data Integration (PDI) suite is a comprehensive data integration and business analytics platform. You can use it to create a JDBC connection to ThoughtSpot.

PDI consists of a core data integration (ETL) engine and GUI applications that allow you to define data integration jobs and transformations. Through Pentaho, we primarily use the JDBC driver to set up a connection. The process is not as complicated as with SSIS, and is much more lenient.

Community and enterprise editions of PDI are available. Using the community edition is sufficient, though you may use the enterprise edition, which is subscription based, and therefore contains extra features and provides technical support.

Use JDBC to connect to the ThoughtSpot Simba server from Pentaho. The connection will be made between a new ThoughtSpot Table Input and Output objects.

Check the ThoughtSpot IP and the simba_server status

Before you begin, you need to know the IP address or DNS name of the server you intend to connect your server to.

1. SSH as `admin` or the `thoughtspot` user to your ThoughtSpot node.
2. Verify the node IP(s).

```
$ tscli node ls  
172.18.231.17  
172.18.231.18
```

3. Make a note of each IP; there may be more than one.
4. Configure the ThoughtSpot firewall to allow connections from your ETL client, by running the following command on any ThoughtSpot node: `tscli firewall open-ports --ports`

12345

5. Exit or close the shell.

Install the Simba drivers in the Pentaho directories

Before starting the Pentaho Data Integration (PDI) client and creating the connection, ensure that the Simba JDBC client libraries are present in the Pentaho client/server machines. This will ensure that the drivers picked up at runtime.

1. Log in to the local machine where you have already installed the Pentaho Data Integration (PDI) client.
2. Click [Here](#) to download the JDBC driver.
3. Click **JDBC Driver** to download the file `thoughtspot_jdbc<version>.jar`.
4. Copy the `thoughtspot_jdbc<version>.jar` file to the following directories:
 - `<Pentaho_install_dir>/server/data-integration-server/tomcat/webapps/pentaho-di/WED-INF/lib/`
 - `<Pentaho_install_dir>/design-tools/data-integration/lib/`
 - `<Pentaho_install_dir>/server/data-integration-server/tomcat/lib/`
 - `<Pentaho_install_dir>/design-tools/data-integration/plugins/spoon/agile-bi/lib/`

Set up the driver

This section explains how to set up the JDBC driver using Pentaho. These instructions use Spoon, the graphical transformation and job designer associated with the PDI suite. It is also known as the Kettle project.

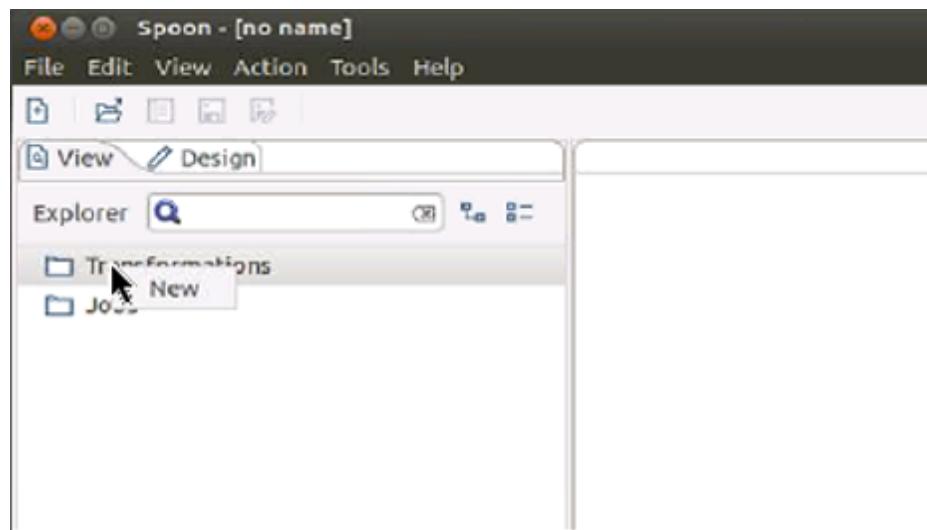
Create a transformation

Do the following on your ETL workstation with the Pentaho client:

1. Open the PDI client.

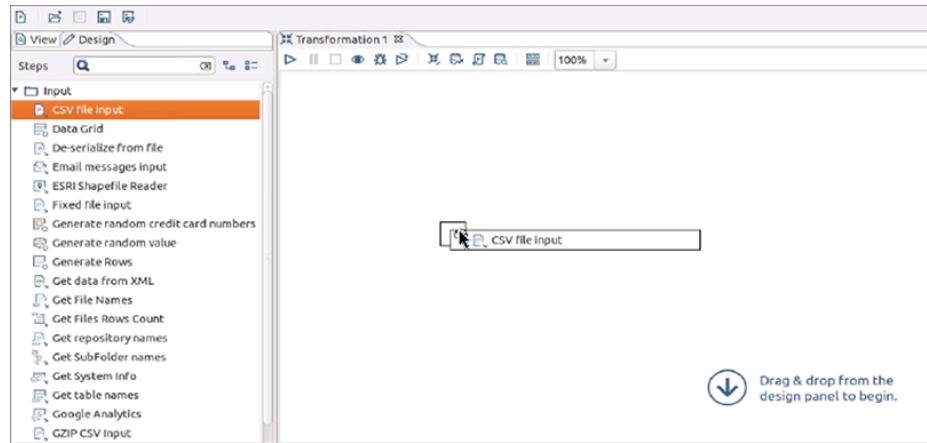
```
./spoon.sh &>/dev/null &
```

2. Right click **View > Transformations** tab.
3. Click **New** to create a new transformation.

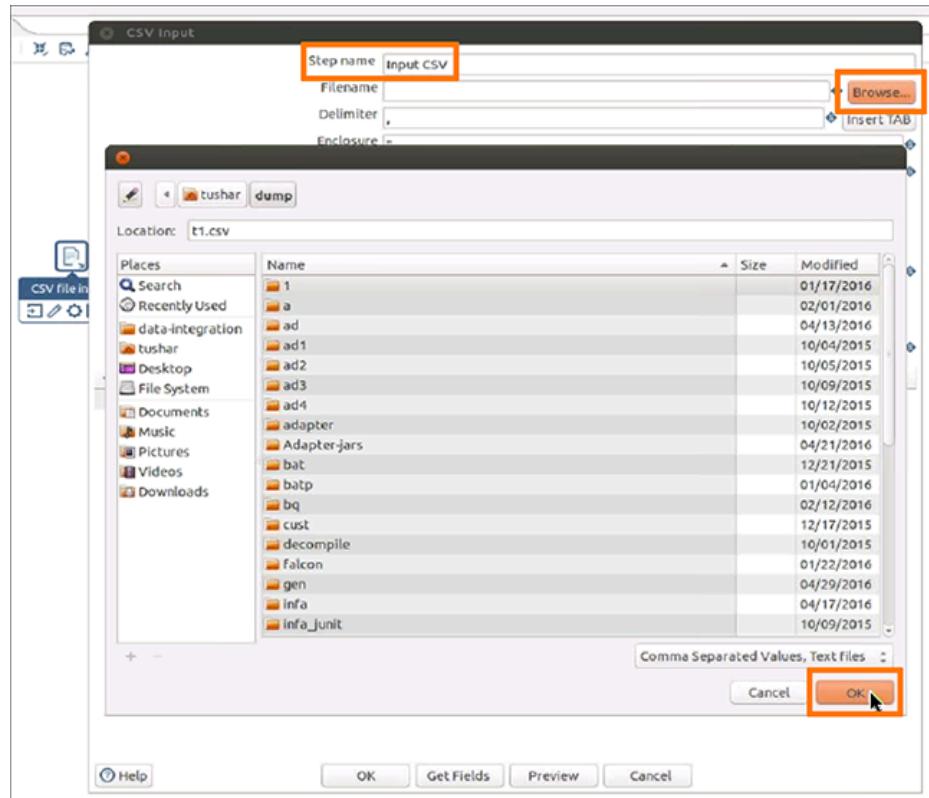


4. Click **Input** under the **Design** tab to expand it.
5. Drag and drop **CSV File Input** to the **Transformation** window.

This opens a new CSV file.



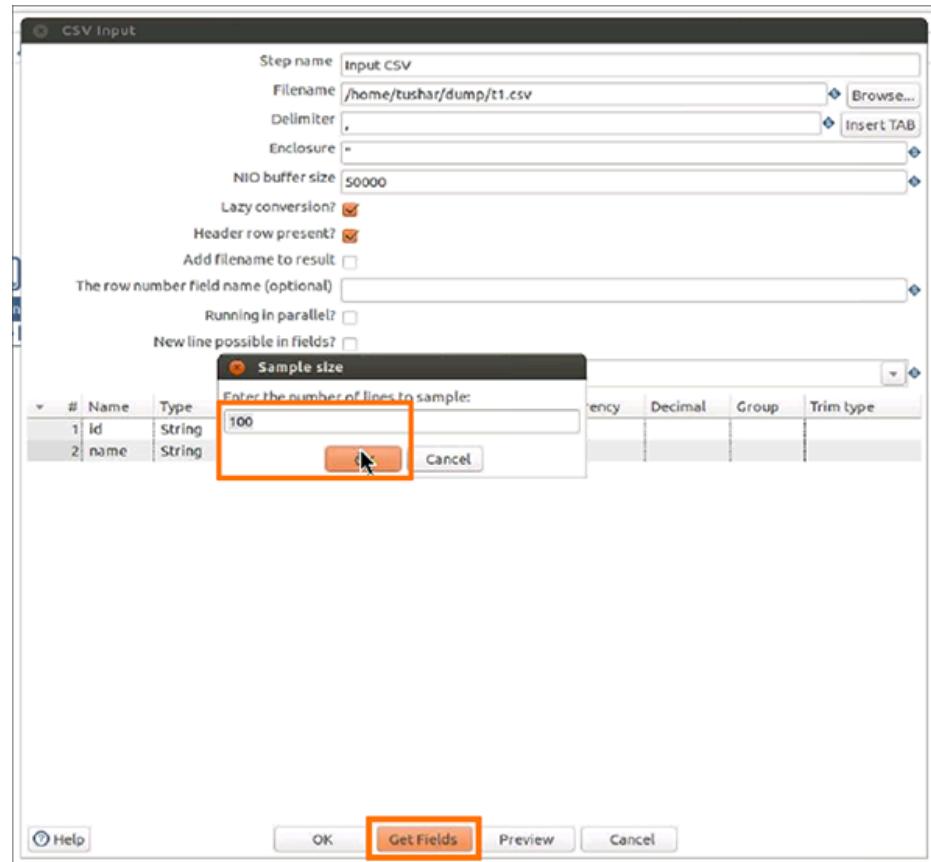
6. Double-click the **CSV File Input** icon to open the **CSV Input** dialog .
7. Name the **Step**.
8. Click **Browse** next to the **Filename** field and provide the file you want to read from.
9. Click **OK**.



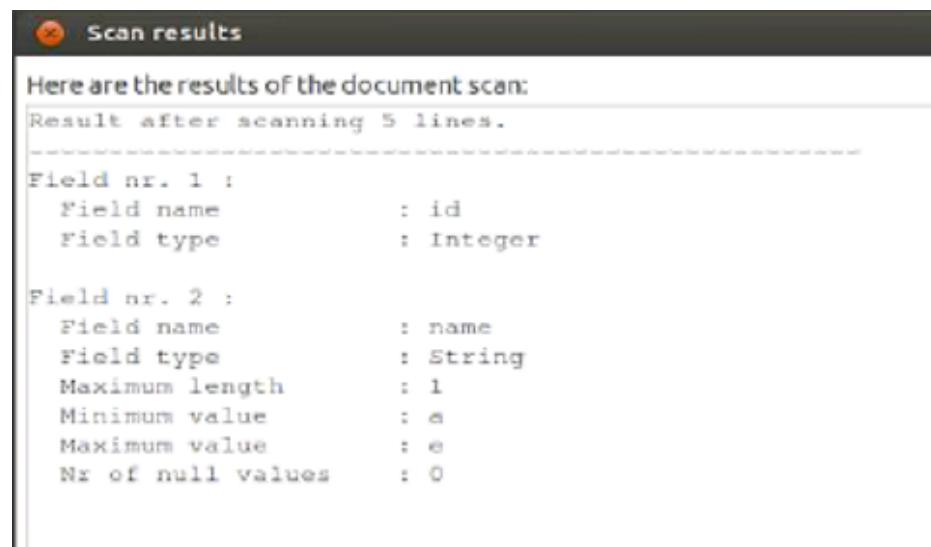
10. In the CSV Input dialog, click **Get Fields**.
11. Enter the number of lines you would like to sample in the Sample size dialog.

The default setting is 100.

1. Click **OK** when you are ready.

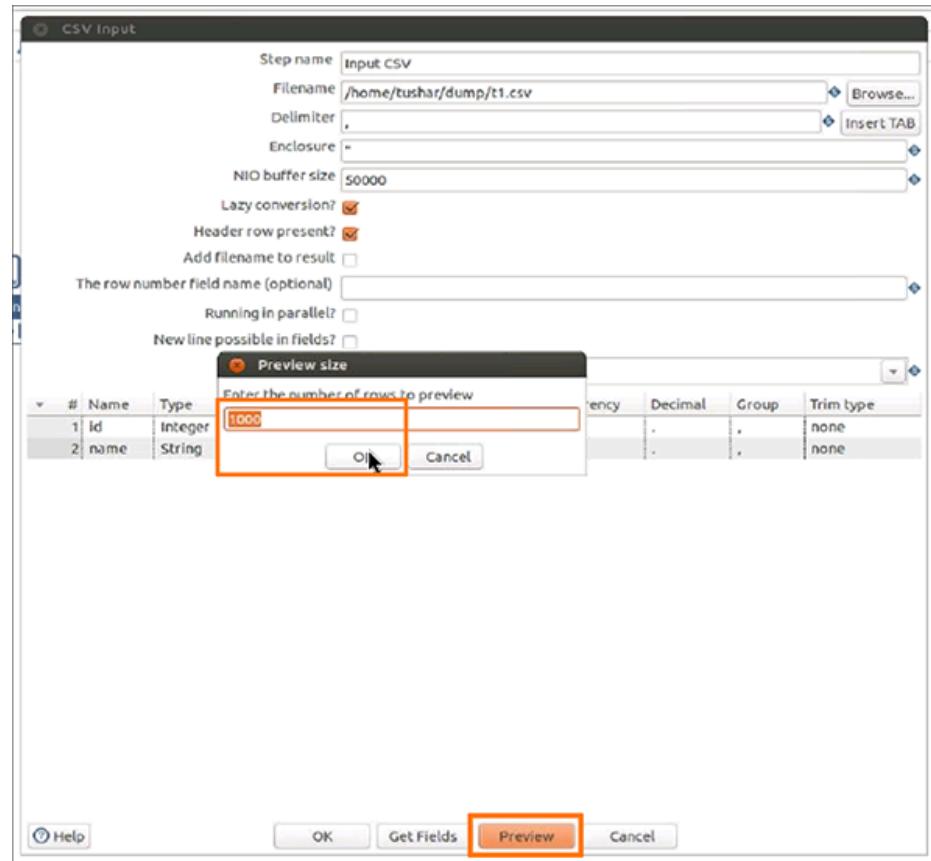


The tool reads the file and suggests the field name and type.



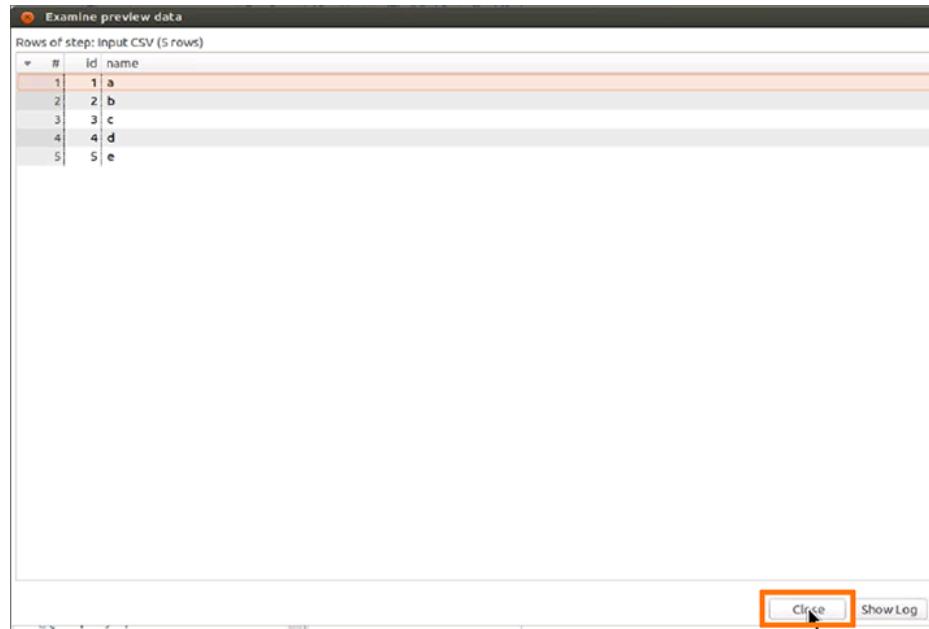
2. Click **Preview** to preview the data.
3. Enter the number of rows to preview in the **Preview size** dialog.

The default setting is 1000. Click **OK** to start the transformation in preview.



4. Examine the preview data, then click **Close**.

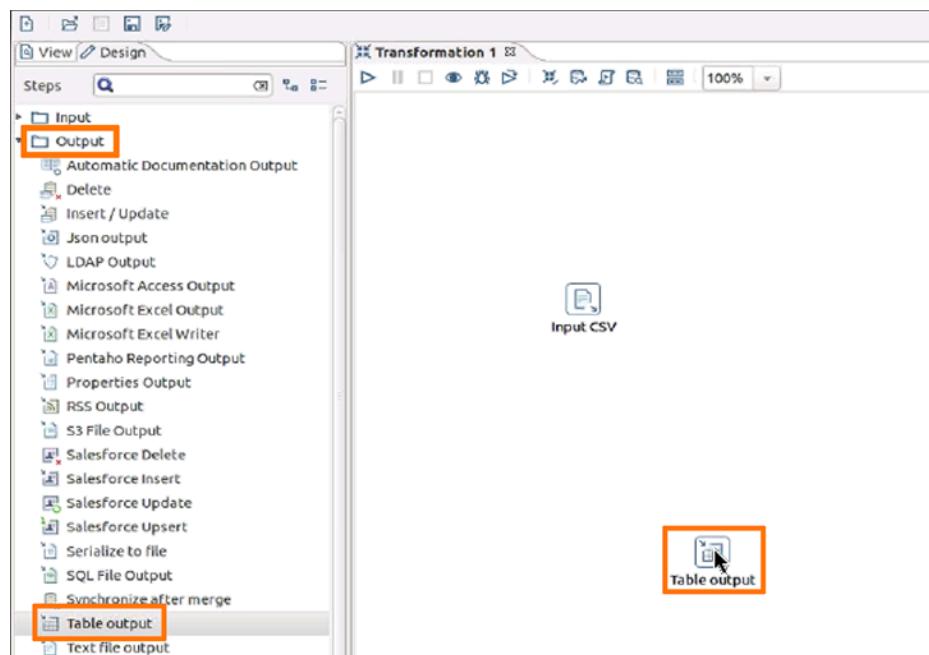
You may want to verify that you are able to read the data using the SQL query from ThoughtSpot.



5. Click **OK** in the CSV Input dialog to confirm your CSV input settings.

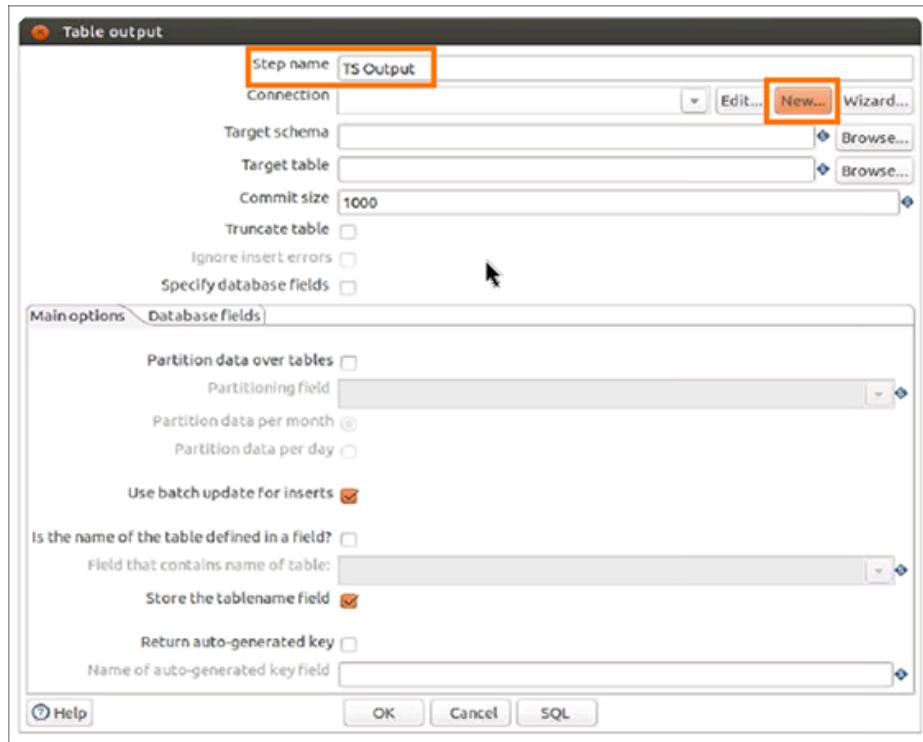
Define the Output

1. Click **Design > Output**.
2. Drag and drop **Table output** to the Transformation window.



3. Double click the **Table output** icon to open the Table output dialog.

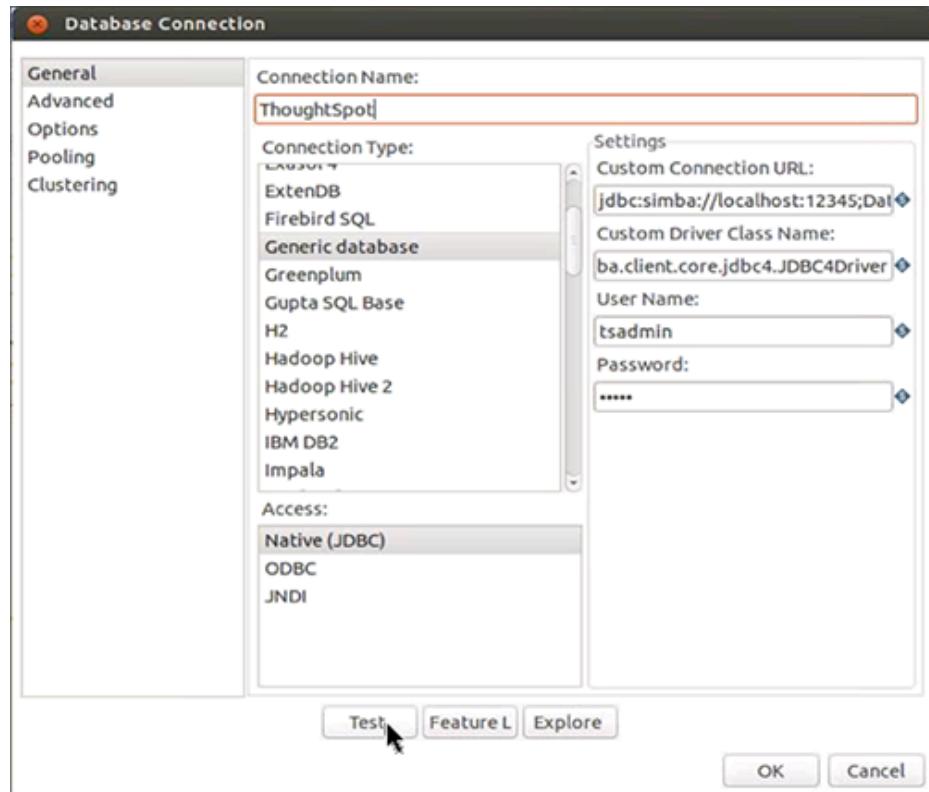
4. Enter a **Step name**.
5. Click **New** to create a new connection.



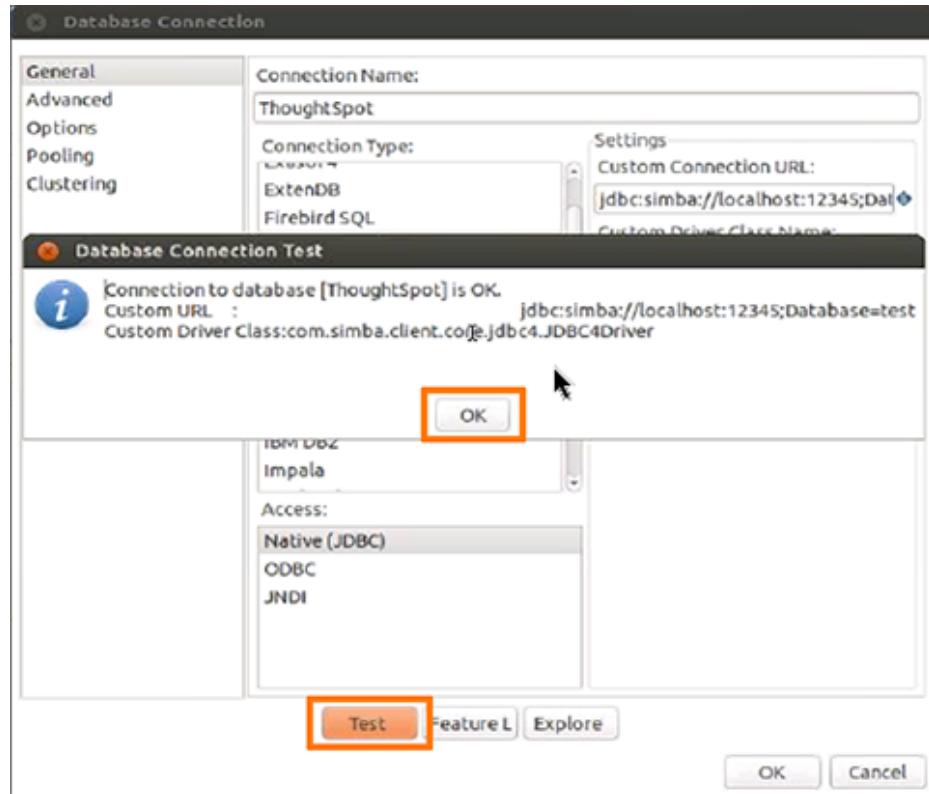
6. Enter or select the following information in the Database Connection dialog:

Field	Description
Connection	Any string.
Name	
Connection Type	Generic database
Access	Native (JDBC)
Custom Connection URL	<code>jdbc:simba://SERVER_IP:12345;Database=DATABASE_or_SCHEMA_NAME </code></code>
URL	The IP is a node in your ThoughtSpot cluster. The name or schema of the database you want to connect to. Use TQL to create a database name if needed. Ensure that there are no leading or trailing spaces.

Custom Driver Class Name	com.simba.client.core.jdbc4.JDBC4Driver Ensure that there are no leading or trailing spaces.
User Name	A ThoughtSpot username. If you leave this empty, you are prompted for it at connection time. This user should have **Data Management** privileges on ThoughtSpot.
Password	The password for the **User Name**. If you leave this empty, you are prompted for it at connection time.



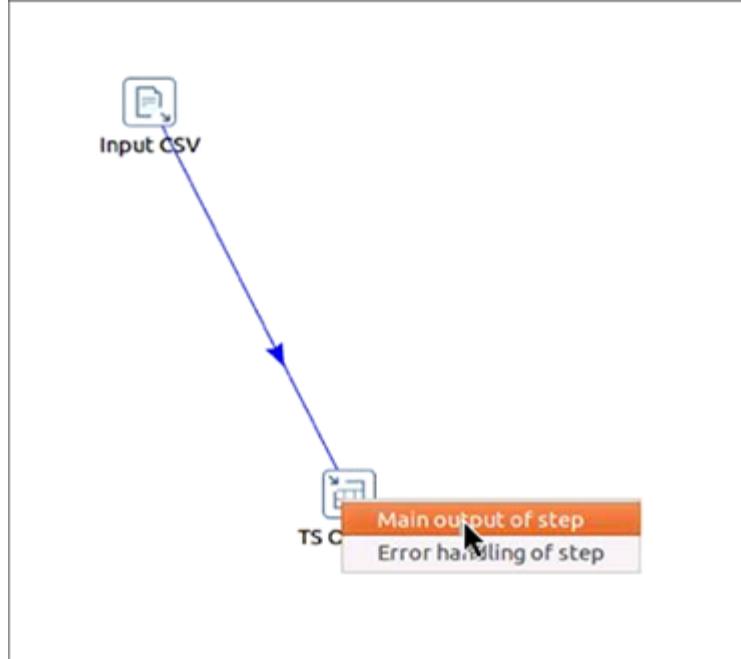
7. Click **Test** to test your database connection.
8. If you are able to make a successful connection to the ThoughtSpot Simba Server, click **OK**.



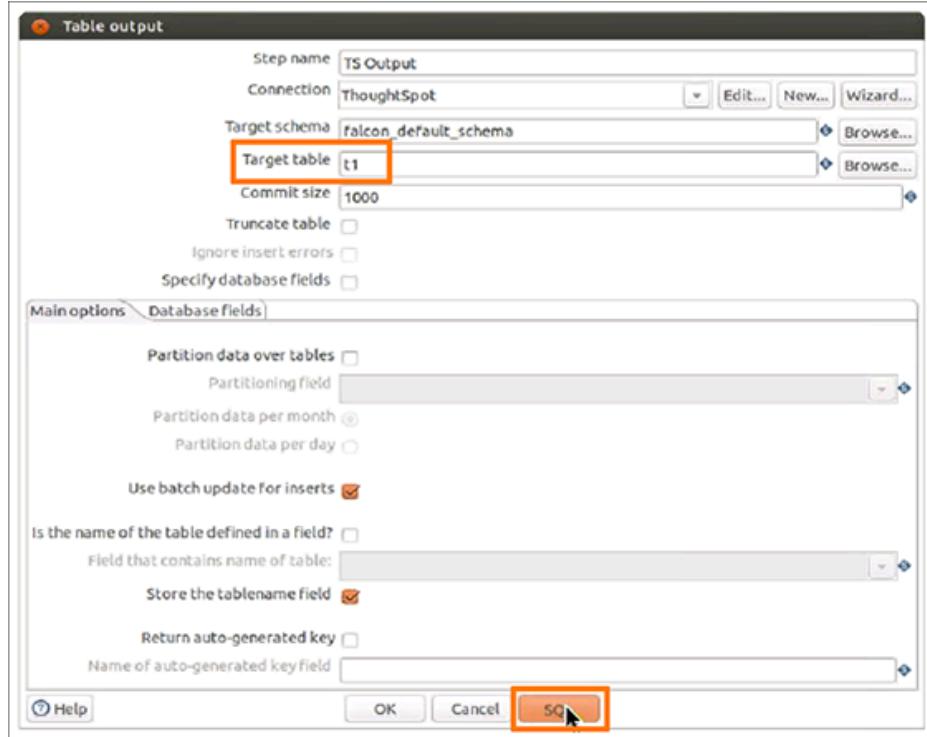
9. Click **OK** in the Database Connection dialog to create the new connection.

Import data

1. In the **Table output** dialog, select the connection you just created.
2. Click **Browse** next to the **Target schema** field and select your **Target schema**.
3. Click **OK** when you are done.
4. Connect the **Input CSV** icon to the **Table output** icon by clicking and dragging an arrow.
5. When prompted, choose **Main output of step**.

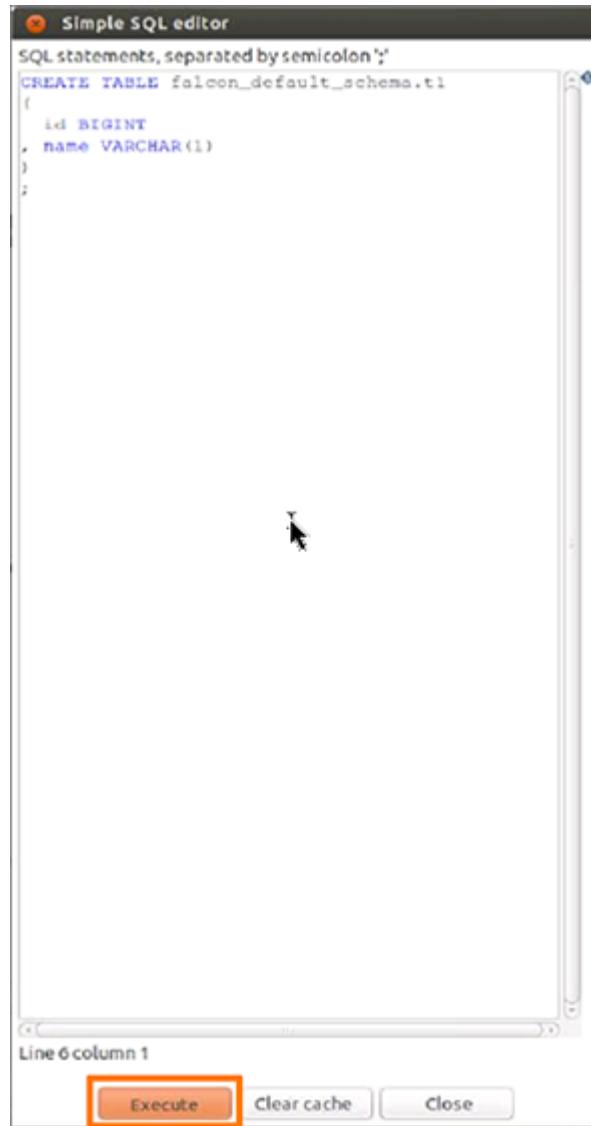


6. Double click the **Table output** icon to reopen the **Table output** dialog.
7. Enter a **Target table name**.
8. Click **SQL**.

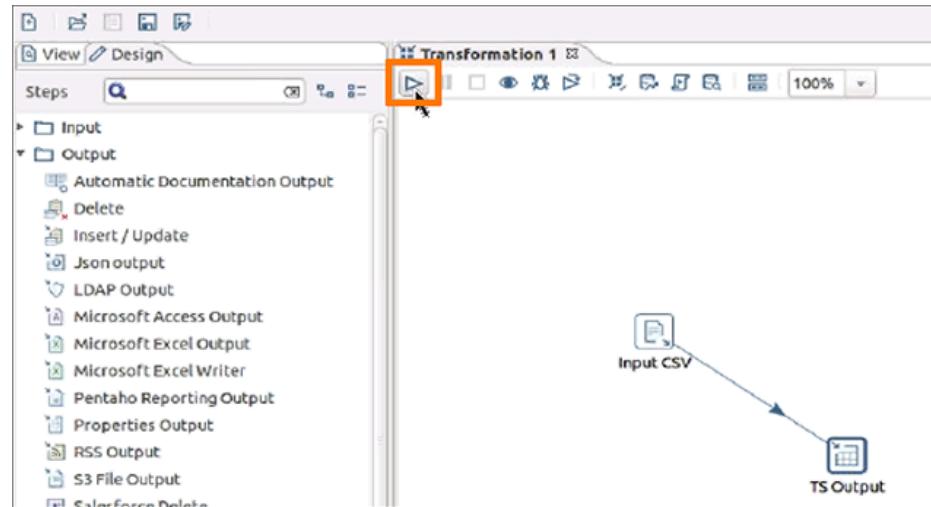


9. In the **Simple SQL editor** dialog, click **Execute**.

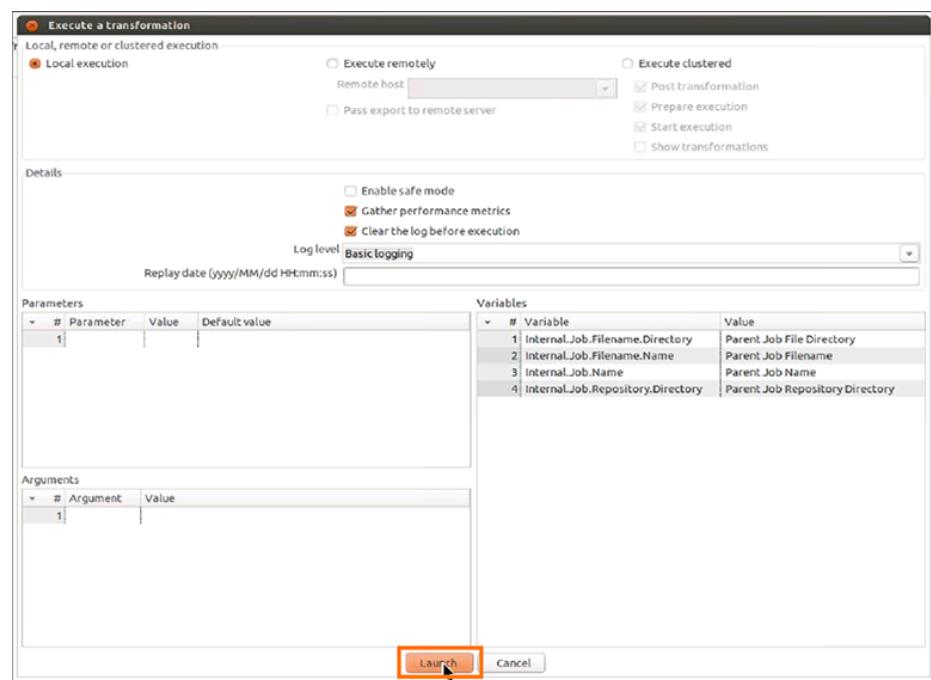
The system processes and then displays the results of the SQL statements.



10. Close all open dialogs.
11. Click the **Play** button at the top of the **Transformation** window to execute the transformation.



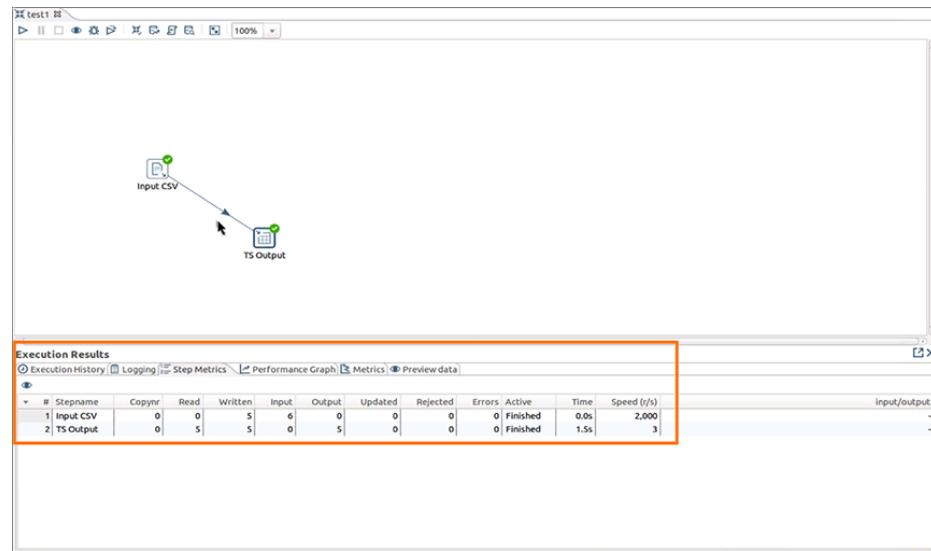
12. Click **Launch** in the **Execute a transformation** dialog.



The system prompts you to save it if you have not already.

13. View the **Execution Results**.

Set up the JDBC driver for Pentaho



Troubleshooting Data Integrations

Summary: Learn how to fix connection issues.

This section can help if you're having trouble creating a connection or need to find out more information about what is going on with ODBC or JDBC.

The information contained here is very basic, and mostly about how to enable logs on the client side. If you need more detailed troubleshooting information or help, please contact ThoughtSpot Support.

- **Enable ODBC Logs**

If you need more information in order to troubleshoot ODBC connections, you can enable logging for ODBC.

- **Enable JDBC Logs**

To enable logging for JDBC, add the logging parameters to the connect string. Logs are stored on ThoughtSpot.

- **Schema not found error with ODBC**

When connecting with ODBC, you need to specify both the database and schema to connect to. If no schema is supplied, you will get an error indicating that the schema could not be found.

- **How to improve throughput of the load**

The transaction/commit size value can improve the throughput of the load when setting up the ODBC Driver.

- **ODBC tracing on Windows**

Using logs to aid in troubleshooting.

Enable ODBC Logs

Summary: Learn how to troubleshoot ODBC connections.

If you need more information in order to troubleshoot ODBC connections, you can enable logging for ODBC on the workstation you use for connecting to ThoughtSpot. There are two points where you can enable logging:

- the workstation where you run your ETL activities
- the server where the Simba service is running

On both workstation and servers, the verbosity of the log is controlled by the `LogLevel` property. This property can be one of the following:

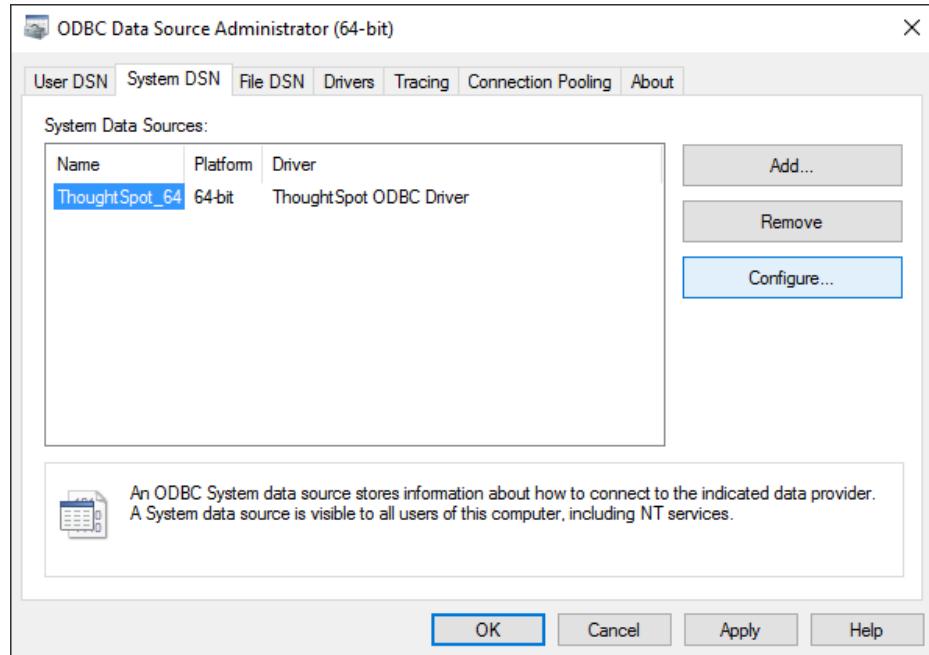
- `0` or `LOG_OFF` : no logging occurs
- `1` or `LOG_FATAL` : only log fatal errors
- `2` or `LOG_ERROR` : log all errors
- `3` or `LOG_WARNING` : log all errors and warnings
- `4` or `LOG_INFO` : log all errors, warnings, and informational messages
- `5` or `LOG_DEBUG` : log method entry and exit points and parameter values for debugging
- `6` or `LOG_TRACE` : log all method entry points

Larger values include the information from lesser values. For example, if you set `3` or `LOG_WARNING`, you log all warnings *and* all errors.

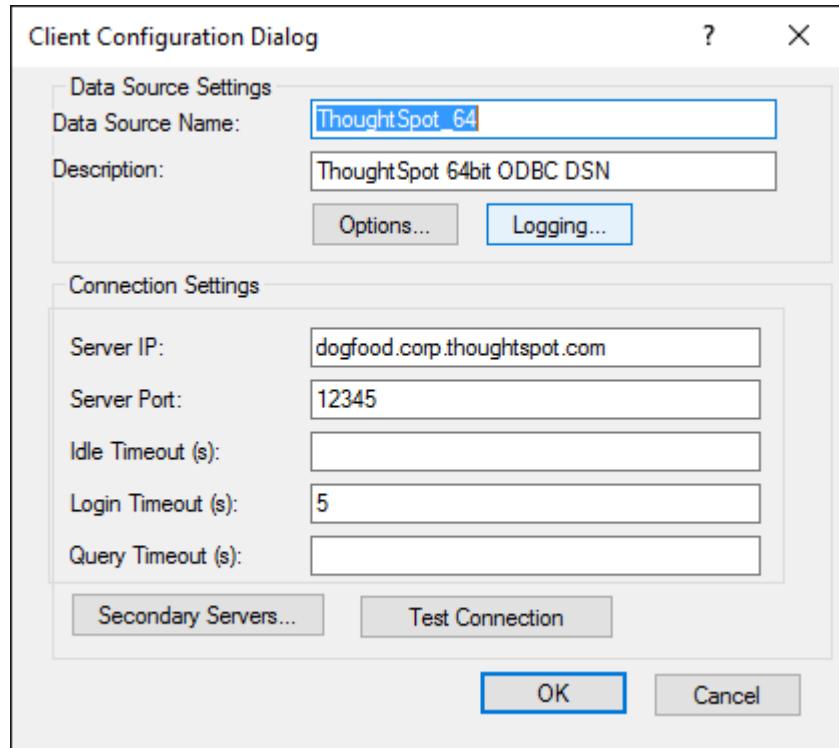
Enable ODBC logs on a Windows workstation

To enable ODBC logs on Windows:

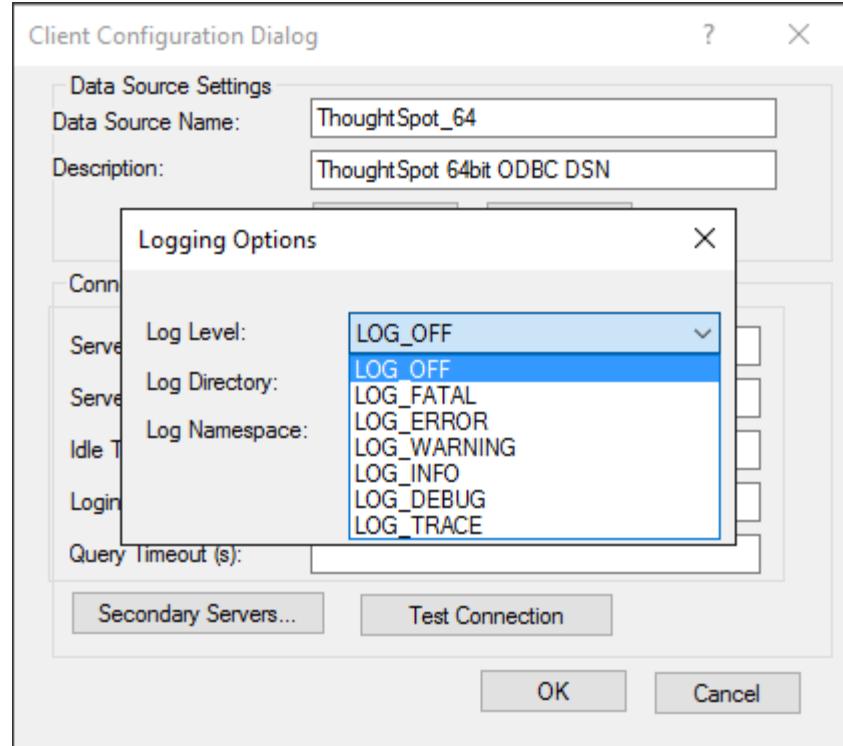
1. Open the **ODBC Data Source Administrator** and select the **System DSN** tab.
2. Select your ThoughtSpot data source and click **Configure**.



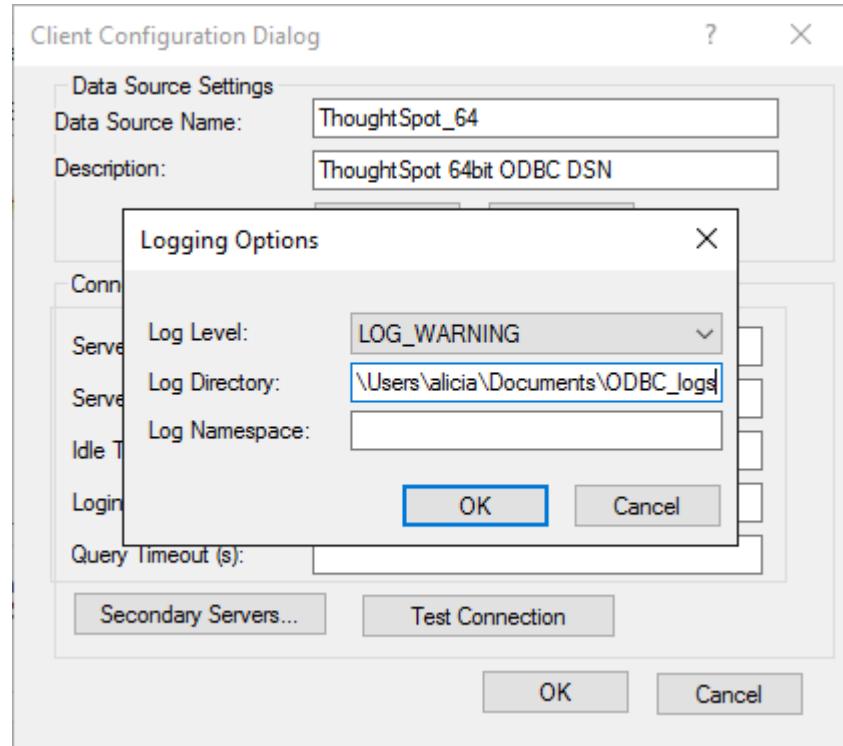
3. In the Client Configuration Dialog, click **Logging**.



4. Choose a **Log Level**, depending on what level of verbosity you want to show in the logs.



5. For **Log Directory**: type in the fully qualified path where you want the logs to be saved.



6. Click **OK** to save your settings, and **OK** again, to dismiss the ODBC Data Source Administrator.
7. Run the ODBC load.
8. Locate the log file that was generated, and send it to ThoughtSpot Support with a description of the problem.

Enable ODBC logs on a Linux workstation

To enable logging on Linux, follow these instructions:

1. Navigate to the directory where you installed ODBC.
2. Open the `odbc.ini` file in a text editor.

This file is the registry and configuration file for ODBC.

3. Locate the `LogLevel` and `LogPath` properties.
4. Uncomment the properties.
5. Enter a value for the `LogLevel`.

Acceptable values are from 1 to 6 with 6 being the most verbose.

6. Enter the fully qualified path for the `LogPath` values.

The log will be written here. Your file will look similar to the following: Example for Linux 64-bit:

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList = 172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The
certificate file can be obtained by extracting the SDK t
arball
LogLevel = 3 # Set log level to enable debug logging
LogPath = /home/admin/odbc-logs # Set the debug log file
s path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

7. Save and close the file.
8. To test the configuration, run the ODBC load and review the log files.

Control logs from the Simba server

You may want to collect logs from the Simba service. Do the following procedure on every ThoughtSpot node running the Simba service.

1. SSH into the ThoughtSpot node.
2. Edit the `/etc/thoughtspot/linux.ini` file.

```
...
[Driver]

## Note that this default DriverManagerEncoding of UT
F-32 is for iODBC. unixODBC uses UTF-16 by default.
## If unixODBC was compiled with -DSQL_WCHART_CONVERT,
then UTF-32 is the correct value.
## Execute 'odbc_config --cflags' to determine if you n
eed UTF-32 or UTF-16 on unixODBC
DriverManagerEncoding=UTF-32
DriverLocale=en-US
ErrorMessagesPath=/usr/home/linux/ErrorMessages/
LogLevel=0
LogNamespace=
LogPath=

....
```

3. Uncomment the `LogLevel` setting.

The `LogLevel` is the level of logging to capture (0-6).

4. Set `LogPath` to a directory to save the logs.

The `LogPath` is the fully qualified path where ThoughtSpot should write the logs.

5. Work with ThoughtSpot Support to restart the Simba service.

The node IP may change because of the restart. If this happens, repeat the entire procedure.

Enable JDBC Logs

Summary: Configure logging parameter strings.

To enable logging for JDBC, add the logging parameters to the connect string. Logs are stored on ThoughtSpot. Before enabling JDBC logging, you need:

- The level of logging you want to capture.
- The path on the ThoughtSpot server where the logs will be written. Make sure the directory has the correct permissions so that the “admin” Linux user can write logs to it.

To enable JDBC logging:

1. When forming the connect string for JDBC, add these two parameter, separated by "&":

For example:

```
jdbc:simba://192.168.2.248:12345;SERVERS=192.168.2.24  
9:12345,  
192.168.2.247:12345;Database=test;Schema=falcon_defaul  
t_schema;**LogLevel=3;LogPath=/usr/local/scaligent/log  
S**
```

The `LogLevel` is the level of logging to capture (0-6). The `LogPath` is the fully qualified path where logs will be written on ThoughtSpot.

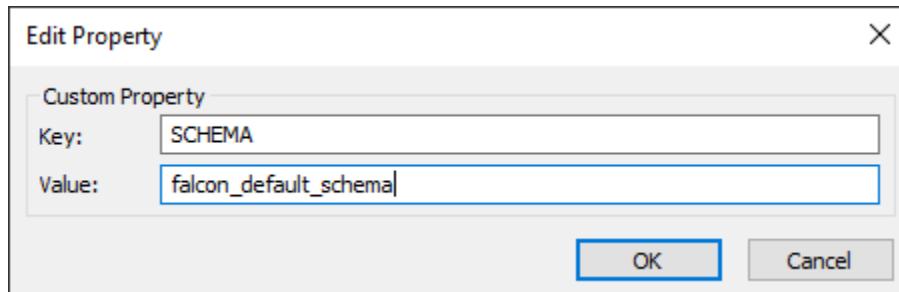
2. Run the JDBC code that uses the connection you modified.
3. Check the `LogPath` directory for logs generated by JDBC.

Schema not found error with ODBC

Summary: Correct schema not found errors.

When connecting with ODBC, you need to specify both the `DATABASE` and `SCHEMA` parameters. This is true even if you do not use schema names in ThoughtSpot. If you don't supply a `SCHEMA`, you get an error indicating that the schema could not be found.

The default schema name in ThoughtSpot is `falcon_default_schema`. To set the `SCHEMA` on Windows, adding a custom property with the key `SCHEMA` and the value `falcon_default_schema`.



On Linux, you can edit the properties in the `odbc.ini` file for the driver you are using:

```
[ThoughtSpot]
Description = ThoughtSpot 64-bit ODBC Driver
Driver = ThoughtSpot
ServerList = 172.18.231.17 12345
Locale = en-US
ErrorMessagesPath = /home/admin/linux/ErrorMessages
UseSsl = 0
#SSLCertFile = # Set the SSL certificate file path. The certificate file can be obtained by extracting the SDK tarball
#LogLevel = 0 # Set log level to enable debug logging
#LogPath = # Set the debug log files path
DATABASE = # Set the default database to connect to
SCHEMA = # Set the default schema to connect to
```

Related information

- [Configuring ODBC on Windows](#)
- [Configuring ODBC on LINUX](#)
- [ODBC and JDBC configuration properties](#)

How to improve throughput

Summary: Adjusting the transaction size may correct poor performance and low throughput.

The transaction/commit size value can improve the throughput of the load when setting up the ODBC Driver.

Adjusting the transaction size may correct poor performance and low throughput issues. The transaction size should be set to match the total number of rows that are expected to be loaded in the load cycle. However, increasing this value even higher should help improve throughput of the load.

Warning: A high transaction size may slow down the ThoughtSpot system.



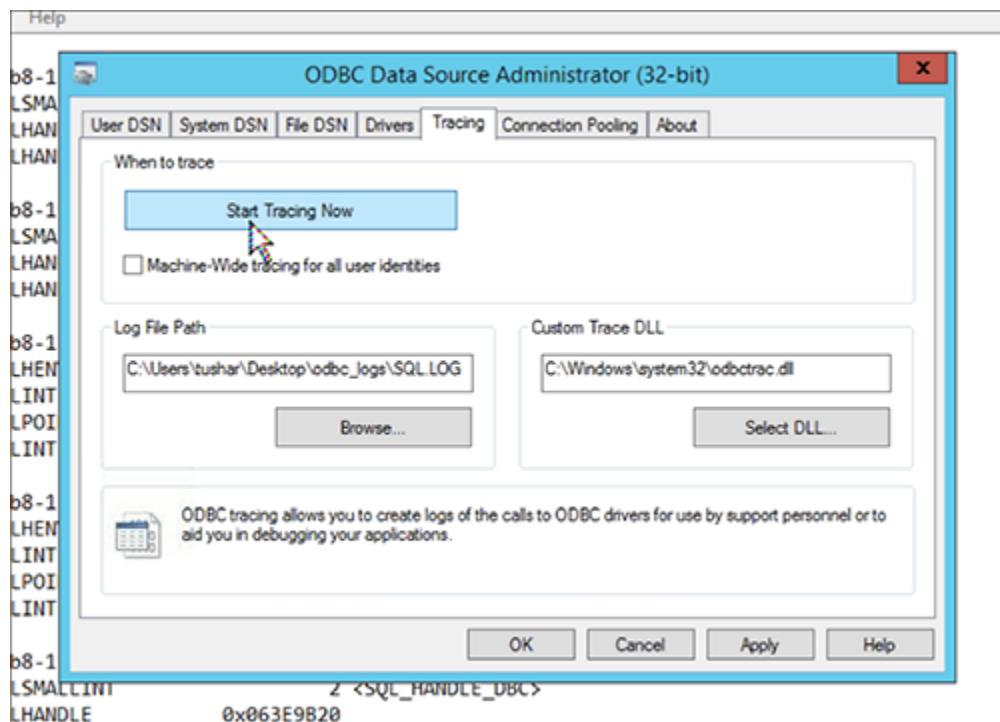
This is where the transaction size field exists for SSIS. Clicking on the ODBC destination reveals the properties on the right hand side, where the **Transaction Size** can be found.

See [Set up the ODBC Driver for SSIS](#) for more details on setting the transaction size.

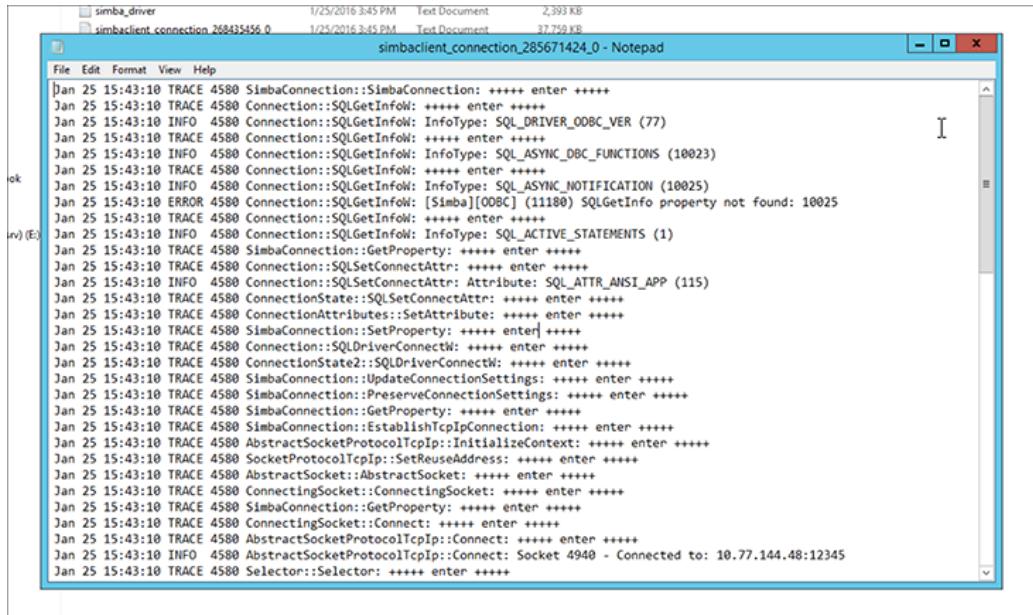
ODBC tracing on Windows

Summary: Using logs to aid in troubleshooting.

Windows shows ODBC specific tracing in the ODBC Data Source Administrator Tracing tab. You can start tracing there by clicking **Start Tracing Now**. This logs every ODBC call from this system, and prints the input and output for the call.



Although this is lower level information, it can still be helpful in troubleshooting. When you are not sure if it is our driver or the tool causing an issue, doing this trace will help narrow the inquiry.



The screenshot shows a Windows Notepad window titled "simbaclient_connection_285671424_0 - Notepad". The window displays a log of ODBC trace messages. The log entries are timestamped and show various calls to SimbaConnection and SQLGetInfoW methods, indicating the connection setup process. The log ends with a successful connection to a socket.

```
File Edit Format View Help
Jan 25 15:43:10 TRACE 4580 SimbaConnection::SimbaConnection: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 Connection::SQLGetInfoW: +++++ enter +++++
Jan 25 15:43:10 INFO 4580 Connection::SQLGetInfoW: InfoType: SQL_DRIVER_ODBC_VER (77)
Jan 25 15:43:10 TRACE 4580 Connection::SQLGetInfoW: +++++ enter +++++
Jan 25 15:43:10 INFO 4580 Connection::SQLGetInfoW: InfoType: SQL_ASYNC_DBC_FUNCTIONS (10023)
Jan 25 15:43:10 TRACE 4580 Connection::SQLGetInfoW: +++++ enter +++++
Jan 25 15:43:10 INFO 4580 Connection::SQLGetInfoW: InfoType: SQL_ASYNC_NOTIFICATION (10025)
Jan 25 15:43:10 ERROR 4580 Connection::SQLGetInfoW: [Simba][ODBC] (11180) SQLGetInfo property not found: 10025
Jan 25 15:43:10 TRACE 4580 Connection::SQLGetInfoW: +++++ enter +++++
Jan 25 15:43:10 INFO 4580 Connection::SQLGetInfoW: InfoType: SQL_ACTIVE_STATEMENTS (1)
Jan 25 15:43:10 TRACE 4580 SimbaConnection::GetProperty: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 Connection::SQLSetConnectAttr: +++++ enter +++++
Jan 25 15:43:10 INFO 4580 Connection::SQLSetConnectAttr: Attribute: SQL_ATTR_ANSI_APP (115)
Jan 25 15:43:10 TRACE 4580 ConnectionState::SQLSetConnectAttr: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 ConnectionAttributes::SetAttribute: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SimbaConnection::SetProperty: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 Connection::SQLDriverConnectW: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 ConnectionState2::SQLDriverConnectW: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SimbaConnection::UpdateConnectionSettings: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SimbaConnection::PreserveConnectionSettings: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SimbaConnection::GetProperty: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SimbaConnection::EstablishTcpIpConnection: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 AbstractSocketProtocolTcpIp::InitializeContext: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SocketProtocolTcpIp::SetReuseAddress: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 AbstractSocket::AbstractSocket: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 ConnectingSocket::ConnectingSocket: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 SimbaConnection::GetProperty: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 ConnectingSocket::Connect: +++++ enter +++++
Jan 25 15:43:10 TRACE 4580 AbstractSocketProtocolTcpIp::Connect: +++++ enter +++++
Jan 25 15:43:10 INFO 4580 AbstractSocketProtocolTcpIp::Connect: Socket 4940 - Connected to: 10.77.144.48:12345
Jan 25 15:43:10 TRACE 4580 Selector::Selector: +++++ enter +++++
```

If you start or stop tracing, make sure you do not have the SSIS client open. Close it, change the trace, and reopen.

Supported SQL commands

Summary: The ThoughtSpot connection drivers support a limited set of SQL commands.

The ODBC and JDBC drivers support a limited set of SQL commands. When developing software that uses a ThoughtSpot ODBC driver, use this reference of supported commands. This reference is intended for developers using other tools (ETL, etc.) to connect to ThoughtSpot through the ODBC or JDBC driver.

Note: ThoughtSpot displays VARCHAR fields using lower case, regardless of what the original casing of your loaded data is.

ODBC

These SQL commands are supported for ODBC:

- `CREATE TABLE`

Creates a table with the specified column definitions and constraints. The table is replicated on each node.

```
CREATE TABLE country_dim (id_number int, country varchar, CONSTRAINT PRIMARY KEY (id_number));
```

- `INSERT`

Creates placeholders in the table to receive the data.

```
INSERT INTO TABLE country_dim (?, ?);
```

- `DELETE FROM <table>`

Deletes `ALL` rows from the specified table. Use the `WHERE` clause to specify only certain rows to be deleted. Example: You could remove all data for sales before a certain date to free up space in ThoughtSpot.

```
DELETE FROM country_dim;
```

- `SELECT <cols_or_expression> FROM <table_list> [<WHERE ><predicates>] [<GROUP BY ><expressions>] [<ORDER BY ><expressions>]`

Fetches the specified set of table data.

```
SELECT id_number, country FROM country_dim WHERE id_number > 200;
```

JDBC

`TRUNCATE` is not supported. Instead, use `DELETE FROM TABLE` which is functionally equivalent to “truncate table” in terms of table compression and so forth.

Connection configuration

Summary: Lists the properties you can set for ODBC or JDBC connections

This section lists the properties you can set for ODBC or JDBC connections.

Setting Properties for ODBC

Not all the parameters Simba accepts are supported by the ThoughtSpot ODBC clients, and ThoughtSpot has added some properties, which are listed separately here. All configuration properties use the type String (text).

You can set these properties on Windows by using the [ODBC Administrator](#) client. For Linux, the properties are located in three files, depending on the property type:

Property Type	Location
DSN	<code>odbc.ini</code> file
Driver	<code>odbsinst.ini</code> file
SimbaSetting Reader	<code>simbaclient.ini</code> file

Setting Properties for JDBC

For JDBC, these properties are passed as key value pairs in the connect string. For more information, see [Use the JDBC Driver](#).

Properties Reference

The following tables summarize the configuration properties.

Property	Type	Description
DATABASE	DSN or Driver	The default database to connect to.
SCHEMA	DSN or Driver	The default schema to connect to.
Description	DSN	A brief, human-readable description of the DSN. This describes the DSN to users who are deciding which DSN to use.
Driver	DSN or Driver	In the driver configuration location, Driver should contain the path to the driver binary. In the DSN configuration location, Driver could contain the path to the driver binary, or it could contain the driver entry in the registry.
IdleTimeout	DSN	The time to wait for a response from the server, in seconds. This property is optional, but SimbaClient will wait indefinitely for SimbaServer to respond to a request made to the server unless you specify a timeout period. IdleTimeout specifies how many seconds that SimbaClient will wait before aborting the attempt and returning to the application with an error. This timeout corresponds to ODBC's CONNECTION_TIMEOUT property and is only used when more specific timeouts, such as QUERY_TIMEOUT or LOGIN_TIMEOUT aren't applicable.
Locale	DSN	The connection locale. If this value is set, it overrides the driver-wide locale. For example, the driver-wide locale could be en-US . If the client would prefer fr-CA , it can set the connection locale to fr-CA . Values are composed of a 2-letter language code (in lower case), and an optional 2-letter country code (in upper case). If the country code is specified, it must be separated from the language code by a hyphen (-).
LoginTimeout	DSN	The timeout, in seconds, to wait for a response from the server when attempting to log in. A value of 0 means no timeout. The default value is 60.
QueryTimeout	DSN	The timeout, in seconds, to wait for a response from the server during Prepare, Execute, and ExecuteDirect. A value of 0 means no timeout. The default value is 60.
ServerList	DSN	A comma separated list of all servers (IP address and port number) to connect to. SimbaClient must be able to find SimbaServer on the network. This property enables server discovery. SimbaClient will try to make a network connection to the servers in the order specified until a connection is made.
LogLevel	SimbaSetting Reader	Controls the granularity of the messages and events that are logged. With this keyword, you can control the amount of log output by controlling the kinds of events that are logged. Possible values (case sensitive): <ul style="list-style-type: none"> • 0 or LOG_OFF : no logging occurs • 1 or LOG_FATAL : only log fatal errors • 2 or LOG_ERROR : log all errors • 3 or LOG_WARNING : log all errors and warnings • 4 or LOG_INFO : log all errors, warnings, and informational messages • 5 or LOG_DEBUG : log method entry and exit points and parameter values for debugging • 6 or LOG_TRACE : log all method entry points

Property	Type	Description
LogPath	SimbaSetting Reader	<p>Specifies the directory where the log files are created. For example:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px; width: fit-content;"> LogPath=C:\Simba Technologies\Temp </div> <p>If this value is not set, the log files are written to the current working directory of the SimbaClient.</p>
LogFileSize	SimbaSetting Reader	The size of each log file, in bytes. The default values is 20971520 bytes. When the maximum size of the file is reached, a new file is created.
LogFileCount	SimbaSetting Reader	The number of log files to create. When the maximum number of log files has been created, the oldest file will be deleted and a new one created. The default value is 50.
username	UID	Part of a user username/password combination. This combination should correspond to a ThoughtSpot application user with permissions appropriate to your ETL requirements. Typically, this user is a user with data management or administrative privileges on the application.
password	Password	Part of a user username/password combination. This combination should correspond to a ThoughtSpot application user with permissions appropriate to your ETL requirements. Typically, this user is a user with data management or administrative privileges on the application.

Keyword reference

You can use keywords when asking a question to help define your search. This reference lists the various keywords. You can also see this list of keywords and examples from within the help center.

Keywords in Other Languages

Currently, we offer the following keyword translations.

日本語	中文 (简体)	Deutsche	Español (latín)	Español (España)
Français (Canada)	Français (France)	Português (Brasil)	Português (Portugal)	Italiano
Dansk	Suomi	Svenska	Norsk	Nederland

Also, see the topic on how to set [locale preferences in your user profile](#) to control language, date, and number formats on the ThoughtSpot UI.

General

Keyword	Examples
top	<ul style="list-style-type: none">• top sales rep by count sales for average revenue > 10000• sales rep average revenue for each region top
bottom	<ul style="list-style-type: none">• bottom revenue average• bottom revenue by state• customer by revenue for each sales rep bottom
n	top 10 sales rep revenue
n	bottom 25 customer by revenue for each sales rep
sort by	<ul style="list-style-type: none">• revenue by state sort by average revenue• revenue by customer sort by region

Date

Keyword	Examples
after	order date after 10/31/2014
before	order date before 03/01/2014
between ... and ...	order date between 01/30/2012 and 01/30/2014
daily year-over-year	growth of revenue by order date daily year-over-year
daily	shipments by region daily
day	count monday restaurant
day of week	revenue by day of week last 6 months
day of week	count shipments Monday
n days for each month	sales last 2 days for each month
n days for each quarter	revenue last 15 days for each quarter
n days for each week	total sold last 2 days for each week
n days for each year	revenue last 300 days for each year
growth of ... by ... daily	growth of sales by order date daily
growth of ... by ... monthly	growth of sales by date shipped monthly sales > 24000
growth of ... by ... quarterly	growth of sales by date shipped quarterly
growth of ... by ... weekly	growth of sales by receipt date weekly for pro-ski2000
growth of ... by ... yearly	growth of sales by date closed yearly
growth of ... by ...	growth of sales by order date
n hours for each day	sales last 2 hours for each day

Keyword	Examples
last day by	customers last day by referrer
last month by	customers last month by day
last <i>n</i> days	visitors last 7 days
last <i>n</i> quarters	visitors last 2 quarters by month by campaign
last <i>n</i> weeks	visitors last 10 weeks by day
last quarter	customers last quarter sale > 300
last week	customers last week by store
last year	top 10 customers last year by sale by store for region west
last <i>n</i> years	visitors last 5 years by revenue for sum revenue > 5000
month to date	sales by product month to date sales > 2400
<i>month year</i>	commission by sales rep February 2014
<i>month</i>	commission January
month	revenue by month last year
monthly year-over-year	growth of revenue by receipt date monthly year-over-year
monthly	commission > 10000 monthly
<i>n</i> months for each quarter	cost last 2 months for each quarter
<i>n</i> months for each year	last 8 months for each year
<i>n</i> days ago	sales 2 days ago
<i>n</i> months ago	sales 2 months ago by region
<i>n</i> months	visitors last 6 months for homepage visits > 30 by month
<i>n</i> quarters ago	sales 4 quarters ago by product name contains deluxe
<i>n</i> weeks ago	sales 4 weeks ago by store
<i>n</i> years ago	sales 5 years ago by store for region west
<i>n</i> years	opportunities next 5 years by revenue

Keyword	Examples
next day	shipments next day by order
next month	appointments next month by day
next n days	shipments next 7 days
next n months	openings next 6 months location
next n quarters	opportunities next 2 quarters by campaign
next n weeks	shipments next 10 weeks by day
next quarter	opportunities next quarter amount > 30000
next week	shipments next week by store
next year	opportunities next year by sales rep
quarter to date	sales by product quarter to date for top 10 products by sales
quarterly year-over-year	growth of revenue by date shipped quarterly year-over-year
quarterly	sales quarterly for each product
n quarters for each year	last 2 quarters for each year
today	sales today by store
week to date	sales by order date week to date for pro-ski200
week	revenue by week last quarter
weekly year-over-year	growth of revenue by date shipped weekly year-over-year
weekly	revenue weekly
n weeks for each month	sales last 3 weeks for each month
n weeks for each quarter	last 2 weeks for each quarter
n weeks for each year	last 3 weeks for each year
year to date	sales by product year to date
year	revenue by product 2014 product name contains snowboard

Keyword	Examples
yearly	shipments by product yearly
yesterday	sales yesterday for pro-ski200 by store

Time

Keyword	Examples
detailed	ship time detailed
last minute	count homepage views last minute
last hour	count unique visits last hour
n minutes	count visitors last 30 minutes
n hours	count visitors last 12 hours
hourly	visitors by page name hourly
n minutes ago	sum inventory by product 10 minutes ago
n hours ago	sum inventory by product by store 2 hours ago

Text

Keyword	Examples
begins with	product name begins with 'pro'
contains	product name contains "alpine" description contains "snow shoe"
ends with	product name ends with 'deluxe'

Keyword	Examples
not begins with	product name not begins with "tom's"
not contains	product color not contains 'tan' product color not contains 'red'
not ends with	product name not ends with "trial"
similar to	course name similar to 'hand'
not similar to	course name not similar to 'hand'

Number

Function	Examples
sum	sum revenue
average	average revenue by store
count	count visitors by site
max	max sales by visitor by site
min	min revenue by store by campaign for cost > 5000
standard deviation	standard deviation revenue by product by month for date after 10/31/2010
unique count	unique count visitor by product page last week
variance	variance sale amount by visitor by product for last year

Comparative

Function	Examples
all	all

Function	Examples
between... and	revenue between 0 and 1000
vs, versus	revenue east vs west
>	sum sale amount by visitor by product for last year sale amount > 2000
<	unique count visitor by product by store for sale amount < 20
>=	count calls by employee lastname >= m
<=	count shipments by city latitude <= 0
=	unique count visitor by store purchased products = 3 for last 5 days
!=	sum sale amount region != canada region != mexico

Location

Keyword	Examples
near	revenue store name county near san francisco
near... within <i>n</i> miles km meters	revenue store name county near alameda within 50 miles
farther than <i>n</i> miles km meters from	average hours worked branch farther than 80 km from scarborough

Location keywords only work for searches where the data source includes latitude/longitude data.

Period

Keyword	Example
quarter (<i>date</i>)	quarter (purchase date)
quarter of year (<i>date</i>)	quarter of year (purchase date)

month of quarter (<i>date</i>)	month of quarter (purchase date)
week of year (<i>date</i>)	week of year (ship date)
week of quarter (<i>date</i>)	week of quarter (ship date)
week of month (<i>date</i>)	week of month (ship date)
day of year (<i>date</i>)	day of year (ship date)
day of quarter (<i>date</i>)	day of quarter (ship date)
day (<i>date</i>)	day (ship date)
day of month (<i>date</i>)	day of month (order date)
day of week (<i>date</i>)	day of week (order date)
hour (<i>datetime</i>)	hour (timestamp)

In

Keyword	Example
in (<i>subsearch</i>)	in (top 10 store name by sales footwear)

TQL reference

TQL is the ThoughtSpot language for entering SQL commands. This reference lists TQL commands you can use to do things like creating a schema or verifying a data load.

About using TQL

You can use TQL either [through the ThoughtSpot application's web interface](#) or the [command line interface](#) in the Linux shell.

Use `--query_results_apply_top_row_count <number>` flag to limit the number of result rows returned by a query. For example:

```
$ tql --query_results_apply_top_row_count 100
```

As a best practice, you should enclose object names (database, schema, table, and column) in double quotes, and column values in single quotes. When referring to objects using fully qualified object names, the syntax is:

```
"database"."schema"."table"
```

To get help on SQL when using TQL, enter `help` on the command line.

You can use TQL to view and modify schemas and data in tables. Remember to add a semicolon after each command. Commands are not case sensitive but are capitalized here for readability.

Note: Worksheets and pinboards in ThoughtSpot are dependent upon the data in the underlying tables. Use caution when modifying tables directly. If you change or remove a schema on which those objects rely, the objects could become invalid.

View schemas and data

Syntax	Description
SHOW DATABASES	<p>Lists all available databases.</p> <p>Examples:</p> <pre>SHOW DATABASES;</pre>
USE <database>	<p>Switches the context to the specified database. This is required if queries do not use fully qualified names (database.schema.table) for specifying tables.</p> <p>Examples:</p> <pre>USE "fruit_database";</pre>
SHOW SCHEMAS	<p>Lists all schemas within the current database. For example:</p> <pre>SHOW SCHEMAS;</pre>
SHOW TABLES	<p>Lists all tables within the current database by schema. For example:</p> <pre>SHOW TABLES;</pre>
SHOW TABLE <table>	<p>Lists all the columns for a table. For example:</p> <pre>SHOW TABLE "locations";</pre>
SCRIPT SERVER	<p>Generates the TQL schema for all tables in all databases on the server. For example:</p> <pre>SCRIPT SERVER;</pre>

Syntax	Description
<pre>SCRIPT DATABASE <database></pre>	Generates the TQL schema for all tables in a database. For example: <pre>SCRIPT DATABASE "fruit_database";</pre>

<pre>SCRIPT TABLE <table></pre>	Generates the TQL schema for a table. For example:
	<pre>SCRIPT TABLE "vendor";</pre>

Syntax	Description
<pre> SELECT <cols_or_expr> FROM <table_list> [WHERE <predicates>] [GROUP BY <expr>] [ORDER BY <expr>]</pre>	<p>Shows specified set of table data.</p> <p>If you do not specify the TOP number of rows to select, the top 50 rows will be returned by default. The number of rows to return can be set using the TSQL command line flag:</p> <pre>--query_results apply_top_row_count</pre> <p>You can use the following aggregation functions:</p> <ul style="list-style-type: none"> • sum • count • count distinct • stddev • avg • variance • min • max <p>You can use the following date functions:</p> <ul style="list-style-type: none"> • absyear • absmonth • absday • absquarter • date • time

For example:

```

SELECT TOP 10 "quantity" FROM "sales_fact";

SELECT COUNT(*) FROM "vendor";

SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY
"vendor";

SELECT "vendor", SUM("amount") FROM "vendor", "sales_fact"
    WHERE "sales_fact"."vendorid" = "vendor"."vendorid"
        AND "amount" > 100 GROUP BY "vendor" ORDER BY "amount"
DESC;

SELECT "vendor", SUM("quantity") FROM "sales_fact"
GROUP BY "vendor" LIMIT 10;
```

Schema creation

Syntax	Description
<pre>CREATE DATABASE <data- base></pre>	<p>Creates a database. For example:</p> <pre>CREATE DATABASE "fruit_database";</pre>
<pre>CREATE SCHEMA <schema></pre>	<p>Creates a schema within the current database. For example:</p> <pre>CREATE SCHEMA "fruit_schema";</pre>
<pre>CREATE TABLE <table> (<column_def- itions> [<con- straints>]) [PARTI- TION BY HASH (<num- ber>) [KEY ("<col- umn>")]]</pre>	<p>Creates a table with the specified column definitions and constraints.</p> <p>Use PARTITION BY HASH to shard a table across all nodes. If no KEY is specified, the table will be randomly sharded.</p> <p>Do not specify relationship constraints (FOREIGN KEY or RELATIONSHIP) in the CREATE TABLE statement. Instead, define these using ALTER TABLE statements at the end of your TQL script, after creating your tables. This method guarantees that tables are created before they are referenced in the constraint definitions. For example:</p> <pre>CREATE TABLE "vendor" ("vendorid" int, "name" var- char(255)); CREATE TABLE "sales_fact" ("saleid" int, "locationid" int, "vendorid" int, "quantity" int, "sale_amount" double, "fruitid" int, CONSTRAINT PRIMARY KEY("saleid")) PARTITION BY HASH(96) KEY ("saleid");</pre>

Schema modification

Syntax	Description
<code>DROP DATABASE <data-base></code>	Drops a database and all of its schemas and tables. For example: <code>DROP DATABASE "fruit_database";</code>
<code>DROP SCHEMA <schema></code>	Drops a schema within the current database, and drops all of the tables in the schema. For example: <code>DROP SCHEMA "fruit_schema";</code>
<code>DROP TABLE <table></code>	Drops a table. For example: <code>DROP TABLE "location";</code>
<code>TRUNCATE TABLE <table></code>	Removes all data from a table, but preserves its metadata, including all GUIDs, relationships, etc. This can be used to force a new schema for a table without losing the metadata. However, this operation removes all existing data from the table and must be used with caution. You must reload the data following a <code>TRUNCATE</code> , or all dependent objects (worksheets and pinboards) in ThoughtSpot will become invalid. For example: <code>TRUNCATE TABLE "location";</code>

Syntax	Description
<pre>ALTER TABLE <table> ADD DROP RENAME COLUMN <column></pre>	<p>Alters a table to add, drop, or rename a column.</p> <p>When you add a column to an existing table, you must provide a default value to use for existing rows. For example:</p> <pre>ALTER TABLE "cart" ADD COLUMN "nickname" varchar(255) DE- FAULT 'no nickname';</pre> <pre>ALTER TABLE "cart" DROP COLUMN "nickname";</pre> <pre>ALTER TABLE "cart" RENAME COLUMN "nickname" TO "shortname";</pre>

ALTER TABLE <table> DROP CON- STRAINT PRIMARY KEY;	Drops the primary key from a table. Note that if you then add a new primary key, the same upsert behavior will be applied as with adding any primary key. This can result in data deletion, so make sure you understand how the upsert will affect your data ahead of time. For example:
	<pre>ALTER TABLE "sales" DROP CONSTRAINT PRIMARY KEY;</pre> <pre>ALTER TABLE "sales" ADD CONSTRAINT PRIMARY KEY ("P0_num- ber");</pre>

ALTER TABLE <table> DROP CON- STRAINT [FOREIGN KEY RELATIONSHIP] <name>;	Drops the named foreign key or relationship between two tables. For example:
	<pre>ALTER TABLE "sales_fact" DROP CONSTRAINT FOREIGN KEY "FK_P0_number";</pre> <pre>ALTER TABLE "fruit_dim" DROP RELATIONSHIP "REL_dates";</pre>

Syntax	Description
<pre>ALTER TABLE <table> DROP [CONSTRAINT FOREIGN KEY [<table_name>] RELATIONSHIP [WITH <table_name>];</pre>	<p>You must use this syntax when dropping relationships between tables created before ThoughtSpot version 3.2. This is because relationships could not be named in older versions.</p> <p>Drops the foreign key or relationship between two tables where you cannot reference it by relationship name. If the relationship was created without a name, use:</p> <ul style="list-style-type: none"> • the name of the referenced table, for a foreign key. • the name of the related table, for a relationship. <p>If you drop a foreign key without specifying the referenced table, all foreign keys from the table you are altering will be dropped.</p>

Examples:

```
ALTER TABLE "shipments" DROP CONSTRAINT FOREIGN KEY "orders";
```

```
ALTER TABLE "wholesale_buys" DROP RELATIONSHIP WITH "retail_sales";
```

Drops all relationships that have wholesale_buys as a source.

```
ALTER TABLE "wholesale_buys" DROP RELATIONSHIP;
```

Drops all foreign keys from wholesale_buys.

```
ALTER TABLE "wholesale_buys" DROP CONSTRAINT FOREIGN KEY;
```

Syntax	Description
<pre>ALTER TABLE <table> [SET DI- MENSION SET FACT [PARTITION BY HASH [(<shards>)] [KEY(<column>)]]]</pre>	<p>Changes the partitioning on a table by doing one of:</p> <ul style="list-style-type: none"> • re-sharding a sharded table • changing a replicated table to a sharded table • changing a sharded table to a replicated (unsharded) table <p>By default, ThoughtSpot does not shard dimension tables.</p>
	<p>To change the partitioning on a table, or to change a dimension table to a sharded table, use <code>ALTER TABLE...SET FACT PARTITION BY HASH...;</code></p> <p>To make a sharded table into a dimension table (replicated on every node), use <code>ALTER TABLE...SET DIMENSION;</code> command.</p>
	<p>Examples of this statement:</p> <pre>ALTER TABLE "sales_fact" SET FACT PARTITION BY HASH (96) KEY ("PO_number"); ALTER TABLE "fruit_dim" SET DIMENSION;</pre>

<pre>ALTER TABLE <table> MODIFY COLUMN <column> <new_data_type>;</pre>	<p>Changes the data type of a column. This can have implications on sharding and primary key behavior. See About data type conversion. For example:</p> <pre>ALTER TABLE fact100 MODIFY COLUMN product_id int;</pre>
--	--

Modify data

Syntax	Description
<pre>INSERT INTO <table> VALUES ...</pre>	<p>Inserts values into a table. Only use this for testing. Do not use <code>INSERT</code> on a production system. For example:</p> <pre>INSERT INTO "vendor" VALUES ('helen rose', 'jacob norse', 'eileen ruff', 'manny gates');</pre>

Syntax	Description
<pre>ALTER TABLE <table> SET LOAD PRIORITY <value> <new_da- ta_type>;</pre>	<p>Sets the load priority for a table. Load priority determines the order in which a table is loaded on a cluster restart. You can set any value from 1–100 . The system default for all tables is 50 . For example:</p> <pre>ALTER TABLE 'sales_facts' SET LOAD PRIORITY 1;</pre>
<pre>UPDATE <table> ... SET ... [WHERE ...]</pre>	<p>Updates rows in a table that match optionally provided predicates. Predicates have the form column = value connected by the AND keyword. Sets the column values to the specified values. For example:</p> <pre>UPDATE "location" SET "borough" = 'staten island', "city" = 'new york' WHERE "borough" = 'staten isl' AND city = 'NY';</pre>
<pre>DELETE FROM <table> [WHERE...]</pre>	<p>Deletes rows from a table that match optionally provided predicates. Predicates have the form column = value connected by the AND keyword. For example.</p> <pre>DELETE FROM "vendor" WHERE "name" = 'Joey Smith' AND "ven- dorid" = '19463';</pre>

Constraints and relationships

Constraints and relationships in ThoughtSpot are used to define the relationships between tables (how they can be joined). However, constraints are not enforced, as they would be in a transactional database. You can define the following constraints when creating a table with `CREATE TABLE`, or add them to an existing table using the `ADD CONSTRAINT` syntax:

Syntax	Description
PRIMARY KEY	<p>Designates a unique, non-null value as the primary key for a table. This can be one column or a combination of columns. If values are not unique, an upsert will be performed if a row includes a primary key that is already present in the data. Some examples:</p> <pre>CREATE TABLE "schools" ("schoolID" varchar(15), "schoolName" varchar(255), "schoolCity" varchar(55), "schoolState" varchar(55), "schoolNick" varchar(55), CONSTRAINT PRIMARY KEY ("schoolID")) ;</pre> <pre>ALTER TABLE "cart" ADD CONSTRAINT PRIMARY KEY ("cart_id");</pre> <pre>ALTER TABLE "cart" DROP CONSTRAINT PRIMARY KEY "cart_id";</pre>

FOREIGN KEY

Defines a relationship where the value(s) in the table are used to join to a second table. Uses an equality operator. The foreign key must match the primary key of the table that is referenced in number, column type, and order of columns.

When creating a foreign key, give it a name. You can reference the foreign key name later, if you want to remove it.

Examples of this statement:

```
ALTER TABLE "batting" ADD CONSTRAINT "FK_player" FOREIGN KEY ("playerID")
REFERENCES "players" ("playerID");
```

```
ALTER TABLE "batting" ADD CONSTRAINT "FK_lg_team" FOREIGN KEY ("lgID" , "teamID")
REFERENCES "teams" ("lgID" , "teamID");
```

```
ALTER TABLE "shipment" ADD CONSTRAINT "FK_P0_vendor" FOREIGN KEY ("po_number",
"vendor") REFERENCES "orders" ("po_number", "vendor");
```

```
ALTER TABLE "shipment" DROP CONSTRAINT "FK_P0_vendor";
```

Syntax	Description
RELATIONSHIP	<p>Defines a relationship where the value(s) in the table can be used to join to a second table, using an equality condition (required) and one or more range conditions (optional). These conditions act like a WHERE clause when the two tables are joined. They are applied using AND logic, such that all conditions must be met for a row to be included.</p> <p>You may add multiple relationships between tables. When creating a relationship, give it a name. You can reference the relationship name later, if you want to remove it.</p> <p>Examples of this statement:</p> <pre>ALTER TABLE "wholesale_buys" ADD RELATIONSHIP "REL_fruit" WITH "retail_sales" AS "wholesale_buys"."fruit" = "retail_sales"."fruit" AND ("wholesale_buys"."date_order" < "retail_sales"."date_sold" AND "retail_sales"."date_sold" < "wholesale_buys"."expire_date"); ALTER TABLE "wholesale_buys" DROP RELATIONSHIP "REL_fruit";</pre>

Data types

ThoughtSpot supports a simplified list of data types:

Syntax	Description	Examples
Character	<ul style="list-style-type: none"> VARCHAR(<i>n</i>) 	Specify the maximum number of characters, as in VARCHAR(255). The size limit is 1GB for VARCHAR values.
Floating point	<ul style="list-style-type: none"> DOUBLE FLOAT 	DOUBLE is recommended.
Boolean	<ul style="list-style-type: none"> BOOL 	Can be true or false .

Syntax	Description	Examples
Integer	<ul style="list-style-type: none">• INT• BIGINT	<p>INT holds 32 bits.</p> <p>BIGINT holds 64 bits.</p>
Date or time	<ul style="list-style-type: none">• DATE• DATETIME• TIMESTAMP• TIME	<p>DATETIME, TIMESTAMP, and TIME are stored at the granularity of seconds</p> <p>TIMESTAMP is identical to DATETIME, but is included for syntax compatibility.</p>

tsload flag reference

For recurring data loads and for scripting loads, use `tsload` (the ThoughtSpot Loader). This reference section lists all the flags that can be used to modify the behavior of `tsload`.

General tsload flags

Flag	Description	Notes
<code>--target_database <database></code>	Specifies the pre-existing target database into which tsload should load the data.	
<code>--target_schema <schema></code>	Specifies the target schema. Default is “falcon_default_schema”.	
<code>--target_table <table></code>	Specifies the tables that you want to load into the database. The tables must exist in the database specified by <code>--target_database</code> .	
<code>--empty_target</code>	Specifies that any data in the target table is to be removed before the new data is loaded.	If supplied, any rows that exist in the table specified by <code>--target_database</code> and <code>--target_table</code> will be deleted before this data load. To perform an “upsert” on the existing data, omit this flag or specify <code>--noempty_target</code> .
<code>--max_ignored_rows <number></code>	Specifies the maximum number of rows that can be ignored if they fail to load.	If the number of ignored rows exceeds this limit, the load will be aborted.
<code>--bad_records_file <path_to_file>/<file_name></code>	Specifies the file to use for storing rows that failed to load. Input rows that do not conform to the defined schema in ThoughtSpot will be ignored and inserted into this file.	
<code>--date_format <date_formatmask></code>	Specifies the format string for date values.	The default format is <code>yearmonthday</code> e.g. “Dec 30th, 2001” and is represented as <code>20011230</code> . Use the date format specifications supported in the strftime library function .
<code>--date_time_format <date_formatmask>/<time_formatmask></code>	Specifies the format string for datetime values.	The default is <code>yearmonthday hour:minute:second</code> e.g. Dec 30th, 2001 1:15:12 and is represented as <code>20011230 01:15:12</code> . Use the datetime format specifications supported in the strftime library function .
<code>--time_format <time_formatmask></code>	Specifies the format string for time values.	The default is <code>hour:minute:second</code> . Use the time format specifications supported in the strftime library function .

Flag	Description	Notes
--v=[0 1 2 3]	Specifies the verbosity of log messages.	Provide a value for verbosity level. By default, verbosity is set to the minimum, which is 0. This value is similar to a volume control. At higher levels your log receives more messages and that log more frequently. This is used for debugging. You should not change this value unless instructed by ThoughtSpot Support.
--skip_second_fraction	Skips fractional seconds when loading data.	If supplied, the upserts logic may be affected, especially if the date time being loaded is a primary key, and the data has millisecond granularity. Load the data twice, first time as a string with a primary key, and again with second granularity date time. There is no support to store fractional seconds in the ThoughtSpot system.

File loading tsload flags

The following flags are used when loading data from an input file:

Flag	Description	Notes
--source_file <path_to_file>/<file_name>	Specifies the location of the file to be loaded.	
--source_data_format [csv delimited]	Specifies the data file format.	Optional. The default is csv.
--field_separator "<delimiter>"	Specifies the field delimiter used in the input file.	
--trailing_field_separator	Specifies that the field separator appears after every field, including the last field per row.	Example row with trailing field separator: a,b,c,The default is false.
--null_value "<null_representation>"	Specifies how null values are represented in the input file.	These values will be converted to NULL upon loading.
--date_converted_to_epoch [true false]	Specifies whether the “date” or “datetime” values in the input file are represented as epoch values.	

Flag	Description	Notes
--boolean_representation [true_false 1_0 T_F Y_N]	Specifies the format in which boolean values are represented in the input file.	The default is T_F. You can also use this flag to specify other values. For example, if your data used Y for true and NULL for false, you could specify: --boolean_representation Y_NULL
--has_header_row	Indicates that the input file contains a header row.	If supplied, column names in the header row are used to match column names in the target table in ThoughtSpot. If not supplied, the first row of the file is loaded as data, the same as all subsequent rows.
--escape_character "<character>"	Specifies the escape character used in the input file.	If no value is specified, the default is "(double quotes).
--enclosing_character "<character>"	Specifies the enclosing character used in the input file.	If the enclosing character is double quotes, you need to escape it, as in this example: --enclosing_character "\""
--use_bit_boolean_values = [true false]	Specifies how boolean values are represented in the input file.	If supplied, the input CSV file uses a bit for boolean values, i.e. the false value is represented as 0x0 and true as 0x1. If omitted or set to false, boolean values are assumed to be T_F, unless you specify something else using the flag --boolean_representation [true_false 1_0 T_F Y_N] .

tscli command reference

Summary: The ThoughtSpot command line interface, or tscli, is an administration interface for the cluster. Use tscli to take snapshots (backups) of data, apply updates, stop and start the services, and view information about the system. This reference defines each subcommand.

The command returns `0` on success, and a non-zero exit codes on failure. The `tscli` command logs errors to the `stderr` directory.

How to use the tscli command

The `tscli` command has the following syntax:

```
tscli [-h]
      [--helpfull]
      [--verbose]
      [--noautoconfig]
      [--autoconfig]
      [--yes]
      [--cluster ]
      [--zoo ]
      {access, alert, ansible, backup, backup-policy, calendar, callhome, cassandra,
       cluster, command, dr-mirror, etl, event, feature, fileserver,
       firewall, hdfs, ipsec, ldap, logs, map-tiles, monitoring, nas,
       node, notification, onboarding, patch, rpackage, saml, scheduled-pinboards, set, smtp, snapshot,
       snapshot-policy, socialproof spot, sssd, ssl, storage, support,
       tokenauthentication}
```

The `tscli` command has several subcommands, such as `alert`, `backup`, and so on.

Issue subcommands using the following format:

```
tscli [subcommand]
```

Subcommands have their own additional options and actions, such as `tscli backup create` or `tscli backup delete`.

Each subcommand may have several options.

To view help for a subcommand, type `-h` after the subcommand option:

```
tscli [subcommand] -h
```

tscli subcommands

This section lists each subcommand and its syntax.

access

```
tscli access [-h] {list}
```

This subcommand has the following option:

tscli access list

Lists objects by last access time, with the following parameters:

--type TYPE

Type of object, either answer or pinboard.

--limit LIMIT

The number of objects to fetch.

The default is 30.

--offset OFFSET

Offset to use to skip objects for batched results.

The default is 0.

--ascending

Sorts the answers by access time ascending.

The default is `True`.

alert

```
tscli alert [-h] {count,info,list,off,on,refresh,silence,status,unsilence}
```

This subcommand has the following options:

tscli alert count

Lists counts of generated alerts by type.

tscli alert info

Lists all alerts. Add `silenced` to list only silenced alerts, `active` to list only active alerts, or `detailed` to get detailed alert information.

tscli alert list

Lists the generated alerts, with these parameters:

--limit LIMIT

Specifies the number of recent alerts to display.

--since SINCE

Lists all alerts raised since a specified time period, in the form of a human readable duration string, such as 4h (4 hours) or 4m (4 minutes).

tscli alert off

Disables all alerts from the cluster in the cluster's timezone.

tscli alert on

Enables alerts from the cluster.

tscli alert refresh

Refreshes alert metadata on the cluster.

tscli alert silence --name NAME

Silences the alert with `NAME`. For example, `DISK_ERROR`. Silenced alerts are still recorded in postgres, however emails are not sent out.

tscli alert status

Shows the status of cluster alerts.

```
tscli alert unsilence --name NAME
```

Unsilences the alert with `NAME`. For example, `DISK_ERROR`.

ansible

```
tscli ansible [-h] {checkout,commit} [--local]
```

This subcommand has the following options:

```
tscli ansible checkout --host HOST
```

Checks out Ansible playbook with the target `HOST` that is running the `ts_ansible` service.

```
tscli ansible commit --host HOST
```

Commits Ansible playbooks with the target `HOST` that is running the `ts_ansible` service.

Use this subcommand to install and configure third-party software on the ThoughtSpot cluster.

For details, see these articles:

- [About third party security and monitoring software](#)
- [Installing third party security and monitoring software](#)

backup

```
tscli backup [-h] {create,delete,ls,restore}
```

This subcommand has the following options:

```
tscli backup create [-h] [--mode {full,light,dataless}] [--type {full,incremental}]  
[--base BASE]  
[--storage_type {local,nas}] [--remote] [--no-orion-master]
```

Pulls a snapshot and saves it as a backup, with these parameters:

```
--mode {full,light,dataless}
```

Mode of backups.

The default is `full`.

--type {full,incremental}

Type of backup.

Note: `incremental` is not implemented.

The default setting is `full`.

--base BASE

Based snapshot name for incremental backup.

Note: Because `incremental` is not implemented, neither is this option.

There is no default setting.

--storage_type {local,nas}

Storage type of output directory.

The default setting is `local`.

--remote

Takes backup through orion master.

The default setting is `True`.

--no-orion-master

Determines whether orion master is available during backup.

The default is `False`.

tscli backup delete name

Deletes the named backup.

tscli backup ls

Lists all backups taken by the system.

tscli backup restore

Restores cluster using backup, with the following parameters:

--release RELEASE

Restore the cluster on a specific release number.

--disable_rotate_keys

Disables cluster rotate key configurations.

The default is `False`.

--enable_cloud_storage

Determines whether to enable Cloud Storage setup.

--heterogeneous

Should be set for heterogeneous clusters.

The default is `False`.

backup-policy

```
tscli backup-policy [-h] {create,delete,disable,enable,ls,show,status,update}
```

Manages the backup policy.

This subcommand has the following options:

tscli backup-policy create

Prompts an editor for you to edit the parameters of a new periodic backup policy, with the following parameter:

--config CONFIG

Specifies the text format of the periodic backup policy config.

tscli backup-policy delete name

Deletes the backup policy `name`.

tscli backup-policy disable name

Disables the policy `name`.

tscli backup-policy enable name

Enables the policy `name`.

tscli backup-policy ls

Lists backup policies.

tscli backup-policy show name

Shows the backup policy `name`.

tscli backup-policy status name

Shows the status of the backup policy `name`.

tscli backup-policy update name

Prompts an editor for you to edit the backup policy `name`.

calendar

```
tscli calendar [-h] {create,delete,disable,enable,generate,ge  
t,list,update}
```

This subcommand has the following options:

tscli calendar create

Creates a new custom calendar, with the following parameters:

--file_path FILE_PATH

Path to the CSV file holding custom calendar data.

The default is `None`.

--name NAME

Custom calendar name.

The default is `None`.

--separator SEPARATOR

The separator used in the CSV file.

The default is `,`.

--no-header-row

Flag to indicate that the CSV file has no header row.

The default is `True`.

--username USERNAME

The admin username for ThoughtSpot login.

The default is `None`.

tscli calendar delete

Deletes a custom calendar table from the system, with the following parameters:

--name NAME

Deletes the custom calendar NAME.

The default is `None`.

--username USERNAME

The admin username for ThoughtSpot login.

The default is `None`.

tscli calendar disable

Disables custom calendar on the cluster.

tscli calendar enable

Enables custom calendar on the cluster.

tscli calendar generate

Creates a custom calendar table based on given specifications, with the following parameters:

--name NAME

A name to create the custom calendar CSV file with.

The default is `None`.

--start_date START_DATE

The start date to begin the custom calendar with in the form mm/dd/yyyy.

The default is `None`.

--end_date END_DATE

The end date to end the custom calendar with in the form mm/dd/yyyy.

The default is `None`.

--calendar_type {MONTH_OFFSET,4-4-5,4-5-4,5-4-4}

The type of custom calendar to create.

The default is `MONTH_OFFSET`.

--month_offset

`{January,February,March,April,May,June,July,August,September,October,November,December}`

The month offset to start the year from, if the calendar is the MONTH_OFFSET type.

The default is `January`.

--start_day_of_week

The day the week starts on.

The default is `Sunday`.

--quarter_name_prefix

The string to prefix a quarter name with.

--year_name_prefix YEAR_NAME_PREFIX

The string to prefix a year name with.

--username USERNAME

The admin username for ThoughtSpot login.

The default is `None`.

tscli calendar get

Procures data of a custom calendar as a CSV file, with the following parameters:

--name NAME

Procures data of custom calendar `NAME`

The default is `None`.

--username USERNAME

Admin username for ThoughtSpot login.

The default is `None`.

tscli calendar list

Procures a list of custom calendars present in the cluster, with the following parameter:

--username USERNAME

Admin username for ThoughtSpot login.

The default is `None`.

tscli calendar update

Updates a custom calendar table in the system, with the following parameters:

--file_path FILE_PATH

Path to the CSV file holding custom calendar data.

The default is `None`.

--name NAME

Custom calendar name.

The default is `None`.

--separator SEPARATOR

The separator used in the CSV file.

The default is `,`.

--no-header-row

Flag to indicate that the CSV file has no header row.

The default is `True`.

--username USERNAME

The admin username for ThoughtSpot login.

The default is `None`.

callhome

```
tscli callhome [-h] {disable,enable,generate-bundle}
```

This subcommand has the following options:

tscli callhome disable

Turns off the periodic call home feature.

tscli callhome enable --customer_name CUSTOMER_NAME

Enables the "call home" feature, which sends usage statistics to ThoughtSpot.

This feature is enabled by default.

The parameter `customer_name` takes the form `Shared/CUSTOMER_NAME/stats`.

The default is `None`.

tscli callhome generate-bundle [--d D] [--since SINCE]

Generates the callhome stats tar file, with the following parameters:

--d D

Destination folder for the tar file.

There is no default setting.

--since SINCE

Grabs `callhome` data from the specified time window in the past.

This should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

This option generates a `tar` file of the cluster metrics and writes it to the specified directory, where `SINCE` is how many days back the file must start.

There is no default setting.

cassandra

```
tscli cassandra [-h] {backup,restore}
```

Backs up cassandra.

This subcommand has the following options:

tscli cassandra backup

Takes a backup of cassandra, with the following parameters:

--keyspaces KEYSPACES

Comma separated list of keyspaces to take a backup of.

The default is `None`.

backup_dir BACKUP_DIR

The path to the backup directory to write the backup.

The default is `None`.

tscli cassandra restore

Restores cassandra from a backup, with the following parameter:

--backup_dir BACKUP_DIR

The path to the backup directory to write the backup.

The default is `None`.

cluster

```
tscli cluster [-h] abort-reinstall-os,abort-update,bucket- name,check,create,download-release,get-config,list-available-releases,list-downloaded-releases,load,reinstall-os,restore,resume-reinstall-os,resume-update,set-config,set-min-resource-spec,setup-release-host,setup-release-host-key,show-resource-spec,start,status,stop,update,update-hadoop}
```

This subcommand has the following options:

tscli cluster abort-reinstall-os

Aborts in-progress reinstall.

tscli cluster abort-update

Aborts an ongoing cluster update, if safe.

tscli cluster bucket-name

Returns the name of the s3 bucket associated with the cluster, if there is one.

tscli cluster check [--path PATH] [--includes INCLUDES] [--retry RETRY] [--localhost] [--disable-events]

Checks the status of all nodes in the cluster, with the following parameters:

--path PATH

Specifies the working directory of the diagnostic tool.

The default is `/usr/local/scaligent/release`.

--includes INCLUDES

Specifies the comma-separated component(s) to be included in the check.

The default is `all`.

--retry RETRY

The maximum number of retry times if the node is unreachable.

The default is `10`.

--localhost

Runs cluster checks only on localhost.

The default is `False`.

--disable-events

Disables raising configuration events.

The default is `False`.

tscli cluster create release

Creates a new cluster from the release file specified by adding the release number.

Used by ThoughtSpot Support when installing a new cluster. For example, `tscli cluster create 5.3.2.tar.gz`.

This command has the following parameters:

--disable_rotate_keys

Disables cluster rotate key configuration.

The default is `False`.

--enable_cloud_storage {s3a,gcs}

Determines whether to enable Cloud Storage setup, and which storage format to use.

heterogeneous

Should be set for heterogeneous clusters.

The default is `False`.

tscli cluster download-release release

Downloads the specified release to the Hadoop Distributed File System (HDFS) for later upgrading.

tscli cluster get-config

Gets current cluster network and time configuration. Prints JSON configuration to `stdout`.

If the system cannot be connected to all interfaces, the command returns an error but continues to function.

This command has the following parameters:

--local

Gets the config for the local host only.

The default is `False`.

--nodes NODES

A comma separated list of specified nodes to get the config for.

The default is `None`.

tscli cluster list-available-releases

Lists the available releases to update to on the cluster.

tscli cluster list-downloaded-releases

Lists the releases downloaded to the cluster.

tscli cluster load backupdir

Loads the state from a specified backup directory onto an existing cluster.

tscli cluster reinstall-os

Reinstalls OS on all nodes of the cluster, with the following parameters:

--secondary SECONDARY

A secondary drive for reinstall.

The default is `sdd`.

--stdin

Command to take JSON configuration from stdin.

tscli cluster restore --release RELEASE backupdir

Restores a cluster on the specified release number using the backup in the specified directory `backupdir`. If you're restoring from a dataless backup, you must supply the release tarball for the corresponding software release.

This command has the following parameters:

--disable_rotate_keys

Disables cluster rotate key configurations.

The default is `False`.

--enable_cloud_storage {s3a,gcs}

Determines whether to enable Cloud Storage setup.

--heterogenous

Should be set for heterogenous clusters.

The default is `False`.

tscli cluster resume-reinstall-os

Resumes in-progress reinstall.

tscli cluster resume-update

Resumes in-progress updates, with the following parameter:

--ignore_if_unhealthy

Comma separated list of node IPs on which upgrade is not attempted if they are found to be unhealthy. If a node outside of this list is found unhealthy, the upgrade is aborted.

The default is `None`.

tscli cluster set-config

Sets cluster network and time configuration. Takes JSON configuration from stdin.

This subcommand has the following parameters:

--ipv4-only

Only use ipv4 for node communication. Requires passing ipMap in config unless no-network-change is also specified.

The default is `False`.

--no-network-change

This flag ensures that a change made with set-config does not update network settings.

The default is `False`.

tscli cluster set-min-resource-spec

Sets the minimum resource configuration of the cluster.

tscli cluster show-resource-spec

Prints default or min.

tscli cluster start

Starts the cluster.

tscli cluster status

Gives the status of the cluster, including release number, date last updated, number of nodes, pending tables time, and services status.

tscli cluster stop

Pauses the cluster (but does not stop storage services).

tscli cluster update

Update existing cluster.

```
tscli cluster update-hadoop
```

Updates Hadoop/Zookeeper on the cluster.

command

```
tscli command [-h] {run}
```

Command to run on all nodes.

This subcommand has the following option:

```
tscli command run [-h] [--nodes NODES] --dest_dir DEST_DIR [--copyfirst COPYFIRST]
[--timeout TIMEOUT] command
```

These are the parameters:

```
--nodes NODES
```

Space-separated IPs of nodes where to run the command.

The default setting is `all`.

```
--dest_dir DEST_DIR
```

Directory to save the files that contain the output from each node.

This is a mandatory parameter.

```
--copyfirst COPYFIRST
```

Copy the executable to required nodes first.

The default setting is `False`.

```
--timeout TIMEOUT
```

Timeout waiting for the command to finish.

The default setting is `60`.

dr-mirror

```
tscli dr-mirror [-h] {start,status,stop}
```

This subcommand has the following options:

```
tscli dr-mirror start
```

Starts a mirror cluster which will continuously recover from a primary cluster.

```
tscli dr-mirror status
```

Checks whether the current cluster is running in mirror mode.

```
tscli dr-mirror stop
```

Stops mirroring on the local cluster.

etl

```
tscli etl [-h] {change-password,disable-lw,download-agent,enabl  
e-lw,show-lw}
```

This subcommand has the following options:

```
tscli etl change-password --admin_username admin_user --username Informatica_user
```

Changes the Informatica Cloud account password used by ThoughtSpot Data Connect.

Required parameters are:

```
--admin_username admin_user
```

Specifies the Administrator username for ThoughtSpot.

```
--username Informatica_user
```

Specifies the username for the Informatica Cloud.

```
tscli etl disable-lw
```

Disables ThoughtSpot Data Connect.

```
tscli etl download-agent
```

Downloads the ThoughtSpot Data Connect agent to the cluster.

```
tscli etl enable-lw [-h] --username USERNAME --thoughtspot_url THOUGHTSPOT_URL --  
admin_username ADMIN_USERNAME [--groupname GROUPNAME] --org_id ORG_ID [--pin_to  
PIN_TO] [--proxy_host PROXY_HOST] [--proxy_port PROXY_PORT] [--proxy_username  
PROXY_USERNAME] [--max_wait MAX_WAIT]
```

Contact ThoughtSpot Support for assistance in setting this up.

Required parameters are:

--username *USERNAME*

Username for Informatica Cloud

--thoughtspot_url *THOUGHTSPOT_URL*

URL to reach thoughtspot.

--admin_username *ADMIN_USERNAME*

Admin username for ThoughtSpot

--groupname *GROUPNAME*

--org_id *ORG_ID*

Specifies the Informatica `id` of the company.

--pin_to *PIN_TO*

Specifies the IP address to pin to. If you specify an IP to pin to, that node becomes sticky to the Informatica agent, and will always be used. Defaults to the public IP address of the localhost where this command was run.

--proxy_host *PROXY_HOST*

Proxy server host for network access.

--proxy_port *PROXY_PORT*

Proxy server port.

--proxy_username *PROXY_USERNAME*

Proxy server username.

--max_wait *MAX_WAIT*

Maximum time in seconds to wait for Data Connect agent to start.

tscli etl show-lw

Shows the status of ThoughtSpot Data Connect.

It also returns the Informatica username and OrgId.

event

tscli event [-h] {list}

This subcommand and its options manage event notifications.

This subcommand has the following option:

tscli event list [-h] [--include *INCLUDE*] [--since *SINCE*] [--from *FROM*] [--to *TO*] [--

```
limit LIMIT] [--detail] [--summary_contains SUMMARY_CONTAINS] [--detail_contains
DETAIL_CONTAINS] [--attributes ATTRIBUTES]
```

The `event` subcommand accepts these optional flags:

--include INCLUDE

Options are all, config, notification. Default config. (default: config)

--since SINCE

Grab events from this time window in the past. Should be a human readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

--from FROM

Begin timestamp. Must be of the form: `yyyymmdd-HH:MM`.

--to TO

End timestamp. Must be of the form: `yyyymmdd-HH:MM`.

--limit LIMIT

Maximum number of events to fetch.

The default setting is `0`.

--detail

Print events in detail format. This is not tabular. Default is a tabular summary.

The default setting is `False`.

--summary_contains SUMMARY_CONTAINS

Summary of the event will be checked for this string. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matchesALL`. Put single quotes around the param value to prevent undesired glob expansion.

--detail_contains DETAIL_CONTAINS

Details of the event will be checked for this string. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matches ALL`. Put single quotes around the param value to prevent undesired glob expansion.

--attributes ATTRIBUTES

Specify attributes to match as key=value. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matches ALL`. Put single quotes around the param value to prevent undesired glob expansion.

feature

```
tscli feature [-h] {get-all-config}
```

This subcommand has the following option:

tscli feature get-all-config

Gets the configured features in a cluster. The command returns a list of features, such as custom branding, Data Connect, and call home, and informs whether they are enabled or disabled.

fileserver

```
tscli fileserver [-h] {configure,download-release,purge-config,show-config,upload}
```

This subcommand has the following options:

tscli fileserver configure [-h] --user *USER* [--password *PASSWORD*]

Configures the secure file server username and password for file upload/download, and the call home feature.

You only have to issue this command one time, to set up the connection to the secure file server. Reissue this command if the password changes.

The parameter *PASSWORD* is optional. If a password is not specified, you will be prompted to enter it.

tscli fileserver download-release [-h] [--user *USER*] [--password *PASSWORD*] *release*

Downloads the specified release file, including its checksum, and verifies the integrity of release bundle.

You must specify the exact release number, such as `5.1.3`.

Before using this command for the first time, you must set up the file server connection using `tscli fileserver configure`. You can then work with a member of the ThoughtSpot Support team because a privileged user and a corresponding password must be specified to download releases.

tscli fileserver purge-config

Removes the file server configuration.

tscli fileserver show-config

Shows the file server configuration.

```
tscli fileserver upload [-h] [--user USER] [--password PASSWORD] --file_name FILE_NAME --server_dir_path SERVER_DIR_PATH
```

Uploads the specified file to the directory specified on the secure file server.

You may optionally specify the `user` and `password` to bypass the credentials specified when configuring the file server connection with `tscli fileservice configure`. Before using this command for the first time, you must set up the file server connection using `tscli fileservice configure`.

This uses the following flags:

--user *USER*

Username of the fileserver.

--password *PASSWORD*

Password of the fileserver. This is required and the command prompts you for it if you do not supply it.

--file_name *FILE_NAME*

Local file to upload.

--server_dir_path *SERVER_DIR_PATH*

Directory path on fileserver. The `SERVER_DIR_PATH` parameter specifies the directory for file upload. It is based on customer name, and takes the form `/Shared/support/customer_name`.

firewall

```
tscli firewall [-h] {close-ports,disable,enable,open-ports,stat  
us}
```

This subcommand has the following options:

tscli firewall close-ports

Closes specified ports through firewall on all nodes.

Accepts a comma-separated list of ports. Only closes ports that were previously opened using `open-ports`, and ignores ports that were not opened with `open-port`, or closed ports.

Some essential ports are always kept open, such as `ssh`; they are not affected by this command or by `open-ports`.

tscli firewall disable

Disable firewall.

tscli firewall enable

Enable firewall.

tscli firewall open-ports --ports *ports*

Opens specified ports through a firewall on all nodes.

Accepts a comma-separated list of ports.

Ignores open ports.

Some essential ports are always kept open, such as `ssh`; they are not affected by this command or by `close-ports`.

tscli firewall status

Shows whether firewall is currently enabled or disabled.

hdfs

```
tscli hdfs [-h] {leave-safemode}
```

This subcommand has the following option:

tscli hdfs leave-safemode

Command to get HDFS namenodes out of `safemode`.

ipsec

```
tscli ipsec [-h] {disable,enable,status}
```

This subcommand has the following options:

tscli ipsec disable

Disable IPSec

tscli ipsec enable

Enable IPSec

tscli ipsec status

Show IPSec status on all nodes

ldap

```
tscli ldap [-h] {add-cert,configure,purge-configuration}
```

This subcommand has the following options:

tscli ldap add-cert *name certificate*

Adds an SSL certificate for LDAP. Use only if LDAP has been configured without SSL and you wish to add it. Use `name` to supply an alias for the certificate you are installing.

tscli ldap configure

Configures LDAP using an interactive script.

tscli ldap purge-configuration

Purges (removes) any existing LDAP configuration.

logs

```
tscli logs [-h] {collect,runcmd}
```

Manages the logging behavior.

This subcommand has the following options:

```
tscli logs collect [-h] [--include INCLUDE] [--exclude EXCLUDE] [--since SINCE] [--from FROM] [--to TO] [--out OUT] [--maxsize MAXSIZE] [--sizeonly] [--nodes NODES]
```

Extracts logs from the cluster. Does not include any logs that have been deleted due to log rotation.

These parameters have the following values:

--include INCLUDE

Specifies a comma separated list of logs to include. Each entry is either a "selector" or a glob for matching files. Selectors must be among: `all`, `orion`, `system`, `ts`. Anything starting with `/` is assumed to be a glob pattern, and it is interpreted through `find(1)`. Other entries are ignored. Put single quotes around the param value to prevent undesired glob expansion (default: `all`)

--exclude EXCLUDE

Comma separated list of logs to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include`.

--since SINCE

Grab logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None)

--from FROM

Timestamp where collection begins; must be of the form `yyyymmdd-HH:MM`.

--to TO

Timestamp where collection ends; must be of the form `yyyymmdd-HH:MM`.

--out OUT

Tarball path for writing logs from each node.

The default setting is `/tmp/logs.tar.gz`.

--maxsize MAXSIZE

Only fetch logs if size is smaller than this value. Can be specified in megabytes or gigabytes, such as `100MB`, `10GB`.

--sizeonly

Do not collect logs. Just report the size.

The default setting is `False`.

--nodes NODES

Comma separated list of nodes from where to collect logs. Skip this to use all nodes.

```
tscli logs runcmd [-h] --cmd CMD [--include INCLUDE] [--exclude EXCLUDE] [--since SINCE] [--from FROM] [--to TO] [--outfile OUTFILE] [--outdir OUTDIR] [--cmd_infmt CMD_INFMT] [--cmd_outfmt CMD_OUTFMT] [--nodes NODES]
```

Runs a Unix command on logs in the cluster matching the given constraints. Results are reported as text dumped to standard out, the specified output file, or as tarballs dumped into the specified directory.

Accepts these optional flags:

--cmd CMD

Unix-Command to be run on the selected logs. Use single quotes to escape spaces etc. Note the language used to specify CMDSTR has following rules.

- A logfile and its corresponding result file can be referred by keywords `SRCFILE` and `DSTFILE`. For example, `cp SRCFILE DSTFILE`.
- Without any reference to `DSTFILE` in CMDSTR, `DSTFILE` will be appended to CMDSTR for output redirection. For example, `du -sch SRCFILE` gets auto-transtalted to `du -sch SRCFILE > DSTFILE`.
- Without any reference to `SRCFILE`, content of log is streamed to CMDSTR by pipe. For example, `tail -n100 | grep ERROR` gets auto-transtalted to `cat SRCFILE | tail -n100 | grep ERROR > DSTFILE`.

--include INCLUDE

Comma-separated list of logs to include,each entry is either a "selector" or a glob for matching files.

Selectors must be one of `all`, `orion`, `system`, `ts`.

Anything that starts with `/` (forward slash) is assumed to be a glob pattern and interpreted through `find(1)`. Other entries are ignored.

TIP: use single quotes around the parameter value to prevent undesired glob expansion.

The default setting is `all`.

--exclude EXCLUDE

Comma separated list of logs to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include`.

--since SINCE

Grab logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

--from FROM

Timestamp where collection begins; must be of the form `yyyymmdd-HH:MM`.

--to TO

Timestamp where collection ends; must be of the form `yyyymmdd-HH:MM`.

--outfile OUTFILE

File path for printing all results. By default printed to `stdout`

--outdir OUTDIR

Directory path for writing results with original directory structure from each node. Used as an alternative to printing output to `outfile/stdout`.

--cmd_infmt CMD_INFMT

Specify if the input file should be compressed or uncompressed before running `CMD`. `C` for compressed, `U` for uncompressed. Don't use this flag if `CMD` works on both.

--cmd_outfmt CMD_OUTFMT

Specify if `OUTFILE` generated by `CMD` should be compressed or uncompressed. `C` for compressed, `U` for uncompressed. Don't use this flag if output file is of the same format as the input file.

--nodes NODES

Comma separated list of nodes where to run command. Skip this to use all nodes.

map-tiles

```
tscli map-tiles [-h] {disable,enable,status}
```

This subcommand has the following options:

```
tscli map-tiles enable [-h] [--online] [--offline] [--tar TAR] [--md5 MD5]
```

Enables ThoughtSpot's map tiles, used when constructing geomap charts.

If you don't have internet access, you must download the map tiles tar and md5 files, and append the following to the `tscli` command:

--online

Download `maptiles` tar from internet.

The default setting is `True`

--offline

Using `maptiles` tar from local disk.

The default setting is `False`

--tar TAR

Specified tar file for map-tiles.

--md5 MD5

Specified md5 file for map-tiles.

tscli map-tiles disable

Disable map-tiles functionality.

tscli map-tiles status

Check whether map-tiles are enabled.

monitoring

```
tscli monitoring [-h] {set-config,show-config}
```

This subcommand has the following options:

```
tscli monitoring set-config [-h] [--email EMAIL] [--clear_email] [--heartbeat_interval HEARTBEAT_INTERVAL] [--heartbeat_disable] [--report_interval REPORT_INTERVAL] [--report_disable]
```

Sets the monitoring configuration.

The `monitoring` subcommand accepts the following optional flags:

--email EMAIL

Comma separated list (no spaces) of email addresses where the cluster will send monitoring information.

--clear_email

Disable emails by clearing email configuration. (default: False)

--heartbeat_interval HEARTBEAT_INTERVAL

Heartbeat email generation interval in seconds. Should be greater than 0.

--heartbeat_disable

Disable heartbeat email generation. (default: False)

--report_interval REPORT_INTERVAL

Cluster report email generation interval in seconds.

Should be greater than 0 .

--report_disable

Disable cluster report email generation.

The default setting is False .

tscli monitoring show-config

Shows the monitoring configuration.

nas

```
tscli nas [-h] {ls,mount-cifs,mount-nfs,unmount}
```

This subcommand has the following options:

tscli nas ls [-h]

List mounts managed by NAS mounter service.

```
tscli nas mount-cifs [-h] --server SERVER [--path_on_server PATH_ON_SERVER] --
mount_point MOUNT_POINT--username USERNAME --password PASSWORD [--uid UID] [--gid GID]
[--options OPTIONS]
```

Mounts a CIFS device on all nodes.

Accepts the following optional flags:

--server SERVER

IP address or DNS name of CIFS service.

For example, 10.20.30.40 .

--path_on_server PATH_ON_SERVER

Filesystem path on the CIFS source server to mount NAS.

For example, `/a`.

The default setting is `/` (forward slash).

--mount_point MOUNT_POINT

Directory on all cluster nodes where to mount the NFS filesystem on the target.

If this directory does not exist, the command creates it. If this directory already exists, the command uses it for mounting.

For example, `/mnt/external`.

--username USERNAME

Username to connect to the CIFS filesystem

--password PASSWORD

CIFS password for `--username`

--uid UID

The `UID` that owns all files or directories on the mounted filesystem when the server does not provide ownership information.

See `man mount.cifs` for more details.

The default setting is `1001`.

--gid GID

The `GID` that owns all files or directories on the mounted filesystem when the server does not provide ownership information.

See `man mount.cifs` for more details.

The default is `1001`.

--options OPTIONS

Other command-line options to forward to the `mount.cifs` command.

The default setting is `noexec`.

```
tscli nas mount-nfs [-h] --server SERVER [--protocol PROTO --path_on_server  
PATH_ON_SERVER] --mount_point MOUNT_POINT [--options OPTIONS]
```

Mounts a NFS device on all nodes.

Accepts the following optional flags:

--server SERVER

IP address or DNS name of NFS service. For example, `10.20.30.40`.

--path_on_server PATH_ON_SERVER

Filesystem path on the NFS source server.

For example, `/a/b/c/d`.

The default setting is `/`.

--mount_point MOUNT_POINT

Directory on all cluster nodes of the target system.

If this directory does not exist, the command creates it. If this directory already exists, the command uses it for mounting.

For example, `/mnt/external`.

--options OPTIONS

Command-line options to mount.

The default setting is `noexec`.

--protocol PROTO

One of `nfs` or `nfs4`.

The default is `nfs`.

tscli nas umount [-h] --dir DIR

Unmounts all devices from the specified directory, `DIR`.

This command returns an error if nothing is currently mounted on this directory through `tscli nas mount`.

node

```
tscli node [-h] {check,ls,reinstall-os,resume-reinstall-os,stat,us}
```

This subcommand has the following options:

```
tscli node check [-h] [--select {reinstall-preflight}] [--secondary SECONDARY]
```

Run checks per node.

Accepts the following flags:

```
--select {reinstall-preflight}
```

Select the type of node check

The default setting is `reinstall-preflight`.

```
--secondary SECONDARY
```

Secondary drive for `reinstall-preflight`.

The default setting is `sdd`.

```
tscli node ls [-h] [--type {all,healthy,not-healthy}]
```

Filter by node state.

The default setting is `all`.

```
tscli node reinstall-os [-h] [--secondary SECONDARY] [--cluster]
```

Reinstall OS on a node.

Accepts the following flags:

```
--secondary SECONDARY
```

Secondary drive for reinstall.

The default setting is `sdd`.

```
--cluster
```

The node part of a cluster.

The default setting is `False`.

```
tscli node resume-reinstall-os
```

Resume in-progress reinstall

onboarding

```
tscli onboarding [-h] {configure,purge-configuration}
```

Onboarding helps application administrators to bulk update user information. In particular, it configures various in-app email options.

This subcommand has the following options:

```
tscli onboarding --help
```

Prints help for the onboarding configuration

```
tscli onboarding configure
```

Configures the onboarding through series of steps.

Asks the user to provide information necessary for onboarding-related functionality, such as the following:

1. Company name
2. Product name
3. Should welcome emails be enabled?
 - Send welcome emails to new users
 - Support email
 - Custom message to include in emails
 - URL of the ThoughtSpot instance
 - URL of the ThoughtSpot documentation

```
tscli onboarding purge-configuration
```

This command removes all onboarding configuration.

patch

```
tscli patch [-h] {apply,ls,resume-apply,resume-rollback,rollbac  
k}
```

This subcommand has the following options:

```
tscli patch apply [-h] [release]
```

Apply the patch on an existing cluster.

Accepts the following flag:

```
release
```

The relative path to the patch tar ball.

```
tscli patch ls [-h] [--applied] [--rolled_back] [--service SERVICE] [--md5 MD5] [--
```

history]

Lists the patches currently applied.

Accepts the following flags:

--applied

Show only the patches applied since last full release.

The default setting is `False`.

--rolled_back

Show only the patches rolled back since last full release.

The default setting is `False`.

--service SERVICE

Show patches filtered by service.

The default setting is `None`.

--md5 MD5

Shows the details of the patch specified.

The default setting is `None`.

--history

Shows the history of all patch apply/rollback release.

The default setting is `False`.

tscli patch resume-apply [-h]

Resume patch apply

tscli patch resume-rollback [-h]

Resume patch roll-backup

tscli patch rollback [-h]

Rollback the patch from an existing cluster

rpackage

tscli rpackage [-h] {add,delete,list}

Manages R packages available to SpotIQ.

This subcommand has the following options:

```
tscli rpackage add [-h] [--repo REPO] [--timeout TIMEOUT] [--dest_dir DEST_DIR] [--nodes NODES] package_name
```

Command to add an R package, `package_name`, to the cluster.

Accepts the following flags:

--repo REPO

Specify the url of a specific repository to download packages.

-timeout REPO

Timeout waiting for the R Package to be installed (default: 60)

--dest_dir REPO

Directory where output of this command will be placed

--nodes NODES

Space-separated list of IPs for nodes where to run the command.

The default setting is `all`.

```
tscli rpackage add [-h] [--timeout TIMEOUT] [--dest_dir DEST_DIR] [--nodes NODES] package_name
```

Command to delete an installed R package from the cluster.

Accepts the following flags:

--timeout REPO

Timeout waiting before removing the R package.

The default is 60

--dest_dir REPO

Directory where to save the output of this command.

--nodes NODES

Space-separated list of node IPs where to run the command.

The default setting is `all`.

```
tscli rpackage list [-h] [--detailed]
```

List all R packages installed on the cluster.

saml

```
tscli saml [-h] {configure,purge-configuration}
```

This subcommand has the following options:

```
tscli saml configure [-h]
```

Configures SAML.

```
tscli saml purge-configuration
```

Purges any existing SAML configuration.

To see a list of prerequisites, refer to [Configure SAML](#).

scheduled-pinboards

```
tscli scheduled-pinboards [-h] {disable,enable}
```

This subcommand has the following options:

```
tscli scheduled-pinboards disable [-h]
```

Disable scheduled pinboards for this cluster.

```
tscli scheduled-pinboards enable [-h]
```

Enables scheduled pinboards, which is disabled in prod clusters by default.

ⓘ Note: When you enable scheduled pinboards, you should also configure a whitelist of intended email domains. Contact ThoughtSpot Support for help on how to configure a whitelist.

smtp

```
tscli smtp [-h] {remove-mailfromname,remove-mailname,remove-relayhost,remove-saslcredentials,reset-canonical-mapping,set-canonical-mapping,set-mailfromname,set-mailname,set-relayhost,set-saslcredentials,show-canonical-mapping,show-mailfromname,show-mailname,show-relayhost}
```

This subcommand has the following options:

tscli smtp remove-mailfromname

Removes current cluster mail from name.

tscli smtp remove-mailname

Removes current cluster mail name.

tscli smtp remove-relayhost

Removes current cluster relay host.

tscli smtp remove-saslcredentials

Clears SASL credentials and disables SMTP AUTH.

tscli smtp reset-canonical-mapping

Deletes the current postmap mapping.

tscli smtp set-canonical-mapping [-h] new_key new_value

Sets a new Postmap mapping.

tscli smtp set-mailfromname mailfromname

Sets the name and an email address from where email alerts are sent for the cluster.

tscli smtp set-mailname mailname

Sets the mailname and a domain from where email alerts are sent for the cluster.

tscli smtp set-relayhost [-h] [--force FORCE] relayhost

Sets the Relay Host for SMTP (email) sent from the cluster.

Accepts the following flag:

--force FORCE

Set even if relay host is not accessible.

The default setting is `False`.

tscli smtp set-saslcredentials

Sets SASL credentials and enables SMTP AUTH

tscli smtp show-canonical-mapping

Shows the current postmap mapping.

tscli smtp show-mailfromname

Shows the mailname, from which email alerts are sent, for the cluster.

tscli smtp show-mailname

Shows the mailname, where email alerts are sent, for the cluster.

tscli smtp show-relayhost

Shows the for SMTP (email) sent from the cluster.

If there a relay host is not configured, the command returns `NOT FOUND`.

snapshot

```
tscli snapshot [-h] {backup,create,delete,ls,pin,restore,unpin,update-ttl}
```

To learn more about snapshots and backups, see the [Understand the backup strategies](#) documentation.

This subcommand has the following options:

```
tscli snapshot backup [-h] [--mode {full,light,dataless}] [--type {full,incremental}]  
[--base BASE] [--storage_type {local,nas}] [--remote] name out
```

Pull snapshot out as a backup.

Accepts the following flags:

--mode {full,light,dataless}

Mode of backups. (default: 'full')

name

Name of snapshot to pull out as a backup. To list all snapshots, run `tscli snapshot ls`.

out

Directory where backup will be written, must not already exist.

```
--type {full,incremental}
```

Type of backup.

Incremental backup is not implemented.

The default setting is `full`.

```
--base BASE
```

Based snapshot name for incremental backup.

Incremental backup not implemented yet.

```
storage_type {local,nas}
```

Storage type of output directory. (default: `local`)

```
--remote
```

Take backup through Orion master. (default: `True`)

```
tscli snapshot create [-h] name reason ttl
```

Creates a new snapshot with the specified `name` and `reason`.

This command does not accept `.` (periods). It does accept `-` (dashes or hyphens).

The `ttl` parameter is the number of days after which this snapshot is automatically deleted. A value of `-1` disables automatic deletion.

```
tscli snapshot pin [-h] name
```

Pins a snapshot so it cannot be deleted or garbage collected.

```
tscli snapshot delete [-h] name
```

Deletes the named snapshot.

```
tscli snapshot ls [-h]
```

Lists available snapshots.

```
tscli snapshot restore [-h] [--allow_release_change] [--only_service_state] name
```

Restores cluster to an existing snapshot.

Accepts the following flags:

```
--allow_release_change
```

Allow restoration to a snapshot at a different release. (default: `False`)

```
--only_service_state
```

Restore only service state. (default: `False`)

```
tscli snapshot unpin [-h] name
```

Unpin a snapshot so it can be deleted or garbage-collected

```
tscli snapshot update-ttl [-h] [--disable DISABLE] name ttl
```

Updates manual snapshot garbage collection policy.

Accepts the following flags:

```
name
```

Specifies which snapshot to update.

```
ttl
```

Extends the manual snapshot

```
ttl
```

This is the "time-to-live" value.

Use a positive value to increase `ttl`. Use negative values to decrease it.

```
--disable DISABLE
```

Disable manual snapshot garbage collection.

Setting this value to `True` will override any `ttl` value. (default: False)

snapshot-policy

```
tscli snapshot-policy [-h] {disable,enable,show,update}
```

This subcommand has the following options:

```
tscli snapshot-policy disable [-h]
```

Disable snapshot policy.

```
tscli snapshot-policy enable -h
```

Enable specified snapshot policy.

```
tscli snapshot-policy show [-h]
```

Show snapshot policy.

```
tscli snapshot-policy update [-h] [--config CONFIG]
```

Update periodic snapshot config. This takes the following parameter

--config CONFIG

Text format of periodic backup policy config.

spot

```
tscli spot [-h] {enable}
```

Enables Spot integration.

This subcommand has the following option:

```
tscli spot enable [-h] --token TOKEN --thoughtspot_url THOUGHTSPOT_URL [--cache_timeout CACHE_TIMEOUT]
```

The `spot` subcommand accepts the following optional flags:

--token TOKEN

Slack authorization token for Spot bot. This is required. You receive this token when your Slack administrator adds the Spot application.

--thoughtspot_url THOUGHTSPOT_URL

URL for the ThoughtSpot application.

This is required.

--cache_timeout CACHE_TIMEOUT

Internal cache timeout.

The default setting is `60000`.

ssl

```
tscli ssl [-h] {add-cert,clear-min-tls-version,off,on,rm-cert,set-min-tls-version,status,tls-status,add-valid-hosts}
```

This subcommand manages the SSL configuration.

To use SSL, the following ports must be open:

- 443

- 80

This subcommand has the following options:

```
tscli ssl add-cert [-h] key certificate
```

Adds an SSL certificate, key pair.

```
tscli ssl clear-min-tls-version [-h]
```

Clears any customizations for the minimum TLS version to support.

```
tscli ssl off
```

Disables SSL. Disabling SSL will stop users from seeing a security warning when accessing ThoughtSpot from a browser if there is no SSL certificate installed.

```
tscli ssl on [-h]
```

If SSL is enabled and there is no certificate, users will see a security warning when accessing ThoughtSpot from a browser.

```
tscli ssl rm-cert
```

Removes the existing SSL certificate, if any.

```
tscli ssl set-min-tls-version [-h] {1.0,1.1,1.2}
```

Sets the minimum supported TLS version. Sets the minimum SSL version to be supported by the ThoughtSpot application. Please ensure that client browsers are enabled for this version or newer.

```
tscli ssl status
```

Shows whether SSL authentication is enabled or disabled.

```
tscli ssl tls-status [-h]
```

Prints the status of TLS support.

```
tscli ssl add-valid-hosts [-h] VALID_HOSTS
```

Enables host validation for the specified host(s). Helps improve security. This feature is for all customers that have or are planning to enable SSL. Multiple hosts must be separated by a comma (,).

Examples:

1. If you want to make sure the valid host is **cluster1.corp.example.com**, you would run the command:

```
tscli ssl add-valid-hosts cluster1.corp.example.com
```

2. If you want to allow all hosts which have the suffix **corp.example.com**, you would run the command:

```
tscli ssl add-valid-hosts *.corp.example.com
```

. This wild card should be used within the hostname.

3. If you want to allow multiple valid hosts, for example both ***.corp.example.com** and **cluster1**, you would run the command:

```
tscli ssl add-valid-hosts *.corp.thoughtspot.com,cluster1
```

sssd

```
tscli sssd {enable, disable, set-sudo-group, clear-sudo-group}
```

This subcommand uses system security services daemon (SSSD), and has the following options:

```
tscli sssd enable --user USER --domain DOMAIN
```

Enables system Active Directory (AD) user access on a single node. You will be prompted for password credentials. The user must have permission to join a computer or VM to the domain.

```
tscli sssd disable
```

```
tscli sssd set-sudo-group ACTIVE_DIRECTORY_GROUP_NAME
```

Allows `sudo` permissions for AD group.

```
tscli sssd clear-sudo-group ACTIVE_DIRECTORY_GROUP_NAME
```

Clears any set AD sudo group.

For more about setting up Active Directory access, see [Enable Active Directory based access](#).

storage

```
tscli storage [-h] {gc,df}
```

This subcommand has the following options:

```
tscli storage gc [-h] [--log_age LOG_AGE] [--force] [--localhost_only]
```

Garbage collect unused storage.

Before issuing this command, you must stop the cluster using `=tscli cluster stop`.

After garbage collection finishes, you can restart the cluster with `tscli cluster start`.

This command frees space in the following directories:

```
/tmp
/usr/local/scaligent/logs/
/export/logs/orion
/export/logs/oreo
/export/logs/hadoop
/export/logs/zookeeper
cores
```

The `storage` subcommand accepts these optional flags:

--log_age *LOG_AGE*

Deletes logs older than these many hours. Use a non-zero value, because zero deletes all temporary files, including the ones that are closed temporarily, while they are passed from one component to the next.

The default setting is `4`.

--force

Forces deletion of all logs and temporary files regardless of age. This must only be run on a stopped cluster.

The default setting is `False`.

--localhost_only

If used, only the logs on the localhost will be removed. If not specified, the command acts on the entire cluster.

tscli storage df [--mode disk|hdfs]

Checks the disk usage on the relevant mounts. Returns output similar to the Linux system command `df -h directory`.

support

```
tscli support [-h]
{bundle, restart-remote, rm-admin-email, rm-admin-phone, rm-
feedback-email,
  set-admin-email, set-admin-phone, set-debug-ui-password, se
t-feedback-email,
  set-remote, show-admin-email, show-admin-phone, show-feedba
ck-email,
  show-remote, start-remote, stop-remote}
```

This subcommand has the following options:

```
tscli support bundle [-h] [--include INCLUDE] [--exclude EXCLUDE] [--list_selectors]
[--since SINCE] [--from FROM] [--to TO] [--out OUT] [--nodes NODES]

--include INCLUDE
```

Comma-separated list of selectors to include. Each entry is either a "selector" or a glob for matching files. To see the list of valid selectors, run this command with `--list_selectors`. You may also specify `all` to get all selectors and logs, and `basic` to get only the basic selectors.

Selectors can be used for logs collection: `all`, `orion`, `system`, `ts`, or the name of a service.

Anything that starting with `/` (forward slash) is assumed to be a glob pattern, and it is interpreted through `find(1)`. Other entries are ignored.

TIP: Use single quotes around the param value to prevent undesired glob expansion. Use `all` to collect all selectors and all logs.

The default setting is `all_but_logs`.

```
--exclude EXCLUDE
```

Comma-separated list of selectors to exclude. Applies to the list selected by `--include`. Params are interpreted in the same manner as in `--include`.

Use the special keyword `logs` to exclude logs collection altogether.

There is no default setting.

```
--list_selectors
```

List the selectors available for `--include` and `--exclude`, and then exit.

The default setting is `False`.

```
--since SINCE
```

Grabs logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

There is no default setting.

```
--from FROM
```

Timestamp when collection begins. Must be of the form: `yyyymmdd-HH:MM`.

There is no default setting.

--to *T0*

Timestamp when collection ends. Must be of the form: `yyyymmdd-HH:MM`.

There is no default setting.

--out *OUT*

Tarball path for dumping the support bundle.

The default setting is `/tmp/support_bundle.tar.gz`.

--nodes *NODES*

Comma separated list of nodes from where to collect logs. Skip this to use all nodes.

There is no default setting.

tscli support restart-remote

Restarts remote support.

tscli support rm-admin-email

Removes the email address for contacting the customer administrator. Replaces it with the default ThoughtSpot Support email address.

tscli support rm-feedback-email

Removes the email address for product feedback. Replaces it with the default ThoughtSpot Support email address

tscli support rm-admin-phone

Removes the phone number for contacting the customer administrator. Replaces it with the default ThoughtSpot Support phone number

tscli support rm-feedback-email

Removes the email for sending feedback out of the system. To set a blank email address, issue the command `tscli support set-feedback-email`.

tscli support set-admin-email *email*

Sets the email address for contacting the customer administrator. To display a blank email address, issue the command `tscli support set-admin-email`.

tscli support set-feedback-email *email*

Sets the email address for sending feedback. To display a blank email address, issue the command `tscli support set-feedback-email`.

tscli support set-admin-phone *phone_number*

Sets the phone number for contacting the customer administrator. Specify a phone number using any value, such as `+1 800-508-7008 Ext. 1`. To display a blank phone number, issue the command `tscli support set-admin-phone`.

```
tscli support set-remote [-h] [--addr ADDR] [--user USER]
```

Configures the cluster for remote support through SSH tunneling, where `ADDR` is the address of support, such as `tunnel.thoughtspot.com`, and `USER` is the support username.

```
tscli support show-admin-email
```

Shows the email address for customer administrator, if set

```
tscli support show-feedback-email
```

Shows the email address for product feedback, if set

```
tscli support show-admin-phone
```

Shows the phone number for customer administrator, if set

```
tscli support show-remote
```

Shows the status and configuration of remote support

```
tscli support start-remote
```

Starts remote support

```
tscli support stop-remote
```

Stops remote support

tokenauthentication

```
tscli cli tokenauthentication [-h] {disable,enable}
```

This subcommand has the following options:

```
tscli cli tokenauthentication enable
```

Generates a token

```
tscli cli tokenauthentication disable
```

Purges token login configuration

Date and time formats reference

This is a reference for the date and time contexts and formats you can use with ThoughtSpot. You define data formats in specific contexts and, depending on the context, your choices in data formatting differ.

You must understand date and time when you load data in these contexts:

- using data upload from the browser
- through `tsload` command
- through an extract, transform, load (ETL) tool

Data loading formats do not change how data is displayed in tables and charts.

The context where you *can control* date and time formats is data modeling. Data modeling controls how data is displayed in search and their resulting answers.

Data loading formats through `tsload`

When loading through the `tsload` command you must specify `date` and `timestamp` formats using the format specifications defined in the [strftime library function](#). Data is imported based on the timezone of the node from which `tsload` is run.

For `date` data types, the default format is `%Y%m%d`, which translates to `yearmonthday`. For example, `Dec 30th, 2001` is represented as `20011230`. For `time` and `datetime` data types, the default is `%Y%m%d %H:%M:%S` which translates to `yearmonthday hour:minute:second`, for example, `Dec 30th, 2001 1:15:12` is represented as `20011230 01:15:12`.

Data modeling formats for browser data upload

These date and time formats are supported in a CSV file when uploading through the browser. You cannot specify the date format; ThoughtSpot will pick the format that fits your data best:

- 1/30/2014
- 2014-01-30
- 2014-1-30
- 30-Jan-2014
- 2014-Jan-30

- 2014-01-30 10:32 AM
- 2014-01-30 14:52
- 2014-01-30 10:32:22
- 2014-01-30 10:32:22 AM
- 2014-01-30 10:32:22.0
- 2014-01-30 10:32:22.0 AM
- 2014-01-30 10:32:22.000
- 2014-01-30 10:32:22.000 AM
- 1/30/2014
- 30-Jan-14
- 01-Mar-02 (assumes 2002)
- 30/1/2014 10:32 AM
- 30/1/2014 14:52
- 30/1/2014 10:32:22
- 30/1/2014 10:32:22 AM
- 30/1/2014 10:32:22.0
- 30/1/2014 10:32:22.0 AM
- 30/1/2014 10:32:22.000
- 30/1/2014 10:32:22.000 AM
- 30-Jan-14 10:32 AM
- 30-Jan-14 14:52
- 30-Jan-14 10:32:22
- 30-Jan-14 10:32:22 AM
- 30-Jan-14 10:32:22.0
- 30-Jan-14 10:32:22.0 AM
- 30-Jan-14 10:32:22.000
- 30-Jan-14 10:32:22.000 AM
- Fri Jan 30 2014 3:26 PM
- Fri Jan 30 2014 13:46
- Fri Jan 30 2014 10:32:22
- Fri Jan 30 2014 10:32:22 AM
- Fri Jan 30 2014 10:32:22.0
- Fri Jan 30 2014 10:32:22.0 AM
- Fri Jan 30 2014 10:32:22.000
- Fri Jan 30 2014 10:32:22.000 AM
- 14:52

- 10:32 AM
- 10:32:22
- 10:32:22 AM
- 10:32:22.0
- 10:32:22.000
- 10:32:22.0 AM
- 10:32:22.000 AM

Data loading formats through an ETL tool

Data loads through ETL uses ODBC or JDBC connections. After you extract the data from the source but before you load it into ThoughtSpot, you must transform any date or timestamp columns into a format that is valid for ThoughtSpot. After the data transformation completes, there is no requirement for explicit data masking. See the data integration guide for more information on loading data through ODBC and JDBC.

Data modeling formats

A user with administrative rights can configure data modeling for data on one or all files. You can set number, date, and currency display formats. These formats define how these value types display in tables and charts. See the Admin Guide for more information about data modeling settings. The following format strings are available for use:

Format mask	Description
YYYY or yyyy	four digit year such as 2017
YY or yy	last two digits of year such as 17
M	month with no leading zero 1 - 12
MM	Two digit month 01 - 12
MMM	Three letter month such as Jan
D	Day of year without a leading zero 0 - 365
DD	Day of year with up to one leading zero 01 - 365
DDD	Day of year with up to two leading zeroes 001 - 365

Format mask	Description
d	Day of month with no leading zero 1 - 31
dd	Two digit day of month 01 - 31
HH	Two digit 24 hour representation of hour 00 - 23
hh	Two digit 12 hour representation of hour 01 - 12
H	24 hour representation of hour with no leading zero 0 - 23
h	12 hour representation of hour with no leading zero 1 - 12
mm	Minutes 00 - 59
m	Minutes with no leading zero 0 - 59
ss	Seconds 00 - 59
s	Seconds with no leading zero 0 - 59
a	AM/PM indicator

Valid delimiters include most non-alphabet characters. This includes but is not limited to:

- \ (forward slash)
- / (backward slash)
- | (pipe symbol)
- : (colon)
- – (dash)
- _ (underscore)
- = (equal sign)

Examples of valid format masks you can produce for display are as follows:

- MM/dd/yyyy
- MMM
- DD/MM/yyyy
- MM/dd/yyyy HH:mm
- DD/MM/yyyy HH:mm

Row level security rules reference

ThoughtSpot allows you to create row level security rules using expressions. If an expression evaluates to “true” for a particular row and group combination, that group will be able to see that row. This reference lists the various operators and functions you can use to create rules.

For information on how to use the row level security functions and operators, see [About Rule-Based Row Level Security](#). There is a special variable called `ts_groups`, which you can use when creating row level security rules. It fetches a list of the groups that the currently logged in user belongs to. For each row, if the expression in the rule evaluates to ‘true’ for any one of these groups, that row will be shown to the user.

You can also see this list of operators and examples from within the Rule Builder by selecting **Rule Assistant**.

Conversion functions

These functions can be used to convert data from one data type to another. Conversion to or from date data types is not supported.

Function	Description	Examples
<code>to_bool</code>	Returns the input as a boolean data type (true or false).	<code>to_bool (0) = false</code> <code>to_bool (married)</code>

Function	Description	Examples
to_date	<p>Accepts a date represented as an integer or text string, and a second string parameter that can include strftime date formatting elements.</p> <p>Replaces all the valid strftime date formatting elements with their string counterparts and returns the result.</p> <p>Does not accept epoch formatted dates as input.</p>	<code>to_date (date_sold, '%Y-%m-%d')</code>
to_double	Returns the input as a double data type.	<code>to_double ('3.14') = 3.14</code> <code>to_double (revenue * .01)</code>
to_integer	Returns the input as an integer.	<code>to_integer ('45') + 1 = 46</code> <code>to_integer (price + tax - cost)</code>
to_string	Returns the input as a text string. To convert a date data type to a string data type, specify the date format you want to use.	<code>to_string (45 + 1) = '46'</code> <code>to_string (revenue - cost)</code> <code>to_string (date, ('%m/%d/%y'))</code>

Date functions

Function	Description	Examples
add_days	Returns the result of adding the specified number of days to the given date.	<code>add_days (01/30/2015, 5) = 02/04/2015</code> <code>add_days (invoiced, 30)</code>
add_minutes	Returns the result of adding the specified number of minutes to input date/date-time/time.	<code>add_minutes (01/30/2015 00:10:20 , 5) = 01/30/2015 00:11:20</code> <code>add_minutes (invoiced , 30)</code>

Function	Description	Examples
add_months	Returns the result of adding the specified number of months to the given date.	add_months (01/30/2015, 5) = 06/30/2015 add_months (invoiced_date , 5)
add_seconds	Returns the result of adding the specified number of seconds to the given date.	add_seconds (01/30/2015 00:00:00, 5) = 06/30/2015 00:00:05 add_seconds (invoiced_date , 5)
add_weeks	Returns the result of adding the specified number of weeks to the given date.	add_weeks (01/30/2015, 2) = 02/13/2015 add_weeks (invoiced_date , 2)
add_years	Returns the result of adding the specified number of years to the given date.	add_years (01/30/2015, 5) = 01/30/2020 add_years (invoiced_date , 5)
date	Returns the date portion of a given date.	date (home visit)
day	Returns the number (1-31) of the day for the given date.	day (01/15/2014) = 15 day (date ordered)
day_number_of_quarter	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_quarter (01/30/2015) = 30 day_number_of_quarter (01/30/2015, 'fiscal') = 91
day_number_of_week	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	day_number_of_week(01/15/2014) = 3 day_number_of_week (shipped)

Function	Description	Examples
day_number_of_year	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_year (01/30/2015) = 30 day_number_of_year (01/30/2015, 'fiscal') = 275 day_number_of_year (invoiced)
day_of_week	Returns the day of the week for the given date.	day_of_week (01/30/2015) = Friday day_of_week (serviced)
diff_days	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	diff_days (01/15/2014, 01/17/2014) = -2 diff_days (purchased, shipped)
diff_time	Subtracts the second date from the first date and returns the result in number of seconds.	diff_time (01/01/2014, 01/01/2014) = -86,400 diff_time (clicked, submitted)
hour_of_day	Returns the hour of the day for the given date.	hour_of_day (received)
is_weekend	Returns true if the given date falls on a Saturday or Sunday.	is_weekend (01/31/2015) = true is_weekend (emailed)
month	Returns the month from the given date.	month (01/15/2014) = January month (date ordered)

Function	Description	Examples
month_number	Returns the number (1-12) of the month from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number (09/20/2014) = 9 month_number (09/20/2014, 'fiscal') = 5 month_number (purchased)
month_number_of_quarter	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number_of_quarter (02/20/2018) = 2 month_number_of_quarter (02/20/2018, 'fiscal') = 1
now	Returns the current timestamp.	now ()
quarter_number	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	quarter_number (04/14/2014) = 2 quarter_number (04/14/2014, 'fiscal') = 4 quarter_number (shipped)

Function	Description	Examples
start_of_month	Returns MMM yyyy for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as MM/dd/yyyy . Speak with your ThoughtSpot administrator for information on doing this.	start_of_month (01/31/2015) = Jan FY 2015 start_of_month (shipped)
start_of_quarter	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	start_of_quarter (04/01/2014) = Apr 2014 start_of_quarter (04/01/2014, 'fiscal') = Feb 2014 start_of_quarter (sold)
start_of_week	Returns the date for the first day of the week for the given date.	start_of_week (06/01/2015) = 05/30/2015 Week start_of_week (emailed)
start_of_year	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	start_of_year (04/01/2014) returns Jan 2014 start_of_year (04/01/2014, 'fiscal') returns May 2013 start_of_year (joined)
time	Returns the time portion of a given date.	time (3/1/2002 10:32) = 10:32 time (call began)

Function	Description	Examples
week_number_of_month	Returns the week number for the given date in a month.	week_number_of_month(03/23/2017) = 3
week_number_of_quarter	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_quarter (04/03/2017) = 1 week_number_of_quarter (04/03/2017, 'fiscal') = 10
week_number_of_year	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_year (01/17/2014) = 3 week_number_of_year (01/17/2014, 'fiscal') = 38
year	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	year (01/15/2014) = 2014 year (12/15/2013, 'fiscal') = 2014 year (date ordered)

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
<code>!=</code>	Returns true if the first value is not equal to the second value.	<code>3 != 2 = true</code> <code>revenue != 1000000</code>
<code><</code>	Returns true if the first value is less than the second value.	<code>3 < 2 = false</code> <code>revenue < 1000000</code>
<code><=</code>	Returns true if the first value is less than or equal to the second value.	<code>1 <= 2 = true</code> <code>revenue <= 1000000</code>
<code>=</code>	Returns true if the first value is equal to the second value.	<code>2 = 2 = true</code> <code>revenue = 1000000</code>
<code>></code>	Returns true if the first value is greater than the second value.	<code>3 > 2 = true</code> <code>revenue > 1000000</code>
<code>>=</code>	Returns true if the first value is greater than or equal to the second value.	<code>3 >= 2 = true</code> <code>revenue >= 1000000</code>
<code>greatest</code>	Returns the larger of the values.	<code>greatest (20, 10) = 20</code> <code>greatest (q1 revenue, q2 revenue)</code>
<code>least</code>	Returns the smaller of the values.	<code>least (20, 10) = 10</code> <code>least (q1 revenue, q2 revenue)</code>

Number functions

Function	Description	Examples
<code>*</code>	Returns the result of multiplying both numbers.	<code>3 * 2 = 6</code> <code>price * taxrate</code>
<code>+</code>	Returns the result of adding both numbers.	<code>1 + 2 = 3</code> <code>price + shipping</code>

Function	Description	Examples
-	Returns the result of subtracting the second number from the first.	$3 - 2 = 1$ revenue - tax
/	Returns the result of dividing the first number by the second.	$6 / 3 = 2$ markup / retail price
^	Returns the first number raised to the power of the second.	$3 ^ 2 = 9$ width ^ 2
abs	Returns the absolute value.	abs (-10) = 10 abs (profit)
acos	Returns the inverse cosine in degrees.	acos (0.5) = 60 acos (cos-satellite-angle)
asin	Returns the inverse sine (specified in degrees).	asin (0.5) = 30 asin (sin-satellite-angle)
atan	Returns the inverse tangent in degrees.	atan (1) = 45 atan (tan-satellite-angle)
atan2	Returns the inverse tangent in degrees.	atan2 (10, 10) = 45 atan2 (longitude, latitude)
cbrt	Returns the cube root of a number.	cbrt (27) = 3 cbrt (volume)
ceil	Returns the smallest following integer.	ceil (5.9) = 6 ceil (growth rate)
cos	Returns the cosine of an angle (specified in degrees).	cos (63) = 0.45 cos (beam angle)
cube	Returns the cube of a number.	cube (3) = 27 cube (length)
exp	Returns Euler's number (~2.718) raised to a power.	exp (2) = 7.38905609893 exp (growth)
exp2	Returns 2 raised to a power.	exp2 (3) = 8 exp2 (growth)
floor	Returns the largest previous integer.	floor (5.1) = 5 floor (growth rate)
ln	Returns the natural logarithm.	ln (7.38905609893) = 2 ln (distance)

Function	Description	Examples
log10	Returns the logarithm with base 10.	<code>log10 (100) = 2</code> <code>log10 (volume)</code>
log2	Returns the logarithm with base 2 (binary logarithm).	<code>log2 (32) = 5</code> <code>log2 (volume)</code>
mod	Returns the remainder of first number divided by the second number.	<code>mod (8, 3) = 2</code> <code>mod (revenue, quantity)</code>
pow	Returns the first number raised to the power of the second number.	<code>pow (5, 2) = 25</code> <code>pow (width, 2)</code>
random	Returns a random number between 0 and 1.	<code>random () = .457718</code> <code>random ()</code>
round	Returns the first number rounded to the second number (the default is 1).	<code>round (35.65, 10) = 40</code> <code>round (battingavg, 100)</code> <code>round (48.67, .1) = 48.7</code>
safe_divide	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	<code>safe_divide (12, 0) = 0</code> <code>safe_divide (total_cost, units)</code>
sign	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	<code>sign (-250) = -1</code> <code>sign (growth rate)</code>
sin	Returns the sine of an angle (specified in degrees).	<code>sin (35) = 0.57</code> <code>sin (beam angle)</code>
spherical_distance	Returns the distance in km between two points on Earth.	<code>spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96</code> <code>spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)</code>
sq	Returns the square of a numeric value.	<code>sq (9) = 81</code> <code>sq (width)</code>
sqrt	Returns the square root.	<code>sqrt (9) = 3</code> <code>sqrt (area)</code>
tan	Returns the tangent of an angle (specified in degrees).	<code>tan (35) = 0.7</code> <code>tan (beam angle)</code>

Operators

Operator	Description	Examples
and	Returns true when both conditions are true , otherwise returns false .	(1 = 1) and (3 > 2) = true lastname = 'smith' and state ='texas'
		Note: Not available for row-level security (RLS) formulas.
if...then...else	Conditional operator	if (3 > 2) then 'bigger' else 'not bigger' if (cost > 500) then 'flag' else 'approve'
ifnull	Returns the first value if it is not null , otherwise returns the second value.	ifnull (cost, 'unknown')
isnull	Returns true if the value is null .	isnull (phone)
not	Returns true if the condition is false , otherwise returns false .	not (3 > 2) = false not (state = 'texas')
or	Returns true when either condition is true , otherwise returns false .	(1 = 5) or (3 > 2) = true state = 'california' or state ='oregon'

Text functions

Function	Description	Examples
concat	Returns two or more values as a concatenated text string. Use single quotes around each literal string, not double quotes.	concat ('hay' , 'stack') = 'haystack' concat (title, ' ', first_name , ' ', last_name)

Function	Description	Examples
contains	Returns true if the first string contains the second string, otherwise returns false.	<code>contains ('broomstick', 'room') = true</code> <code>contains (product, 'trial version')</code>
edit_distance	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	<code>edit_distance ('attorney', 'atty') = 4</code> <code>edit_distance (color, 'red')</code>
edit_distance_with_cap	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	<code>edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4</code> <code>edit_distance_with_cap (event, 'burning man', 3)</code>
similar_to	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	<code>similar_to ('hello world', 'hello swirl') = true</code> <code>similar_to (current team, drafted by)</code>
similarity	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	<code>similarity ('where is the burning man concert', 'burning man') = 46</code> <code>similarity (tweet1, tweet2)</code>

Function	Description	Examples
spells_like	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	spells_like ('thouhgspot', 'thoughtspot') = true spells_like (studio, distributor)
strlen	Returns the length of the text.	strlen ('smith') = 5 strlen (lastname)
strpos	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	strpos ('haystack_with_needles', 'needle') = 14 strpos (complaint, 'lawyer')
substr	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	substr ('persnickety', 3, 7) = snicket substr (lastname, 0, 5)

Variables

These variables can be used in your expressions.

Function	Description	Examples
ts_groups	Returns a list of all the groups the current logged in user belongs to. For any row, if the expression evaluates to true for any of the groups, the user can see that row.	ts_groups = 'east'
ts_username	Returns the user with the matching neame.	ts_username != 'mark'

Formula function reference

ThoughtSpot allows you to create derived columns in worksheets using formulas. You create these columns by building formulas using the **Formula Assistant**. An individual formula is constructed from n combination of operators and functions.

This reference lists the various operators and functions you can use to create formulas.

Operators

Operator	Description	Examples
and	Returns true when both conditions are true , otherwise returns false .	(1 = 1) and (3 > 2) = true lastname = 'smith' and state ='texas'
		ⓘ Note: Not available for row-level security (RLS) formulas.
if...then...else	Conditional operator	if (3 > 2) then 'bigger' else 'not bigger' if (cost > 500) then 'flag' else 'approve'
ifnull	Returns the first value if it is not null , otherwise returns the second value.	ifnull (cost, 'unknown')
isnull	Returns true if the value is null .	isnull (phone)
not	Returns true if the condition is false , otherwise returns false .	not (3 > 2) = false not (state = 'texas')
or	Returns true when either condition is true , otherwise returns false .	(1 = 5) or (3 > 2) = true state = 'california' or state ='oregon'

Aggregate functions (group aggregate)

Use the following functions to aggregate data.

Function	Description	Examples
average	Returns the average of all the values of a column.	average (revenue)
average_if	Returns the average of all the columns that meet a given criteria.	average_if(city = "San Francisco", revenue)
count	Returns the number of rows in the table containing the column.	count (product)
count_if	Returns the number of rows in the table containing the column.	count_if(region = 'west', region)
cumulative_average	Takes a measure and one or more attributes. Returns the average of the measure, accumulated by the attribute(s) in the order specified.	cumulative_average (revenue, order date, state)
cumulative_max	Takes a measure and one or more attributes. Returns the maximum of the measure, accumulated by the attribute(s) in the order specified.	cumulative_max (revenue, state)
cumulative_min	Takes a measure and one or more attributes. Returns the minimum of the measure, accumulated by the attribute(s) in the order specified.	cumulative_min (revenue, campaign)
cumulative_sum	Takes a measure and one or more attributes. Returns the sum of the measure, accumulated by the attribute(s) in the order specified.	cumulative_sum (revenue, order date)

Function	Description	Examples
group_aggregate	<p>Takes a measure and optional attributes and filters. Used to aggregate measures with different granularities and filters than the columns used in the search. Commonly used in comparison analysis.</p> <p>This formula takes the following form:</p> <pre>group_aggregate (<aggregation(measure)>, <groupings>, <filters>)</pre> <p>Define lists using curly brackets, { }. Optional list functions query_groups or query_filters specify the lists or filters used in the original search. Use + (plus) and - (minus) to add or exclude specific columns for query groups.</p> <p>See Flexible aggregation functions.</p>	<pre>group_aggregate (sum (revenue), {ship mode, date}, {})</pre> <pre>group_aggregate (sum (revenue), {ship mode , date}, {day_of_week (date) = 'friday'})</pre> <pre>group_aggregate (sum (revenue), query_groups(), query_filters())</pre> <pre>group_aggregate (sum (revenue), query_groups() + {date}, query_filters())</pre>
group_average	Takes a measure and one or more attributes. Returns the average of the measure grouped by the attribute(s).	<pre>group_average (revenue, customer region, state)</pre>
group_count	Takes a measure and one or more attributes. Returns the count of the measure grouped by the attribute(s).	<pre>group_count (revenue, customer region)</pre>
group_max	Takes a measure and one or more attributes. Returns the maximum of the measure grouped by the attribute(s).	<pre>group_max (revenue, customer region)</pre>
group_min	Takes a measure and one or more attributes. Returns the minimum of the measure grouped by the attribute(s).	<pre>group_min (revenue, customer region)</pre>

Function	Description	Examples
group_stddev	Takes a measure and one or more attributes. Returns the standard deviation of the measure grouped by the attribute(s).	group_stddev (revenue, customer region)
group_sum	Takes a measure and one or more attributes. Returns the sum of the measure grouped by the attribute(s).	group_sum (revenue, customer region)
group_unique_count	Takes a measure and one or more attributes. Returns the unique count of the measure grouped by the attribute(s).	group_unique_count (product , supplier)
group_variance	Takes a measure and one or more attributes. Returns the variance of the measure grouped by the attribute(s).	group_variance (revenue, customer region)
max	Returns the maximum value of a column.	max (sales)
max_if	Returns the maximum value among columns that meet a criteria.	max_if((revenue > 10) , customer region)
min	Returns the minimum value of a column.	min (revenue)
min_if	Returns the minimum value among columns that meet a criteria.	min_if((revenue < 10) , customer region)

Function	Description	Examples
moving_average	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1...Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the average of the measure over the given window. The attributes are the ordering columns used to compute the moving average.	<code>moving_average (revenue, 2, 1, customer region)</code>
moving_max	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1...Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the maximum of the measure over the given window. The attributes are the ordering columns used to compute the moving maximum.	<code>moving_max (complaints, 1, 2, store name)</code>

Function	Description	Examples
moving_min	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1...Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the minimum of the measure over the given window. The attributes are the ordering columns used to compute the moving minimum.	<code>moving_min (defects, 3, 1, product)</code>
moving_sum	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1...Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the sum of the measure over the given window. The attributes are the ordering columns used to compute the moving sum.	<code>moving_sum (revenue, 1, 1, order date)</code>
rank	Returns the rank for the current row. Identical values receive an identical rank. Takes an aggregate input for the first argument. The second argument specifies the order, ‘asc’ ‘desc’ .	<code>rank (sum (revenue) , 'asc')</code> <code>rank (sum (revenue) , 'desc')</code>
rank_percentile	Returns the percentile rank for the current row. Identical values are assigned an identical percentile rank. Takes an aggregate input for the first argument. The second argument specifies the order, ‘asc’ ‘desc’ .	<code>rank_percentile (sum (revenue) , 'asc')</code> <code>rank_percentile (sum (revenue) , 'desc')</code>

Function	Description	Examples
stddev	Returns the standard deviation of all values of a column.	stddev (revenue)
stddev_if	Returns a standard deviation values filtered to meet a specific criteria.	stddev_if((revenue > 10) , (revenue/10.0))
sum	Returns the sum of all the values of a column.	sum (revenue)
sum_if	Returns sum values filtered by a specific criteria.	sum_if(region='west', revenue)
unique_count	Returns the number of unique values of a column.	unique_count (customer)
unique_count_if	Returns the number of unique values of a column provided it meets a criteria.	unique_count_if((revenue > 10) , order date)
variance	Returns the variance of all the values of a column.	variance (revenue)
variance_if	Returns the variance of all the values of a column provided it meets a criteria..	variance_if((revenue > 10) , (revenue/10.0))

Conversion functions

Use these functions to convert data from one data type into another data type.

ThoughtSpot does not support *date* data type conversion.

Function	Description	Examples
to_bool	Returns the input as a boolean data type (true or false).	to_bool (0) = false to_bool (married)

Function	Description	Examples
to_date	<p>Accepts a date represented as an integer or text string, and a second string parameter that can include strftime date formatting elements.</p> <p>Replaces all the valid strftime date formatting elements with their string counterparts and returns the result.</p> <p>Does not accept epoch formatted dates as input.</p>	<code>to_date (date_sold, '%Y-%m-%d')</code>
to_double	Returns the input as a double data type.	<code>to_double ('3.14') = 3.14</code> <code>to_double (revenue * .01)</code>
to_integer	Returns the input as an integer.	<code>to_integer ('45') + 1 = 46</code> <code>to_integer (price + tax - cost)</code>
to_string	Returns the input as a text string. To convert a date data type to a string data type, specify the date format you want to use.	<code>to_string (45 + 1) = '46'</code> <code>to_string (revenue - cost)</code> <code>to_string (date, ('%m/%d/%y'))</code>

Date functions

Function	Description	Examples
add_days	Returns the result of adding the specified number of days to the given date.	<code>add_days (01/30/2015, 5) = 02/04/2015</code> <code>add_days (invoiced, 30)</code>
add_minutes	Returns the result of adding the specified number of minutes to input date/date-time/time.	<code>add_minutes (01/30/2015 00:10:20 , 5) = 01/30/2015 00:11:20</code> <code>add_minutes (invoiced , 30)</code>

Function	Description	Examples
add_months	Returns the result of adding the specified number of months to the given date.	add_months (01/30/2015, 5) = 06/30/2015 add_months (invoiced_date , 5)
add_seconds	Returns the result of adding the specified number of seconds to the given date.	add_seconds (01/30/2015 00:00:00, 5) = 06/30/2015 00:00:05 add_seconds (invoiced_date , 5)
add_weeks	Returns the result of adding the specified number of weeks to the given date.	add_weeks (01/30/2015, 2) = 02/13/2015 add_weeks (invoiced_date , 2)
add_years	Returns the result of adding the specified number of years to the given date.	add_years (01/30/2015, 5) = 01/30/2020 add_years (invoiced_date , 5)
date	Returns the date portion of a given date.	date (home visit)
day	Returns the number (1-31) of the day for the given date.	day (01/15/2014) = 15 day (date ordered)
day_number_of_quarter	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_quarter (01/30/2015) = 30 day_number_of_quarter (01/30/2015, 'fiscal') = 91
day_number_of_week	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	day_number_of_week(01/15/2014) = 3 day_number_of_week (shipped)

Function	Description	Examples
day_number_of_year	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_year (01/30/2015) = 30 day_number_of_year (01/30/2015, 'fiscal') = 275 day_number_of_year (invoiced)
day_of_week	Returns the day of the week for the given date.	day_of_week (01/30/2015) = Friday day_of_week (serviced)
diff_days	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	diff_days (01/15/2014, 01/17/2014) = -2 diff_days (purchased, shipped)
diff_time	Subtracts the second date from the first date and returns the result in number of seconds.	diff_time (01/01/2014, 01/01/2014) = -86,400 diff_time (clicked, submitted)
hour_of_day	Returns the hour of the day for the given date.	hour_of_day (received)
is_weekend	Returns true if the given date falls on a Saturday or Sunday.	is_weekend (01/31/2015) = true is_weekend (emailed)
month	Returns the month from the given date.	month (01/15/2014) = January month (date ordered)

Function	Description	Examples
month_number	Returns the number (1-12) of the month from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number (09/20/2014) = 9 month_number (09/20/2014, 'fiscal') = 5 month_number (purchased)
month_number_of_quarter	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number_of_quarter (02/20/2018) = 2 month_number_of_quarter (02/20/2018, 'fiscal') = 1
now	Returns the current timestamp.	now ()
quarter_number	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	quarter_number (04/14/2014) = 2 quarter_number (04/14/2014, 'fiscal') = 4 quarter_number (shipped)

Function	Description	Examples
start_of_month	Returns MMM yyyy for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as MM/dd/yyyy . Speak with your ThoughtSpot administrator for information on doing this.	start_of_month (01/31/2015) = Jan FY 2015 start_of_month (shipped)
start_of_quarter	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	start_of_quarter (04/01/2014) = Apr 2014 start_of_quarter (04/01/2014, 'fiscal') = Feb 2014 start_of_quarter (sold)
start_of_week	Returns the date for the first day of the week for the given date.	start_of_week (06/01/2015) = 05/30/2015 Week start_of_week (emailed)
start_of_year	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	start_of_year (04/01/2014) returns Jan 2014 start_of_year (04/01/2014, 'fiscal') returns May 2013 start_of_year (joined)
time	Returns the time portion of a given date.	time (3/1/2002 10:32) = 10:32 time (call began)

Function	Description	Examples
week_number_of_month	Returns the week number for the given date in a month.	week_number_of_month(03/23/2017) = 3
week_number_of_quarter	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_quarter (04/03/2017) = 1 week_number_of_quarter (04/03/2017, 'fiscal') = 10
week_number_of_year	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_year (01/17/2014) = 3 week_number_of_year (01/17/2014, 'fiscal') = 38
year	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	year (01/15/2014) = 2014 year (12/15/2013, 'fiscal') = 2014 year (date ordered)

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
<code>!=</code>	Returns true if the first value is not equal to the second value.	<code>3 != 2 = true</code> <code>revenue != 1000000</code>
<code><</code>	Returns true if the first value is less than the second value.	<code>3 < 2 = false</code> <code>revenue < 1000000</code>
<code><=</code>	Returns true if the first value is less than or equal to the second value.	<code>1 <= 2 = true</code> <code>revenue <= 1000000</code>
<code>=</code>	Returns true if the first value is equal to the second value.	<code>2 = 2 = true</code> <code>revenue = 1000000</code>
<code>></code>	Returns true if the first value is greater than the second value.	<code>3 > 2 = true</code> <code>revenue > 1000000</code>
<code>>=</code>	Returns true if the first value is greater than or equal to the second value.	<code>3 >= 2 = true</code> <code>revenue >= 1000000</code>
<code>greatest</code>	Returns the larger of the values.	<code>greatest (20, 10) = 20</code> <code>greatest (q1 revenue, q2 revenue)</code>
<code>least</code>	Returns the smaller of the values.	<code>least (20, 10) = 10</code> <code>least (q1 revenue, q2 revenue)</code>

Number functions

Function	Description	Examples
<code>*</code>	Returns the result of multiplying both numbers.	<code>3 * 2 = 6</code> <code>price * taxrate</code>
<code>+</code>	Returns the result of adding both numbers.	<code>1 + 2 = 3</code> <code>price + shipping</code>

Function	Description	Examples
-	Returns the result of subtracting the second number from the first.	$3 - 2 = 1$ revenue - tax
/	Returns the result of dividing the first number by the second.	$6 / 3 = 2$ markup / retail price
^	Returns the first number raised to the power of the second.	$3 ^ 2 = 9$ width ^ 2
abs	Returns the absolute value.	abs (-10) = 10 abs (profit)
acos	Returns the inverse cosine in degrees.	acos (0.5) = 60 acos (cos-satellite-angle)
asin	Returns the inverse sine (specified in degrees).	asin (0.5) = 30 asin (sin-satellite-angle)
atan	Returns the inverse tangent in degrees.	atan (1) = 45 atan (tan-satellite-angle)
atan2	Returns the inverse tangent in degrees.	atan2 (10, 10) = 45 atan2 (longitude, latitude)
cbrt	Returns the cube root of a number.	cbrt (27) = 3 cbrt (volume)
ceil	Returns the smallest following integer.	ceil (5.9) = 6 ceil (growth rate)
cos	Returns the cosine of an angle (specified in degrees).	cos (63) = 0.45 cos (beam angle)
cube	Returns the cube of a number.	cube (3) = 27 cube (length)
exp	Returns Euler's number (~2.718) raised to a power.	exp (2) = 7.38905609893 exp (growth)
exp2	Returns 2 raised to a power.	exp2 (3) = 8 exp2 (growth)
floor	Returns the largest previous integer.	floor (5.1) = 5 floor (growth rate)
ln	Returns the natural logarithm.	ln (7.38905609893) = 2 ln (distance)

Function	Description	Examples
log10	Returns the logarithm with base 10.	<code>log10 (100) = 2</code> <code>log10 (volume)</code>
log2	Returns the logarithm with base 2 (binary logarithm).	<code>log2 (32) = 5</code> <code>log2 (volume)</code>
mod	Returns the remainder of first number divided by the second number.	<code>mod (8, 3) = 2</code> <code>mod (revenue, quantity)</code>
pow	Returns the first number raised to the power of the second number.	<code>pow (5, 2) = 25</code> <code>pow (width, 2)</code>
random	Returns a random number between 0 and 1.	<code>random () = .457718</code> <code>random ()</code>
round	Returns the first number rounded to the second number (the default is 1).	<code>round (35.65, 10) = 40</code> <code>round (battingavg, 100)</code> <code>round (48.67, .1) = 48.7</code>
safe_divide	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	<code>safe_divide (12, 0) = 0</code> <code>safe_divide (total_cost, units)</code>
sign	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	<code>sign (-250) = -1</code> <code>sign (growth_rate)</code>
sin	Returns the sine of an angle (specified in degrees).	<code>sin (35) = 0.57</code> <code>sin (beam_angle)</code>
spherical_distance	Returns the distance in km between two points on Earth.	<code>spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96</code> <code>spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)</code>
sq	Returns the square of a numeric value.	<code>sq (9) = 81</code> <code>sq (width)</code>
sqrt	Returns the square root.	<code>sqrt (9) = 3</code> <code>sqrt (area)</code>
tan	Returns the tangent of an angle (specified in degrees).	<code>tan (35) = 0.7</code> <code>tan (beam_angle)</code>

Text functions

Function	Description	Examples
concat	Returns two or more values as a concatenated text string. Use single quotes around each literal string, not double quotes.	concat ('hay' , 'stack') = 'haystack' concat (title, ' ', first_name , ' ', last_name)
contains	Returns true if the first string contains the second string, otherwise returns false.	contains ('broomstick', 'room') = true contains (product, 'trial version')
edit_distance	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	edit_distance ('attorney', 'atty') = 4 edit_distance (color, 'red')
edit_distance_with_cap	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4 edit_distance_with_cap (event, 'burning man', 3)
similar_to	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	similar_to ('hello world', 'hello swirl') = true similar_to (current team, drafted by)

Function	Description	Examples
similarity	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	similarity ('where is the burning man concert', 'burning man') = 46 similarity (tweet1, tweet2)
spells_like	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	spells_like ('thouhgtspot', 'thoughtspot') = true spells_like (studio, distributor)
strlen	Returns the length of the text.	strlen ('smith') = 5 strlen (lastname)
strpos	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	strpos ('haystack_with_needles', 'needle') = 14 strpos (complaint, 'lawyer')
substr	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	substr ('persnickety', 3, 7) = snicket substr (lastname, 0, 5)

Alerts code reference

This reference identifies the messages that can appear in the **System Health > Overview > Critical Alerts** and in the **Alerts** dashboard.

Informational alerts

TASK_TERMINATED

Msg: Task {{.Service}}.{{.Task}} terminated on machine {{.Machine}}

Type: INFO

This alert is raised when a task terminates.

DISK_ERROR

Msg: Machine {{.Machine}} has disk errors

Type: INFO

Raised when a machine has disk errors.

ZK_AVG_LATENCY

Msg: Average Zookeeper latency is more than {{.Num}} msec

Type: INFO

Raised when average Zookeeper latency is above a threshold.

ZK_MAX_LATENCY

Msg: Max Zookeeper latency is more than {{.Num}} msec

Type: INFO

Raised when max Zookeeper latency is above a threshold.

ZK_MIN_LATENCY

Msg: Min Zookeeper latency is more than {{.Num}} msec

Type: INFO

Raised when min Zookeeper latency is above a threshold.

ZK_OUTSTANDING_REQUESTS

Msg: Number of outstanding Zookeeper requests exceeds {{.Num}}

Type: INFO

Raised when there are too many outstanding Zookeeper requests.

ZK_NUM_WATCHERS

Msg: Number of Zookeeper watchers exceeds {{.Num}}

Type: INFO

Raised when there are too many Zookeeper watchers.

MASTER_ELECTION

Msg: {{.Machine}} elected as Orion Master

Type: INFO

Raised when a new Orion Master is elected.

PERIODIC_BACKUP

Msg: {{.Process}} periodic backup for policy {{.Name}} failed.

Type: INFO

Raised when periodic backup fails.

PERIODIC_SNAPSHOT

Msg: {{.Process}} periodic snapshot {{.Name}} failed.

Type: INFO

Raised when a periodic snapshot fails.

HDFS_CORRUPTION

Msg: HDFS root directory is in a corrupted state.

Type: INFO

Raised when HDFS root directory is corrupted.

APPLICATION_INVALID_STATE

Msg: {{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}

Type: INFO

Raised when Application raises invalid state alert.

UPDATE_START

Msg: Starting update of ThoughtSpot cluster {{.Cluster}}

Type: INFO

Raised when update starts.

UPDATE_END

Msg: Finished update of ThoughtSpot cluster {{.Cluster}} to release {{.Release}}

Type: INFO

Raised when update completes.

Errors

TIMELY_JOB_RUN_ERROR

Msg: Job run {{.Message}}

Type: ERROR

Raised when a job run fails.

TIMELY_ERROR

Msg: Job manager {{.Message}}

Type: ERROR

Raised when a job manager runs into an inconsistent state.

Warnings

DISK_SPACE

Msg: Machine {{.Machine}} has less than {{.Perc}}% disk space free

Type: WARNING

Raised when a disk is low on available disk space. Valid only in the 3.2 version of ThoughtSpot.

ROOT_DISK_SPACE

Msg: Machine {{.Machine}} has less than {{.Perc}}% disk space free on root partition

Type: WARNING

Raised when a machine is low on available disk space on root partition.

BOOT_DISK_SPACE

Msg: Machine {{.Machine}} has less than {{.Perc}}% disk space free on boot partition

Type: WARNING

Raised when a machine is low on available disk space on boot partition.

UPDATE_DISK_SPACE

Msg: Machine {{.Machine}} has less than {{.Perc}}% disk space free on update partition

Type: WARNING

Raised when a machine is low on available disk space on update partition.

EXPORT_DISK_SPACE

Msg: Machine {{.Machine}} has less than {{.Perc}}% disk space free on export partition

Type: WARNING

Raised when a machine is low on available disk space on export partition.

HDFS_NAMENODE_DISK_SPACE

Msg: Machine {{.Machine}} has less than {{.Perc}}% disk space free on HDFS namenode drive

Type: WARNING

Raised when a machine is low on available disk space on HDFS namenode drive.

MEMORY

Msg: Machine {{.Machine}} has less than {{.Perc}}% memory free

Type: WARNING

Raised when a machine is low on free memory.

OS_USERS

Msg: Machine {{.Machine}} has more than {{.Num}} logged in users

Type: WARNING

Raised when a machine has too many users logged in.

OS_PROCS

Msg: Machine {{.Machine}} has more than {{.Num}} processes

Type: WARNING

Raised when a machine has more too many processes.

SSH

Msg: Machine {{.Machine}} doesn't have an active SSH server

Type: WARNING

Raised when a machine has more than 600 processes.

DISK_ERROR_EXTERNAL

Msg: Machine {{.Machine}} has disk errors

Type: WARNING

Raised when more than 2 disk errors happen in a day.

ZK_FD_COUNT

Msg: Zookeeper has more than {{.Num}} open file descriptors

Type: WARNING

Raised when there are too many open Zookeeper files.

ZK_EPHEMERAL_COUNT

Msg: Zookeeper has more than {{.Num}} ephemeral files

Type: WARNING

Raised when there are too many Zookeeper ephemeral files.

HOST_DOWN

Msg: {{.Machine}} is down

Type: WARNING

Raised when a host is down.

TASK_UNREACHABLE

Msg: {{.ServiceDesc}} on {{.Machine}} is unreachable over HTTP

Type: WARNING

Raised when a task is unreachable over HTTP.

TASK_NOT_RUNNING

Msg: {{.ServiceDesc}} is not running

Type: WARNING

Raised when a service task is not running on any machine in the cluster.

Critical alerts

TASK_FLAPPING

Msg: Task {{.Service}}.{{.Task}} terminated {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}

Type: CRITICAL

This alert is raised when a task is crashing repeatedly. The service is evaluated across the whole cluster.

So, if a service crashes 5 times in a day across all nodes in the cluster, this alert is generated.

OREO_TERMINATED

Msg: Oreo terminated on machine {{.Machine}}

Type: CRITICAL

This alert is raised when the Oreo daemon on a machine terminates due to an error. This typically happens due to an error accessing Zookeeper, HDFS, or a hardware issue.

HDFS_DISK_SPACE

Msg: HDFS has less than {{.Perc}}% space free

Type: CRITICAL

Raised when a HDFS cluster is low on total available disk space.

ZK_INACCESSIBLE

Msg: Zookeeper is not accessible

Type: CRITICAL

Raised when Zookeeper is inaccessible.

PERIODIC_BACKUP_FLAPPING

Msg: Periodic backup failed {{._actual_num_occurrences}} times in last
{{._earliest_duration_str}}

Type: CRITICAL

This alert is raised when a periodic backup failed repeatedly.

PERIODIC_SNAPSHOT_FLAPPING

Msg: Periodic snapshot failed {{._actual_num_occurrences}} times in last
{{._earliest_duration_str}}

Type: CRITICAL

This alert is raised when periodic snapshot failed repeatedly.

APPLICATION_INVALID_STATE_EXTERNAL

Msg: {{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}

Type: CRITICAL

Raised when Application raises invalid state alert.

User action code reference

This reference identifies the user action codes that can appear in the **System Health** pages and in logs or other reports.

answer_unsaved	User makes a change to tokens in the search bar.
answer_saved	User opens an existing saved answer and makes changes to tokens in the search bar.
answer_pinboard_context	User opens an existing saved pinboard, edits a context viz and makes a change to tokens in the search bar.
answer_aggregated_worksheet	User opens an existing saved aggregated worksheet and makes changes to tokens in the search bar.
answer_upgrade	Requests made for the sole purpose of upgrade.
pinboard_view	User opens an existing saved pinboard.
pinboard_filter	User adds, removes or applies values to a pinboard filter.
pinboard_ad_hoc	User drills down in a pinboard viz.
data_chart_config	Request for new data being generated following a chart config change.
data_show_underlying_row	Request to show underlying data for a data row(s).
data_export	Request to export data.
pinboard_tspublic_runtime_filter	Request to TSPublic/pinboarddata with runtime filters.
answer_aggregated_worksheet_save	User updates aggregated worksheet.
answer_add_new_filter	User adds a filter using the UI.
data_show_underlying_viz	Request to show underlying data for a data row(s).
answer_view	User opens an existing, saved answer.

answer_viz_context_view	User opens an existing saved pinboard, edits a context viz.
pinboard_insight_view	User opens SpotIQ tab pinboards.
pinboard_admin_view	User opens admin tab pinboards.
pinboard_embed_view	User opens embed pinboard from a URL.
pinboard_homepage_view	On loading of homepage pinboard.
pinboard_learn_view	On loading learn pinboard.
pinboard_tspublic_no_runtime_filter	Request to TSPublic/pinboard data without run-time filters.

Frequently asked questions

Where can I find the version of ThoughtSpot I am using?

Users with administrative privileges can see this displayed on the **Admin > System Health > Overview** page.

I'm not seeing certain columns/values in the drop-down, why?

It could be the index has not built with the latest data or something is causing the column to be dropped.

- Verify the the column is available using the **Data** page.
- View the table columns and check the **INDEX TYPE** value. If it is set to `DONT_INDEX`, change it.
- Check the column's **INDEX PRIORITY** — make sure it is `1`.

To learn more about modeling data see [modeling data](#) in this documentation.