

Tableau Server Administrator Guide

Version 9.2; Last Updated in 2015

Copyright © 2015 Tableau Software, Incorporated and its licensors. All rights reserved.

This product is Client Software as defined in Tableau Software's End User Software License Agreement.



Before you install...

Make sure the computer on which you're installing Tableau Server meets the following requirements:

- **Supported operating systems**—Tableau Server is available in 32-bit and 64-bit versions. You can install Tableau Server on Windows Server 2008 or higher, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 7, Windows 8, Windows 8.1, or Windows 10. The 64-bit version of Tableau Server is recommended on a 64-bit operating system. You may install Tableau Server on virtual or physical platforms.
- **Minimum requirements**—The computer you install Tableau Server on must meet or exceed the minimum hardware requirements. Tableau Server will not install if your computer does not meet the minimum requirements.

Minimum requirements are appropriate for testing and prototyping. For production environments your computers should meet or exceed the minimum recommendations. For more information, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#) on page 59.

- **Administrative account**—The account under which you install Tableau Server must have permission to install software and services.
- **Optional: Run As Account**—A Run As User account for the Tableau Server service to run under is useful if you're using NT Authentication with data sources or if you're planning on doing SQL Server impersonation. For more information, see [Run As User](#) on page 525 and [SQL Server Impersonation](#) on page 534.
- **IIS and port 80**—Tableau Server's gateway listens on port 80, which is also used by Internet Information Services (IIS) by default. If you are installing Tableau Server on a machine that's also running IIS, you should modify the Tableau's gateway port number to avoid conflict with IIS. See [Tableau Server Ports](#) on page 540 and [Edit the Default Ports](#) on page 547 for details.
- **Static IP addresses**—Any computer running Tableau Server, whether it's a single server installation or part of a cluster, must have a static IP address. For more information, see [Hostname Support in Tableau Server](#) on page 75.

Configuration Information

When you install and configure Tableau Server you may be asked for the following information:

Option	Description	Your Information
Server	The server must have a user account that the service	Username:

Option	Description	Your Information
Account	can use. The default is the built-in Windows Network Service account. If you use a specific user account you'll need the domain name, user name, and password.	Password: Domain:
Active Directory	Instead of using Tableau's built-in user management system, you can authenticate through Active Directory. If so, you'll need the fully-qualified domain name .	Active Directory Domain:
Open port in Windows firewall	When selected Tableau Server will open the port used for http requests in the Windows Firewall software to allow other machines on your network to access the server.	<input type="checkbox"/> - Yes <input type="checkbox"/> - No

Ports

By default Tableau Server requires several TCP/IP ports to be available to the server. See the topic [Tableau Server Ports on page 540](#) for the full list, including which ports must be available for all installations vs. distributed installations or failover-ready installations. The default ports can be changed if there is a conflict. See [Edit the Default Ports on page 547](#) to learn how.

Drivers

You may need to install additional database drivers. Download drivers from www.tableau.com/support/drivers.

What's New and What's Changed

Find out about the new and changed features in Tableau Server:

- See the What's New in Tableau Server topic in the Tableau Server online help for information about key new features.
- See [What's Changed - Things to Know Before You Upgrade on page 1](#) for information about changes that may impact your users.

Minimum Hardware Requirements and Recommendations for Tableau Server

The following minimum hardware requirements and recommendations apply to all computers running Tableau Server, including physical hardware and virtual machines (VMs):

- **Minimum requirements** are the minimum hardware your computer must have in order for Setup to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping.
- **Minimum recommendations** are higher than minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the Setup program will warn you but you can continue the installation.

In addition, Tableau Server should not be installed on a physical computer or on a VM instance that is also running resource-intensive applications such as databases or application servers.

Note: If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see [Server Process Defaults and Limits on page 67](#)

Minimum Hardware Requirements

The computer on which you are installing or upgrading Tableau Server must meet the minimum hardware requirements. If the Setup program determines that your computer does not meet the following requirements, you will not be able to install Tableau Server. For more information on how the Setup program determines hardware, see "Determining Computer Hardware," below.

These minimum requirements are appropriate for prototyping and testing of Tableau Server and apply to single-node installations and to each computer in a distributed installation.

Server Version	CPU	RAM	Free Disk Space
64-bit Tableau Server	4-core	8 GB	15 GB
32-bit Tableau Server	2-core	4 GB	15 GB

For the requirements:

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space.
- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

If you cannot install the 64-bit Tableau Server because of hardware requirements but your computer meets the minimum hardware requirements for the 32-bit version of Tableau Server, you may be able to install the 32-bit version.

Note: For Tableau Server 9.2 on a 64-bit virtual machine, you need a minimum of 4 physical cores. If you are installing on an Amazon EC2 instance, this means 8 vCPUs. For more information, see [Amazon EC2 Instances](#).

Minimum Hardware Recommendations

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors including number of users, and number and size of extracts.

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Single node	64-bit	8-core, 2.0 GHz or higher	32 GB	50 GB
Multi-node and Enterprise deployments	Contact Tableau for technical guidance. Nodes must meet or exceed the minimum hardware recommendations, except nodes running backgrounder, where 4 cores may be acceptable.			

Determining Computer Hardware

The Tableau Server Setup program determines how many physical cores a computer has by querying the operating system . To view hardware information the Setup program detected on your computer, open the `tabadmin.log` file located on the computer where you are installing Tableau Server:

```
<install directory>\ProgramData\Tableau\Tableau Server-logs\tabadmin.log
```

In `tabadmin.log`, look for lines similar to the following to check the physical and logical cores that Setup detected and used to determine the core count that is being used for licensing:

```
2015-04-09 14:22:29.533 -0700 DEBUG_10.36.2.32:<machine name>:_  
pid=21488_0x2cd83560__user=__request=__ Running hardware check  
  
2015-04-09 14:22:29.713 -0700 DEBUG_10.36.2.32:<machine name>:_  
pid=21488_0x2cd83560__user=__request=__ Detected 12 cores and  
34281857024 bytes of memory  
  
2015-04-09 14:22:29.716 -0700 DEBUG_10.36.2.32:<machine name>:_  
pid=21488_0x2cd83560__user=__request=__ Hardware meets recom-  
mended specifications. Default values will be used.
```

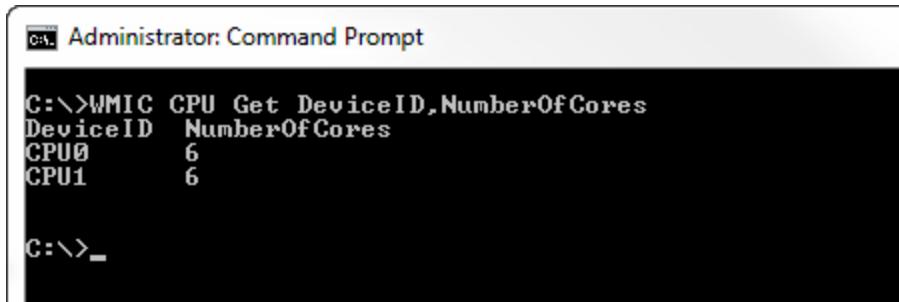
Manually determining the number of cores on your computer

You can use the Windows Management Instrumentation Command-line tool (WMIC) to determine how many physical cores your server has. This is useful if you do not know whether your computer will meet the minimum hardware requirements for installing Tableau Server.

1. Open a command prompt.
2. Enter the following command:

```
WMIC CPU Get DeviceID,NumberOfCores
```

The output will display the device id or ids and the number of physical cores the computer has:



```
C:\>WMIC CPU Get DeviceID,NumberOfCores  
DeviceID  NumberOfCores  
CPU0      6  
CPU1      6  
  
C:\>_
```

In the above example there are two CPUs, each with six cores, for a total of twelve physical cores. This computer would satisfy the minimum hardware requirements for installing 64-bit Tableau Server.

A longer command will list the logical processors as well as the physical cores:

```
WMIC CPU Get  
DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesign  
ation
```

```
Administrator: Command Prompt  
C:\>WMIC CPU Get DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesignation  
DeviceID  NumberOfCores  NumberOfLogicalProcessors  SocketDesignation  
CPU0      6              12                      CPU 1  
CPU1      6              12                      CPU 2  
C:\>-
```

In the above example, in addition to the twelve physical cores, there are 24 logical cores.

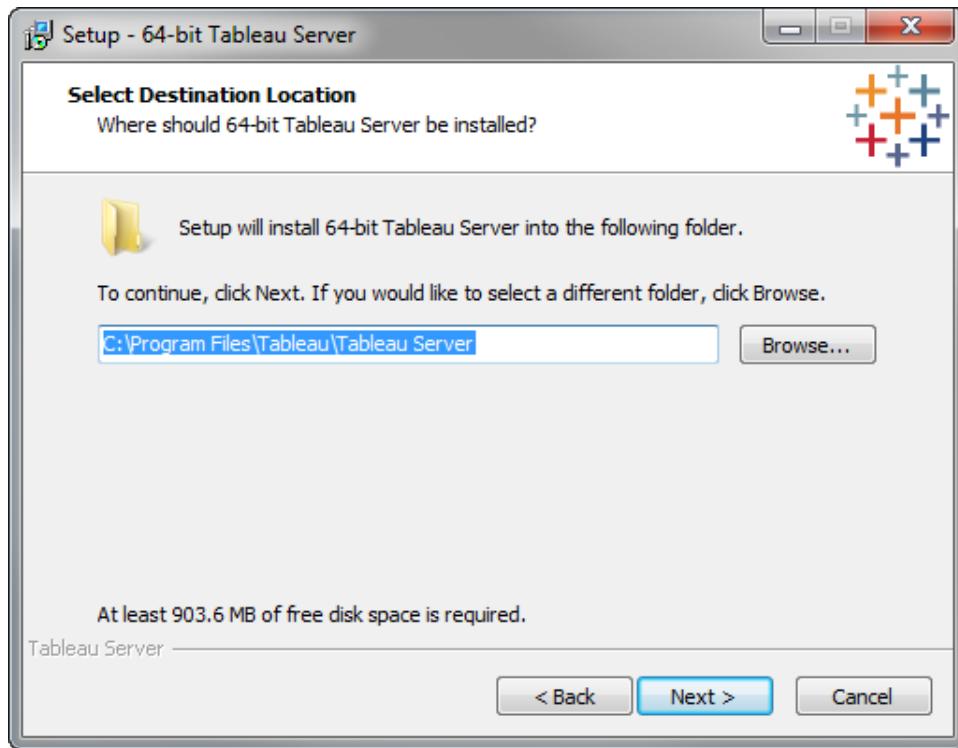
Install and Configure

Here are the main steps you need to take to install and configure Tableau Server:

Run Server Setup

After you download the Tableau Server installation file, follow the instructions below to install the server.

1. Double-click the installation file.
2. Follow the on-screen instructions to complete Setup and install the application.



Note: If you are upgrading and your original installation was not the default location, when you browse to the location, do not include the Tableau Server folder. If you include the Tableau Server folder you will install to a second Tableau Server folder, for example *install-drive\Program Files\Tableau\Tableau Server\Tableau Server*. Verify the path in the Setup program after you select the location. For more information on upgrading, see [Upgrade to 9.2 on page 62](#)

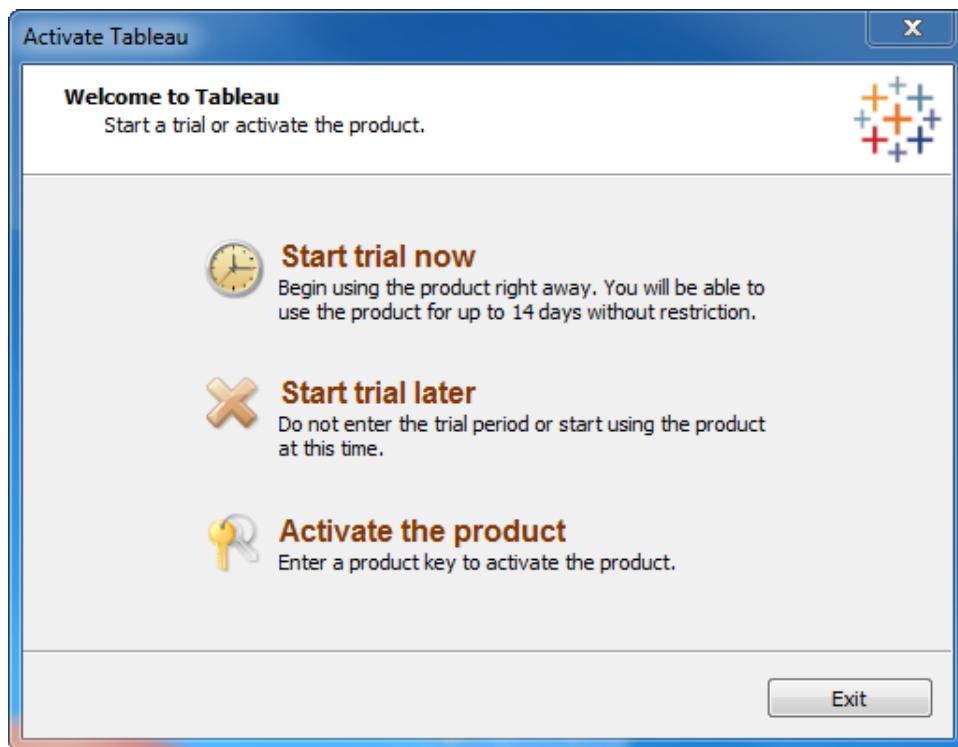
3. After the installation completes, click **Next** to open the Product Key Manager window.

If you need to support characters that are not the Latin-1 set, install the Windows Language Packs via **Control Panel > Regional and Language Options**. The language packs will need to be installed on the primary server as well as any worker machines.

Activate Tableau

Tableau Server requires at least one product key that both activates the server and specifies the number of license levels you can assign to users. You can access your product keys from the [Tableau Customer Account Center](#). After installing and configuring the server, the product key manager automatically opens so you can enter your product key and register the product. If you need to activate the product on a computer that is offline, see [Activate Tableau Offline below](#).

1. Select Activate and paste in your product key:



2. Refer to the [download help page](#) on the web site for step-by-step instructions.

Activate Tableau Offline

If you are working offline you can follow the steps below to complete offline activation.

1. When the product key manager opens click **Activate the product**.
Paste your server product key into the corresponding text box and click Activate. You can get your product key from the [Tableau Customer Portal](#).
2. When you are offline, activation will fail and you are given the option to save a file that you can use for offline activation. Click **Save**.
3. Select a location for the file and click **Save**. The file is saved as **offline.tlq**.
4. Back in Tableau click **Exit** to close the Activation dialog box.
5. From a computer that has Internet access, open a web browser and visit the [Product Activations](#) page on the Tableau website. Complete the instructions to submit your offline.tlq file.

After you submit your offline.tlq file online, while your browser is still displaying the Product Activations page, a file called **activation.tlf** is created, and Tableau prompts you to save the file to your computer.
6. Save the activation.tlf file and move it to the computer where you are installing Tableau Server.
7. On the computer where you are installing Tableau Server, open a command prompt as an administrator and run the following command:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```
8. Next, type `tabadmin activate --tlf <path>\activation.tlf`, where `<path>` is the location of the response file you saved from the Product Activations page. For example:

```
tabadmin activate --tlf \Desktop\activation.tlf
```

Keep the command prompt window open.
9. After the license is initialized, you are prompted to activate the product again. On Tableau Server, click **Start > All Programs > Tableau Server 9.2**
10. Right-click **Manage Product Keys** and select **Run as Administrator**.

Even if you are logged into the Tableau Server computer as an administrator, you need to do this to avoid a potential registration error.
11. Click **Activate the product**.
12. Enter your product key again (the same one you entered in step 1).
13. Save the .tlq file.
14. From a computer that has Internet access, open a web browser and visit the [Product Activations](#) page again on the Tableau website. Complete the instructions.

Tableau will again create a file called **activation.tlf** and prompt you to save it.

15. Save the file and move it to the computer where you are installing Tableau Server.
16. Back in the command prompt window on Tableau Server, type `tabadmin activate --tlf <path>\activation.tlf`, where `<path>` is the location of the second response file you saved from the Product Activations page. For example:

```
tabadmin activate --tlf \Desktop\activation.tlf
```

Tableau Server is now activated. If you need additional assistance, [contact Tableau Customer Service](#).

Configure Tableau Server

The Tableau Server Configuration utility opens during a Tableau Server installation. You can set configuration options at this time, as part of the installation, before the server starts. The server is started at the end of the installation process.

You can also run the utility after installing Tableau Server by selecting **All Programs > Tableau Server 9.2 > Configure Tableau Server** on the Windows Start menu. You need to stop the server before making any configuration changes. See [Reconfigure the Server on page 35](#) for steps.

There are two things to keep in mind about the settings you specify in the Configuration dialog box:

- **Settings are system-wide:** The settings you enter apply to the entire server. If the server is running multiple sites, these settings affect every site.
- **User Authentication is "permanent":** The **User Authentication** setting (on the **General** tab) can only be set when you are installing Tableau Server for the first time. You can change all of the other settings after installation by stopping the server and rerunning the Configuration utility.

See the topics below for details on the different Configuration tabs:

Configure General Server Options

Use the steps below to configure options on the General tab:

1. By default, Tableau Server runs under the Network Service account. To use an account that will accommodate NT authentication with data sources, specify a user name and password. The user name should include the domain name. See [Run As User on page 525](#) to learn more about using a specific user account.

Server Run As User

Tableau Server requires a Windows account that it can run under.

User: MYCO\TableauServer Password: *****

Example: DOMAIN\username

2. Select whether to use **Active Directory** to authenticate users on the server. Select **Use Local Authentication** to create users and assign passwords using Tableau Server's built-in user management system. You cannot switch between Active Directory and Local Authentication later.

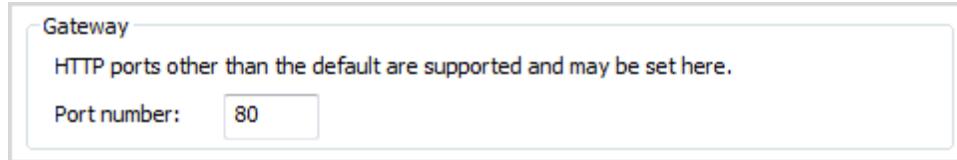
User Authentication Tableau Server can manage user names and passwords or use an existing Active Directory. <input checked="" type="radio"/> Use Active Directory <input type="radio"/> Use Local Authentication	Active Directory Domain: myco.lan Nickname: MYCO <input checked="" type="checkbox"/> Enable automatic logon
--	---

3. If you use Active Directory:

- You can optionally **Enable automatic logon**, which uses Microsoft SSPI to automatically sign in your users based on their Windows username and password. This creates an experience similar to single sign-on (SSO). Do not select **Enable automatic logon** if you plan to configure Tableau Server for **SAML**, **trusted authentication**, or for a **proxy server**.
- Be sure to type the fully qualified domain name (FQDN) and nickname.

To determine the FQDN: Select **Start > Run** then type `sysdm.cpl` in the Run textbox. In the System Properties dialog box, select the **Computer Name** tab. The FQDN is shown near the middle of the dialog box. The first time your users sign in, they will need to use the fully qualified domain name (for example, `myco.lan\jsmith`). On subsequent sign-ins, they can use the nickname (`myco\jsmith`).

4. The default port for web access to Tableau Server (via HTTP) is port 80. You may need to change the port number if you have another server running on port 80 or other networking needs. For example, you may have a hardware firewall or proxy in front of the Tableau Server host, which might make running a back-end system on port 80 undesirable.



5. Select whether to open a port in Windows firewall. If you do not open this port, users on other machines may not be able to access the server.



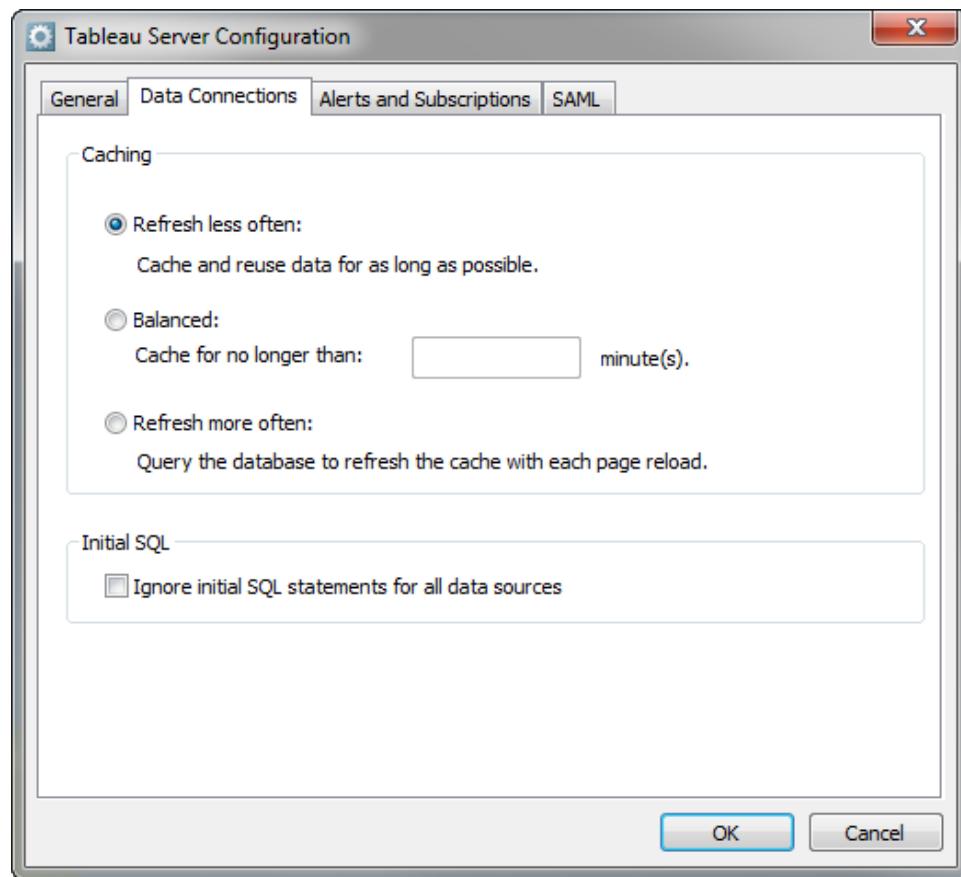
6. Select whether to include sample data and users. The **Include sample data and users** option installs several sample workbooks and data, which can help you get familiar with Tableau Server (especially if you are installing a trial version of the product). If you select **Include sample data and users**, the first user created in Tableau Server will be assigned as the owner of sample workbooks and data. To change the assigned owner, see [Manage Ownership](#) on page 331.
7. Optionally continue to the next page to configure Caching and Initial SQL options. If you do not want to configure these options click **OK**.

Configure Data Connections

Use the options on the Data Connections tab to configure caching and specify how you want to handle initial SQL statements from data sources.

Caching

Views published to Tableau Server are interactive and sometimes have a live connection to a database. As users interact with the views in a web browser, the data that is queried gets stored in a cache. Subsequent visits will pull the data from this cache if it is available. The Data Connections tab is where you configure aspects of caching that will apply to all data connections:



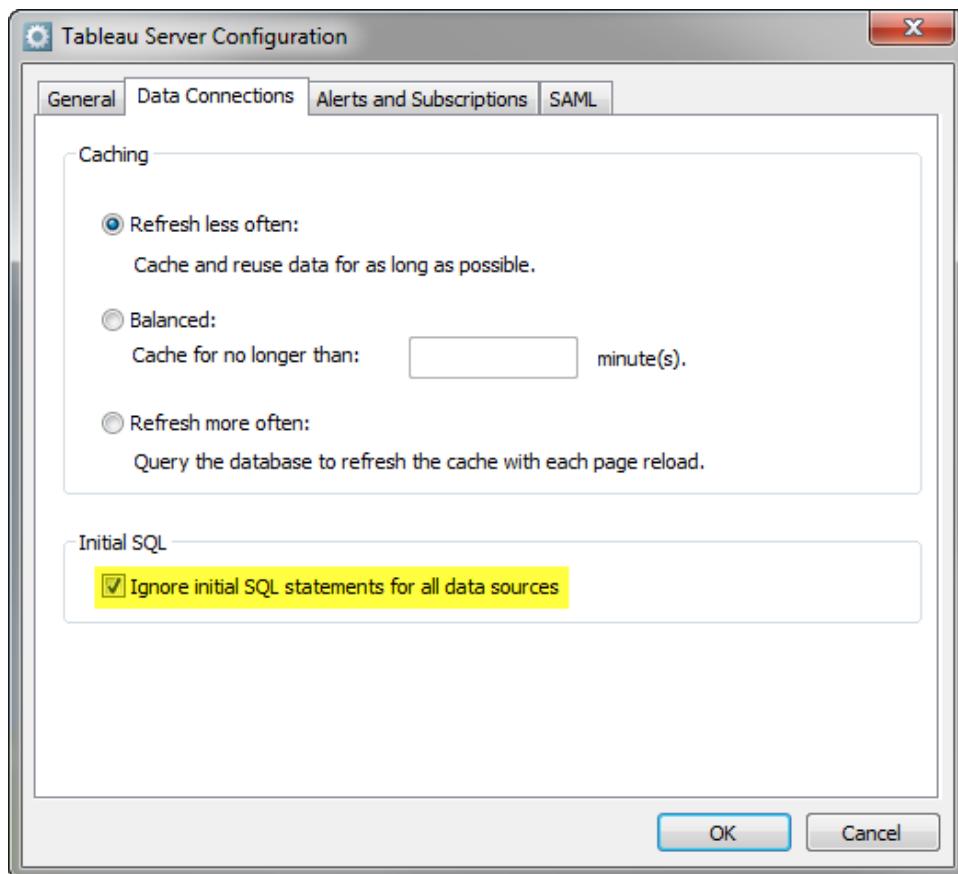
To configure caching, select from one of the following options:

- **Refresh Less Often**—Data is cached and reused whenever it is available regardless of when it was added to the cache. This option minimizes the number of queries sent to the database. Select this option when data is not changing frequently. Refreshing less often may improve performance.
- **Balanced**—Data is removed from the cache after a specified number of minutes. If the data has been added to the cache within the specified time range the cached data will be used, otherwise new data will be queried from the database.
- **Refresh More Often**—The database is queried each time the page is loaded. The data is still cached and will be reused until the user reloads the page. This option will ensure users see the most up to date data; however, it may decrease performance.

Regardless of how caching is configured, the user can click the **Refresh Data** button on the toolbar to force the server to send a query and retrieve new data.

Initial SQL

For views that connect to Teradata data sources, workbook creators can specify a SQL command that will run once, when the workbook is loaded in the browser. This is called an initial SQL statement. For performance or security reasons, some administrators may want to disable this functionality. The **Data Connections** tab is where you do this:



To disable initial SQL functionality, select the **Ignore initial SQL statements for all data sources** checkbox. Workbooks created with initial SQL statements will still open but the initial SQL commands will not be sent.

Configure Alerts and Subscriptions

Tableau Server can send email to alert system administrators if there is a system failure and can email subscriptions (snapshots of selected views) to system users. Configure the SMTP server Tableau Server uses to send email for alerts and subscriptions on the **Alerts and Subscriptions** tab.

Note: Encrypted SMTP connections are not supported for alerts or subscriptions.

Configure Email Alerts

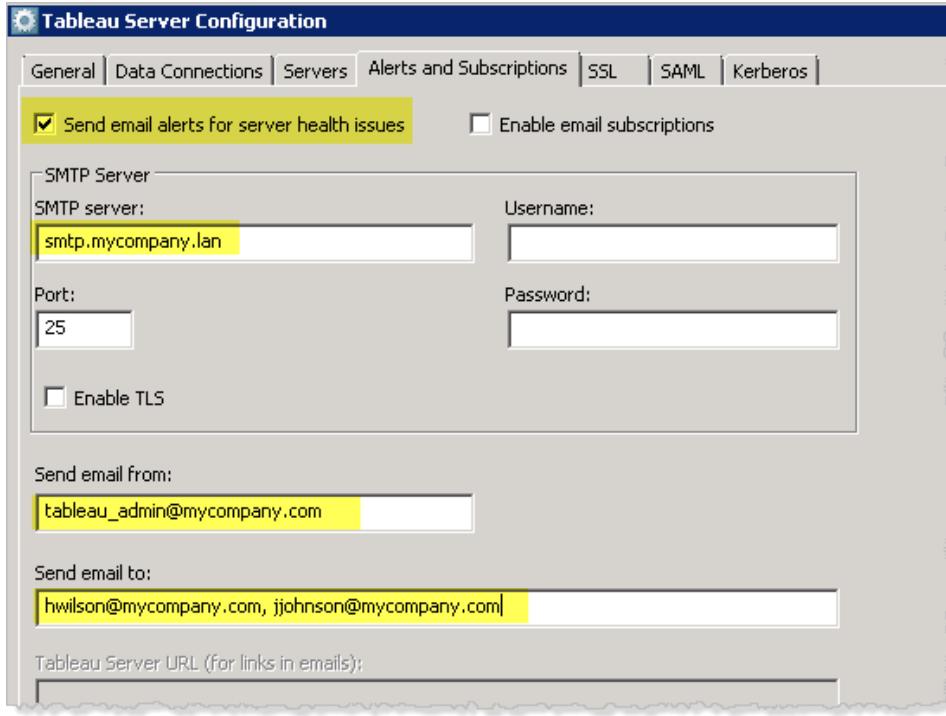
When you configure alerts, Tableau Server sends an email to the recipients under Send email to any time the data engine, repository, or gateway server processes stop or restart, or any time the primary Tableau Server stops or restarts. If you are running a single-server installation (all processes on the same machine), health alerts are only sent when Tableau Server is up. No DOWN alerts are sent. If you are running a distributed installation that's configured for failover (see [Configure for Failover and Multiple Gateways on page 94](#)) a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

To configure email alerts

1. Select **Send email alerts for server health issues**.



2. Under **SMTP Server**:
 - a. Enter the name of your SMTP server.
 - b. (Optional) Enter a **Username** and **Password** for your SMTP server account only if it requires one (some do, some do not).
 - c. The default SMTP port value is 25. Change this only if you know you are not using port 25.
 - d. For **Send email from**, enter the email address that will send an alert if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.com or noreply@mycompany), but it does not have to also be an actual email account on Tableau Server.
 - e. Leave the **Enable TLS** box cleared so that the connection to your mail server is unencrypted.



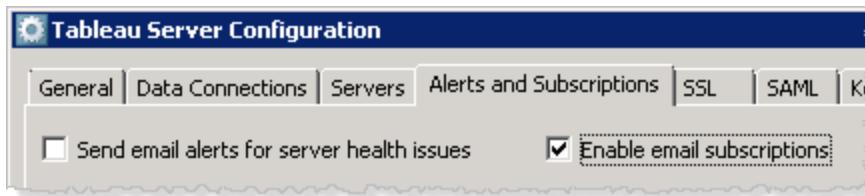
3. For **Send email to**, enter at least one email address that will receive the alerts. If you enter multiple addresses, separate them with commas.
4. Click **OK**.

When you **start the server** it will trigger an email alert and this confirms that you have set up alerts correctly.

Configure SMTP for Email Subscriptions

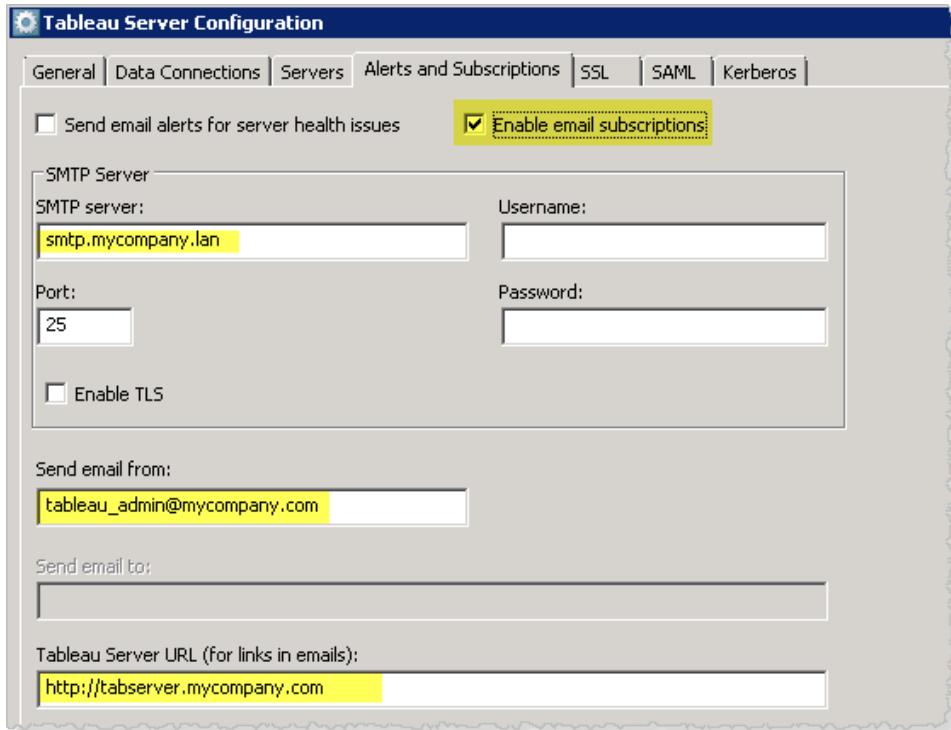
To configure email subscriptions

1. Select **Enable email subscriptions**.



2. Under **SMTP Server**, enter the name of your SMTP server. Enter a **Username** and **Password** for your SMTP server account only if it requires one (some do, some do not). The default SMTP port value is 25. Under **Send email from**, enter the email address that will send subscriptions to Tableau Server users.

- a. Enter the name of your SMTP server.
- b. (Optional) Enter a **Username** and **Password** for your SMTP server account only if it requires one (some do, some do not).
- c. The default SMTP port value is 25. Change this only if you know you are not using port 25.
- d. For **Send email from**, enter the email address that will send an alert if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.com or noreply@mycompany), and Tableau Server does not require it to be an actual email account (however, some SMTP servers may require it to be an actual email account). You can override this system-wide **Send email from** address on a per-site basis for subscriptions. See [Add or Edit Sites on page 117](#) for details.
- e. Leave the **Enable TLS** box cleared so that the connection to your mail server is unencrypted.
- f. For **Send email to**, enter at least one email address that will receive the alerts. If you enter multiple addresses, separate them with commas.



3. Under **Tableau Server URL**, enter `http://` or `https://`, followed by the name of the Tableau Server. This name will be used for the footer of subscription emails.
4. Click **OK**.

Configure External SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the web browser and the server or Tableau Desktop and the server is protected. Steps on how to configure the server for SSL are described in the topic below; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server. If you are running a Tableau Server cluster and you want to use SSL, see [Configure SSL for a Cluster on page 22](#), below, for recommendations.

1. Acquire an Apache SSL certificate from a trusted authority (for example, Verisign, Thawte, Comodo, GoDaddy). You can also use an internal certificate issued by your company. Wildcard certificates, which allow you to use SSL with many host names within the same domain, are also supported.

Some browsers will require additional configuration to accept certificates from certain providers. Refer to the documentation provided by your certificate authority.

2. Place the certificate files in a folder named SSL, parallel to the Tableau Server 9.2 folder. For example:

```
C:\Program Files\Tableau\Tableau Server\SSL
```

This location gives the account that's running Tableau Server the necessary permissions for the files.

Note: You may need to create this folder.

3. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 9.2 > Configure Tableau Server** on the Start menu.
4. In the Configuration Tableau Server dialog box, select the **SSL** tab.
5. Select **Use SSL for server communication** and provide the location for each of the following certificate files:

- **SSL certificate file**—Must be a valid PEM-encoded x509 certificate with the extension .crt.

SSL certificate key file—Must be a valid RSA or DSA key that has an embedded passphrase, and is not password protected with the file extension .key.

SSL certificate chain file (Optional for Tableau Server, required for Tableau Mobile and Tableau Desktop on the Mac)—Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate. All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

6. (optional) If you are using SSL for server communication and want to configure SSL communication between Tableau Server and clients using certificates on both the server and clients:

- Select **Use mutual SSL and automatic login with client certificates**.
- In **SSL CA certificate file**, browse to the location for the certificate file. The SSL CA certificate file must be a valid PEM-encoded x509 certificate with the extension .crt.

Note: If you have multiple trusted Certificate Authorities (CAs) you can copy and paste the entire contents of each CA certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines, into a new file, then save the file as CAs.crt. In **SSL CA certificate file**, browse to the location of this new file.

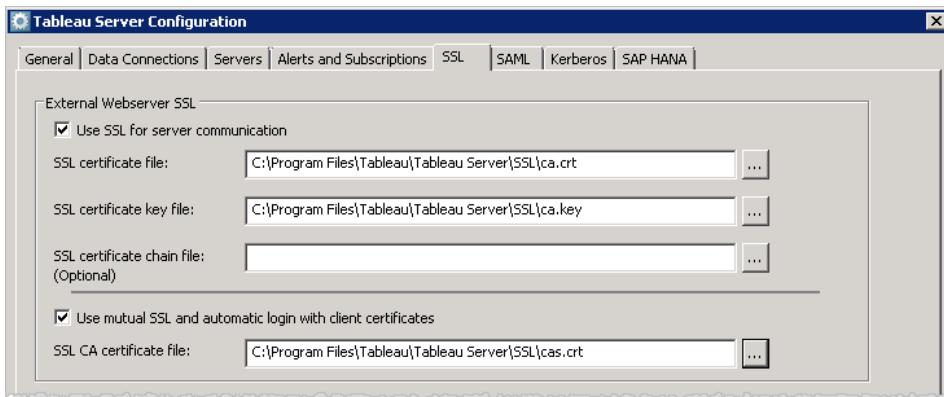
7. Click **OK**. The changes will take effect the next time the server is restarted.

When the server is configured for SSL, it accepts requests to the non-SSL port (default is port 80) and automatically redirects to the SSL port 443.

Note: Tableau Server only supports port 443 as the secure port. It cannot run on a computer where another application is using port 443.

SSL errors are logged in the install directory at the following location. Use this log to troubleshoot validation and encryption issues:

C:\ProgramData\Tableau\Tableau
Server\data\tabsvc\logs\httpd\error.log



Configure SSL for a Cluster

You can configure a Tableau Server cluster to use SSL. If the primary Tableau Server computer is the only node that is running the gateway process (which it does by default), then that's the only place where you need to configure SSL. See the procedure above for steps.

SSL and Multiple Gateways

A highly available Tableau Server cluster can include multiple gateways, fronted by a load balancer ([learn more](#)). If you are configuring this type of cluster for SSL, you have two choices:

- **Configure your load balancer for SSL.** Traffic is encrypted from the client web browsers to the load balancer. Traffic from the load balancer to the Tableau Server gateway processes is not encrypted. No SSL configuration in Tableau Server is required, it's all handled by your load balancer.
- **Configure Tableau Server for SSL:** Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. See the procedure below for details.

Configure a Server Cluster for SSL

When you configure a Tableau Server cluster to use SSL, you place the SSL certificate and key files on every computer that's running a gateway process. To configure a Tableau Server cluster to use SSL:

1. Configure the load balancer for SSL passthrough. Refer to your load balancer's documentation for assistance.
2. Make sure that the SSL certificate you use was issued for the load balancer's host name.
3. Configure the primary Tableau Server node as described in the procedure above.
4. Place the same SSL certificate and key file that you used for the primary on each Tableau Server worker node that is running a gateway process. Use the same folder location on the workers that you used on the primary.

If you are using mutual ssl, place the SSL CA certificate file you used for the primary on each worker node that is running a gateway process. Use the same folder location that you used on the primary.

You do not need to do any additional configuration on the workers.

For example, say you have a cluster that includes a primary Tableau Server node and three worker nodes with gateway processes are running on the primary, Worker 2 and Worker 3. In this situation, you [configure the primary Tableau Server for SSL](#), then copy the same SSL certificate and key files to Worker 2 and Worker 3. Because these files are in C:\Program Files\Tableau\Tableau Server\SSL folder on the primary, they are in that same location on Worker 2 and Worker 3.

You can configure a Tableau Server cluster to use SSL. If the primary Tableau Server computer is the only node that is running the gateway process (which it does by default), then that's the only place where you need to configure SSL. See the procedure above for steps.

Configure Internal SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository.

1. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**.
2. In the Tableau Server Configuration dialog box, click the **SSL** tab.
3. Select one of the following options:

- **Required for all connections**

When this option is selected, Tableau Server uses SSL for communications between the repository database and other server components. In addition, direct connections to Tableau Server (connections using the "tableau" or "readonly" users) must use SSL.

- **Optional for direct user connections**

This option configures Tableau Server to use SSL between the repository and other server components and supports but does not require SSL for direct connections by "tableau" or "readonly" users.

- **Off for all connections** (the default)

This option disables SSL for internal communications and direct connections.

4. Click **OK**.

For more information on downloading the public certificate for direct connections, see [Configure SSL for Direct Connections on page 495](#).

Configure SSL for Direct Connections

When Tableau Server is configured to use SSL internally, SSL connections are either optional or required for client machines making direct connections to the Tableau Server repository database. Direct connections include those using the "tableau" user or the "readonly" user.

To use SSL with direct connections, generate the SSL certificate file and copy it to the computer from which you will be making the direct connections.

1. Generate the SSL certificate file using the [regenerate_internal_tokens](#) on page 607 command.
2. Locate the SSL cert file by looking in the workgroup.yml file on the primary Tableau Server node.

The workgroup.yml file is located on the primary Tableau Server node in the `\ProgramData\Tableau\Tableau Server\data\tabsvc\config` folder.

The location of the SSL certificate and key files are listed in the file. For example:

```
pgsql.ssl.cert.file: C:/ProgramData/Tableau/Tableau Server-/data/tabsvc/config/pgsql/server.crt
```

```
pgsql.ssl.key.file: C:/ProgramData/Tableau/Tableau Server-/data/tabsvc/config/pgsql/server.key
```

3. Copy the cert file to the computer that will be making the direct connection and import them into the computer's certificate store using the documentation from the operating system manufacturer.

Note: Do not copy the key file. This file should only be on the server.

Configure SAML

You can configure Tableau Server to use an external identity provider (IdP) to authenticate Tableau Server users over SAML. All user authentication is done outside of Tableau, regardless of whether you're using Active Directory or local authentication in Tableau Server to manage your user accounts on Tableau Server. This allows you to provide a single sign-on experience across all the applications in your organization.

Before you configure Tableau Server for SAML, make sure you meet the [SAML Requirements](#) on page 475.

Configure SAML

To configure Tableau Server to use SAML:

1. Place the certificate files in a folder named SAML, parallel to the Tableau Server 9.2 folder. For example:

```
C:\Program Files\Tableau\Tableau Server\SAML
```

You should use this this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. If you are configuring SAML during Tableau Server setup, go to the SAML tab in the configuration utility.

If you are configuring SAML after you installing Tableau Server, open the Tableau Server Configuration Utility (**Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**) and then click the **SAML** tab.

3. On the SAML tab, select **Use SAML for single sign-on** and provide the location for each of the following:

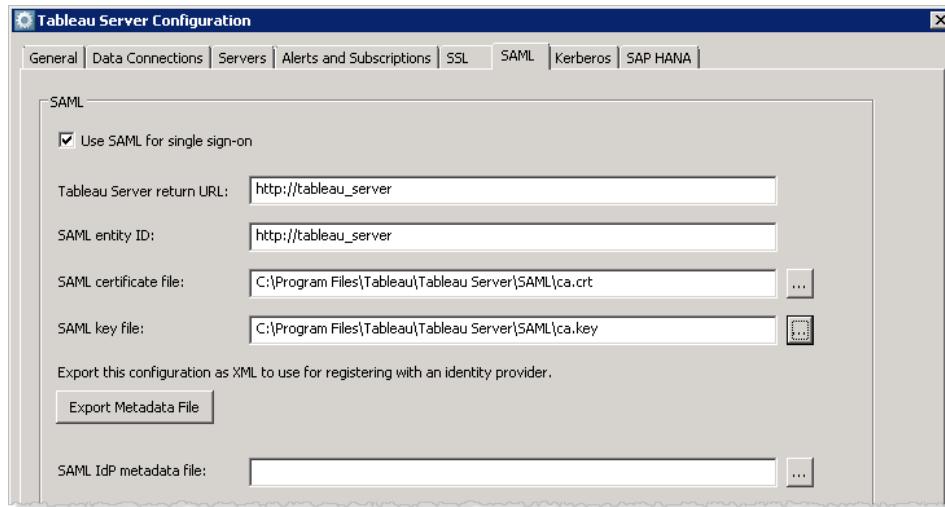
Tableau Server return URL—The URL that Tableau Server users will be accessing, such as `http://tableau_server`. Using `http://localhost` is not recommended. Using a URL with a trailing slash (for example, `http://tableau_server/`) is not supported.

SAML entity ID—The entity ID uniquely identifies your Tableau Server installation to the IdP. You can enter your Tableau Server URL again here, if you like, but it does not have to be your Tableau Server URL.

SAML certificate file—A PEM-encoded x509 certificate with the file extension **.crt**. This file is used by Tableau Server, not the IdP.

SAML certificate key file—An RSA or DSA private key file that is not password protected, and that has the file extension **.key**. This file is used by Tableau Server, not the IdP.

4. Leave the **SAML IdP metadata file** text box empty for now and click **Export Metadata File**.

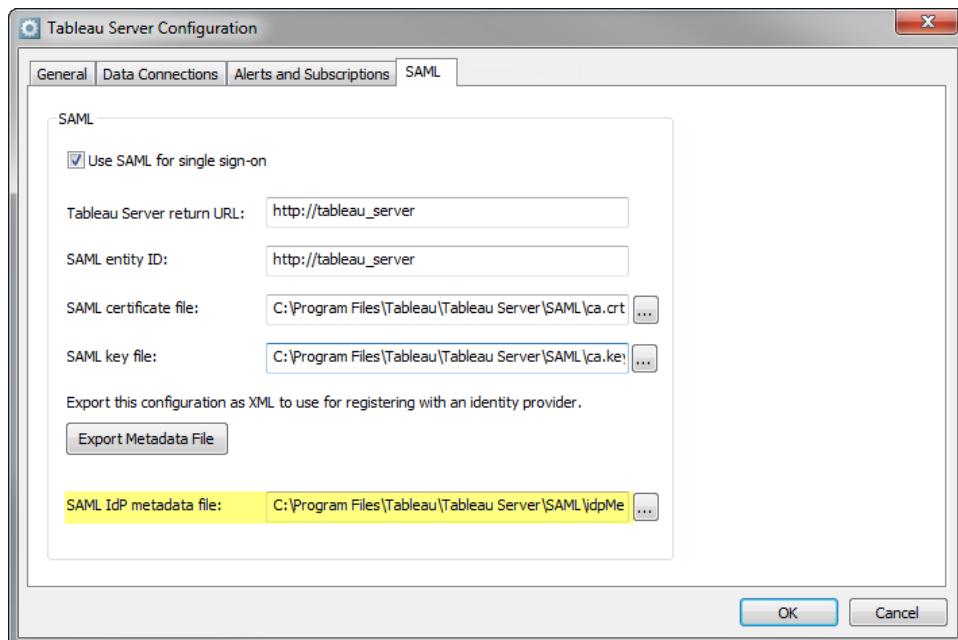


5. A dialog box opens that allows you to save Tableau Server's SAML settings as an XML file. At this point, metadata from your IdP is not included.

Save the XML file with the name of your choice.

6. On your IdP's website or in its application:

- Add Tableau Server as a Service Provider. Refer to your IdP's documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the file you saved in step 5.
 - Confirm that your IdP uses **username** as the attribute element to verify.
7. Still within your IdP, export your IdP's metadata XML file.
- It's a good idea to verify that the metadata XML you get from the IdP includes a **SingleSignOnService** element in which the binding is set to HTTP-POST, as in the following example:
- ```
<md:SingleSignOnService Bind-
ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://SERVER-NAME:9031/idp/SSO.saml2"/>
```
8. Copy your IdP's metadata XML file to the following folder on the computer where Tableau Server is installed:
- ```
C:\Program Files\Tableau\Tableau Server\SAML
```
9. On the SAML tab in the Tableau Server Configuration dialog box, enter the location to the file in the **SAML IdP metadata file** text box:



10. Click OK. Tableau Server is now configured for SAML authentication.

Configure a Server Cluster for SAML

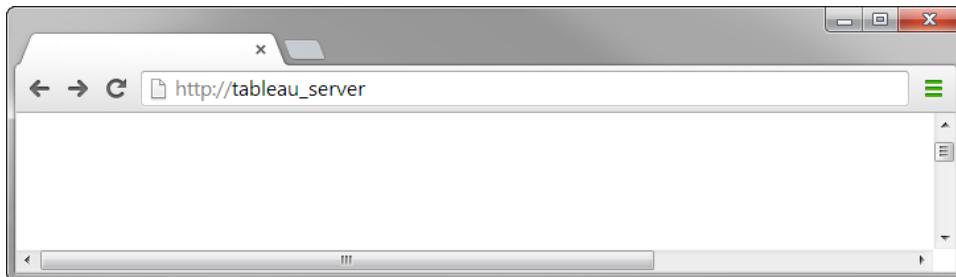
When you configure a Tableau Server cluster to use SAML, you place the same SAML certificate, SAML key, and SAML IdP metadata files on every computer that's running a Tableau application server process (also known as `vizportal.exe`). To configure a Tableau Server cluster to use SAML:

1. Configure the primary Tableau Server as described in the procedure above.
2. Place the same SAML certificate, SAML key, and SAML IdP metadata files that you used for the primary on each Tableau Worker that is running an application server process. Use the same folder location on the workers that you used on the primary. You do not need to do any additional configuration on the workers.

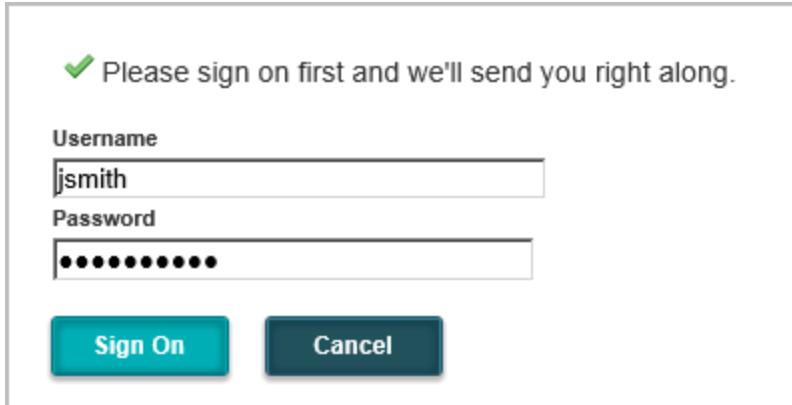
For example, consider a cluster that includes a primary Tableau Server and two workers. Application server processes are running on the primary and on Worker 2 and Worker 3. In this situation, you [configure the primary Tableau Server for SAML](#), and then copy the same SAML certificate, SAML key, and SAML IdP metadata files to the Worker 2 and Worker 3 computers. On the worker computers, put the SAML files in the the `C:\Program Files\Tableau\Tableau Server\SAML` folder, just as they are on the primary computer.

Test Your Configuration

Test your SAML configuration by opening a new web browser instance and typing the Tableau Server name in the URL window:



You should note that the sign in prompt that appears is from your IdP and not Tableau Server:



Configure Kerberos

You can configure Tableau Server to use Kerberos. This allows you to provide a single sign-on experience across all the applications in your organization. Before you configure Tableau Server for Kerberos make sure you meet the [Kerberos Requirements on page 506](#).

1. Open a command prompt as an administrator and change directories to the location of Tableau Server's bin directory. The default location is C:\Program Files\Tableau\Tableau Server\9.0\bin.
2. Type the following command to stop Tableau Server:
`tabadmin stop`
3. Open the Tableau Server Configuration Utility (**Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**), and then click the **Kerberos** tab.
4. Select **Enable Kerberos for single sign-on**.
5. Click **Export Kerberos Configuration Script**. The generated script configures your Active Directory domain to use Kerberos with Tableau Server. For more information, see [Kerberos Configuration Script on page 510](#).



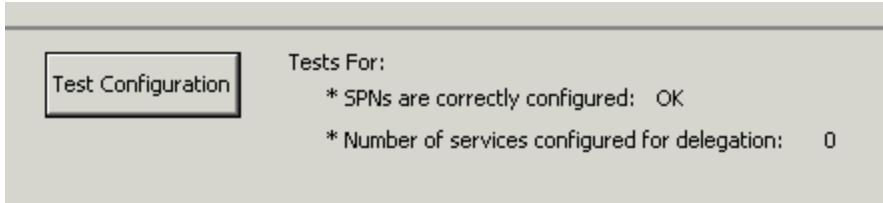
Note: Verify the host names in the setspn lines of the script. If you are using an external load balancer or a reverse proxy, the host names should match the name you used when you configured Tableau Server for the load balancer or proxy. If you have not configured Tableau Server for your proxy or external load balancer, do that and then re-export the Kerberos configuration script to ensure it has the correct host names. See [Add a Load Balancer on page 106](#) and [Configure Tableau to Work with a Proxy Server on page 449](#).

6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the .keytab file. The domain administrator should do the following:
 - Review the script to verify it contains correct values.
 - Run the script at a command prompt on any computer in the domain by typing the script name (not by double-clicking the script in Windows Explorer).

The script creates a file, `kerberos.keytab`, in a `\keytabs` folder in the location that the script was run.
7. Save a copy of the .keytab file created by the script to the Tableau Server computer. In Step 3, enter the path to the .keytab file, or click the browse button to navigate to the file. The keytab file will be copied to all the gateway nodes in your Tableau Server installation when you click **OK** in the Configuration utility.

Note: Do not rename the .keytab file. The script creates a file named `kerberos.keytab` and you need to save it with this name.

8. (optional) Click **Test Configuration** to confirm that your environment is configured correctly to use Kerberos with Tableau Server.

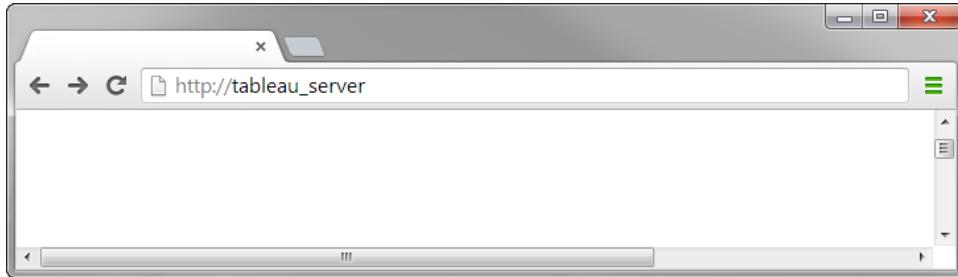


If you have not configured any data sources for Kerberos delegation, 0 is shown for the **Number of services configured for delegation**.

9. Click **OK** to save your Kerberos configuration.
10. Start Tableau Server.

Confirm Your SSO Configuration

Once Tableau Server has restarted, test your Kerberos configuration from a web browser on a different computer by typing the Tableau Server name in the URL window:



You should be automatically authenticated to Tableau Server.

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide Single Sign-on (SSO) for SAP HANA. HANA SSO is not dependent on SAML authentication to Tableau Server.

Note: You do not need to use SAML sign on with Tableau Server in order to use HANA SSO.

You can sign in to Tableau Server using whatever method you choose.

With SSO for SAP HANA, Tableau Server functions as an Identity Provider (IdP) and this configuration allows you to provide a single sign-on experience for users making SAP HANA connections. As part of the configuration, you need to acquire a SAML certificate and key file for Tableau Server (these should be a public key certificate and private key). You need to also install the signed certificate in HANA. You can generate the certificate and key yourself, or get them from a Certificate Authority. For more information on generating a certificate/private key and configuring SAP HANA, see the [Tableau Knowledgebase](#).

Note: The SAP HANA driver version 1.00.9 or later must be installed on Tableau Server in order to use SSO for SAP HANA. The driver cannot encrypt the SAML assertion, so you may want to enable encryption for the SAML connections. For more information, see the [Tableau Knowledgebase](#).

Configure SSO for SAP HANA

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 9.2 folder.
For example:

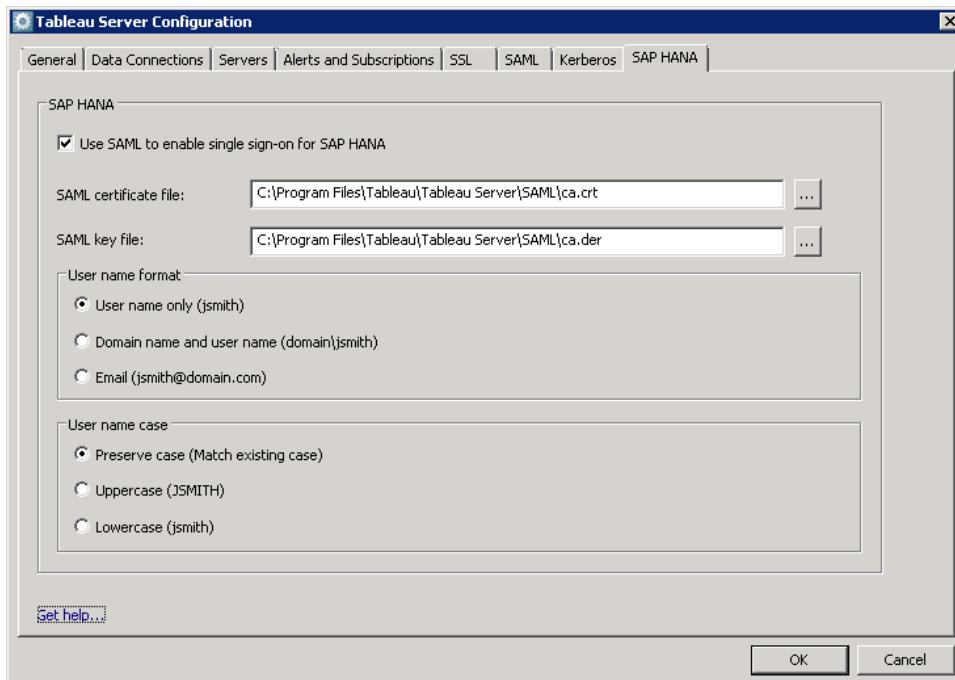
C:\Program Files\Tableau\Tableau Server\SAML

You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. After you install Tableau Server, run the Configuration utility (**Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**), and then click the **SAP HANA** tab.
3. Select **Use SAML to enable single sign-on for SAP HANA** and provide the location for each of the following:

SAML certificate file—A PEM-encoded x509 certificate with the file extension **.crt** or **.cert**. This file is used by Tableau Server, and must also be installed on HANA.

SAML private key file—A DER-encoded private key file that is not password protected, and that has the file extension **.der**. This file is only used by Tableau Server.



4. Select the format of the user name.
5. Select the case for the user name. This determines the case of the name when it is forwarded to the SAP HANA identity provider (IdP).

Configure Tableau Server for OpenID Connect

This topic describes how to configure Tableau Server to use OpenID Connect for single-sign on (SSO). This is one step in a multi-step process. The following topics provide information about

configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect on page 517](#)
- [Configure the Identity Provider \(IdP\) for OpenID Connect on page 518](#)
- Configure Tableau Server for OpenID Connect (you are here)
- [Signing In to Tableau Server Using OpenID Connect on page 522](#)
- [Changing IdPs in Tableau Server for OpenID Connect on page 524](#)

Note: Before you perform the steps described here, you must configure the OpenID identity provider (IdP) as described in [Configure the Identity Provider \(IdP\) for OpenID Connect on page 518](#).

Important notes

Before you configure Tableau Server for OpenID Connect, make sure you read these notes.

- You can use OpenID Connect with Tableau Server only if the server is configured to use local authentication. OpenID Connect is not available if the server is configured to use Active Directory authentication. For more information, see [Configure General Server Options on page 12](#).
- We recommend that you configure Tableau Server to use SSL for external communications. This helps to maintain secure communications between Tableau Server and the IdP during the exchange of authentication information. For details, see [Configure External SSL on page 491](#).

If you are configuring OpenID Connect during the initial configuration of Tableau Server (the first time the configuration utility runs), there is no option to set up SSL. In that case, we recommend that you finish the installation, then return to the configuration to set up SSL and then configure OpenID.

Note If you want to use external SSL for Tableau Server, it's generally more convenient to do that before you configure OpenID Connect. If you configure SSL after you've already configured OpenID, you need to return to the IdP and update the configuration that you made previously. For example, you need to change the protocol for the Tableau Server external URL from `http://` to `https://`.

Configure the server

To configure Tableau Server for OpenID Connect, follow these steps.

1. Log in as an administrator to the computer where Tableau Server is running.
2. If the server is running, stop it (Windows Start > All Applications > Tableau Server >

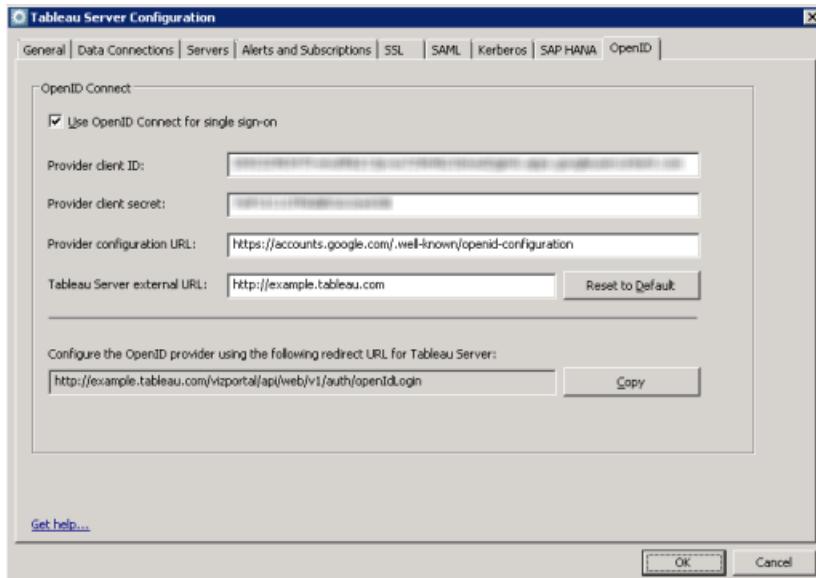
Stop Tableau Server).

Tip: You can also stop the server by using the `tabadmin stop` command.

3. Run the Tableau Server Configuration tool (Windows Start > All Applications > Tableau Server > Configure Tableau Server).
4. Click the **OpenID** tab.
5. Select the **Use OpenID Connect for single sign-on** option.
6. Fill in the **Provider client ID** and **Provider client secret** boxes with the values you recorded earlier.
7. In the **Provider configuration URL** box, enter the URL that the IdP uses for OpenID Connect discovery.
8. In the **Tableau Server external URL** box, enter the URL of your server. This is typically the public name of your server, such as `http://example.tableau.com`.

When you initially configure OpenID, the **Provider configuration URL** box contains a default value that's constructed based on the name of the server (`gateway.public.host`) and the gateway port, if any (`gateway.public.port`). In addition, by default the protocol is set to `https://` if SSL is enabled for the server.

Note: Make sure that you update the external URL if the default value is not the URL for how your server can be reached from an external source.



9. Copy the URL in the box labeled **Configure the OpenID provider using the**

following redirect URL for Tableau Server. You'll use this value in the next procedure to finish configuring the IdP.

10. Start the server (Windows Start > All Applications > Tableau Server > Start Tableau Server).

Tip: You can also start the server by using the tabadmin start command.

Add the redirect URL to the IdP configuration

After you configure Tableau Server, you finish the IdP configuration using the server's redirect URL.

1. Return to the IdP portal where you set up the project or application.
2. Edit the project configuration and find the redirect URL.
3. Enter the redirect URL that you copied in the previous procedure.

Add an Administrator Account

The final step in activating Tableau Server is to add an administrator account. The administrator will have all access to the server including the ability to manage users, groups, and projects. Adding an administrator account differs depending on whether you are using Active Directory or local authentication.

Active Directory

If you are using Active Directory, type the **Username** and **Password** for an existing Active Directory user who will be the administrator. Then click **Add user**.

The screenshot shows a web-based setup interface for Tableau Server. At the top, it says "Tableau Server Setup Tasks". To the right is the Tableau logo and the word " tableau". Below that, there's a section titled "Add Administrator Account". It has two input fields: "Username:" and "Password:", each with a corresponding text input box. At the bottom is a large orange button labeled "Add User".

Note:

If the administrator account is in the same domain as the server simply type the username without the domain. Otherwise you should include the fully qualified domain name. For example, test.lan\username.

Local Authentication

If you are using Local Authentication, create an administrative account by typing a **Username**, **Display Name**, and a **Password** (twice) of your choosing. Then click **Add user**.

The screenshot shows the 'Tableau Server Setup Tasks' interface. At the top, there is a logo consisting of a grid of colored dots followed by the word 'tableau'. Below the logo, the title 'Tableau Server Setup Tasks' is displayed. A red banner at the top says 'Add Administrator Account'. The main form contains four input fields: 'Username' (set to 'Administrator'), 'Display Name' (set to 'Administrator'), 'Password' (represented by four asterisks), and 'Confirm password' (also represented by four asterisks). At the bottom right of the form is a large orange button labeled 'Add user'.

Reconfigure the Server

When you install Tableau Server for the first time, you do initial configuration of the server as part of the installation. You can run the Tableau Server Configuration utility after installing Tableau Server to make additional configuration changes. Some configuration options are only available when you run the configuration utility after installation. You can also use the [tabadmin on page 583](#) command line tool to make configuration changes. Configuration setting changes are written to the `tabsvc.yml` file located in the `<install drive>:\ProgramData\Tableau\Tableau Server\config` directory.

Note: You cannot switch between Active Directory and Local Authentication. These options can only be configured during the initial installation of Tableau Server.

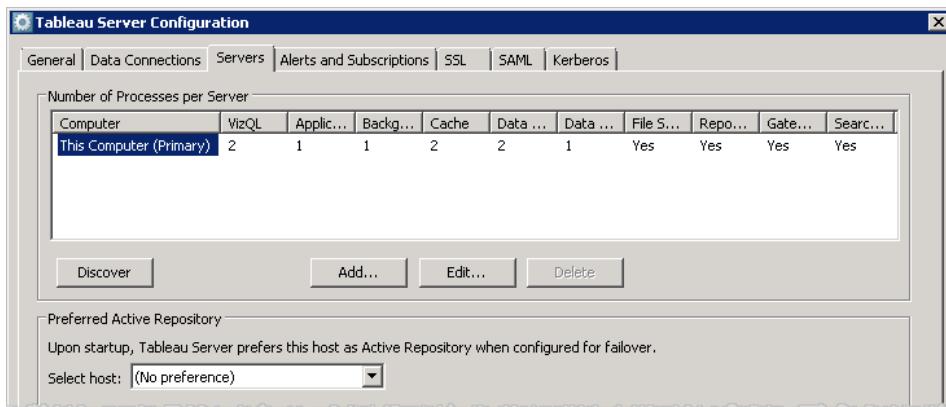
To change a Tableau Server configuration setting:

1. Stop the server by selecting **All Programs > Tableau Server 9.2 > Stop Tableau Server** on the Windows Start menu.
2. Select **Configure Tableau Server** on the Windows Start menu.
3. If you are using an Active Directory account for the server's Run As User account, enter its password on the **General** tab.
4. Make your configuration change.
5. Click **OK**.
6. Start the server by selecting **All Programs > Tableau Server 9.2 > Start Tableau Server** on the Windows Start menu.

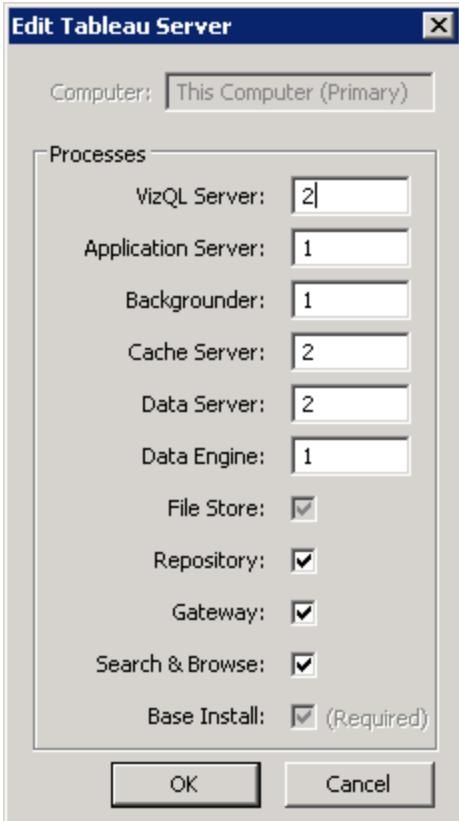
Reconfigure Processes

To change how processes are configured for a single server installation, follow the steps below. If you are changing how processes are configured for a worker, refer to [Install and Configure Worker Nodes](#) on page 76.

1. You will need to stop Tableau Server to make this configuration change. From the Start menu, click **All Programs > Tableau Server 9.2 > Stop Tableau Server**.
2. Open the Tableau Server Configuration dialog box from the Start menu by navigating to **All Programs > Tableau Server 9.2 > Configure Tableau Server**.
3. Enter your **Password**, if necessary, on the **General** tab then click the **Servers** tab:



4. Highlight **This Computer** and click **Edit**:
5. The Edit Tableau Server dialog box is where you change the number of processes:



You can run up to eight instances of the VizQL, application server, data server, or background processes—although this limit can be changed if necessary. See [Server Process Defaults and Limits on page 67](#) for more information. You need to have at least one instance of backgrounder installed. Also, for Tableau Server to function, there must always be one active instance of the data engine (and associated file store) and the repository. For steps on how to move them to another machine, see [Move the Data Engine and File Store Processes on page 66](#). For steps on how to configure additional instances of them, refer to [High Availability on page 82](#).

After you make your changes, click **OK**.

6. If you want to designate a specific computer as the preferred active repository, select the computer from the **Select host** list. If you add workers, you need to save the configuration and restart the Configuration utility for the workers to display in the list. For more information about the repository, see [Tableau Server Repository on page 48](#).
7. Click **OK** to close the Configuration utility.
8. Start Tableau Server again. From the Start menu, click **All Programs > Tableau Server 9.2 > Start Tableau Server**.

Tableau Server Processes

There are Tableau Server processes whose default configuration you can change to achieve different results. The topics [Improve Server Performance](#) on page 430 and [High Availability](#) on page 82 describe some of the approaches you can take. High-level status for each process is displayed on the server's Status page and more detailed information related to some of the processes—such as the background process—is in the [Administrative Views](#) on page 289 topic.

Note: Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node. File store is installed when you install data engine and cannot be installed separately. Every instance of a data engine process will always have one instance of the file store process present as well.

Architecturally, the 64-bit version of Tableau Server uses native, 64-bit processes; the 32-bit version of Tableau Server uses 32-bit processes. The exception is the data engine. If the 32-bit version of Tableau Server is installed on a 64-bit operating system, the 64-bit version of the data engine process is used.

For information on log files generated by these processes, see [Server Log File Locations](#) on page 645.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
API Server	wgserver.exe	Handles REST API calls	Yes	Unless you are using REST APIs for critical business processes, this service can be down without impacting the overall health of Tableau Server.
Application Server	vizportal.exe	Handles the web application, supports browsing and searching	Yes	Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.
Background	backgrounder.exe	Executes server	No	A single-threaded process where multiple processes can be run on any or

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
		tasks, including extract refreshes, 'Run Now' tasks, and tasks initiated from tabcmd		all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.
Cache Server	redis-server.exe	Query cache	No	A query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and API server and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.
Cluster Controller	clustercontroller.exe	Responsible for monitoring various components, detecting failures, and executing failover when needed	n/a	Included in the base install on every node.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
Coordination Service	zookeeper.exe	In distributed installations, responsible for ensuring there is a quorum for making decisions during failover	n/a	Included in the base install on every node.
Data Engine	tdeserver64.exe	Stores data extracts and answers queries	Yes	The data engine's workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory consumption is primarily based on the size of the data extracts being loaded. The 64-bit binary is used as the default on 64-bit operating systems, even if 32-bit Tableau Server is installed. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.
	tdeserver.exe (32-bit)			
Data	dataserver.exe	Manages	Yes	Because it's a proxy, it's normally only

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
Server		connections to Tableau Server data sources		bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.
File Store	filestore.exe	Automatically replicates extracts across data engine nodes	n/a	Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.
Repository	postgres.exe	Tableau Server database, stores workbook and user metadata	n/a	Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository on page 48.
Search & Browse	searchserver.exe	Handles fast search, filter, retrieval, and display of content metadata on the server	Yes	The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.
VizQL Server	vizqlserver.exe	Loads and renders views,	Yes	Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O

Process	File Name	Purpose	Multi-threaded?	Performance Characteristics
		computes and executes queries		bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.

Tableau Server Coordination Service

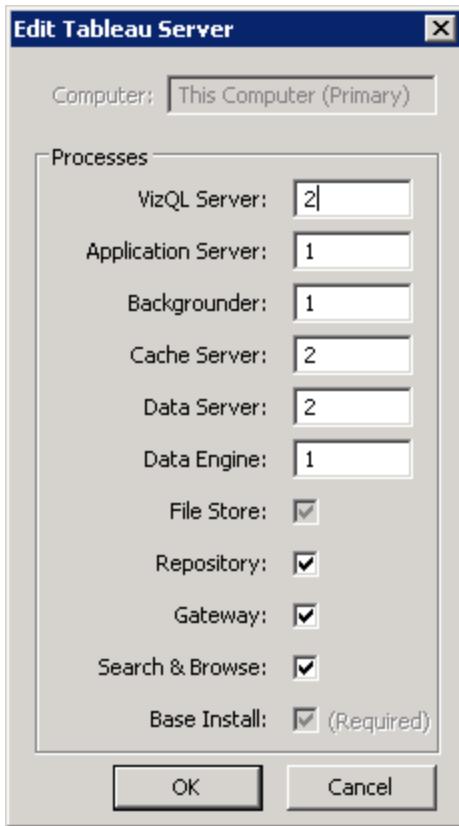
If Tableau Server is configured as a distributed environment, one of the [processes](#) that runs on each node in the cluster is the Coordination Service. Tableau Server uses the Coordination Service to coordinate activities on the server, including for high availability installations. The Coordination Service is built on [Apache ZooKeeper](#), an open-source project.

The hardware for your cluster can have some effect on how well the Coordination Service runs. In particular:

- Memory. The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.
- Disk speed. Because the service stores state information on disk, it benefits from fast disk speed on the individual node computers.
- Connection speed between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Configuration for the Coordination Service

The Coordination Service runs automatically on each node of Tableau Server. You do not have to explicitly configure the coordination service, and there are no settings you can make for the service. As a consequence, when you [add a node](#) to your cluster, you do not see the configuration service listed as process—for example, you do not see the coordination service listed in the **Add Tableau Server** dialog box:



The **Base Install** option includes the Coordination Service and Cluster Controller. As you can see, this option is disabled, because you cannot choose not to install those services.

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a *quorum*—a minimum number of instances of the service. In a Tableau Service installation, the Coordination Services must be running on more than 50% of the nodes in the cluster. This has an effect if you reduce the number of nodes in your installation. It also has an important effect if the number of computers in the cluster is reduced.

If you reduce the number of nodes

If you reduce the nodes in your cluster from three (or more) to two nodes, a warning tells you Tableau Server can no longer support high availability:

A minimum of three Tableau Server nodes are required for high availability. You can add a third node now, or continue with only two nodes. Continuing with only two nodes means Tableau Server will not be highly available. You can always add a third node later. Click OK to continue with 2 nodes, or Cancel to go back and add a node.

If you continue, Tableau Server will run, but you will not have any automatic failover of the repository.

If the server experiences a problem

If the Coordination Service detects that fewer than half of the instances of the service are running, it shuts the server down, since this means that the nodes are unable to synchronize and the server installation cannot manage failover. If this occurs, you can determine that the Coordination Service was responsible for shutting down the server down in the following ways:

- **Alerts.** If you have configured the server to send you alerts, the coordination services causes an email to be sent that indicates that the number of instances of Coordination Service fell below the quorum. For more information about configuring alerts, see [Configure Alerts and Subscriptions on page 16](#).
- **Log files.** You can examine log files, specifically the log files for the Coordination Service (zookeeper) and the cluster Controller service. For more information, see [Server Log File Locations on page 645](#).

Viewing Coordination Service Status

The Coordination Service is not included in the listing when you [view server process status](#). To see the state of the service, you can use the following `tabadmin` command:

```
tabadmin status --verbose
```

The output from the command shows you whether the service is running:

```
10.32.139.21:  
  Status: RUNNING  
  'Tableau Server Data Engine 0' (2456) is running.  
  'Tableau Server Vizqlserver 0' (3336) is running.  
  'Tableau Server Backgrounder 0' (11976) is running.  
  'Tableau Server CacheServer 0' (2508) is running.  
  'Tableau Server Dataserver 0' (3572) is running.  
  'Tableau Server Application Server 0' (804) is running.  
  'Tableau Server API Server 0' (3584) is running.  
  'Tableau Server Coordination Service 0' (2624) is running.  
  'Tableau Server Search and Browse 0' (2744) is running.  
  'Tableau Server Gateway' (2824) is running.  
  'Tableau Server Cluster Controller' (2840) is running.  
  'Tableau Server Repository' (2032) is running (Active Repository).  
  'Tableau Server File Store' (2964) is running.
```

Performing Cleanup for the Coordination Service

The Coordination Service maintains state information about the server, such as transaction logs of activities on the server. This information is written to disk, and when the server is restarted, the information on disk is used to restart the Coordination Service and to determine state information such as whether multiple repositories have been synchronized.

If the data maintained by the service is corrupted (for example, due to hardware problems) or if there is some other problem with the Coordination Service that affects server startup, you can perform a cleanup operation on the service's information. To do so, run the following `tabadmin` command:

```
tabadmin cleanup --reset-coordination
```

This command will perform a normal [cleanup](#) on page 590 as well as removing Coordination Service files.

Note: This command can only be run when the server is stopped.

Tableau Server File Store

The Tableau Server File Store process is installed along with the Data Engine and controls the storage of extracts. In highly available (HA) environments, the File Store ensures that extracts are synchronized to other file store nodes so they are available if one file store node stops running.

Process	File Store
File name	<code>filestore.exe</code>
Status	Status of the File Store process is visible on the Status Page. For more information, see View Server Process Status on page 240
Logging	Logs are located in <code>\logs\filestore</code> . For more information, see Server Log File Locations on page 645

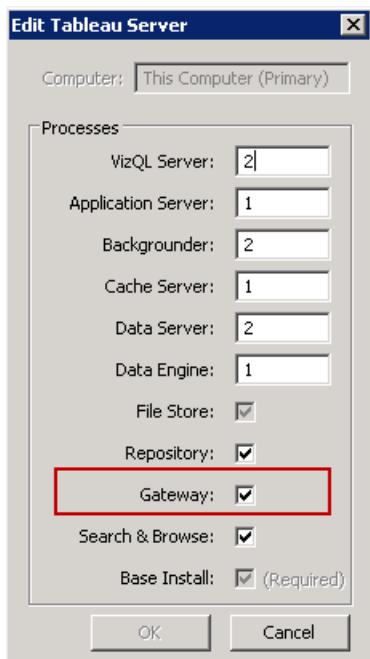
The decommission Command

If you want or need to remove a file store you should decommission the file store first, using the `decommission` command. Decommissioning puts the file store into read-only mode and copies any unique data contained in the file store to the other file store(s) in the cluster. While a file store is being decommissioned, this shows on the Status page, and once all unique content has been copied to other file store nodes, the decommissioned node shows as ready to be removed.

Tableau Server Gateway Process

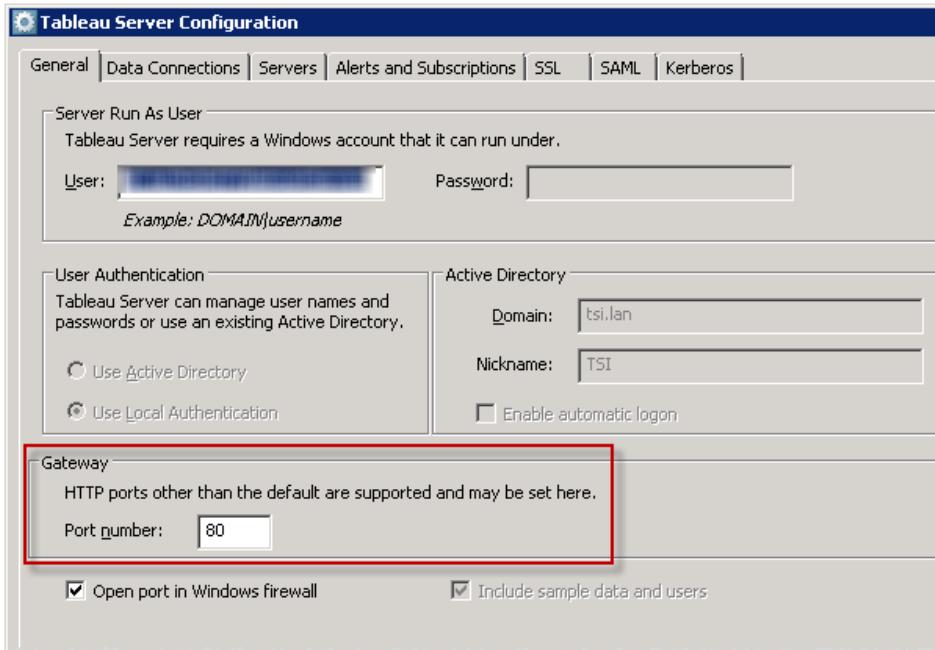
The Tableau Server gateway process is an Apache web server component (`httpd.exe`). Its role is to handle requests to the server from all clients—Tableau Desktop, mobile devices, a proxy, a load balancer, etc.

The server runs a single instance of the gateway process; you can't run more than one per machine.



Post assignment

By default, the gateway process listens for requests on port 80 (for HTTP requests) and 443 (for SSL requests). When you install Tableau Server on a computer, part of the server configuration makes sure that this port is open in the computer's firewall. If the computer is running a different process that requires port 80, you can change the port assignment for the gateway process. You can do this in the Tableau Server Configuration tool:



Alternatively, you can run the following `tabadmin` command, where *nn* is the new port number:

```
tabadmin gateway.public.port nn
```

Log files for the gateway process

The gateway process creates two sets of log files in the `\logs\httpd` folder of the log file archive:

- Activity logs. The name for these log files has the format `access.yy_mm_dd_hh_mm_ss.log`.
- Error logs. All errors are logged in a single file named `error.log`.

For more information, see [Archive Log Files](#) on page 641.

Gateway processes in a cluster

If your server environment is distributed across multiple machines, you can run a single gateway process on each node of the cluster. The most common scenario for running a gateway process on multiple computers in the cluster is that you have a load balancer in front of the cluster. In this scenario, the load balancer distributes requests to any gateway in the cluster. If you need to take a node off line (for example, to perform maintenance on that node), you can disable the load balancer's routing to that machine. When the maintenance is complete, you can re-enable the node on the load balancer.

You must have a gateway process running on at least one computer in the cluster. If you remove the gateway process from the primary server, you must make sure that another

computer in the cluster is running the gateway process. You must also make sure that that computer is reachable by clients.

If the Tableau Server is configured to use SSL, you must make sure that the certificate for SSL support is in the same location on each computer in the cluster that has the gateway process running. For more information about using SSL, see [Configure External SSL on page 491](#).

Similarly, if the server installation uses a custom logo, the logo must be in the same location on every computer that is running the gateway process.

If you need to change the port number that the gateway process listens on, as explained earlier, you can use the configuration dialog box or run the following command for each worker computer that is running the gateway process:

```
tabadmin workerN.gateway.port nn
```

Additional information

[Proxy Servers on page 448](#)

[Add a Load Balancer on page 106](#)

[Configure for Failover and Multiple Gateways on page 94](#)

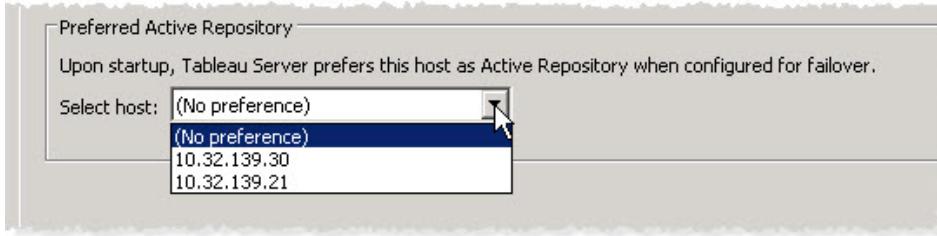
Tableau Server Repository

Tableau Server Repository is a database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.

Process	Repository
File name	postgres.exe
Status	Status of the Repository is visible on the Status Page. For more information, see View Server Process Status on page 240
Logging	Logs generated by the repository are located in \logs\repository. For more information, see Server Log File Locations on page 645

[Preferred active repository](#)

When you configure Tableau Server after the initial installation, you have the option to specify a **Preferred Active Repository**. This is an optional step, and if you do not specify a preferred active repository, Tableau Server will select the active repository on startup.



Configure a preferred active repository if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using [custom administrative views](#). Custom administrative views have embedded connection information that refers to the repository for which you created the views.

The failoverrepository Command

If failover occurs and your passive repository becomes the active repository, it remains the active repository until either Tableau Server restarts or you use the `failoverrepository` command to switch back. Specify the repository you want to be the active one, or specify that the preferred active repository (if configured) should be made active again. For more information, see [failoverrepository on page 599](#).

Server Process Defaults and Limits

The number of installed process instances depends on the computer where you are installing Tableau Server.

If the computer meets or exceeds the minimum hardware recommendations, the default installed number of most processes is two instances. If the computer meets the minimum hardware requirements but does not have at least 8 cores and 16 GB of system memory, the defaults are reduced to one instance of each process. This is intentional and is intended to match the software to the available hardware. For more information on server hardware requirements and recommendations, see [Minimum Hardware Requirements and Recommendations for Tableau Server on page 59](#).

Note: You should not change the number of processes without increasing associated RAM. If you are unsure, contact Tableau Support for guidance.

In a distributed installation, you can have a maximum of two repository instances (active and passive). You can also run Tableau Server with one repository, but doing this means there is no failover available for the repository.

If you are running the 64-bit version of Tableau Server (which is available starting with version 8.1), two instances of a process is the recommended maximum for most

configurations. For more information on when you might want more than two instances of a process, see [Improve Server Performance](#) on page 430.

If you are running the 32-bit version of Tableau Server and the default settings aren't sufficient, you can change them to up to eight instances either during Setup (for upgrades only) or after Setup, using the [Tableau Server Configuration utility](#).

Changing the default upper limit

Eight instances of a process is the default upper limit. If your machine has enough RAM and CPU cores, and you want to go above this limit, you can change the limit using the `service.max_procs` tabadmin setting. For each process instance, Tableau recommends that the machine running the process have at least 1 GB of RAM and 1 logical CPU core.

To change the maximum number of processes allowed:

1. After Setup, [stop the server](#).
2. In the Tableau Server bin directory, type the following command, where `number` is the maximum number of process instances you want to allow:

```
tabadmin set service.max_procs <number>
```

For example:

```
tabadmin set service.max_procs 10
```

3. Still in the bin directory, type:

```
tabadmin config
```

4. [Start the server](#) so the changes can take effect.

Upgrade to 9.2

Use the following topics to upgrade your Tableau Server software to version 9.2. If you are upgrading from a version earlier than Version 8.2, please refer to the [Tableau Knowledge Base](#).

Pre-Upgrade Checklist

Before you upgrade Tableau Server to version 9.2.x you should read [What's Changed - Things to Know Before You Upgrade on page 1](#) and this topic and perform the steps described here.

Note: A new version of tabcmd is released with every release of Tableau Server. If you installed the command line utility on computers that are not running Tableau Server, you may need to upgrade tabcmd on those computers when you upgrade Tableau Server.

For more information see [Install tabcmd on page 552](#).

Credentials and Setup Files

Before you upgrade, make sure you have the following:

- **User account credentials:** For each computer you're upgrading, you need credentials for a user account with local admin permissions.
- **Run As account credentials:** Confirm that you have the user name and password for Tableau Server's [Run As](#) account. If you are using NT AUTHORITY\NetworkService (the default), no password is required.
- **Setup files:** In addition to having the .exe for the upgrade you're about to perform, you should locate or re-download the Setup .exe for the server version you currently have in production (see [Downloading Tableau Products](#)). If something unexpected happens during the upgrade, this can help you recover more quickly.

Customizations

While Tableau retains configuration settings during an upgrade, it's a best practice to also note any customizations you've made so that you can verify them later. These include [configuring SSL](#), changing Tableau's default [port](#) and [time out](#) values, as well as using [custom logos](#). Also, if you added your current Tableau Server version to your Windows PATH environment variable, you will need to update that entry after upgrading so that it refers to the newer version of Tableau Server.

Hardware Requirements (cores, RAM, and free disk space)

Before you upgrade, make sure the computers you are upgrading meet or exceed the minimum hardware requirements. See [Hardware Requirements](#) for more information.

Bit Version

Starting with version 8.1, Tableau Server is provided as a native 64-bit application as well as a 32-bit application. Earlier versions of Tableau Server were only available as 32-bit.

If you were previously running the 32-bit version of Tableau Server on a 64-bit operating system, upgrading to the 64-bit version of Tableau Server is recommended. See [Before you install... on page 3](#) for the minimum requirements.

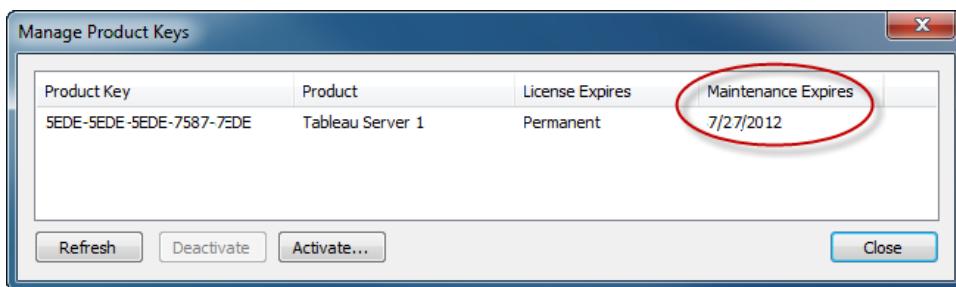
If you are upgrading a distributed installation of Tableau Server, the entire cluster must run the same bit version—either all 32-bit or all 64-bit Tableau server software. When upgrading from the 32-bit version of Tableau Server to the 64-bit version, you must first uninstall the 32-bit version on each worker before installing the 64-bit version of the worker software. For more information, see [Upgrading a distributed installation of Tableau Server from 32-bit to 64-bit on page 65](#).

Check Your Product Maintenance Status

If you attempt to upgrade Tableau Server from a server whose maintenance has expired, the result will be an unlicensed instance of Tableau Server.

To see whether your server's maintenance has expired:

- Select **Start > All Programs > Tableau Server > Manage Product Keys** and look under the **Maintenance Expires** column.



If your maintenance has expired, select the key and click **Refresh**. If the maintenance date doesn't update, contact [Tableau Customer Support](#). Reactivating the product key will be part of Setup. See [Activate Tableau on page 10](#) for details. If your server doesn't have internet access, refer to [Activate Tableau Offline on page 10](#).

Create a “Clean” Backup

As a best practice you should always create a backup just before upgrading Tableau Server, in addition to your regular backups. Before you create the backup, run the `tabadmin cleanup` command to remove non-essential files from your backup. See [Running Cleanup](#) and [Back Up the Tableau Data](#) on page 631 for steps.

Distributed Installations Only: Whether to Remove Workers Before Creating the Backup

The Tableau backup file (`.tsbak`) includes configuration information as well as data. Therefore, a backup of a distributed installation of Tableau Server will include configuration information about the worker nodes, including their IP addresses. If you don’t want this information as part of your backup (for example, because you are migrating worker nodes to new hardware as part of your upgrade), you can do one of two things:

- Remove the workers from the Tableau Server configuration before creating the backup.
- Plan on using the `--no-config` option when you restore the backup file to your new installation. Note that with this option, no configuration information is restored—including for the primary Tableau Server node.

Note: You should uninstall Tableau Server from any workers that you are not including in your new installation to avoid conflicts between the older workers and the new installation.

If you are running a distributed installation of Tableau Server and have a worker running Windows XP or Windows Server 2003 SP1 or SP2, you must remove it from the configuration before upgrading. These operating systems are not supported platforms in version 9.2. Note that Windows Server 2008 or higher *is* supported.

To delete a worker from your Tableau Server configuration:

1. [Stop the server](#) on the primary Tableau Server.
2. On the primary server, open the configuration utility by selecting **Tableau Server <version> > Configure Tableau Server** on the Start menu.



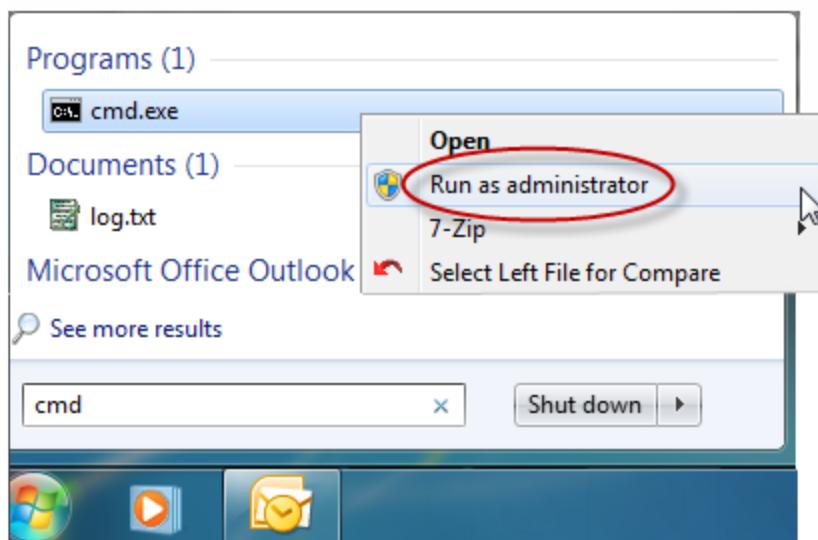
3. In the configuration utility, select the **Servers** tab.
4. If the worker is hosting the data engine or the repository, move those processes onto another machine before continuing. See [Move the Data Engine and File Store Processes on page 66](#) for steps.
5. Next, highlight the worker and click **Delete**.
6. Click **OK**.
7. Start the server.

Running Cleanup

Running the `tabadmin cleanup` command removes files from the Tableau Server system that you don't need in your backup file. You should run cleanup once with the server running, which allows it to act on the Tableau database, and once with the server stopped, which allows it to remove log files. For more information, see [Remove Unneeded Files on page 634](#).

To run `tabadmin cleanup`:

1. Open a command prompt as an administrator:



2. Navigate to your Tableau Server bin directory. For example:

```
cd "C:\Program Files\Tableau\Tableau Server\9.1\bin"
```

3. Confirm that the server is running:

```
tabadmin status
```

4. Run cleanup by typing the following:

```
tabadmin cleanup
```

5. Stop the server:

```
tabadmin stop
```

6. Run cleanup again:

```
tabadmin cleanup
```

Keep the server stopped for creating a backup (next).

Create the Backup File

The `tabadmin backup` command creates a `.tsbak` file containing data from your repository, data extracts, and server configuration. After you create the file, store it on a separate computer. See [Back Up the Tableau Data on page 631](#) for steps. Note that if you are creating a backup using Tableau Server version 8.0 or earlier, you must stop the server before creating a backup. Beginning with version 8.1, you can create a backup without stopping the server first.

Distributed installations only: If you removed worker nodes from your server configuration prior to creating your backup and you are not migrating to new hardware as part of your upgrade, you can now add the workers back to your configuration. Follow the steps in [Upgrade to 9.2 on page 62](#). Otherwise, if you are migrating to new hardware as part of your upgrade, leave the workers off the configuration. See [Migrate to New Hardware on page 68](#) for details.

What Changed in Version 9.2

Version 9.2 includes some changes you should know about before upgrading.

For information about what's new in Tableau Server 9.2, see the What's New in Tableau Server topic in the Tableau Server online help.

The updates to Tableau Server 9.2 have the following impact:

Assign Permissions to Contents setting

Because content permissions can be locked to the project, the **Assign Permissions to Contents** button has been removed and is no longer available for projects and workbooks. For more information, see [Quick Start: Lock Project Permissions](#) and [Lock Content Permissions to the Project on page 369](#).

What Changed in Version 9.1

Version 9.1 includes some changes you should know about before upgrading.

For information about what's new in Tableau Server 9.1, see the What's New in Tableau Server topic in the Tableau Server online help.

The updates to Tableau Server 9.1 have the following impact:

SAML authentication - logout

Starting with version 9.1, Tableau Server supports SAML logout. SAML logout is enabled by default and you can disable or enable it using the `tabadmin set wgserver.saml.logout.enabled false/true` command.

If your pre-9.1 Tableau Server is configured for SAML authentication, the logout functionality will not work until you reconfigure the metadata for SAML. You must re-export the SAML metadata file and re-import it into your IDP. For more information about configuring SAML metadata, see [Configure SAML](#) on page 479.

Hidden fields in published data sources - unavailable for workbooks

Starting with version 9.1, workbooks respect hidden fields in published data sources. Prior to 9.1, workbooks using hidden fields automatically exposed these fields.

If a workbook that was created prior to Tableau 9.1 used a published data source with hidden fields, the hidden fields were displayed in the workbook. Starting with Tableau 9.1, the behavior changes:

- If you are creating a new workbook that uses a published data source with hidden fields, those fields remain hidden in the workbook and cannot be used in calculations, sets, groups, and other object creation.
- If you are working with an existing workbook that uses a published data source with hidden fields, those hidden fields are displayed in red in the workbook to indicate that the fields, and therefore the views and calculations that use those fields, are invalid.

You can address this issue in one of two ways, depending on whether you want to show the fields or not:

- Show (unhide) the relevant fields in the data source, and then republish it, or
- Update the relevant workbooks to exclude the hidden fields.

For information on unhiding fields in the Data pane, see [Hide or Unhide Fields](#) in the Tableau Desktop help.

Clickjack protection - enabled by default

Starting with version 9.1, clickjack protection is enabled by default on Tableau Server. The protection has been available for several releases, but had been off by default. For more information on clickjack protection and how it impacts embedded views, see [Clickjack Protection](#) on page 422.

Note: When clickjack protection is enabled, embedded views that use the embed URL copied from the browser address bar might not load. These view URLs usually contain the hash symbol (#) after the server name (for example, `http://myserver/#/views/Sales/CommissionModel?:embed=y`) are blocked when clickjack protection is enabled on Tableau Server. You can fix these views by editing the embed URL. For more information, see [Embedded Views Don't Load If Clickjack Protection is Enabled](#) in the Tableau Knowledge Base.

What Changed in Version 9.0

Tableau Server 9.0 includes some changes you should know about before upgrading.

For information about what's new in Tableau Server 9.0, see the [What's New in Tableau Server](#) topic in the Tableau Server online help.

The updates to Tableau Server 9.0 have the following impact:

Customizations

Default start page

Any user-defined default start page will be reset to the Tableau Server default start page. Users will need to [reset their default start page](#) after the upgrade.

Custom logos

Starting with version 9.0, custom logos have changed in the following ways:

- The background for large custom logos is different based on logo location. On the navigation bar the background is black and on the sign-in screen the background is white. For more information, see [Change the Name or Logo on page 285](#).
- The small logo option has been deprecated. There are no locations in Tableau Server where the small logo is displayed, so the option does not do anything.

Hardware Requirements (cores, RAM, and free disk space)

Beginning with version 9.0, Tableau Server will not install if your computer does not meet the minimum requirements. This is true for upgrades and new installations, and for all computers in a distributed installation. The hardware requirements are:

- **64-bit Tableau Server**—At minimum you must have 4 cores, 8 GB of RAM, and 15 GB of free disk space to install the 64-bit version of Tableau Server.

- **32-bit Tableau Server**—At minimum you must have 2 cores, 4 GB of RAM, and 15 GB of free disk space to install the 32-bit version of Tableau Server.

For more information, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#) on the next page.

Note: If you are upgrading Tableau Server on a computer that does not meet the minimum hardware requirements, you will not be able to install Tableau Server 9.2. If you cannot upgrade 64-bit Tableau Server because of hardware requirements but your computer meets the minimum hardware requirements for 32-bit Tableau Server, you may be able to upgrade to 32-bit Tableau Server.

High availability and failover

As of version 9.0, Tableau Server no longer supports automatic failover with a two-node cluster. To get the benefit of automatic failover, you need to install Tableau Server on a minimum of three nodes. One of these can include a minimal install (the "base install" option).

The option to use an external confirmation host is no longer supported. Any installation that is configured with an external confirmation will be upgraded without that host.

When you upgrade a two-node installation that is configured for high availability (automatic failover), you are given the option to add a third node. You can do so as part of the upgrade process, or at a later time.

The Tableau Software user

Prior to Tableau Server 9.0, if you installed the sample data and users, a user named Tableau Software was created. The Tableau Software user was the owner of the sample data.

Starting with version 9.0, no Tableau Software user is created. If you install the sample data, ownership of that data is assigned to the initial user that is created (the administrator user).

Internal PostgreSQL database password regeneration

Installing Tableau Server or upgrading from a previous version regenerates the password that is used by internal Tableau Server processes for communicating with the PostgreSQL database. This password is only used by internal processes and is not accessible to server administrators or other users. For more information, see [Regenerate a Password](#).

tabadmin restore - Doesn't automatically restart Tableau Server

Starting with version 9.0, a `tabadmin restore` command will not automatically start Tableau Server. If you want the server to start after doing a restore, use the `--restart` option. For more information, see [restore on page 609](#).

"Remember me" option

With version 9.0 of Tableau Server, there is no **Remember me** option on the sign in page.

Session ID in URLs

With version 9.0 of Tableau Server, the session ID at the end of server URLs is now indicated by an "iid" parameter, :iid=<n>. For example,

`http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1`.

This parameter replaces the hash symbol "#<n>" used for the session ID in 8.x versions of Tableau Server.

Changes in view URLs may impact embedded views, API calls, and trusted tickets

In Tableau Server 9.0, view URLs have changed. We recommend that you generate URLs by clicking the **Share** link in a view in Tableau Server 9.0, and then use the resulting URL in embedded views, API calls, or trusted tickets that you created in Tableau Server prior to version 9.0.

Note: If you use view URLs that were created by copying the URL in a browser's address bar rather than using the URL generated by clicking the **Share** link, the views may not work as expected after you upgrade to version 9.0. This issue can be resolved by replacing the view URL with the **Share** link URL.

Minimum Hardware Requirements and Recommendations for Tableau Server

The following minimum hardware requirements and recommendations apply to all computers running Tableau Server, including physical hardware and virtual machines (VMs):

- **Minimum requirements** are the minimum hardware your computer must have in order for Setup to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping.
- **Minimum recommendations** are higher than minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the Setup program will warn you but you can continue the installation.

In addition, Tableau Server should not be installed on a physical computer or on a VM instance that is also running resource-intensive applications such as databases or application servers.

Note: If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see [Server Process Defaults and Limits](#) on page 67

Minimum Hardware Requirements

The computer on which you are installing or upgrading Tableau Server must meet the minimum hardware requirements. If the Setup program determines that your computer does not meet the following requirements, you will not be able to install Tableau Server. For more information on how the Setup program determines hardware, see "Determining Computer Hardware," below.

These minimum requirements are appropriate for prototyping and testing of Tableau Server and apply to single-node installations and to each computer in a distributed installation.

Server Version	CPU	RAM	Free Disk Space
64-bit Tableau Server	4-core	8 GB	15 GB
32-bit Tableau Server	2-core	4 GB	15 GB

For the requirements:

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space.
- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

If you cannot install the 64-bit Tableau Server because of hardware requirements but your computer meets the minimum hardware requirements for the 32-bit version of Tableau Server, you may be able to install the 32-bit version.

Note: For Tableau Server 9.2 on a 64-bit virtual machine, you need a minimum of 4 physical cores. If you are installing on an Amazon EC2 instance, this means 8 vCPUs. For more information, see [Amazon EC2 Instances](#).

Minimum Hardware Recommendations

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general.

Actual system needs for Tableau Server installations can vary based on many factors including number of users, and number and size of extracts.

Install Type	Processor	CPU	RAM	Free Disk Space
Single node	64-bit	8-core, 2.0 GHz or higher	32 GB	50 GB
Multi-node and Enterprise deployments	Contact Tableau for technical guidance. Nodes must meet or exceed the minimum hardware recommendations, except nodes running backgrounder, where 4 cores may be acceptable.			

Determining Computer Hardware

The Tableau Server Setup program determines how many physical cores a computer has by querying the operating system. To view hardware information the Setup program detected on your computer, open the `tabadmin.log` file located on the computer where you are installing Tableau Server:

```
<install directory>\ProgramData\Tableau\Tableau Server-  
\logs\tabadmin.log
```

In `tabadmin.log`, look for lines similar to the following to check the physical and logical cores that Setup detected and used to determine the core count that is being used for licensing:

```
2015-04-09 14:22:29.533 -0700_DEBUG_10.36.2.32:<machine name>:_  
pid=21488_0x2cd83560__user=__request=__ Running hardware check  
  
2015-04-09 14:22:29.713 -0700_DEBUG_10.36.2.32:<machine name>:_  
pid=21488_0x2cd83560__user=__request=__ Detected 12 cores and  
34281857024 bytes of memory  
  
2015-04-09 14:22:29.716 -0700_DEBUG_10.36.2.32:<machine name>:_  
pid=21488_0x2cd83560__user=__request=__ Hardware meets recom-  
mended specifications. Default values will be used.
```

Manually determining the number of cores on your computer

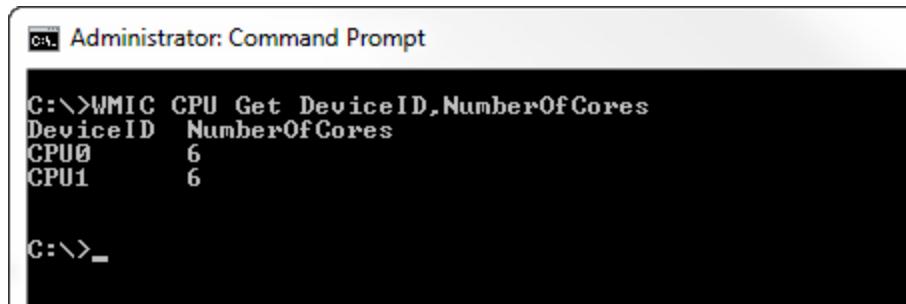
You can use the Windows Management Instrumentation Command-line tool (WMIC) to determine how many physical cores your server has. This is useful if you do not know whether

your computer will meet the minimum hardware requirements for installing Tableau Server.

1. Open a command prompt.
2. Enter the following command:

```
WMIC CPU Get DeviceID,NumberOfCores
```

The output will display the device id or ids and the number of physical cores the computer has:

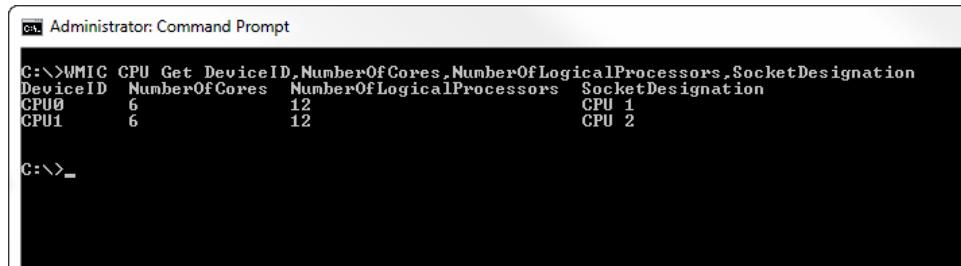


```
C:\ Administrator: Command Prompt  
C:\>WMIC CPU Get DeviceID,NumberOfCores  
DeviceID  NumberOfCores  
CPU0      6  
CPU1      6  
  
C:\>_
```

In the above example there are two CPUs, each with six cores, for a total of twelve physical cores. This computer would satisfy the minimum hardware requirements for installing 64-bit Tableau Server.

A longer command will list the logical processors as well as the physical cores:

```
WMIC CPU Get  
DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesignation
```



```
C:\ Administrator: Command Prompt  
C:\>WMIC CPU Get DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesignation  
DeviceID  NumberOfCores  NumberOfLogicalProcessors  SocketDesignation  
CPU0      6              12                      CPU 1  
CPU1      6              12                      CPU 2  
  
C:\>_
```

In the above example, in addition to the twelve physical cores, there are 24 logical cores.

Upgrade to 9.2

After you've completed the [Pre-Upgrade Checklist](#) on page 51, upgrade your existing Tableau Server installation to version 9.2 by following one of the procedures below. If you are migrating to new hardware as part of your upgrade, refer to [Migrate to New Hardware](#) on page 68 instead of the procedures below.

When you install the newer version of Tableau Server, use the same drive and directory that the earlier version used. This way, data and configuration settings from your earlier version can be automatically imported.

Note: If you are upgrading and your original installation was not the default location, when you browse to the location, select the Tableau folder. Do not include the Tableau Server folder. The Setup program will append the Tableau Server folder to the path. If you include the Tableau Server folder, the Setup program will create a second Tableau Server folder, for example *install-drive\Program Files\Tableau\Tableau Server\Tableau Server*. Verify the path in the Setup program after you select the location. For more information on upgrading, see [Upgrade to 9.2 on the previous page](#).

During the installation of the new version, your existing extracts will be migrated to the new File Store. This process may take a long time (up to several hours if you have a large number of extracts or extracts that are large in size). While this takes place a message displays: "**Migrating extracts to File Store This process may take up to several hours.**" For more information, see [Troubleshoot Tableau Server Install and Upgrade on page 667](#)

If you are upgrading on a server with fewer than eight cores or less than 16 GB of RAM and you have not explicitly set the number of VizQL server processes, the number will be set to one instance. For optimum performance on machines with fewer than eight cores and less than 16 GB of RAM, set the number of VizQL server and data server processes to 1.

If you are upgrading from 32-bit Tableau Server to 64-bit Tableau Server you must uninstall your existing version before installing the new version.

Note: As a best practice you should make a backup of your Tableau Server data before upgrading. For more information, see [Pre-Upgrade Checklist on page 51](#).

Single Server Installations

To upgrade a single server installation of Tableau Server to version 9.2 or 9.2.x:

1. Use Add/Remove Programs on your Tableau Server computer to uninstall the earlier version.
Uninstalling removes the server software but leaves your data and configuration settings intact.
2. Install Tableau Server. Tableau Server Setup will handle importing the data and configuration settings from your earlier version.

Distributed Installations

If you are moving your cluster to the 64-bit version of Tableau Server as part of your upgrade to version 9.2, review the guidelines on "bit version" in the [Pre-Upgrade Checklist on page 51](#).

To upgrade from version 8.2, 8.3, 9.0 or 9.1 to version 9.2 or 9.2.x:

1. Use Add/Remove Programs on the primary Tableau Server computer to uninstall the earlier version.
2. Install Tableau Server on the primary server node.

Tableau Server Setup handles importing the data and configuration settings from your earlier version.

Note: If you are upgrading from a two-node cluster (a primary and one worker) or a configuration that used an external confirmation host, you may see a warning about the limitations of running Tableau Server on two nodes. For more information, see [Distributed Requirements on page 73](#)

3. If prompted by the Configure Tableau Server utility to upgrade the worker nodes, switch to the worker nodes and use Add/Remove Programs to uninstall the earlier version on those worker nodes.
Uninstalling removes the server software but leaves your data and configuration settings intact.
4. Install Tableau Worker Server on each worker node.
5. Return to the primary server and continue the installation.

To upgrade from version 9.2.x to version 9.2.x:

1. Use Add/Remove Programs on your primary Tableau Server computer to uninstall the earlier version.
Uninstalling removes the server software but leaves your data and configuration settings intact.
2. Install Tableau Server on your primary Tableau Server. In most cases, with a "same version" upgrade (version 9.2.x to 9.2.x), the primary Tableau Server pushes updates to the worker servers. so there is no need to uninstall and reinstall server software on the Tableau workers.

Note: If there is an update to PostgreSQL drivers or other third-party software, Tableau workers cannot be upgraded automatically. During upgrading a message tells you that "One or more workers could not be upgraded automatically" and

instructs you to manually upgrade the software on each worker. This can happen even during a "same version" upgrade.

If you are upgrading from 32-bit Tableau Server to 64-bit, you need to uninstall and reinstall. See Upgrading distributed installation of Tableau Server from 32-bit to 64-bit below.

Tableau Server Setup will handle importing the data and configuration settings from your earlier version.

Upgrading a distributed installation of Tableau Server from 32-bit to 64-bit

If you are upgrading a distributed installation from 32-bit to 64-bit, you need to take the following steps:

1. Use Add/Remove Programs on your primary Tableau Server computer to uninstall the 32-bit version from the primary server.
2. Install 64-bit Tableau Server on your primary Tableau Server node.
3. If prompted by the Configure Tableau Server utility to upgrade the worker nodes, switch to the worker nodes and use Add/Remove Programs to uninstall the earlier version on those worker nodes.

Uninstalling removes the server software but leaves your data and configuration settings intact.

4. Install 64-bit Tableau Worker Server on each worker node.
5. Return to the primary server and continue the installation of 64-bit Tableau Server.

Move the Repository Process

If you need to delete a worker node from your Tableau Server configuration and that worker is hosting the only instance of the repository, you must move the process to another computer before deleting the node. There must always be at least one active instance of the repository, so you cannot remove an instance if it is the only instance.

Note: If you are also moving a data engine/file store group, you can move the repository at the same time. See [Move the Data Engine and File Store Processes on the next page.](#)

1. Create a full backup of Tableau Server. For more information, see [Back Up the Tableau Data on page 631](#).
2. If you haven't done so already, [stop Tableau Server](#) and run the Tableau Server Configuration utility ([Start > Tableau Server 9.2 > Configure Tableau Server](#)) on

the primary Tableau Server node.

3. On the **Servers** tab, select the computer (IP address or computer name) onto which you want to move the process and click **Edit**. It can be another worker or the primary (*This Computer (Primary)*).
4. In the **Edit Tableau Server** dialog box, select the **Repository** check box and click **OK** to close the dialog box.
5. Click **OK** in the Tableau Server Configuration utility to save your changes and close the utility.
6. **Start the primary Tableau Server node** so that synchronization completes between the existing repository and the newly added repository.
7. Open the Status page in Tableau Server and wait until the new repository status no longer says "Setting up". When the repository status is "Passive" the synchronization is complete.
8. **Stop the server** and open the Tableau Server Configuration utility.
9. On the **Servers** tab, highlight the computer from which you are removing the process and click **Edit**.
10. Remove the processes you are moving: clear the **Repository** check box and click **OK**.
11. Click **OK** again to save your changes and close the utility.
12. **Start the primary server** so that the changes can take effect.

If you are performing this procedure as part of deleting a worker node from the Tableau Server configuration (as described in the [Pre-Upgrade Checklist](#) on page 51) stop Tableau Server again before proceeding.

Move the Data Engine and File Store Processes

If you need to delete a worker node from your Tableau Server configuration and that worker is hosting the only instance of the data engine and file store (which handle extracts), you must first move the processes to another computer. There must always be at least one instance of the data engine/file store processes, so you cannot remove an instance if it is the only instance.

1. Create a full backup of Tableau Server. For more information, see [Back Up the Tableau Data](#) on page 631.
2. If you haven't done so already, **stop the primary Tableau Server node** and run the Tableau Server Configuration utility (**Start > Tableau Server 9.2 > Configure Tableau Server**) on the primary Tableau Server node.
3. On the **Servers** tab, highlight the computer (IP address or computer name) onto which you want to move the processes and click **Edit**. It can be another worker or the primary (*This Computer (Primary)*).

4. In the **Edit Tableau Server** dialog box, enter the number of **Data Engine** processes, and click **OK** to close the dialog box.

Note: When you install a data engine process on a node, the file store process is also installed. Changing the value of **Data Engine** from 0 automatically selects the **File Store** check box.

5. Click **OK** in the Tableau Server Configuration utility to save your changes and close the utility.
6. **Start the primary Tableau Server node** so that the changes can take effect.
7. Open the Status page in Tableau Server and wait until the new file store status no longer says "Syncing".
8. **Stop the server**.
9. **Decommission** the file store on the worker:

From the Windows command line, in the `C:\Program Files\Tableau\Tableau Server\9.2\bin` directory, run:

```
tabadmin decommission <worker_node>
```

where `<worker_node>` is the name or ip address of the worker you are going to remove, as it appears in the list of servers on the **Servers** tab of the Configuration utility.

10. Open the Tableau Server Configuration utility and on the **Servers** tab, highlight the computer from which you are removing the process and click **Edit**.
11. Remove the processes you are moving: enter 0 for **Data Engine** and click **OK**. The File Store check box will be cleared automatically.
12. Click **OK** again to save your changes and close the utility.
13. **Start the primary server** so that the changes can take effect.

If you are performing this procedure as part of deleting a worker node from the Tableau Server configuration (as described in the [Pre-Upgrade Checklist on page 51](#)) stop Tableau Server again before proceeding.

Server Process Defaults and Limits

The number of installed process instances depends on the computer where you are installing Tableau Server.

If the computer meets or exceeds the minimum hardware recommendations, the default installed number of most processes is two instances. If the computer meets the minimum hardware requirements but does not have at least 8 cores and 16 GB of system memory, the defaults are reduced to one instance of each process. This is intentional and is intended to match the software to the available hardware. For more information on server hardware

requirements and recommendations, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#) on page 59.

Note: You should not change the number of processes without increasing associated RAM. If you are unsure, contact Tableau Support for guidance.

In a distributed installation, you can have a maximum of two repository instances (active and passive). You can also run Tableau Server with one repository, but doing this means there is no failover available for the repository.

If you are running the 64-bit version of Tableau Server (which is available starting with version 8.1), two instances of a process is the recommended maximum for most configurations. For more information on when you might want more than two instances of a process, see [Improve Server Performance](#) on page 430.

If you are running the 32-bit version of Tableau Server and the default settings aren't sufficient, you can change them to up to eight instances either during Setup (for upgrades only) or after Setup, using the [Tableau Server Configuration utility](#).

Changing the default upper limit

Eight instances of a process is the default upper limit. If your machine has enough RAM and CPU cores, and you want to go above this limit, you can change the limit using the `service.max_procs` tabadmin setting. For each process instance, Tableau recommends that the machine running the process have at least 1 GB of RAM and 1 logical CPU core.

To change the maximum number of processes allowed:

1. After Setup, [stop the server](#).
2. In the Tableau Server bin directory, type the following command, where `number` is the maximum number of process instances you want to allow:

```
tabadmin set service.max_procs <number>
```

For example:

```
tabadmin set service.max_procs 10
```

3. Still in the bin directory, type:

```
tabadmin config
```

4. [Start the server](#) so the changes can take effect.

Migrate to New Hardware

Use the following procedure to migrate Tableau Server from one computer to another. Specifically, these steps describe how to move Tableau Server data and configuration settings from your in-production computer to a new computer where Tableau Server version 9.2 is

installed. Before you start, make sure you have followed the steps in the [Pre-Upgrade Checklist](#) on page 51, including creating a [backup](#).

1. Install Tableau Server on the new computer.
2. Copy your `.tsbak` file to the bin folder on your new Tableau Server (for example, `C:\Program Files\Tableau\Tableau Server\9.2\bin`).
3. Next, [stop Tableau Server](#).
4. Restore your in-production data without configuration information to your new Tableau Server installation:

```
tabadmin restore --no-config <filename>
```

where `<filename>` is the name of the `.tsbak` file. For example:

```
tabadmin restore --no-config mybackup.tsbak
```

The `--no-config` option restores the data from your in-production Tableau Server but excludes configuration information. You need to use this option when moving to new hardware because otherwise you will have conflicts with the old configuration. After doing the restore, you may need to reconfigure some options (SMTP or proxy settings, for example).

5. [Start the server](#).
6. **Distributed installations only:** Run the Tableau worker installer on all the additional computers you want to add to your Tableau Server cluster. See [Install and Configure Worker Nodes](#) on page 76 for steps.
7. The same Tableau Server product key can be activated three times: once for a production environment, once for a test environment, and once for a QA environment. After you have tested your new Tableau Server installation and confirmed that it's ready for production, you must deactivate your earlier production version of Tableau Server, and then you must uninstall it. To deactivate the earlier version:
 - Select **Start > All Programs > Tableau Server > Manage Product Keys**.
 - For each product key, select the product key and click **Deactivate**.

Note: If you do not have an internet connection, you are prompted to create an offline activation file to complete the deactivation process. See [Activate Tableau Offline](#) on page 10 for steps.

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see [Troubleshoot Server Processes on page 242](#).

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.
2. Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tabadmin restart` command. This will stop all the processes associated with Tableau Server and then restart them.
3. Clean up files associated with the Coordination Service (ZooKeeper). To clean up Coordination Service files, use the `tabadmin cleanup --reset-coordination` command.

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tabadmin status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tabadmin restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Installing Tableau Server

Install fails due to hardware requirements

Starting with version 9.0, Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The minimum requirements are designed to minimize issues that result from running Tableau Server on under-powered computers. The requirements apply to both primary server computers and worker computers. The minimum requirements are lower for the 32-bit version of Tableau Server. If you are unable to install the 64-bit version due to hardware limitations, you may be able to use the 32-bit version instead.

For details on minimum hardware requirements, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#) on page 59.

Upgrading Tableau Server

Migrating Extracts to the File Store

Tableau Server 9.2 introduced a more reliable storage mechanism for data extracts called the File Store. Upgrading from a previous version requires migration of the extracts. This can take a long time (up to several hours) if you have a large number of extracts or extracts that have a lot of data. During migration a message displays:

```
Migrating extracts to File Store  
This process may take up to several hours.
```

If the migration progress appears to be stalled or stuck, you can verify that migration is continuing by watching the `tabadmin.log`. An entry is written to this log for each extract that is migrated.

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts. As a part of upgrading to version 9.0, the Setup program migrates extracts to the new File Store and this takes space.

To free up disk space:

1. Zip and save logs using the `tabadmin ziplogs` command.

After you create the `ziplogs` file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tabadmin cleanup` command. For more information, see [Remove Unneeded Files](#) on page 634

Reindexing Tableau Server Search & Browse

Other problems that can be solved by reindexing Search & Browse

Other symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, rebuild the Search & Browse index using the `tabadmin reindex` command.

Distributed Environments

With a distributed installation, you install portions of Tableau Server on different computers.

Distributed Requirements

Before you start to configure a Tableau Server cluster, make sure you meet the following requirements.

Hardware

While the computers you use in your cluster must meet the requirements described in [Before you install... on page 3](#), they do not need to be identical.

Hardware Guidelines for High Availability

Here are some guidelines for the systems you use for [failover and high availability](#):

- **Failover—three computers:** To configure a cluster that provides failover support for the data engine and repository processes, you need at least three computers or VMs: one for the primary Tableau Server and two for Tableau worker nodes.

Note: If you install Tableau Server on a two-node cluster (the primary and one worker) with a repository and a data engine/file store on each node, a warning displays to let you know that you will not have failover support with this configuration and asking if you want to add a third node. You are not required to add a third server to the cluster, but with a two-node cluster there is no failover support, and if one of the two nodes goes down, Tableau Server will shut down.

- **Failover & multiple gateway support—three computers and a load balancer:** To configure a cluster that provides the above plus support for multiple gateways, you need at least three computers or VMs, and a load balancer to front the cluster.
- **High availability—four computers and a load balancer:** To configure for high availability, you need the resources described above plus an additional computer to be the backup primary for your primary Tableau Server.
- **Primary computers:** If you configure for high availability, the primary Tableau Server and the backup primary may be running few or no Tableau Server processes. Therefore, the computers that run the primary and backup primary do not need as many cores as the ones running your worker servers. You will, however, need adequate disk space for backups because the primary computer is used during the database backup and restore processes. In addition to the amount of space needed for the backup file, you need

temporary disk space roughly 10 times the size of the backup file (so if your backup is 4 GB, you should have about 40 GB of temporary disk space available).

Software

Tableau Server is available in 32- and 64-bit versions. If you are running a Tableau Server cluster, each computer must run the same bit version—either all 64-bit or all 32-bit. For example, if the primary Tableau Server is running the 64-bit version of Tableau Server, the workers in the cluster must run the 64-bit version of Tableau Server Worker. They can't run the 32-bit version of Tableau Server Worker.

Note: To install Tableau Server on multiple nodes, you must have a Tableau Server—Multi-Machine Core license or a user-based license.

Networking and Ports

- **Ports:** As with any distributed system, the computers or VMs you use need to be able to communicate with one another. See [Tableau Server Ports on page 540](#) for a list of ports that must be available on the gateways and workers.
- **Same domain:** All computers in a cluster must be members of the same domain. The server's [Run As User on page 525](#) account, which is specified on the primary Tableau Server, must be a domain account in this same domain.
- **Static IP addresses:** Any computer running Tableau Server, whether it's a single server installation or part of a cluster, must have a static IP address ([learn more](#)).

Best Practices

Here are some things to keep in mind before you start to install and configure:

- **IP addresses or computer names:** Note the IPv4 addresses or computer names of each computer or VM you'll be working with. You will need to provide them during Tableau Worker Setup and configuration. As mentioned above, each computer in the cluster must use a static IP address, even if you use the computer's name to identify it during configuration.
- **CNAME record:** If you're configuring for high availability and you are not using a load balancer, make sure your primary Tableau Server and backup primary have the same CNAME record so that your Tableau Server users have a smooth experience if one primary fails and you configure the other to take over. If you are using a load balancer, it's the load balancer's name that users will be using as the Tableau Server URL, regardless of the gateway that's actually handling the request.
- **User account credentials:** For each computer, you need credentials for a user

account with local admin permissions. If you're configuring for high availability, the Run As account you use for your primary Tableau Server must be the same as the one you use for your backup primary Tableau Server.

- **Backup:** It's a best practice to create a backup prior to making significant system changes. See [Back Up the Tableau Data](#) on page 631 for steps.

SSL

If you are planning to configure SSL for a highly available Tableau Server cluster with multiple gateways and a load balancer ([learn more](#)), make sure that the SSL certificate you use was issued for the load balancer's host name. See [Configure SSL for a Cluster](#) on page 493 for other details.

Hostname Support in Tableau Server

Starting with version 8.1, hostname support was added to Tableau Server. This means that when you're configuring Tableau Server to work with another computer, you can use the name of that computer to identify it, instead of its static IPv4 address. Internally, however, Tableau Server still relies on IP addresses to communicate with various services, such as Tableau workers or trusted hosts. So even if you provided the name of a computer instead of its IP address, the IP address associated with that computer can't change or be temporary.

If a computer running Tableau Server gets a new IP address—for example, after a VM reboot, or in a network environment that's using DHCP—you need to run `tabadmin config` to update Tableau Server's configuration with the change. See the procedure below for steps.

In addition to DHCP, another item that could result in an IP address changing, post-Setup, is a Windows operating system feature for IPv6 addresses called "temporary IPv6 addresses". See the [Knowledge Base](#) for details on how to identify and disable this feature.

To update the Tableau Server configuration:

1. On the primary Tableau Server, open a command prompt as an administrator.
2. Type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

3. Stop the server:

```
tabadmin stop
```

4. Update the server's configuration by typing the following:

```
tabadmin config
```

5. Start the server:

```
tabadmin start
```

Install and Configure Worker Nodes

After you complete the initial configuration, you can set up Tableau Server to run on multiple computers. This is called a distributed installation, or cluster. Running a distributed installation uses additional ports on the primary Tableau Server and requires that certain ports be available for binding during Setup on the Tableau Worker Servers. See [Tableau Server Ports on page 540](#) for more information. There are also additional requirements to be aware of when you run a distributed installation. See [Distributed Requirements on page 73](#) for details.

Note: If you install Tableau Server on a two-node cluster (the primary and one worker) with a repository and a data engine/file store on each node, a warning displays to let you know that you will not have failover support with this configuration and asking if you want to add a third node. You are not required to add a third server to the cluster, but with a two-node cluster there is no failover support, and if one of the two nodes goes down, Tableau Server will shut down.

To install and configure a Tableau Worker Server

1. Make sure you've installed Tableau Server on the primary computer.
2. Stop Tableau Server on the primary node (see [Tableau Server Monitor on page 264](#) to learn how).
3. Download the Tableau Server Worker software from the [Tableau Customer Account Center](#).
4. Run Tableau Server Worker Setup on all additional computers that you want to add to the Tableau Server cluster.
5. During installation you will be asked to provide the IPv4 addresses or computer name of the primary server. Using a computer name is recommended.

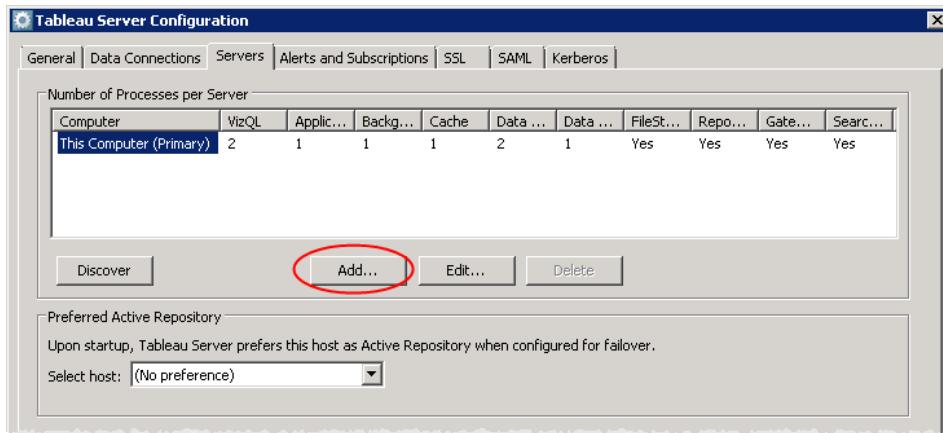
If the primary has multiple network interface cards (NICs) enabled and you choose to enter IPv4 addresses, enter all of the primary's IPv4 addresses, separating each with a comma. The IP address(es) for the computer running the primary must be static, this applies even if you use a computer name to identify the primary ([learn more](#)).

If you have a worker running Windows 7 with Windows Firewall enabled, refer to the [Tableau Knowledge Base](#) before proceeding.

6. Once the Worker software is installed on worker computers, and with the primary Tableau Server still stopped, return to the primary server and open the configuration utility by selecting **Tableau Server 9.2 > Configure Tableau Server** on the Start

menu.

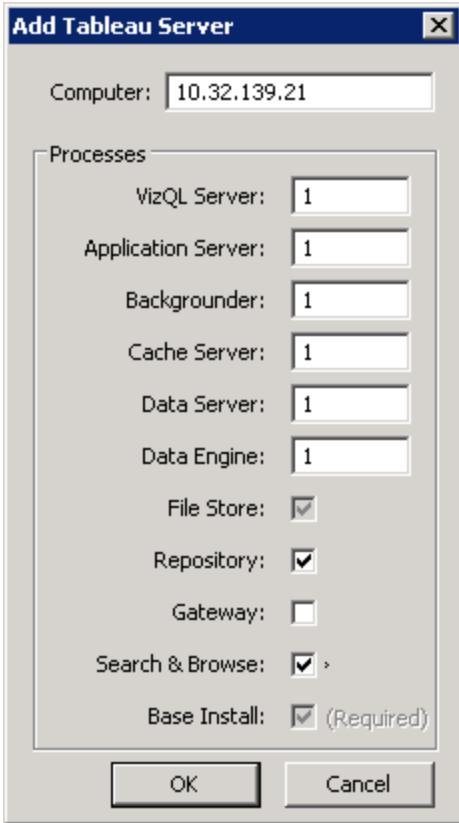
7. In the Configuration Utility, enter your password on the **General** tab then select the **Servers** tab and click **Add**.



Note: Click the **Discover** button to automatically add any worker computers configured in step 5 (above) with the IPv4 address or name of the computer on which you are running the configuration utility.

8. In the next dialog box, type the IPv4 address or computer name for one of the worker computers and specify the number of **VizQL**, **Application Server**, **Background**, **Cache Server**, **Data Server**, **Data Engine**, **File Store**, **Repository**, **Gateway**, and **Search & Browse** processes to allocate to the computer.

With the 64-bit version of Tableau Worker Server, you can run up to two instances of each process. In rare cases and if the server's hardware allows, that limit can be changed. See [Server Process Defaults and Limits](#) on page 67 and [Performance](#) on page 428 for more information.



By default, the data engine and file store, repository, and gateway are hosted on the primary server. Running these processes on an additional server, or moving them off of the primary server, is part of configuring for high availability. See [High Availability](#) on page 82 for more information.

9. Click **OK**. It may take several minutes for the updates to complete.
10. Repeat these steps for each computer you want to add to the distributed environment. When you're finished adding workers, click **OK** to save the changes and close the Configuration utility, then start Tableau Server on the primary node.

Database Drivers

The installers for Tableau Server and Tableau Server Workers automatically install drivers for Oracle and Oracle Essbase databases. If you plan to publish workbooks and data sources that connect to other databases, you will need to make sure that both your primary and worker servers have the corresponding drivers.

Workers running VizQL, application server, data server, or backgrounder processes need these database drivers. For example, if you have a worker dedicated as a VizQL server and another computer dedicated to extract storage, you only need to install drivers on the computer running the VizQL server process.

Server process	Requires data-base driver?
VizQL server	yes
Application server	yes
Data server	yes
Backgrounder	yes
API server	yes
Data engine (extract storage)	no
Repository	no
Gateway	no
Cluster controller	no
Cache server	no
Search & Browse	no
File store	no

Reinstall and Configure Worker Node

You might need to reinstall one of your Tableau worker nodes. To do so, follow one of these procedures. The specific steps you take depend on whether or not the worker you are reinstalling has data engine or repository components on it and whether or not these are duplicated on any other node in the installation.

Note: Reinstalling multiple workers at the same time could lead to data loss.

Use the following procedure to help you reinstall and configure a worker node that is hosting the *only* data engine or repository in the distributed installation. Every Tableau Server installation requires at least one data engine and one repository. If you are reinstalling the worker node that hosts either of these processes, you must first add the process to a second node.

To reinstall the worker node hosting the data engine or repository instance

1. Create a full backup of Tableau Server. For more information, see [Back Up the Tableau Data on page 631](#).
2. Stop Tableau Server on the primary by selecting **Tableau Server 9.2> Stop Tableau**

Server on the Windows Start menu, or by running the `tabadmin stop` on page 613 command from the command line.

3. On the Start menu, select **Tableau Server 9.2 > Configure Tableau Server**.
4. In the Configuration Utility:
 - On the **General** tab, enter your password.
 - On the **Servers** tab, add the data engine and/or repository components that the worker is hosting to another worker or to the primary, and then save your changes.
For example, if the worker you are reinstalling currently hosts the data engine, add the data engine to another node.
5. **Start the primary Tableau Server node** so that synchronization completes between the existing data engine or repository on the worker you will be reinstalling and the newly added instances of those processes.
6. Open the Status page in Tableau Server and check on the components you added:
 - If you added a data engine/file store, wait until the new file store status no longer says "Syncing".
 - If you added a repository, wait until the new repository status says "Passive".
7. Stop Tableau Server.
8. If you are removing a node that hosts data engine, **decommission** the file store you are removing:

From the Windows command line, in the `C:\Program Files\Tableau\Tableau Server\9.2\bin` directory, run:

```
tabadmin decommission <worker_node>
```

where `<worker_node>` is the name or ip address of the worker you are going to remove, as it appears in the list of servers on the **Servers** tab of the Configuration utility.
9. In the Configuration Utility:
 - On the **General** tab, enter your password.
 - On the **Servers** tab, select the worker you want to reinstall and then click **Delete**.
 - Save your changes.
10. Start Tableau Server and verify that everything is working as expected.
11. On the worker:
 - Uninstall the Tableau Server worker software from Windows Control Panel.
 - Delete (or rename) the following folders: `C:\Program Files\Tableau` and `C:\ProgramData\Tableau`. `\ProgramData` is a hidden folder so may not be visible.
 - Install the updated worker software.
12. On the Tableau Server primary, stop Tableau Server, add the worker back into the

configuration, and then save the changes.

Note: The data engine and repository need to remain on at least one node while you are re-adding the worker.

13. Start Tableau Server.

Use the following procedure to help you reinstall and configure a Tableau worker that is either not hosting a data engine or repository, or is hosting a component but there is an additional node that is hosting the same component.

To reinstall and configure the worker node that is either not hosting data engine or file store or hosting one that is also on another node

1. Create a full backup of Tableau Server.
2. Stop Tableau Server on the primary by selecting **Tableau Server 9.2 > Stop Tableau Server** on the Start menu or by running the `tabadmin stop` command at a command prompt.
3. If you are removing a node that includes a data engine/file store pair, **decommission** the file store on that node:

From the Windows command line, in the `C:\Program Files\Tableau\Tableau Server\9.2\bin` directory, run:

```
tabadmin decommission <worker_node>
```

where `<worker_node>` is the name or ip address of the worker you are going to remove, as it appears in the list of servers on the **Servers** tab of the Configuration utility.

4. Open the configuration utility by selecting **Tableau Server 9.2 > Configure Tableau Server** on the Start menu.
5. In the Configuration Utility:
 - On the **General** tab, enter your password.
 - On the **Servers** tab, select the worker you want to reinstall and then click **Delete**.
 - Save your changes.
6. Start Tableau Server and verify that everything is working as expected.
7. On the worker:
 - Uninstall the Tableau Server Worker software from Control Panel.
 - Delete (or rename) the following folders: `C:\Program Files\Tableau` and `C:\ProgramData\Tableau`. `\ProgramData` is a hidden folder so may not be visible.
 - Install the updated worker software.
8. On the primary node, stop Tableau Server, use the configuration utility to add the worker

back into the configuration, and then save the configuration.

Note: The data engine and repository need to remain on at least one node while you are re-adding the worker.

9. Start Tableau Server.

Maintain a Distributed Environment

After you set up a primary and one or more worker servers for a distributed installation, you can perform all subsequent configuration and updates from the primary server, using the command line tools and configuration utility on the primary server. Updates will be pushed to the workers automatically.

When you installed worker servers, you specified the primary's IPv4 address or computer name. If that IP address or computer name changes, you will need to re-install the worker servers.

You can monitor the status of the Tableau Server cluster on the server Maintenance page. See [Server Maintenance on page 240](#) to learn more about maintaining the server.

Server Status			
Process Status			
Process	Primary 10.32.139.22	Worker 1 10.32.139.21	Worker 2 10.32.139.30
Cluster Controller	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Gateway	<input checked="" type="checkbox"/>		
Application Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VizQL Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cache Server	<input checked="" type="checkbox"/>		
Search & Browse	<input checked="" type="checkbox"/>		

High Availability

A high availability installation of Tableau Server is a special type of distributed installation designed to maximize the availability of Tableau Server.

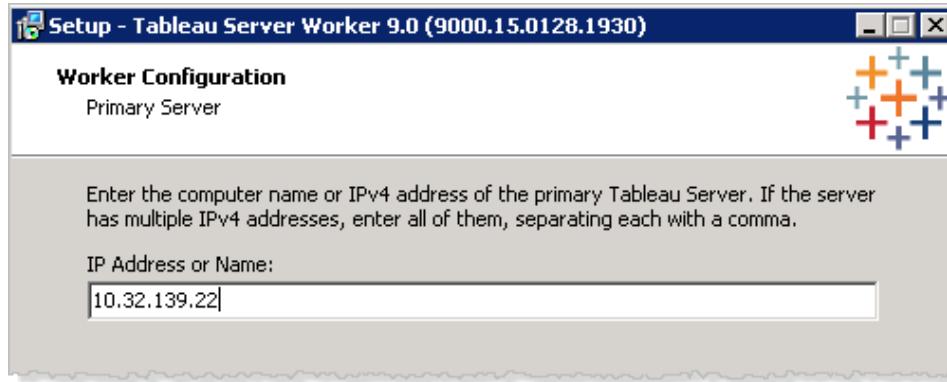
Quick Start: Configuring Failover & Highly Available Gateways

Extracts and repository data can change rapidly and even regular backups may not help you fully recover from a system failure. Another vulnerability is having a single entry point, or gateway, for your Tableau Server cluster. To help with this, distributed Tableau Server

deployments provide real-time content replication and failover support, as well as the ability to run multiple gateways.

1 Install the Servers

Install Tableau Server on the primary computer. After Setup, stop the server and run Tableau Worker Setup on the two additional computers that will provide failover support. During Worker Setup, provide the primary's IPv4 address or name.

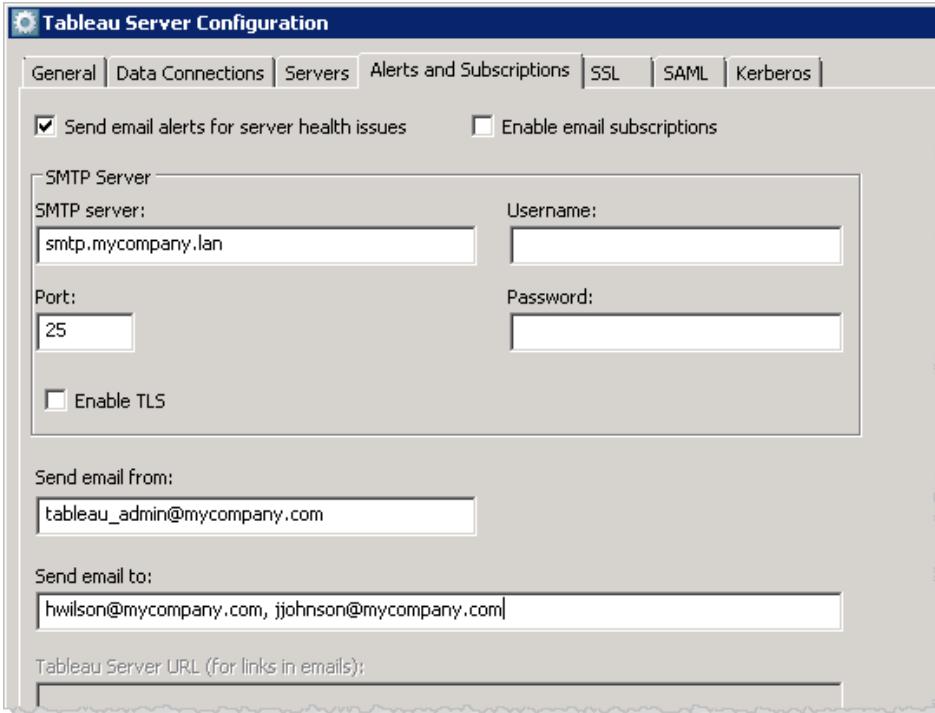


To stop or start the server, at a command prompt, go to the Tableau Server bin folder and type tabadmin stop or tabadmin start.

Stop the primary server and open its Configuration utility.

3 Set Up Email Alerts

After you add the second worker and with the Configuration utility still open, click the **Alerts and Subscriptions** tab in the Configuration utility and select **Send email alerts for server health issues**:



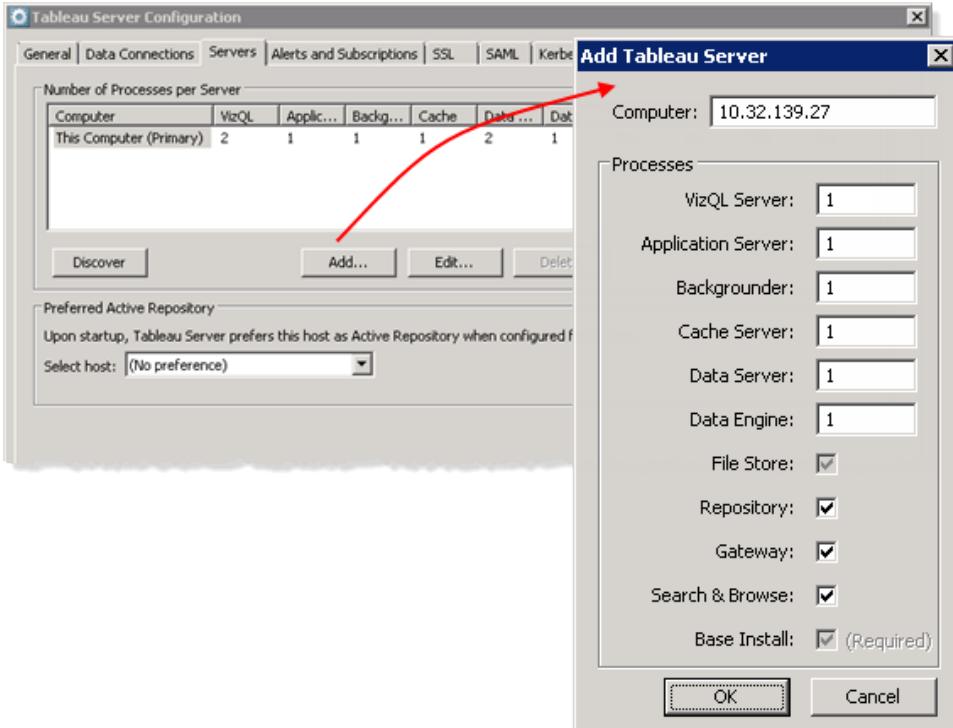
When you test, your email account will receive messages about the services.

Enter the name of your SMTP server—and a username and password if it's required by your SMTP server.

Next, enter the email account that will send an alert if there's a system failure, and the account(s) that will receive it. Click **OK** and start Tableau Server.

2 Configure the Distributed System

1. On the **Servers** tab, click **Add** to add a worker server. Enter its IPv4 address or computer name. Enter 1 for each process. Select **Repository**, **Gateway**, and **Search & Browse**. Click **OK**:



2. Click **Add** to add a second worker server. Enter its IPv4 address or computer name. Enter **1** for every process except the **Data Engine** (set that to **0**). Leave **Repository** cleared but select **Gateway**. Click **OK**.
3. Click **OK** to close the Configuration utility, then start Tableau Server on the primary server so your changes can take effect.
4. Stop the primary server and open the Configuration utility.
5. On the **Servers** tab, select the second worker and click **Edit**. Set **Data Engine** to **1** and select the **Repository** check box. Click **OK**, then **OK** again to close the Configuration utility. Start Tableau Server.
6. Still on the **Servers** tab, select **This Computer (Primary)** and click **Edit**. Set every process to **0**, clear the **Repository** check box but keep **Gateway** selected. Click **OK**.

4 Load Balance the Gateways

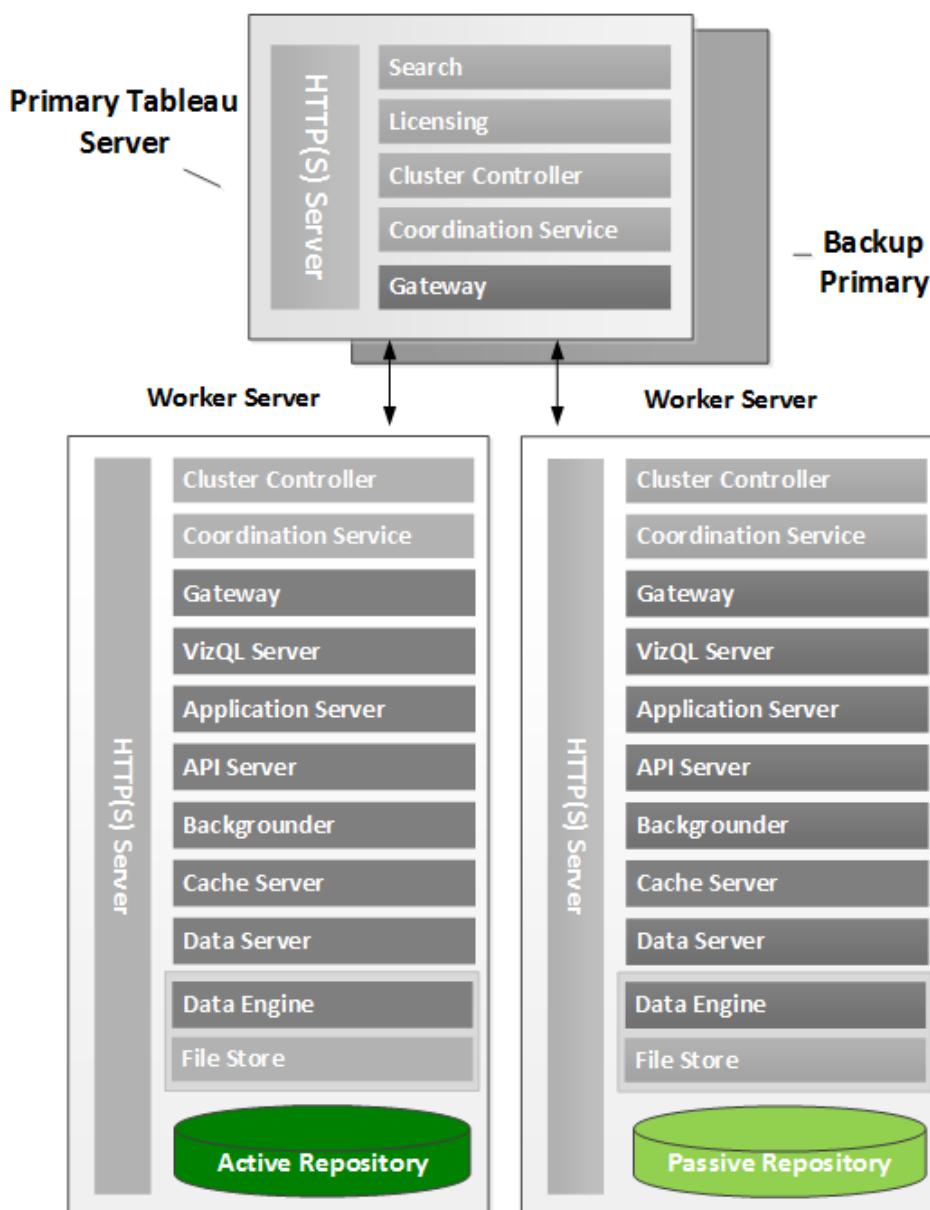
You can optionally use a load balancer to ensure the cluster's availability in the event of gateway failure, and to distribute the cluster's workload.

In your load balancer, enter the IP address for each computer that's running a gateway process (the primary and the two workers), and configure the load balancing method, such as Fastest or Round Robin

Quick Start: Creating a Backup Primary

This Quick Start describes how to create a backup of your primary Tableau Server so that if your current primary fails, it will take just a few steps to bring your backup primary online.

Before beginning, make sure you have configured your environment for failover and highly available gateways, using the [Quick Start: Configuring Failover & Highly Available Gateways on page 82](#) as your guide. You should have two worker servers and a primary Tableau Server. To help ensure a smooth transition for your Tableau Server users, assign the same common name to both your current and backup primary servers.



Configuring Primary Failover

1 Configure the Primary

Stop the server on your primary Tableau Server, then run the following command from the Tableau Server bin directory:

```
tabadmin failoverprimary --primary "<computer1>,<computer2>"
```

`computer1` is the current primary's IPv4 address or computer name. `computer2` is the backup primary's IPv4 address or computer name.

2 Copy the Primary's Config to the Backup

Copy the primary's `tabsvc.yml` file (located in `ProgramData\Tableau\Tableau Server\config`) to a temporary location on the backup primary. In the file, replace the IPv4 address or computer name for the primary (on the `worker.hosts` line) with the IPv4 address or computer name for the backup primary.

3 Install & Disable the Backup Primary

Install Tableau Server on your backup primary. After Setup completes, open a command prompt on the backup primary and stop the server. Next, run the following command:

```
tabadmin autostart off
```

Before you begin the next section, power down your primary to simulate a system failure.

After the Primary Fails

4 Configure the Backup Primary

On your backup primary, use the `tabsvc.yml` file you edited in step 2 to overwrite the locally installed `tabsvc.yml`. (If [web data connectors](#) were imported to the primary server, copy them to the primary backup.) Next, open a command prompt on your backup primary and run the following command from the backup primary's Tableau Server bin directory:

```
tabadmin failoverprimary --primary "<computer2>, <computer1>"
```

`computer2` is the IPv4 address or computer name of your backup primary (soon to be your active primary) and `computer1` is the IPv4 address or computer name for your former primary (soon to be your backup).

5 Start the Backup Primary

Run the following command:

```
tabadmin autostart on
```

Then start the server. Your backup primary is now your primary.

6 View Status

Sign in to Tableau Server on your new primary and view the status of your distributed system on the Status page. In the first row of the Status table you'll see the IP address or computer name of your new primary server.

Understanding High Availability

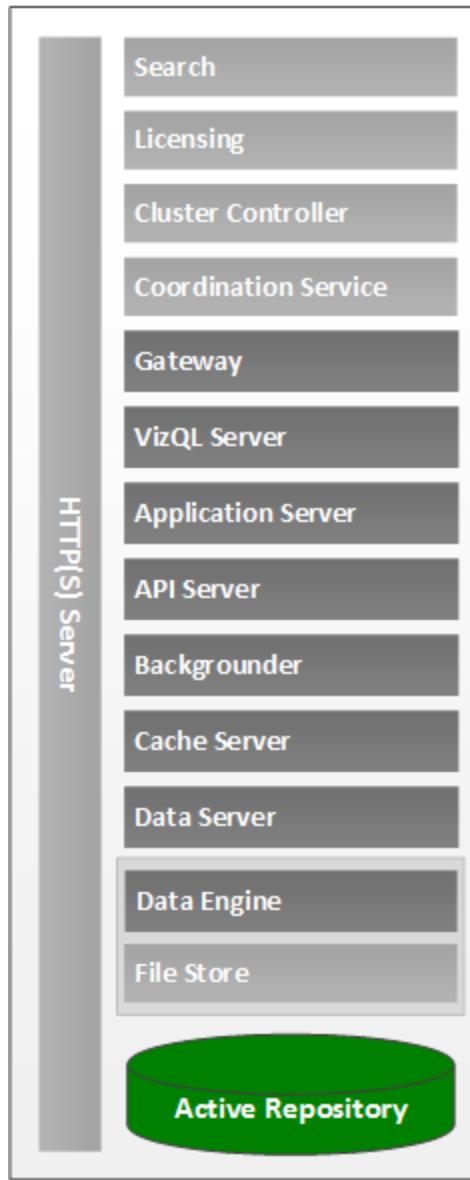
If you're configuring a Tableau Server system for high availability, the steps you perform are all designed to build in redundancy, thus reducing your potential downtime. The four areas that require redundancy are the data engine, repository, and gateway processes, and the primary Tableau Server, which runs the server's licensing component. Because there must always be one active of the repository process, configuring the cluster is a multi-phased procedure that requires the primary Tableau Server to be stopped and restarted at certain points so that settings can take effect. For exact steps, see [Configure for Failover and Multiple Gateways on page 94](#) and [Use a Backup Primary on page 108](#). See [Distributed Requirements on page 73](#) as well.

The topics below summarize how your server system topology evolves as you configure it for high availability. The minimum supported configuration for high availability is a three-node system. This includes a primary server to run licensing and two workers to host the main processes. You can increase reliability of the system by adding a fourth computer to serve as a backup primary. If you run a gateway process on all nodes, it also makes sense to use a load balancer for the gateways.

A Single Server System

After you install the primary Tableau Server, it is running at least one instance of all server processes. This is the most basic configuration of Tableau Server. It has no redundancy.

Primary Tableau Server



Here's what the Process Status table on the Server Status page typically looks like for a single-server system:

Server Status	
Process Status	
The real-time status of processes running in Tableau Server.	
Process	10.32.139.21
Gateway	✓
Application Server	✓
API Server	✓
VizQL Server	✓✓
Cache Server	✓✓
Search & Browse	✓
Backgrounder	✓
Data Server	✓✓
Data Engine	✓
File Store	✓
Repository	✓

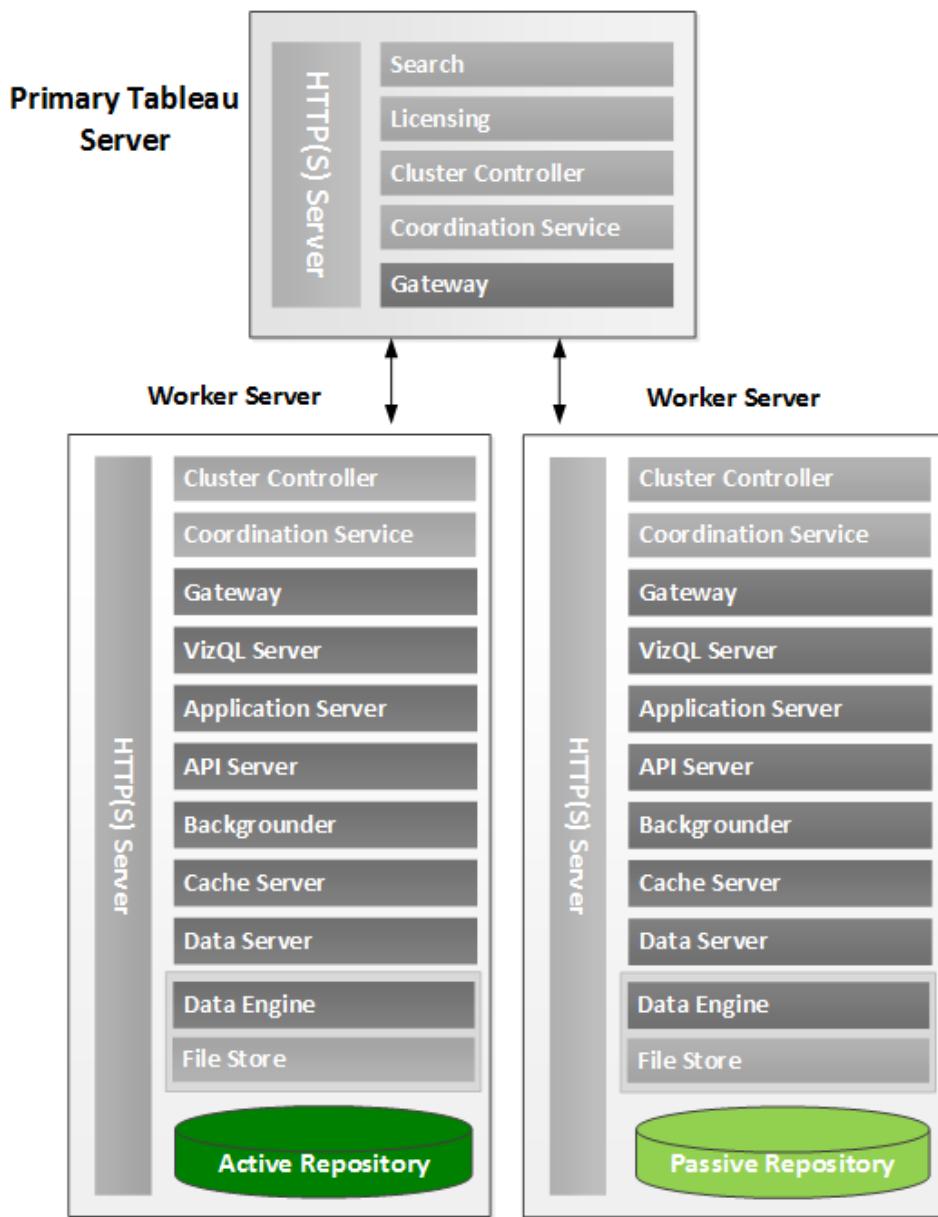
 ✓ Active
 ⌚ Busy
 ✓ Passive
 ⚠ Unlicensed
 ✗ Down
 □ Status unavailable

To build in redundancy, you need to add additional servers to host copies of the repository and data engine/file store processes. In addition, to reduce the system's vulnerability, you can run multiple gateways, and the primary should be isolated on its own node, ideally running as few of the server processes as possible. The fewest number of computers required to achieve this is three (see [A Three-Node System below](#)).

Note: To install Tableau Server on multiple nodes, you must have a Tableau Server—Multi-Machine Core license.

A Three-Node System

A three-node system helps you reduce the primary's vulnerability:



This configuration would look like the following Process Status table on the Server Status page.

Server Status			
Process Status			
The real-time status of processes running in Tableau Server.			
Process	Primary 10.32.139.21	Worker 1 10.32.139.22	Worker 2 10.32.139.30
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server	✓	✓	✓✓
API Server	✓	✓	✓
VizQL Server	✓	✓	✓✓
Cache Server	✓	✓	✓✓
Search & Browse	✓	✓	✓
Backgrounder	✓	✓	✓
Data Server	✓	✓	✓
Data Engine	✓	✓	✓
File Store	✓	✓	✓
Repository	✓	✓	✓

 ✓ Active
 ⌚ Busy
 ✗ Passive
 ⚠ Unlicensed
 ✗ Down
 □ Status unavailable

In a three-node cluster, the Data Engine and Repository processes have been moved from the primary to a worker, and the primary is only running the Gateway and Search & Browse processes. In this configuration, if your active worker fails, the passive worker automatically becomes active. Exactly how to create this three-node cluster, including how to add the workers and remove the processes from the primary, is described in [Configure for Failover and Multiple Gateways on page 94](#). (Licensing functionality is integral to the primary and cannot be removed, so it is not displayed on the Status page. Cluster Controller and Coordination Service are installed on all nodes as part of the "base install" and are not configurable. Coordination Service does not show on the Status page and Cluster Controller only displays if there are two or more nodes in the cluster.)

There are still two things you can do to improve this three-node cluster: 1) add a load balancer to interface with the three active gateways, and 2) create a backup to address the single point of failure: the primary. See the topics below for details.

Add a Load Balancer

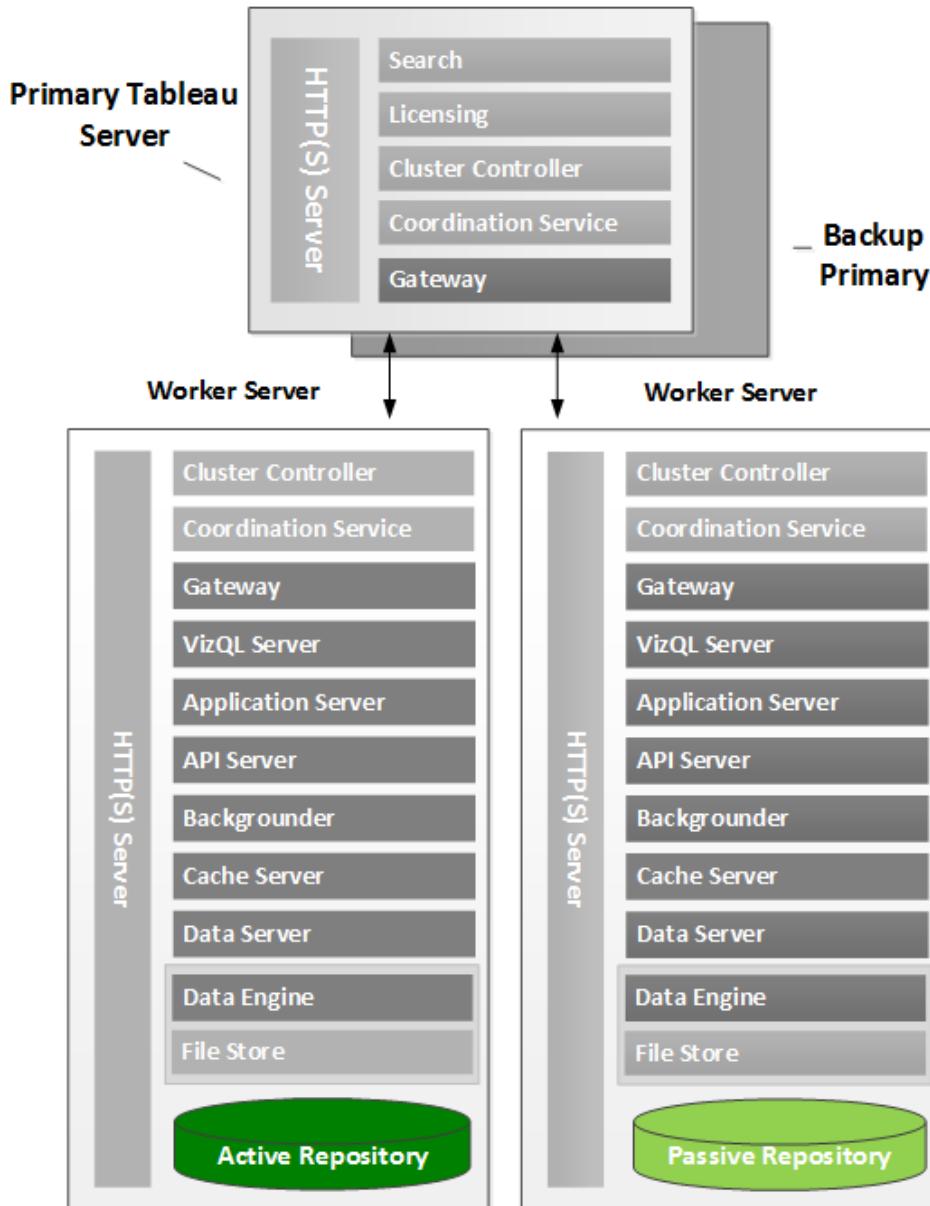
At this point, all three nodes have gateways, which are used to route requests to available server processes. Unlike the repository process, there aren't active and standby gateways. All gateways are active. To further reduce your cluster's potential for downtime, you should [configure a load balancer](#).

Add a Backup Primary

Adding a backup primary provides a safeguard for your system. The backup primary is an additional server added to the system to be ready if your primary fails. While it is not an active server, after you complete the first set of steps in [Use a Backup Primary on page 108](#), it is

ready to be activated. While the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

Here's what the system looks like with a backup primary:



The Process Status table for the configuration shown above looks the same as for a three-node system. If the primary fails and you perform the steps for the backup primary to take over, your system is back online using the new primary:

Server Status			
Process Status			
The real-time status of processes running in Tableau Server.			
Process	Primary 10.32.139.21	Worker 1 10.32.139.22	Worker 2 10.32.139.30
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server	✓	✓	✓✓
API Server	✓	✓	✓
VizQL Server	✓	✓	✓✓
Cache Server	✓	✓	✓✓
Search & Browse	✓	✓	✓
Backgrounder	✓	✓	✓
Data Server	✓	✓	✓
Data Engine	✓	✓	✓
File Store	✓	✓	✓
Repository	✓	✓	✓

 ✓ Active
 ⌚ Busy
 ✗ Passive
 ⚠ Unlicensed
 ✗ Down
 □ Status unavailable

Licensing only runs on the primary Tableau Server and is checked every 8 hours. If the primary goes down and is not running any other processes, you have a window of up to 72 hours to bring the backup primary online. The actual time period depends on when the last licensing check took place, and on whether any processes restart during the licensing window. For example, if the first failed licensing check occurred 71 hours ago, you have 1 hour to bring the backup primary online. If the licensing check occurred 1 minute ago, you have up to 71 hours and 59 minutes. **Note:** Any processes that restart during that 72 hour window will not be licensed and therefore will not be able to respond to service requests.

Configure for Failover and Multiple Gateways

Do the following to configure a three-computer cluster that provides multiple gateways and failover support. In most cases, running multiple gateways makes sense only if you plan to also use a load balancer.

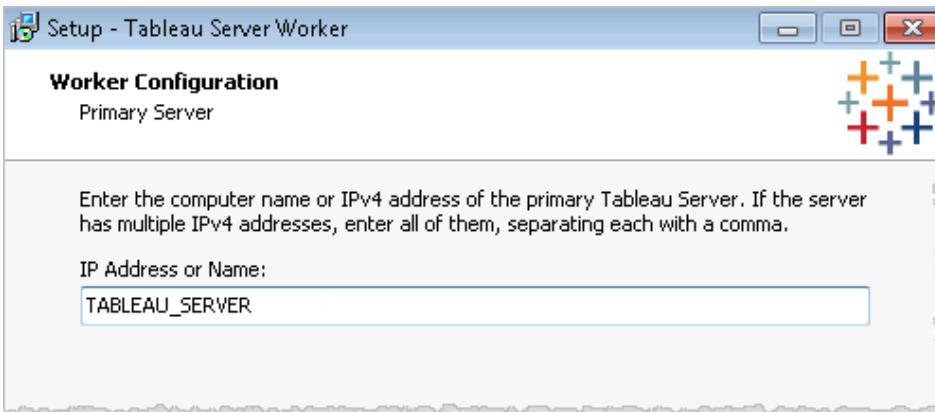
Note: To install Tableau Server on multiple nodes, you must have a Tableau Server—Multi-Machine Core license.

1. [Install Tableau Server](#) on your primary computer.
2. After Setup completes, check the Status page. All the processes should have a green “active” status:

Server Status	
Process Status	
The real-time status of processes running in Tableau Server.	
Process	10.32.139.21
Gateway	✓
Application Server	✓
API Server	✓
VizQL Server	✓ ✓
Cache Server	✓ ✓
Search & Browse	✓
Backgrounder	✓
Data Server	✓ ✓
Data Engine	✓
File Store	✓
Repository	✓

 ✓ Active
 ⌚ Busy
 ✗ Passive
 ⚠ Unlicensed
 ✗ Down
 □ Status unavailable

3. **Stop the server** on the primary computer.
4. Run **Tableau Worker Setup** on the two additional computers or VMs that will provide failover and extra gateway support. During Worker Setup, you will need to provide the computer name (recommended) or IPv4 addresses of the primary Tableau Server. If you enter multiple IPv4 addresses, separate each with a comma.



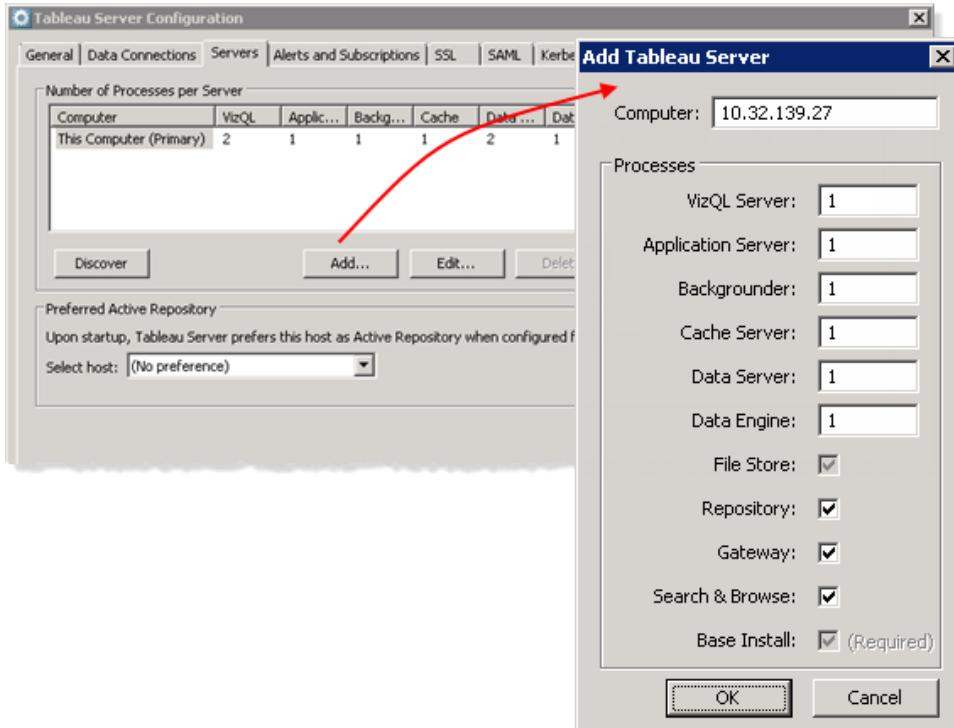
Note: The primary computer must have a static IP address assigned to it, even if you are using the primary's computer name to identify it ([learn more](#)).

5. With the primary server still stopped, start the Tableau Server Configuration utility: **Start > All Programs > Tableau Server > Configure Tableau Server**. On the **General** tab enter the Run As account password.

6. On the **Servers** tab, click **Add** to add a worker.

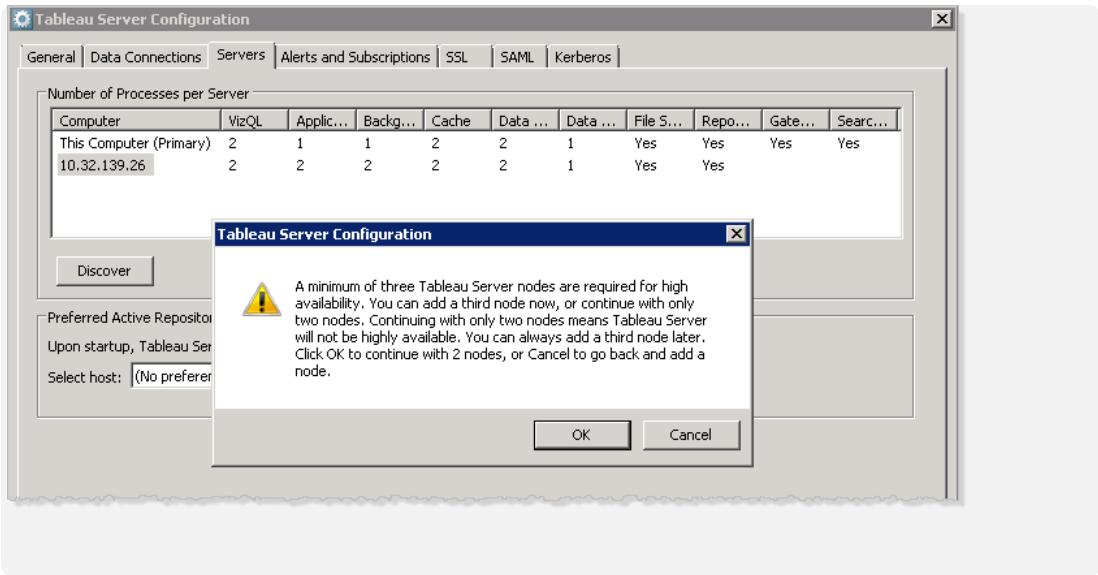
Note: The **Add** button is not functional if you are configuring a server that is licensed with a Tableau Server—Single-Machine Core license.

7. Enter the IPv4 address or computer name of the worker, enter 1 for **Data Engine (File Store** will be automatically selected) and select the **Repository** check box.

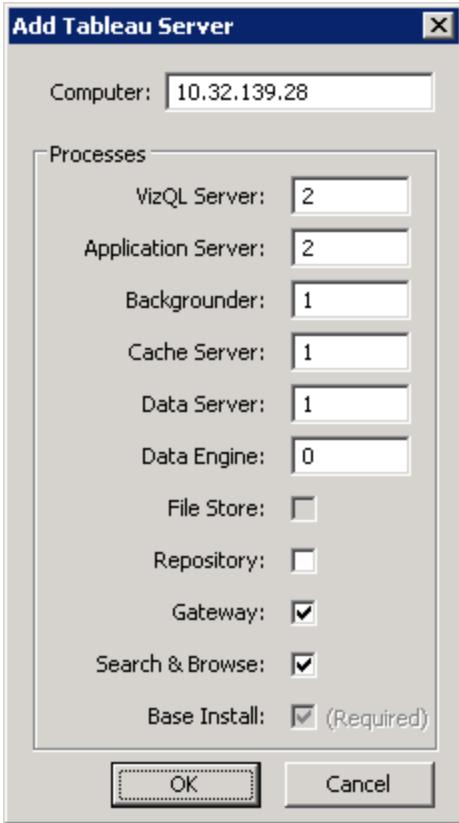


If you want the worker to run other server processes, enter the number of instances you want to run, such as 1 or 2. Click **OK** to close the Add Tableau Server dialog box and click **OK** to save the configuration and close the Configuration utility.

Note: If you have a total of two nodes, and each node has a repository or a data engine or both, a message appears to let you know that at least three server nodes are required for high availability (failover) support: "A minimum of three Tableau Server nodes are required for high availability. You can add a third node now, or continue with only two nodes."

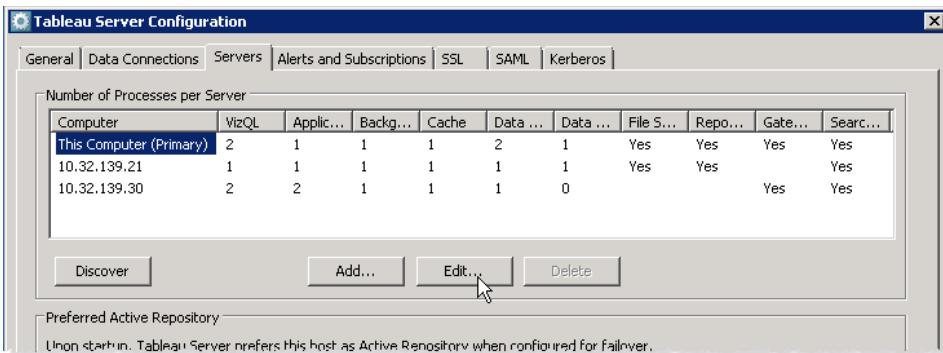


8. **Start the server** on the primary computer.
9. **Important:** Allow several minutes for the server's synchronization processes to copy data. This can take anywhere from 5 minutes to 15 minutes (or even much longer) depending on the size of your installation and the number of extracts.
10. Open the Status page in Tableau Server and check on the components you added:
 - If you added a data engine/file store, wait until the new file store status no longer says "Syncing".
 - If you added a repository, wait until the new repository status says "Passive".
11. After you've confirmed that the synchronization is complete, **stop the server** on the primary.
12. Open the Configuration utility. On the **General** tab enter the Run As account password, then click the **Servers** tab and click **Add** on the **Servers** tab to add another worker.
13. Enter the IPv4 address or computer name of the second worker, enter at least 1 for every process but the **Data Engine** (set that to 0). Clear the **Repository** check box and select **Gateway**.



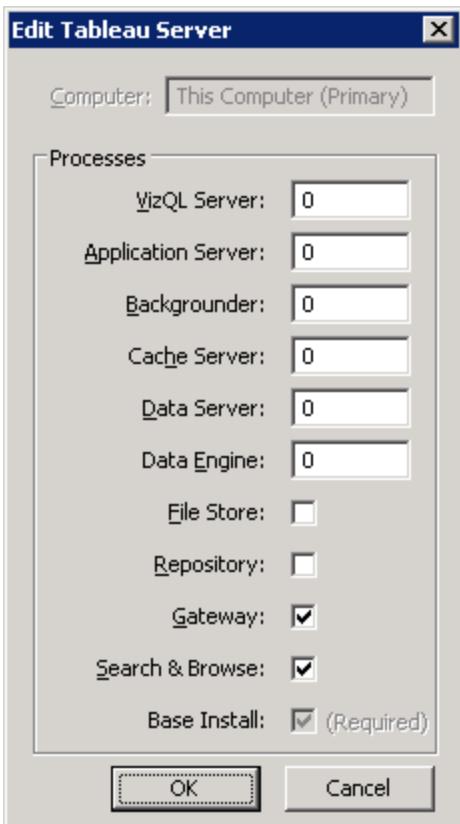
Click **OK**.

14. On the **Servers** tab, select **This Computer (Primary)**, and click **Edit**.



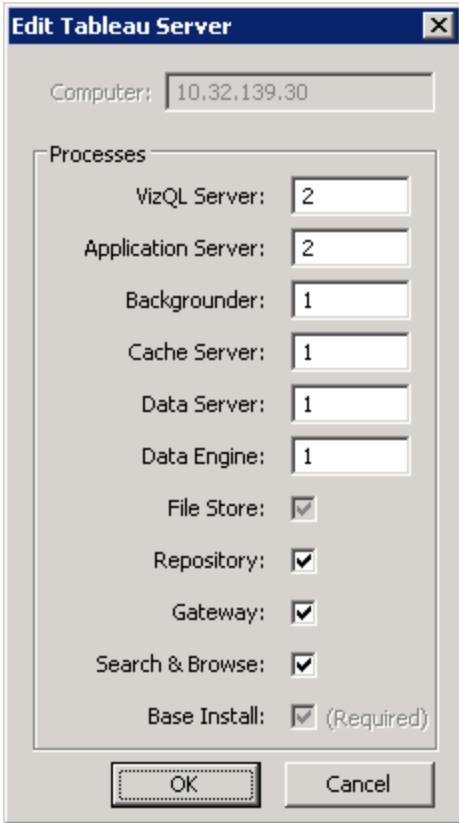
15. In the Edit Tableau Server dialog box, set **Data Engine** to 0 and clear the **Repository** check box. Keep **Gateway** selected. If you want the primary Tableau Server to run nothing but the gateway process (Apache), you can remove the remaining server processes from the primary by entering 0 in each text box.

With a core-based license, the gateway and search & browse processes consume no cores. Configuring the primary Tableau Server to run nothing but the gateway and search & browse is a useful strategy if, for example, you have an 8-core server license and two 4-core workers. You can run three servers (primary plus two workers), but only the worker servers are consuming cores.



Click **OK**.

16. On the **Servers** tab, select the first worker, click **Edit**, and select the **Gateway** check box. Leave the other settings unchanged. Click **OK**.
17. Still on the **Servers** tab, select the second worker and click **Edit**.
18. Set **Data Engine** to 1 (**File Store** will be automatically selected) and select the **Repository** check box.



19. Click **OK**.

The **Servers** tab should now look similar to this:

20. You can also set up email alerts so that you're notified of server failures or changes in status for your data engine and repository processes. To do this, click the **Alerts and Subscriptions** tab in the Configuration utility and follow the steps in [Configure Email Alerts on page 17](#).
21. Click **OK** to close the Configuration utility.

22. If you are removing a data engine/file store (step 14 above), a message appears to let you know that the file store was not decommissioned, and asking if you want to decommission it. Click **Yes** to decommission the file store.
23. **Start the server** on the primary computer (it may take a few minutes for your changes to take effect). Your system is now configured to provide failover support for the repository process. It is also configured for multiple gateways. You can now [use a load balancer](#) to ensure the cluster's availability in the event of a gateway failure—and to distribute the cluster's workload.

The Status page should look similar to this:

Server Status			
Process Status			
	Primary 10.32.139.21	Worker 1 10.32.139.22	Worker 2 10.32.139.30
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server	✓	✓	✓✓
API Server	✓	✓	✓
VizQL Server	✓	✓	✓✓
Cache Server	✓	✓	✓✓
Search & Browse	✓	✓	✓
Backgrounder	✓	✓	✓
Data Server	✓	✓	✓
Data Engine	✓	✓	✓
File Store	✓	✓	✓
Repository	✓	✓	✓

 ✓ Active
 ↻ Busy
 ✗ Passive
 ⚠ Unlicensed
 ✗ Down
 □ Status unavailable

Configure Tableau Server for High Availability with External Coordination Service Nodes

As a part of the Tableau Server installation, a Coordination Service process is installed on each server node. Coordination Service is a service built on Apache ZooKeeper, that coordinates activities on the server. If you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to install Tableau Server in a configuration that uses external Coordination Service nodes. This means installing Coordination Service on nodes that run no other server processes, and removing Coordination Service from the nodes that are running other server processes. This procedure explains how to do this.

To run Tableau Server with external Coordination Service nodes

1. Install Tableau Server on the primary computer (primary node).
2. Install Tableau Server worker software on additional computers.

You need at least three nodes to run Coordination Service, plus the nodes on which you want to run Tableau Server as part of your distributed installation. In the example below, a total of six nodes are used.

3. On the primary node, run the Configuration utility, and add the nodes on which you installed the worker software.
4. In the Configuration utility, edit each server that will run Tableau Server, and specify the processes that should be installed. For more information on how to configure a distributed installation, see [Install and Configure Worker Nodes on page 76](#).
5. In the Configuration utility, edit each server that will run only the Coordination Service process, and configure the node so it is not running any other Tableau Server processes. These nodes are considered "external" to the Tableau Server configuration and will only run the only Coordination Service.

Note: The **Base Install** process is required and installs Coordination Service. You cannot remove it.

6. Close the Configuration utility.
7. On the primary node, at the command line:
 - a. Configure 0 Coordination Service processes on the nodes that are running Tableau Server processes.
 - b. Update the configuration on all nodes.
 - c. Start Tableau Server.

Example

The following example shows how you would configure a three-node distributed installation of Tableau Server along with three external nodes running Coordination Service. If you want failover support in Tableau Server, you must run Coordination Service on a minimum of three nodes so there is a quorum.

Note: To install Tableau Server on multiple nodes, you must have a Tableau Server—Multi-Machine Core license.

1. **Install Tableau Server** on your primary computer.

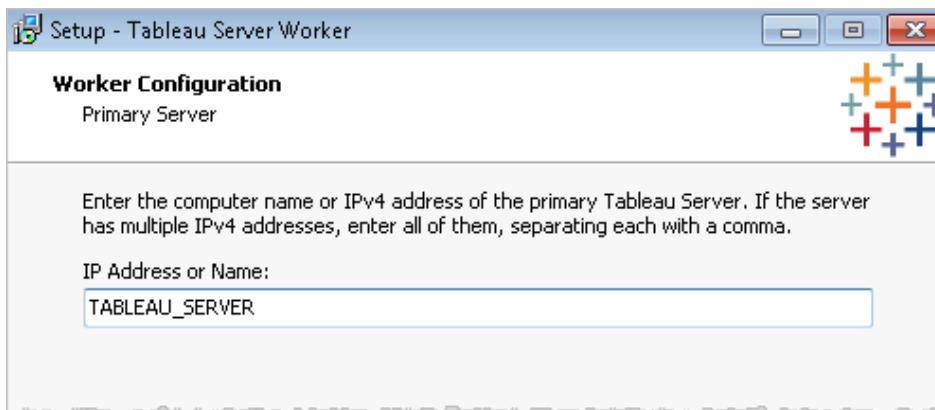
After Setup completes, check the Server Status page. All the processes should have a green “active” status.

Server Status	
Process Status	
The real-time status of processes running in Tableau Server.	
Process	10.32.139.21
Gateway	✓
Application Server	✓
API Server	✓
VizQL Server	✓ ✓
Cache Server	✓ ✓
Search & Browse	✓
Backgrounder	✓
Data Server	✓ ✓
Data Engine	✓
File Store	✓
Repository	✓

2. **Stop the server** on the primary computer.

3. Run **Tableau Worker Setup** on five additional computers or VMs. Two of these will be worker nodes in the installation, run Tableau Server processes, and provide failover support. The other three will run Coordination Service.

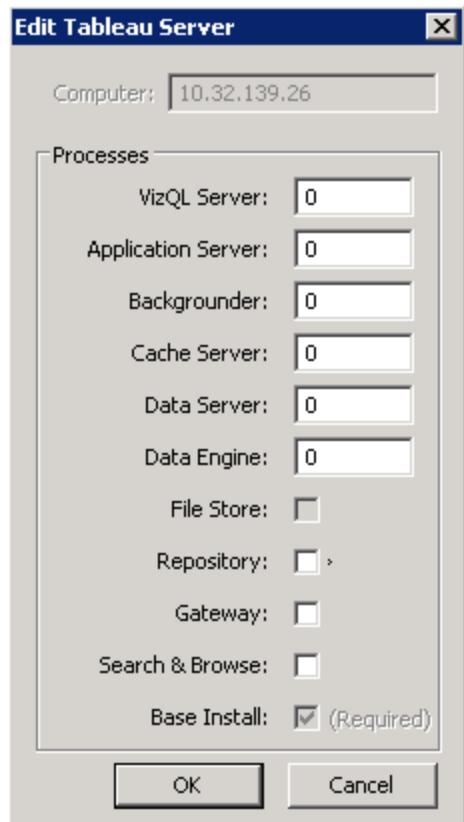
During worker setup, you will need to provide the computer name (recommended) or IPv4 addresses of the primary Tableau Server.



Note: The primary computer must have a static IP address assigned to it, even if you are using the primary's computer name to identify it. For more information, see [Hostname Support in Tableau Server](#) on page 75.

4. Start the Tableau Server Configuration utility: **Start > All Programs > Tableau Server > Configure Tableau Server**. On the **General** tab, enter the Run As account password.
5. On the **Servers** tab, click **Discover** to add the five worker nodes.
6. Select the first worker node, and then click **Edit**. Enter **1** for **Data Engine (File Store)** (File Store will be automatically selected), and then select the **Repository** check box.
Click **OK** to close the Edit Tableau Server dialog box.
7. Select the second worker node, and repeat step 6.
8. For each of the next three computers:
 - a. Select the computer from the **Servers** list, click **Edit**, and then enter **0** for every process. Clear the options for **Repository**, **Gateway**, and **Search & Browse**. **Base Install** will be selected and you cannot change this. These nodes will run only Coordination Service.

The configuration for each of these nodes should look like this:



- b. Click **OK** to close the Edit Tableau Server dialog box.
9. Click**OK** to close the Edit Tableau Server dialog box, and then click **OK** to save the configuration and close the Configuration utility.
10. At a command prompt on the primary computer, remove Coordination Service from the primary node and the two worker nodes that are running Tableau Server:

```
tabadmin set worker0.zookeeper.procs 0  
tabadmin set worker1.zookeeper.procs 0  
tabadmin set worker2.zookeeper.procs 0  
tabadmin config
```

Note: You can find the number of each node from the status page. The primary node is always **worker0**.

11. **Start the server** on the primary computer.

Add a Load Balancer

You can enhance the reliability of a Tableau Server cluster by running multiple gateways and configuring a load balancer to distribute requests across the gateways. Unlike the repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

If you plan to also create a backup primary and that computer will be running a gateway process, be sure to identify that gateway to your load balancer, along with all the other gateways.

Note: If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see [Configure Kerberos](#) on page 508.

Guidelines

Note the following as you configure your load balancer to work with Tableau Server:

- **Tested load balancers:** Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.
If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.
- **Tableau Server URL:** When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the primary Tableau Server.
- **X-Forwarded-For and X-Forwarded-Host headers:** The Tableau Server [User Activity](#) administrative view displays client IP addresses, among other information. For this view to display the IP addresses of clients instead of the cluster's load balancer, the **X-Forwarded-For** and **X-Forwarded-Host** headers may need to be explicitly enabled on the load balancer (some load balancers have it enabled by default, some do not).

Note: The **X-Forwarded-For** header must include the IP addresses of any proxy servers that are traversed between the client and the server.

- **Trusted host settings:** The computer running the load balancer must be identified to Tableau Server as a trusted host. See the procedure below for how to configure Tableau Server.
- **Proxy server configurations:** The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a proxy server. If your Tableau Server

cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. For more information, see [Configure Tableau to Work with a Proxy Server on page 449](#)

- **Persistence:** External load balancer configuration should not include any persistence or affinity unless Active Directory (NTLM) authentication is used. If you are using Active Directory authentication, then use cookie-based persistence for NTLM negotiation requests only.

Note: You can use persistence with Kerberos enabled.

Configure Tableau Server to Work with a Load Balancer

You can configure Tableau Server to work with a load balancer by performing the following steps.

1. [Stop the server](#).
2. In the Tableau Server bin directory, enter the following command, where `name` is the URL that will be used to reach Tableau Server through the load balancer:

```
tabadmin set gateway.public.host "name"
```

For example, if Tableau Server is reached by entering `tableau.example.com` in a browser address bar, enter this command:

```
tabadmin set gateway.public.host "tableau.example.com"
```

3. By default, Tableau assumes that the load balancer is listening on port 80 for external communications. To use a different port, enter the following command, where `port_number` is the port:

```
tabadmin set gateway.port "port_number"
```

4. Now, enter the following command, where `server` is the IPv4 address or computer name of the load balancer:

```
tabadmin set gateway.trusted "server"
```

The value for `server` can be a comma-separated list, for example:

```
tabadmin set gateway.trusted "10.32.139.45, 10.32.139.46,  
10.32.139.47"
```

or

```
tabadmin set gateway.trusted "proxy1, proxy2, proxy3"
```

5. In the next command, you will provide any alternate names for the load balancer, such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases.

These are the names a user might type in a browser. Separate each name with a comma:

```
tabadmin set gateway.trusted_hosts "name1, name2, name3"
```

For example:

```
tabadmin set gateway.trusted_hosts "lb.example.com, lb,  
ftp.example.com, www.example.com"
```

6. Run the `config` command:

```
tabadmin config
```

7. **Start the server** so the changes can take effect.

Use a Backup Primary

Before you follow the procedures in the topics below, follow the steps in [Configure for Failover and Multiple Gateways on page 94](#). After going through those steps, you have two worker servers that are providing failover support. Each server is also running a gateway, for which a load balancer can be configured. The primary Tableau Server is running a gateway process and licensing, which is not exposed or assignable as a process. Now that you have redundancy for the data engine, repository, and gateway, you need to build in redundancy for your primary Tableau Server. You do this by creating a backup of it. While the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

Keep in mind that licensing is checked every 8 hours. If the primary goes down and is not running any other processes, you have a window of up to 72 hours to bring the backup primary online. The actual time period depends on when the last licensing check took place, and on whether any processes restart during the licensing window. For example, if the first failed licensing check occurred 71 hours ago, you have 1 hour to bring the backup primary online. If the licensing check occurred 1 minute ago, you have up to 71 hours and 59 minutes. **Note:** Any processes that restart during that 72 hour window will not be licensed and therefore will not be able to respond to service requests. To see when the last licensing check occurred, look at the `checklicense_lic.log` file and other log files in the `ProgramData\Tableau\Tableau Server\data\tabsvc\logs\licensing` folder.

The first topic below describes how to create a backup of your primary. The second topic walks you through how to switch to your backup primary if your current primary fails.

Create a Backup Primary

Do the following to create a backup primary:

1. **Stop the server** on your primary Tableau Server.
2. On the primary, open a command prompt as an administrator and navigate to the

Tableau Server bin directory:

```
C:\Program Files\Tableau\Tableau Server\9.2\bin
```

3. **Version 8.1.3 and earlier:** Enter the following command, where <primary1> is the current primary's IPv4 address or computer name and <primary2> is the backup primary's IPv4 address or computer name:

```
tabadmin failoverprimary --primary <primary1> --secondary <primary2>
```

Version 8.1.4 and later: Enter the following command, using either the computer names for the current and backup primaries (recommended) or all the IPv4 addresses for the current and backup primaries. If you enter IPv4 addresses, separate each with a comma.

```
tabadmin failoverprimary --primary "primary1_name,primary2_name"
```

or

```
tabadmin failoverprimary --primary "primary1_IP,primary2_IP"
```

For example, if the computer name of the current primary is TABLEAU_SERVER and the computer name of the backup primary is TABLEAU_SERVER2, you would enter the following:

```
tabadmin failoverprimary --primary "TABLEAU_SERVER,TABLEAU_SERVER2"
```

Here's a command example that uses IPv4 addresses. This example assumes that your primary (primary1_IP) has a single IPv4 address of 10.32.139.22 and your backup primary (primary2_IP) has a single IPv4 address of 10.32.139.26:

```
tabadmin failoverprimary --primary "10.32.139.22,10.32.139.26"
```

If the primary and backup primary have multiple IPv4 addresses, enter them all. For example:

```
tabadmin failoverprimary --primary "10.32.139.22,10.32.139.23,10.32.139.26,10.32.139.27"
```

4. Next, copy the tabsvc.yml file on the primary node (located in ProgramData\Tableau\Tableau Server\config) and save that copy in a temporary location on your backup primary computer. You will need to use this file if you are switching to your backup primary.

Note: The tabsvc.yml file contains server configuration settings. It gets updated when you change your configuration settings in the Tableau Server Configuration utility or using tabadmin commands. If you make any configuration changes after making a copy of the tabsvc.yml file, you need to update the copy of tabsvc.yml on your backup primary to ensure you have the latest configurations if you need to failover.

5. On your backup primary, edit the tabsvc.yml file you copied to the backup primary and replace the IP address(es) or computer name for the primary with the IP address(es) or computer name for the backup primary (the computer you're currently on). If the primary is only running the gateway, as described in this procedure, the only line you'll need to edit is `worker.hosts`. If the primary is running additional processes, replace the primary's IP address(es) or name with the backup primary's anywhere it appears.

```

---  

worker0.vizqlserver.procs: 0  

worker0.vizportal.procs: 0  

worker0.backgrounder.procs: 0  

worker0.wgserver.procs: 0  

worker1.dataengine.procs: 1  

worker0.cacheserver.procs: 0  

jdbc.password: 687eb10fc559e55df86432dc25139f2c37bed60d  

worker2.vizportal.procs: 2  

worker0.dataserver.procs: 0  

worker0.dataengine.procs: 0  

worker2.wgserver.port: 8001  

pgsql.readonly_password: 26ca7cd7de1c664d83fba5476779c06f8b294366  

worker2.searchserver.procs: 1  

worker1.searchserver.procs: 1  

worker2.vizqlserver.procs: 2  

worker0.filestore.enabled: false  

worker1.vizqlserver.procs: 1  

worker1.vizportal.procs: 1  

worker1.wgserver.procs: 1  

worker1.dataserver.procs: 1  

worker1.backgrounder.procs: 1  

worker1.cacheserver.procs: 1  

pgsql.host: 10.32.139.21  

worker2.wgserver.procs: 1  

worker1.gateway.enabled: true  

pgsql.adminpassword: 5dd4321b734d419352257815ed0c6c946c90d61d  

worker1.filestore.enabled: true  

worker0.wgserver.port: 8001  

worker2.dataserver.procs: 1  

worker2.backgrounder.procs: 1  

worker2.cacheserver.procs: 1  

worker2.dataengine.procs: 1  

pgsql0.host: 10.32.139.21  

pgsql1.host: 10.32.139.30  

worker2.gateway.enabled: true  

worker1.wgserver.port: 8001  

worker2.filestore.enabled: true  

vizqlserver.initialsql.disabled: false  

worker.hosts: TABLEAU SERVER, 10.32.139.21, 10.32.139.30  

worker0.gateway.port: 8000  

worker1.gateway.port: 8000  

worker2.gateway.port: 8000  

pgsql.remote_password: 1c466e595670edf8b9e275ff825de10acf9d4902  

config.version: 13  

filestore.zookeeper.password: 64fe2ed73d5715743aedeead4ad8a237214d9872  

clustercontroller.zookeeper.password: f18b67ef08c16fa32911f48d7097f07573643b82  

pgsql.preferred_host: 10.32.139.30  

service.init.state: start

```

6. On your backup primary, install Tableau Server. Use the same Run As account and configuration settings that you used when you ran Tableau Server Setup on your primary.

Note: Installing Tableau Server will create a fresh tabsvc.yml file on the backup primary. If you need to fail over to the backup, you will replace this file with the copy you made and updated in Steps 4 and 5 above.

7. After Setup completes, **stop the server** on the backup primary.
8. Still on your backup primary, enter the following command to disable the automatic

starting of the Tableau Server service:

```
tabadmin autostart off
```

9. Type the following command to commit the configuration change:

```
tabadmin config
```

You've finished creating a backup primary. See [Switch to Backup Primary](#) for what to do if your current primary fails.

If you are working in a test environment, this would be a good time to test your configuration by powering down your current primary to simulate a system failure.

Switch to Backup Primary

If your primary node fails, and you have a backup primary configured, you can follow this set of steps to switch to your backup primary. All steps should be performed on the backup primary computer.

Note: After switching to the backup primary, you need to uninstall and reinstall Tableau Server on the original primary computer and configure it as the backup primary. See Step 7 below for more information.

1. On your backup primary, find the tabsvc.yml file you copied and edited in step 5 of [Create a Backup Primary on page 108](#). Copy this from your temporary location to ProgramData\Tableau\Tableau Server\config and replace the existing tabsvc.yml file on the backup primary. You need to do this so the backup primary has the same configuration settings as the primary did.
2. If web data connectors were imported to the primary server, copy them to the following folder on the backup primary:

```
C:\ProgramData\Tableau\Tableau  
Server\data\tabsvc\httpd\htdocs\webdataconnectors
```

Copy the web data connectors from the location from which they were imported to the primary server, or extract the contents of a Tableau Server .tsbak backup file and get them from there.

3. Open a command prompt as an administrator and navigate to the Tableau Server bin directory:

```
C:\Program Files\Tableau\Tableau Server\9.2\bin
```

4. **Version 8.1.3 and earlier:** Enter the following command, where primary2 is the IPv4 address or computer name of your backup primary (soon to be your new primary) and primary1 is the IPv4 address or computer name of your former primary (soon to be your backup):

```
tabadmin failoverprimary --primary <primary2> --secondary  
<primary1>
```

Version 8.1.4 and later: Enter the following command, using either the computer name of your backup primary (soon to be your new primary) or the IPv4 addresses of the backup primary (soon to be your new primary) and the primary (soon to be your backup primary). If you enter IPv4 addresses, separate each with a comma.

```
tabadmin failoverprimary --primary "primary2_name,primary1_  
name"
```

or

```
tabadmin failoverprimary --primary "primary2_IP,primary1_IP"
```

For example, if the computer name of the backup primary is TABLEAU_SERVER2 and the name of the former primary is TABLEAU_SERVER, you would enter the following:

```
tabadmin failoverprimary --primary "TABLEAU_SERVER2,TABLEAU_  
SERVER"
```

Here's an example that uses IPv4 addresses. This example assumes that your backup primary (primary2_IP) has a single IPv4 address of 10.32.139.26 and your former primary (primary1_IP) has a single IPv4 address of 10.32.139.22:

```
tabadmin failoverprimary --primary  
"10.32.139.26,10.32.139.22"
```

If the backup primary and former primary have multiple IPv4 addresses, enter them all. For example:

```
tabadmin failoverprimary --primary  
"10.32.139.26,10.32.139.27,10.32.139.22,10.32.139.23"
```

5. Enter the following command:

```
tabadmin autostart on
```

6. Type the following command to commit the configuration change:

```
tabadmin config
```

7. **Start the server.** Your backup primary is now your primary. When you look at the Status page, you should see that the IP address or computer name for the primary has changed:

Process Status			
The real-time status of processes running in Tableau Server.			
Process	Primary 10.32.139.26	Worker 1 10.32.139.21	Worker 2 10.32.139.30
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server		✓	✓✓

8. For your former primary to now act as your backup primary, you will need to do the following:

- Use Add/Remove Programs to remove Tableau Server from your former primary. At the end of the Uninstall program you will receive a backup error, which you can ignore.
- Delete the Tableau folders under \Program Files and \ProgramData on your former primary.
- Repeat the steps starting with step 4 in [Create a Backup Primary](#) on page 108.

Work with the Server

The following topics describe how to set up and administer Tableau Server.

Sites

A site in Tableau Server is a logical space that isolates content, data, and groups of users from other users on the same server. Tableau Server supports multitenancy by allowing server administrators to create multiple sites on the server for different sets of users and content.

Each site is separate on the server, and permissions can be set per user or group on a project, workbook, view, or data source. All server content is published, accessed, managed, and controlled on a per-site basis. Each site's workbooks, data, and user lists are isolated from those of other sites, and users can only access one site at a time.

Only server administrators can see every site and perform actions such as creating sites and making server-wide changes. Server administrators can create independent sites for various organizations or groups who will be accessing Tableau Server.

Site administrators (who are allowed by the server administrator to create site users) can control site membership. (For details on changing this setting, see step 4 in [Add or Edit Sites on page 117](#)).

After the server administrator creates sites, content owners can publish workbooks, views, and data sources to specific sites on the server.

Users can belong to multiple sites, with different site roles and permissions on each site. Users signing in to the server will see their allowed content in the sites they belong to.

Work with Sites

The topics below describe aspects of working with multiple sites such as which type of authentication is used, as well as things you should know about user licenses, and administrator roles.

Authentication and sign-in credentials

All sites on a server use the same Run As User account and user authentication mode. You choose both of these settings when you install Tableau Server. See [Configure General Server Options on page 12](#) for more information.

Users who belong to more than one site on the same server system use the same credentials for each site. For example, if Jane Smith has a user name of *jsmith* and a password of *MyPassword* on Site A, she uses those same credentials on Site B. When she signs in to Tableau Server, she'll be able to choose which site she wants to access.

The Default site

To help you transition smoothly from a single- to multi-site server system, Tableau Server installs with a site named Default. If you're running in single-site mode, you don't need to explicitly use Default, it happens automatically. However, if you add one or more sites, Default becomes one of the sites you can sign in to when you sign in to Tableau Server. Default differs from sites that you add to the system in the following ways:

- It can never be deleted but, just like sites that you add, it can be renamed.
- It stores the samples and data connections that ship with Tableau Server.
- The URL used for Default does not specify a site. For example, the URL for a view named Profits on a site named Sales is `http://localhost/#/site/sales/views/profits`. The URL for this same view on the Default site would be `http://localhost/#/views/profits`.

Site administrator and server administrator site roles

There are two types of administrators in Tableau Server, server administrators and site administrators. For each site, server administrators can control whether site administrators can add and remove users for the sites they manage (select **Site <name> > Settings**).

Managing Users

Who is allowed to add and remove users.

Only server administrators

Server and site administrators

Limit the number of users to:

Server limit

Custom limit

If **Only server administrators** is selected, site administrators cannot add or remove site users. However, they can still manage groups, projects, workbooks, and data connections within their site. If **Server and site administrators** is selected (the default), site administrators can do all of the above, and add or remove users.

Licensing and user limits

Users can belong to multiple sites, with different site roles and permissions on each site. A user who belongs to several sites, however, does not need a license for each site. Each server user only needs one license.

Server administrators can use the **Limit number of users** setting (select **Site <name> > Settings**) to specify a user limit for the site. Only licensed users are counted; server administrators are excluded. For example, if a site has 90 licensed users, 20 unlicensed users, and one server administrator, the user count is 90. If **Limit number of users** is set to **100**, 10 more licensed users can be added.

Add or Edit Sites

Server administrators can add sites to Tableau Server, or edit existing sites. Even before you add a site, Tableau Server will have a Default site.

1. Open the Sites page. If you are adding the first site on the server, select **Settings > Add a Site**, and then click **Add a Site**.

The screenshot shows the 'Add a Site' tab selected in the 'General' section of the 'Settings' menu. The main heading is 'Host Multiple Sites on Tableau Server'. It includes a sub-instruction: 'Create independent sites for different organizations on a single installation of Tableau Server.' Below that is a note: 'Each site's workbooks, data, and user lists are isolated from those of other sites. As the server administrator, you can manage every site and perform actions such as creating sites and making server-wide changes.' At the bottom is an orange 'Add a Site' button.

Otherwise, select **Server > Sites**, and then click **New Site**.

The screenshot shows the 'Sites' page in the 'Server' menu. It displays a list of eight sites: Customer Support, Default, Development, Documentation - 20 User L..., Finance, Human Resources, Operations, and Sales. Each site entry includes a checkbox, the site name, user count (***), site administrators (0), max users (e.g., 4, 20, 25.7 MB), and storage used (e.g., 0 B, 1.8 MB, 3.8 MB). A search bar is at the top left, and a 'New Site' button is at the top center.

Name	Users	Site administrators	Max users	Storage used
Customer Support	***	0	0	Server limit 0 B
Default	***	43	4	Server limit 25.7 MB
Development	***	0	0	Server limit 0 B
Documentation - 20 User L...	***	0	20	0 B
Finance	***	0	0	Server limit 1.8 MB
Human Resources	***	0	0	Server limit 0 B
Operations	***	0	0	Server limit 0 B
Sales	***	0	0	Server limit 3.8 MB

To edit a site, select **Server > Sites** (multi-site), or click **Settings** (single-site). Select the site you want to modify, and then select **Edit Settings**.

The screenshot shows the 'Sites' section of the administration interface. There are 8 sites listed:

Name	Users
Port	0
Suspend	0
Edit Settings	0
Delete	43
Development	0
Documentation - 20 User L...	0
Finance	0
Human Resources	0
Operations	0
Sales	0

A context menu is open over the 'Finance' site, with the 'Edit Settings' option highlighted.

2. Enter a **Site name** and **Site ID** for the site (if you are editing the Default site, you cannot change the **Site ID**):

Name	
Site Name:	<input type="text" value="Enter Site Name"/>
Site ID:	<input type="text" value="Enter Site ID"/>
URL: http://10.32.139.28/#	

Note The “#/site” in the URL (for example, <http://localhost/#/site/sales>) cannot be changed. In multi-site server systems, it appears in the URL for sites other than the **Default site**.

3. Workbooks, extracts, and data sources all consume storage space on the server. For **Storage**, select either **Server Limit** or **GB**, and enter the number of GB you want as a limit.

Storage

How much space is reserved for content published by users.

Server limit

GB

If you set a server limit and the site exceeds it, publishers will be prevented from uploading new content until the site is under the limit again. Server administrators can track where the site is relative to its limit using the **Max Storage** and **Storage Used** columns on the Sites page.

Name	Users	Site Admins	Max Users	Storage Used	Max Storage	Status
<input type="checkbox"/> Default	10	1	Server limit	12.9 MB	Server limit	Active
<input type="checkbox"/> MyFirstSite	0	0	Server limit	0 B	Server limit	Active
<input type="checkbox"/> MySecondSite	0	0	Server limit	0 B	Server limit	Active

4. Select whether only you, the server administrator, can add and remove users (**Only server administrators**) or if it can be done by both types of administrators (**Server and site administrators**).

Managing Users

Who is allowed to add and remove users.

Only server administrators

Server and site administrators

Limit the number of users to:

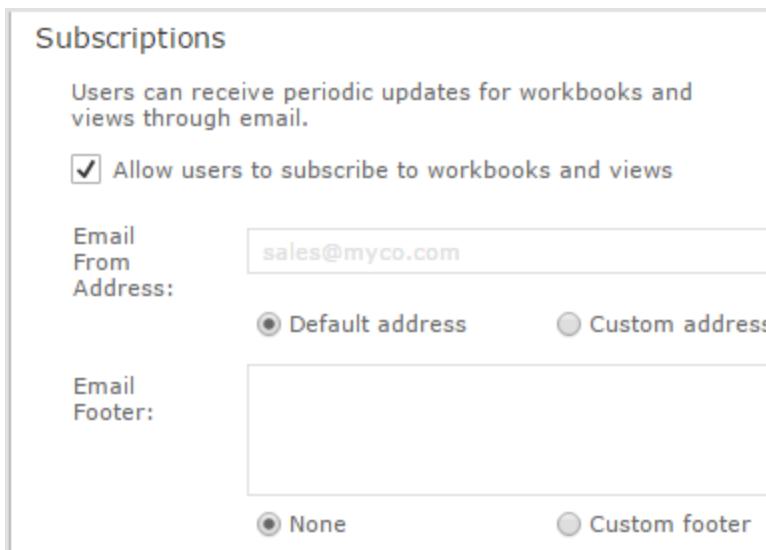
Server limit

users

If you are allowing site administrators to add users, specify how many users they can add to the site by selecting one of the following:

- **Server limit:** For a server with user-based licensing, the limit is the number of available server seat licenses. For a server with core-based licensing, there is no limit to the number of users that can be added. For more information, see [View Licenses](#) on page 255 and [Handle an Unlicensed Server](#) on page 654.

- <n> **users**: Allows a site administrator to add users up to a limit you specify. See [Work with Sites](#) for information on licensing and user limits.
5. Leave **Allow users to use web authoring** selected or clear it to disable authoring for content in the site (not server-wide).
- Disabling web authoring means that users cannot edit published workbooks from the server web environment. To update a workbook published to the server, a Tableau Desktop user must re-publish it. For more information, see [Disable Web Authoring](#) on page 383.
6. For **Subscriptions**, keep **Allow users to subscribe to workbooks and views** selected if you want site users to be able to subscribe to views. This option is only visible if you have also [configured subscription settings](#) in the Configuration dialog box.



You can also enter a custom **From address** for the subscriptions. While the address you enter should use valid email address syntax (such as `bizdev@myco.com` or `noreply@sales`), Tableau Server does not require it to correspond to a real email account (some SMTP servers may require it to be an actual address, however).

For **Email footer**, select **Custom footer** and enter the text you want to display above the Tableau Server URL in subscription footers.



7. Select **Allow recording of workbook performance metrics** to permit your site users to collect metrics on how workbooks perform, such as how quickly they load
In addition to having this check box selected for the site, to initiate recording, users must add a parameter to the workbook's URL. For more information, see [Create a Performance Recording](#) on page 443.
8. Click **New Site** or **Save**.

Note: As a server administrator, when you add your first site to Tableau Server, a **Server** menu is displayed, along with a **Site** menu. When **Server** is selected, the Users page displays the label **Server Users**, because it pertains to all users on the server. When **Site** is selected, the Users page displays the label **Site Users**. As a server administrator, you can add users to the server, or to individual sites. For more information, see [Users](#) on page 173 and [Sites](#) on page 115.

Add Users to a Site

Administrators can add users to sites in the following ways:

- By adding a local user account or a user account from Active Directory, as described in this topic. You can also add users by importing an Active Directory group. For details, see [Create a Group via Active Directory](#) on page 161.
- By importing a CSV file that contains user information. For details, see [Import Users](#) on page 194 and [CSV Import File Guidelines](#) on page 200.

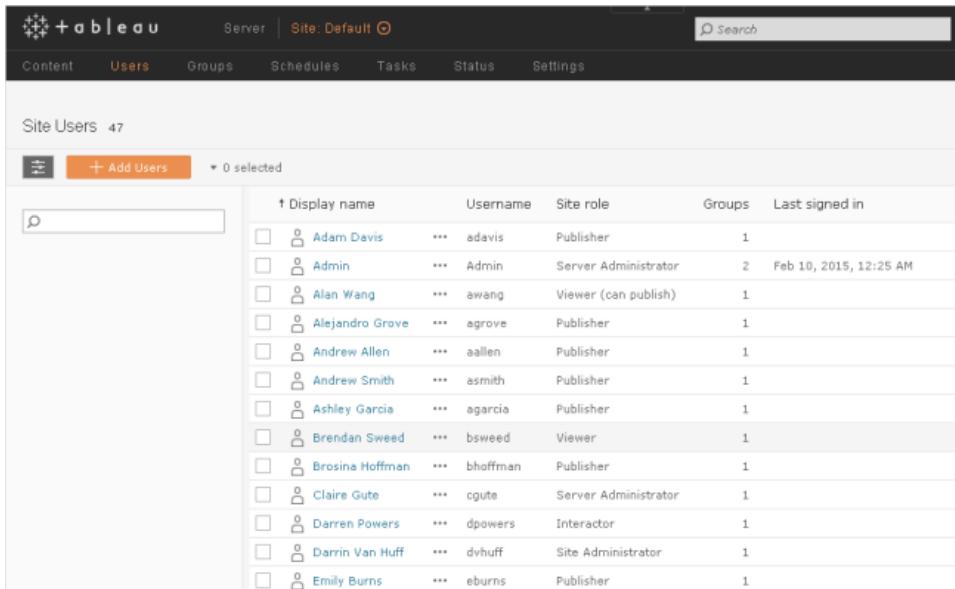
In a single-site environment, administrators can add users to a site on the Users page. In a multi-site environment, you will use the Site Users page. Server administrators must give site

administrators the ability to add users to sites. This setting can be enabled or disabled by the server administrator (see step 4 in [Add or Edit Sites on page 117](#)).

Note: Users can be added to sites, or to the server. To add users to the server, see [Add Users to the Server on page 188](#). The options available for adding users depends on the authentication method that you select when you first configure Tableau Server. If you are using local authentication, you cannot add Active Directory users. If you are using Active Directory, you cannot add local users.

On the **Users** (single-site) or **Site Users** (multi-site) page you can see the users on the site you're currently signed into. You can add users to (or remove them from) the current site only. If a user belongs to more than one site, you can remove that user from the current site.

Note: When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn't own any content.



The screenshot shows the Tableau Server interface with the 'Site: Default' selected. The top navigation bar includes 'Content', 'Users' (which is highlighted in orange), 'Groups', 'Schedules', 'Tasks', 'Status', and 'Settings'. A search bar is at the top right. Below the navigation is a header 'Site Users 47'. A toolbar below the header includes a refresh icon, a 'Add Users' button (which is orange), and a dropdown menu showing '0 selected'. A search input field is also present. The main content area is a table with the following columns: 'Display name', 'Username', 'Site role', 'Groups', and 'Last signed in'. The table lists 17 users, each with a checkbox and a small profile icon. The users are: Adam Davis (adavis, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM), Admin (Admin, Server Administrator, 2 groups, last signed in Feb 10, 2015, 12:25 AM), Alan Wang (awang, Viewer (can publish), 1 group, last signed in Feb 10, 2015, 12:25 AM), Alejandro Grove (agrove, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM), Andrew Allen (aallen, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM), Andrew Smith (asmith, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM), Ashley Garcia (agarcia, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM), Brendan Sween (bsweed, Viewer, 1 group, last signed in Feb 10, 2015, 12:25 AM), Brosins Hoffman (bhoffman, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM), Claire Gute (cgute, Server Administrator, 1 group, last signed in Feb 10, 2015, 12:25 AM), Darren Powers (dpowers, Interactor, 1 group, last signed in Feb 10, 2015, 12:25 AM), Darrin Van Huff (dvhuff, Site Administrator, 1 group, last signed in Feb 10, 2015, 12:25 AM), and Emily Burns (eburns, Publisher, 1 group, last signed in Feb 10, 2015, 12:25 AM).

Note: This screenshot is from a multi-site environment. In a single-site environment, this would be the Users page.

To add local users to a site

1. In a site, click **Users**, click **Add Users**, and then click **Local User**.

The screenshot shows the 'Site Users' page with 18 users. At the top, there's a navigation bar with tabs: Content, Users (which is selected and highlighted in orange), Groups, Schedules, Tasks, Status, and Settings. Below the navigation bar is a toolbar with a 'New Site' icon, a search bar, and a 'Add Users' button, which is also highlighted with a red box. A dropdown menu next to it shows '0 selected'. A modal window titled 'Add Users to Site' is open. It contains two buttons: 'Local User' (highlighted with a red box) and 'Import From File'. A descriptive text says 'Create a local user account on Tableau Server.' To the right of the buttons, there's a note: 'Import users from a CSV file.' Below the buttons is a list of existing users:

User	Role
Henry Wilson	hwilson
Interactor	Interactor
Jane Johnson	jjohnson
Laura Rodriguez	lrodriquez
Michelle Kim	Publisher Interactor

Note: This screenshot is from a multi-site environment. In a single-site environment, this would be the Users page.

2. Enter a user name. If the server is configured for local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, `jsmith@example.com` instead of `jsmith`).

Add Local User

Username: Username available

Display name:

Password:

Confirm password:

Email (optional):

Site role: ①

Also enter information in the following fields:

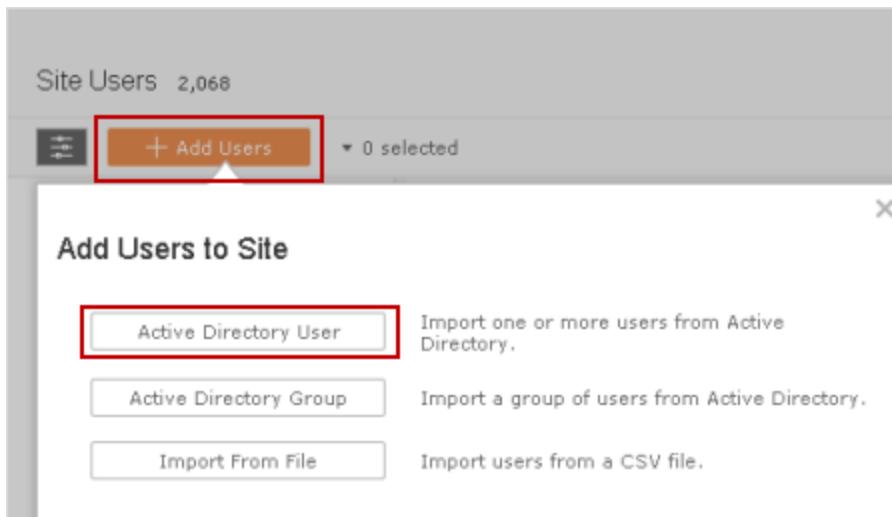
- **Display Name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.

3. Select a site role. For details on site roles, see [Site Roles for Users](#) on page 176.
4. Click **Add User**.

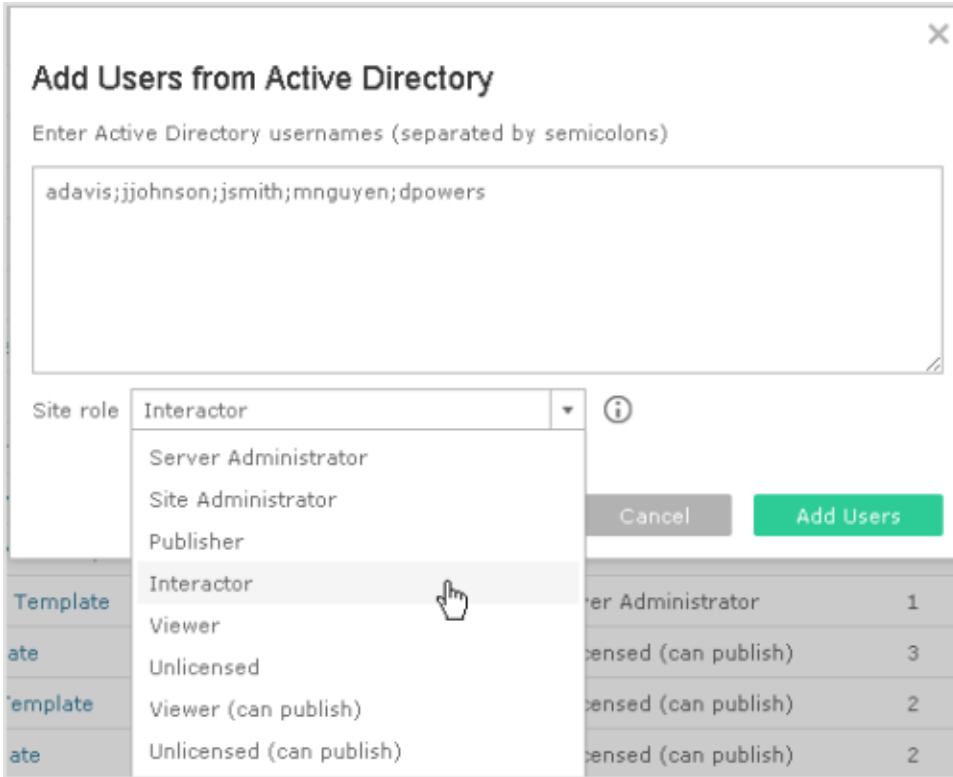
Note for multi-site servers: A site administrator can edit an existing local user account only if the administrator has control over all of the sites the user is a member of. For example, if User1 is a member of sites A and B, an administrator of site B only cannot edit User1's full name or reset the password.

To add Active Directory users to a site

1. In a site, click **Users**, and then click **Add Users**, and then click **Active Directory User**.



1. Enter one or more user names (separated by semicolons). If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server's domain will be assumed.



If there is a two-way trust set up between the server's domain and another domain, you can add users from both domains. The first time you add a user from the "non-server domain," use the fully-qualified domain name with the username. Subsequent users can be added using [the domain's nickname](#). For example, assuming a "non-server domain" of *mybiz.lan*, enter the first user from that domain as *user1@mybiz.lan* or *mybiz.lan\user1*. The next user can be entered using the domain's nickname, such as *user2@mybiz* or *mybiz\user2*.

Note: Do not enter the user's full name in this field; it can cause errors during the importing process.

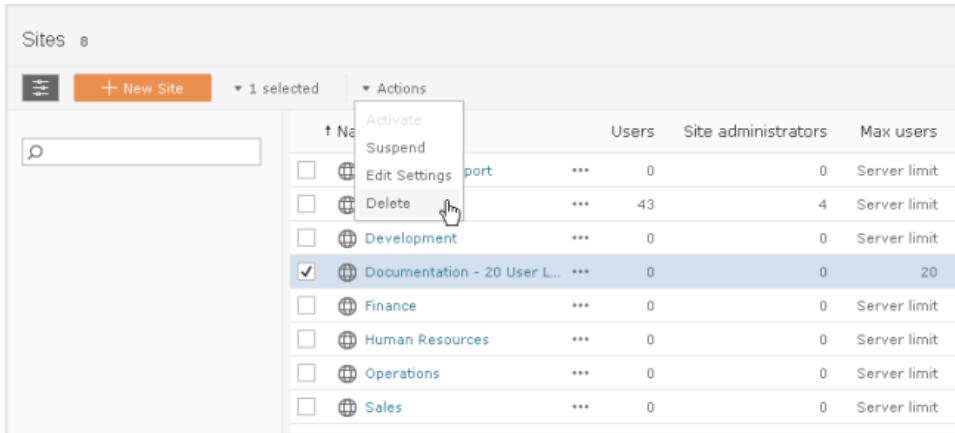
2. Select a site role. For details on site roles, see [Site Roles for Users](#) on page 176.
3. Click **Add Users**.

Delete Sites

Server administrators can delete sites that have been added to Tableau Server. Deleting a site also removes workbooks and data sources that were published to the site, as well as users. If a user belongs to additional sites, they will not be removed. To permanently delete a user, go to the Server Users page.

Note: The Default site cannot be deleted.

1. Click **Server > Sites**.



Name	Users	Site administrators	Max users
port	***	0	0 Server limit
Development	***	43	4 Server limit
Documentation - 20 User L...	***	0	0 20
Finance	***	0	0 Server limit
Human Resources	***	0	0 Server limit
Operations	***	0	0 Server limit
Sales	***	0	0 Server limit

2. Select the site you want to remove, and click **Delete**.
3. Click **Delete** in the confirmation dialog box that appears.

Import or Export a Site

You can provision a new Tableau Server site by exporting an existing site to a file and importing the file into a new site. The site you export is called the *source site*. The site into which you import is called the *target site*.

The source site can come from Tableau Online, which is a cloud-based installation of Tableau Server that is hosted by Tableau, or it can come from a Tableau Server deployment that you administer. When you import a site, all of the source site's resources—including workbooks, projects, data sources, users—come with it. The import also includes any permissions, subscriptions, or user favorites lists that have been created. All site-specific settings from the source site (including site quota, subscription and web authoring settings) are preserved in the target site.

Before you export

Before you export a site, note the following:

Delete unused items. Make sure the source site contains only what you want to import. Delete any unused workbooks, projects or data sources.

Remove unused users. Confirm that all users are licensed and remove any who no longer represent actual users. Any user you export from the source site must be imported to the target site. You can't remove users during the import.

Create user accounts on the target server. The site import process assigns users to a target site. The users must already have user accounts on the target server. If you are exporting one site into another on the same Tableau Server, you will have all the user accounts

you need. If you are exporting a site from Tableau Online or from a different Tableau server, you must create user accounts on the target server before you can perform the import.

Check user authentication. User authentication is a server-wide setting and all sites on a server must use the same setting. You can export from and import to servers that are using different user authentication methods, but you will need to modify the mapping files used for the import. This step is built into the import process and described in [Verify the site mappings on page 131](#). Because Tableau Online sites use a custom user authentication method, exporting from a Tableau Online site requires edits to the user-specific mapping files. This ensures a clean import, regardless of how the target server is configured.

Check schedules. The Schedules page on Tableau Server lists the default schedules you can use for extract refreshes and schedules:

Schedules 5						
		+ New Schedule ▾ 0 selected				
Name	Frequency	Task type	Tasks	Execution	Next run at	
<input type="checkbox"/> End of the month	***	Monthly	Extract Refresh	0 Parallel	Feb 28, 2015, 11:00 PM	
<input type="checkbox"/> Monday morning	***	Weekly	Subscription	Parallel	Feb 16, 2015, 6:00 AM	
<input type="checkbox"/> Saturday night	***	Weekly	Extract Refresh	0 Parallel	Feb 14, 2015, 11:00 PM	
<input type="checkbox"/> Weekday early mornings	***	Weekly	Extract Refresh	0 Parallel	Feb 10, 2015, 4:00 AM	
<input type="checkbox"/> Weekday mornings	***	Weekly	Subscription	Parallel	Feb 10, 2015, 6:00 AM	

Refreshes and subscriptions assigned to default schedules on the source site will be automatically mapped to the same schedules on the target site. If the source site has custom schedules, they are imported to the target site and can optionally be renamed when you edit the mapping files.

Configure the target server to deliver subscriptions. Subscriptions will be imported to the new site, but you must configure the target server to deliver the subscriptions, if it isn't already configured. For more information, see [Configure Alerts and Subscriptions on page 16](#).

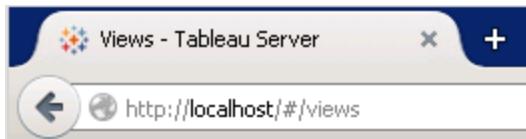
Create or identify the target site. Before you can import a site file, you must already have a target site on Tableau Server. Anything that exists in the target site that does not also exist in the source site will be removed during the import. Because of this, an empty site is recommended. For more information about creating or making changes to sites, see [Add or Edit Sites](#).

Note: If the target site is not empty, workbooks and data sources with identical names on both target and source sites will be replaced by workbooks, data sources, and permissions from the source site, and can be verified by the timestamp.

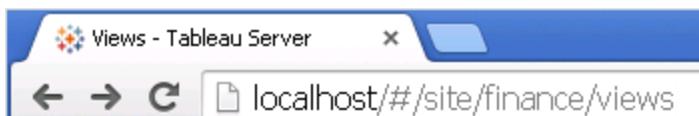
Locate Site IDs. The commands you use to export or import a site require a site ID as a parameter. A site ID uniquely identifies a site to Tableau Server. When you are signed in to a

site, the site ID is displayed after the # in the URL.

If the server isn't running multiple sites, the web browser URL displayed will include #, but not site or the **site ID**. If you see this, you are using Tableau's built-in site, which is named Default.



If the server is running multiple sites, the web browser URL will displayed will include **#/site** followed by the **site ID** for your site.



Tips for importing to a target with fewer users or schedules than the source site

When you import a source site to a target site that has fewer users or schedules than the source site, many-to-one importing is not supported. Consider the following options:

- Remove the extra users or schedules from the target site prior to exporting (preferred option).
- Add the missing users or schedules to the target site before beginning the import.
- Add the missing users or schedules to the target site in the middle of the import process and manually update the mapping files.
- Manually map the users or schedules to different users and schedules in the target site during the import process. This option is required if a user name differs between servers—for example, the exported user named *adavis* is defined on the target site as *davisa*.

Export a Site

You don't need to stop Tableau Server during the export or import process.

1. Open a command prompt as an administrator and navigate to the bin directory on Tableau Server. For example:

```
C:\Program Files\Tableau\Tableau Server\9.2\bin
```

2. Type the following command:

```
tabadmin exportsite <site ID> --file <filename or path>.
```

For example, to export a site with site ID **wsales** to the following file C:\sites\exported_sites\sales_export.zip, type the following:

```
tabadmin exportsite wsales --file C:\sites\exported_sites\--sales_export.zip
```

For examples of other options you can use with the `exportsite` command, see [exportsite on page 597](#).

During the export, Tableau Server locks the site.

Import a Site

If you don't already have a target site for the import, create one. See [Add or Edit Sites](#) for steps.

Importing a site is a three-step process. First, run the `tabadmin importsite` command to generate the files that will be imported. Next, verify files that show how the site will be imported. Finally, run the `tabadmin importsite_verified` command to finish the import.

Before you begin, you will need the exported site file and the site ID for the target site. The site ID for the Tableau Server default site is "" (double quotation marks, no space). If you are running commands within Windows PowerShell, delimit the Default site double quotes with single quotes (' " " ').

While there's no need to stop Tableau Server during the import process, the site receiving the import will be locked until the import completes.

Start the site import process

1. Open a command prompt as an administrator and navigate to the bin directory on Tableau Server. For example:

```
C:\Program Files\Tableau\Tableau Server\9.2\bin
```

2. Type the following command:

```
tabadmin importsite <site ID> --file <filename or path>
```

where `<site ID>` is the site ID of the target site and `<filename or path>` is the full path to the exported site file.

For example, to import the file C:\sites\exported_sites\sales_export.zip into a site with the site ID **esales**, type the following:

```
tabadmin importsite esales --file C:\sites\exported_sites\--sales_export.zip
```

For examples of other options you can use with the `importsite` command, see [importsite on page 600](#).

3. After you enter the command, the mapping files for you to verify are placed in

ProgramData\Tableau\Tableau Server\data\tabsvc\temp\import_<site ID>_<datetime>\mappings. Note this location for the next procedure.

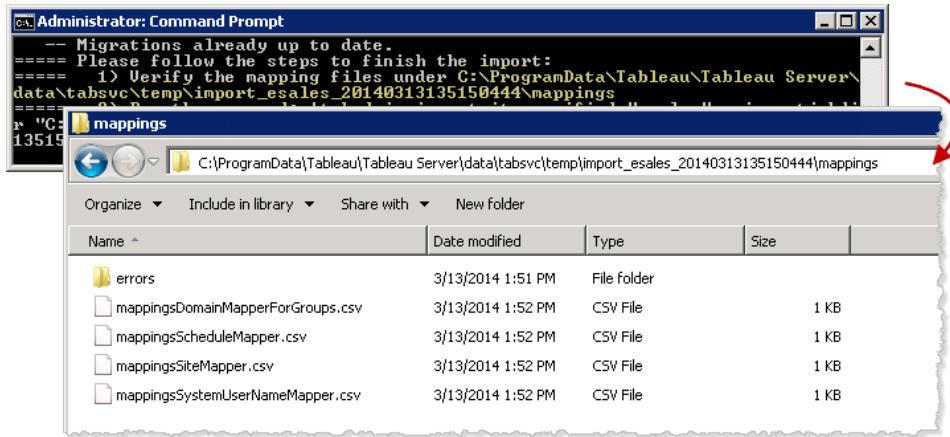
Verify the site mappings

The mapping files that are generated after you initiate a site import with the `importsite` command show you how the site's resources will be assigned once the import is complete. Items that Tableau Server was unable to map, and which need editing, are marked in the CSV files with question marks (???). Before you can run the final `importsite_verified` command you must change the question marks so that they represent valid assignments on the target site.

Note: You can't add or remove users as part of your changes. All user names for the users that you import must already exist on the target server.

To verify a site's mapping files:

1. Navigate to the directory that was displayed after you entered the `importsite` command:



2. Using Microsoft Excel (recommended) or a text editor, open each CSV file in the mappings folder.

Each file shows how items from the source site will be mapped, or handled, once the import to the target site is complete.

3. Verify that the mappings are correct. Replace any entry consisting of question marks (???) with a valid value. Use this table as a guide:

CSV file name	Column title	Can it be edited	Description
---------------	--------------	------------------	-------------

d?

map-pingsDomainMapperForGroups	source_name	No	A user group name on the source site.
	source_domain_name	No	The user authentication type on the source site: either local (for Local Authentication) or a domain name (for Active Directory).
	target_domain_name	Yes*	The user authentication type on the source site: either local for Local Authentication, or a domain name (such as example.com or example.lan) for Active Directory.
<p>*Do not edit the target_domain_name value for All Users. Keep its value of local, even if your target server is configured for Active Directory user authentication. The All Users group is a special default user group that must exist on every Tableau Server.</p>			
mappingsScheduleMapper	source_name	No	The names of custom and default extract or subscription schedules on the source site.
	source_scheduled_action_type	No	The type of schedule, either Extract , for extract refreshes, or Subscription , for subscription deliveries on the source site.

	target_name	Yes	The names of custom schedules on the target site. You can edit this value. For example, if the schedule is named Friday Update on the source site you can rename it Friday Refresh on the target site.
	target_scheduled_action_type	No*	The type of schedule, either Extract , for extract refreshes, or Subscription , for subscription deliveries on the target site.
<small>*In rare cases, there may be question marks (???) in this column. If there are, replace them with either Extract or Subscription, matching the entry you see under source_scheduled_action_type.</small>			
mappingsSiteMapper	source_url_namespac-e	No	The site ID of the source site.
	target_url_namespac-e	No	The site ID of the target site.
map-pingsSystemUserNameMapper	source_name	No	The username of a user on the source site.
	source_domain_name	No	The user authentication type on the source site: either local , for Local Authentication, a domain name (such as example.com or example.lan) for Active

		Directory, or external (for a Tableau Online site).
target_name	Yes	Usernames for users who will be assigned to the target site upon import.
		Confirm that all the usernames listed exist on the target server system and replace any question marks (???) with a valid username from the target server.
		You can't create usernames by adding rows to the CSV file. Similarly, you can't remove usernames by deleting rows.
		You can edit a username in the target_name column to be different from its source username as long as it already exists on the target server system using that different name. For example, a user can have a source_name value of jsmith@myco.com and a target_name value of johnsmith@example.co m as long as the username johnsmith@example.co m exists on the target server.
		You can't map a user on the source site to more than one username on the

		target site.
target_	Yes	The user authentication type on the target site: either local , for Local Authentication, or a domain name (such as example.com or example.lan) for Active Directory.
domain_		
name		

4. If you make edits, save your changes and preserve the CSV files' formatting. Leave the mapping files in their current location.

Complete the site import

1. Open a command prompt as an administrator and navigate to the bin directory on Tableau Server. For example:

```
C:\Program Files\Tableau\Tableau Server\9.2\bin
```

2. Type the following command:

```
tabadmin importsite_verified <site ID> --importjobdir <PATH>
where <site ID> is the site ID of the target site and <PATH> is the directory that's one level up from the mappings directory you used in Verify the site mappings on page 131. For example:
```

```
tabadmin importsite_verified esales --importjobdir
C:\ProgramData\Tableau\Tableau
Server\data\tabsvc\temp\import_esales_20140409185810071
```

For examples of other options you can use with the `importsite_verified` command, see [importsite_verified](#) on page 602.

3. Open the new site that you just imported and confirm that everything came in as expected.

Site Availability

A site can become suspended or locked due to a site import failure, or because a server administrator chooses to suspend the site for a period of time.

When a site is suspended, the only server user who can access it is the server administrator. Only the server administrator can activate the site to make it available again.

To activate or suspend a site

1. Click **Server > Sites**.
2. Select the site, and then select **Actions > Activate or Suspend**.

The screenshot shows the 'Sites' page in Tableau Server with 8 sites listed. A specific site, 'Sales', has a checkmark in its selection box and is highlighted with a red box. A context menu is open over this site, also enclosed in a red box. The menu items are: 'Activate' (with a cursor icon pointing at it), 'Suspend', 'Edit Settings', and 'Delete'. The 'Activate' option is the top item in the list.

Name	Users
Development	0
Documentation - 20 User Li...	0
Finance	1
Human Resources	7
Operations	0
Sales	0

Projects

As an administrator, you can create *projects* to collect and organize related content. *Content* in Tableau Server refers to workbooks, views, and data sources, and the projects that contain them.

You access projects from the Content page in Tableau Server.

As an administrator, you can do the following for projects:

- Create projects
- Rename projects
- Change project owners
- Set permissions for projects and their content
- Lock content permissions

Note: Only administrators can create and own projects.

Project Leader

Users who have the **Project Leader** permission in a project can:

- Control who has access to project content by setting permissions for project content at any time, even when content permissions are locked to the project
- Lock content permissions to the project
- Move workbooks between projects

Default project

Tableau creates every site with a **Default** project. The Default project serves as a template for new projects in that site. It defines the default settings and permissions that are applied to new projects and to the workbooks and data sources within those projects.

When you create a new project, the new project uses a copy of the Default project permissions.

The screenshot shows the Tableau Content page. At the top, there are navigation links: Content, Users, Groups, Schedules, Tasks, and Status. Below these are summary counts: Projects 10, Workbooks 14, Views 62, and Data Sources 2. A search bar and user profile are also present. The main area displays four projects: 'Default' (selected), 'Documentation', 'Finance', and 'General Purpose'. Each project card includes a folder icon, a description, and counts for Workbooks, Views, and Data Sources. The 'Default' project card is highlighted with a red border.

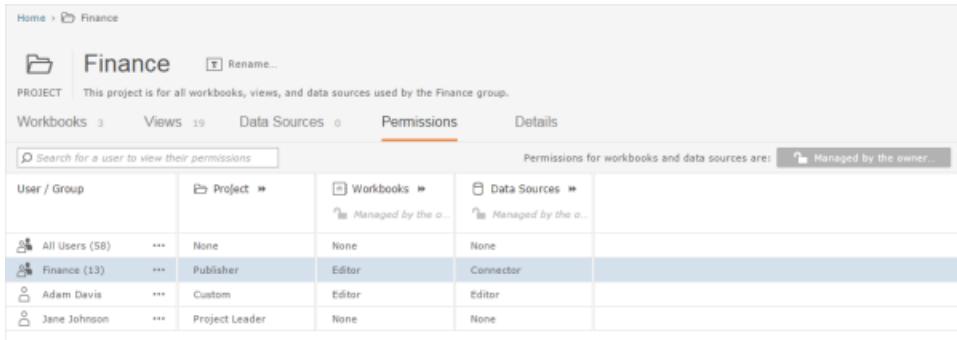
Project	Description	Workbooks	Views	Data Sources
Default	The default project that was automatically created by Tableau.	7	31	2
Documentation		3	7	0
Finance	This project is for all workbooks, views, and data sources used by the Finance group.	3	19	0
General Purpose		1	5	0

Default permissions

As an administrator or project leader, you can set permissions for every project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project.

Each project can have its own set of default permissions.

For more information, see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.



The screenshot shows the 'Permissions' tab of the 'Finance' project settings. It displays a table of users and their permissions across three categories: Project, Workbooks, and Data Sources. The 'Project' column shows 'Managed by the owner'. The 'Workbooks' and 'Data Sources' columns show 'Managed by the owner'. The table includes rows for 'All Users (58)', 'Finance (13)', 'Adam Davis', and 'Jane Johnson'.

User / Group	Project	Workbooks	Data Sources
All Users (58)	Managed by the owner	Managed by the owner	Managed by the owner
Finance (13)	Publisher	Editor	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

Only administrators and Project Leaders can edit the default permissions for a project and its workbooks and data sources.

For information on using projects to control permissions for content, see [Create Project-Based Permissions](#) on page 380.

Project content permissions

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.

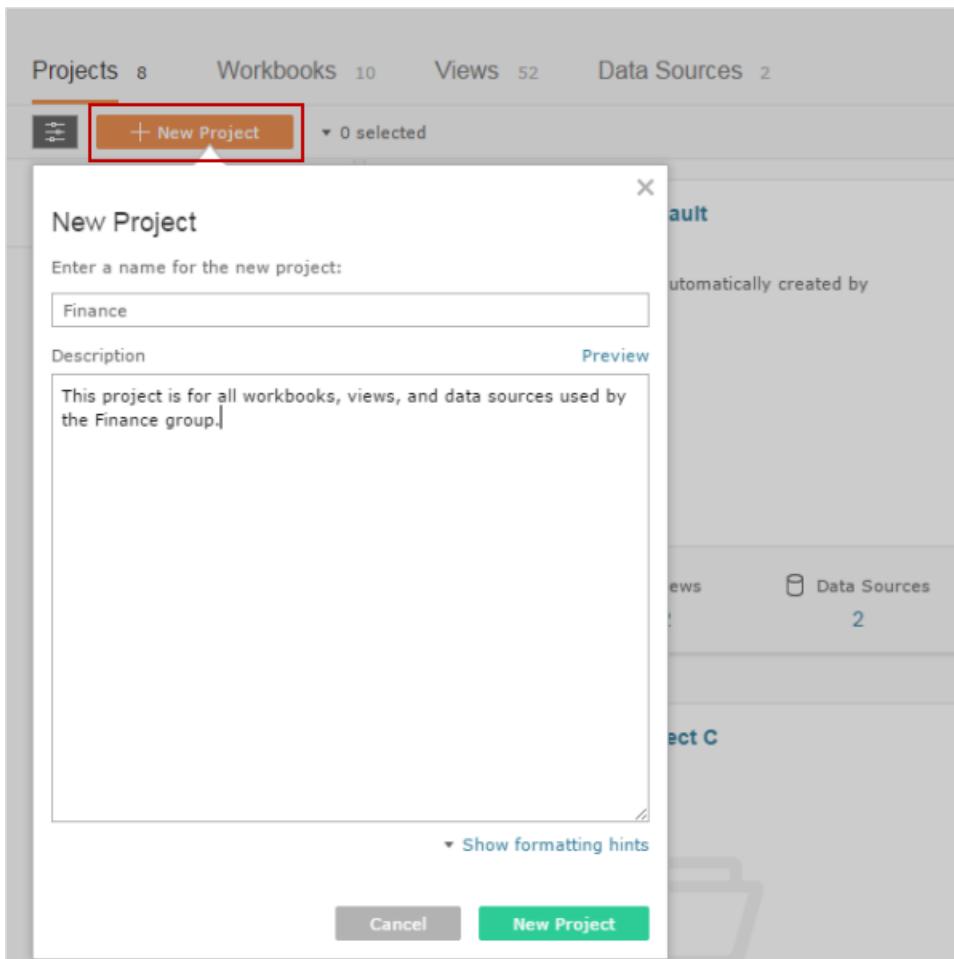
When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

Note: If a workbook or data source with editable permissions is moved to a locked project, the default permissions in the locked project are applied to the moved content and its permissions will then be locked.

For more information, see [Lock Content Permissions to the Project](#) on page 369.

Add Projects

1. On a Content page, click **Projects**, and then click **New Project**.



2. Enter a name and description for the project, and then click **New Project**.

You can include formatting and hyperlinks in the project description. Click **Show formatting hints** for syntax.

To edit a project, click the Project to open it, click **Details**, and then click **Edit Description**.

Move Workbooks into Projects

All workbooks must be in a project. By default, workbooks are added to the **Default** project. After you create your own projects, you can move workbooks from one project to another. You can move workbooks into projects if you are an administrator, or if you have the site role of Publisher or Interactor and at least one of the following is true:

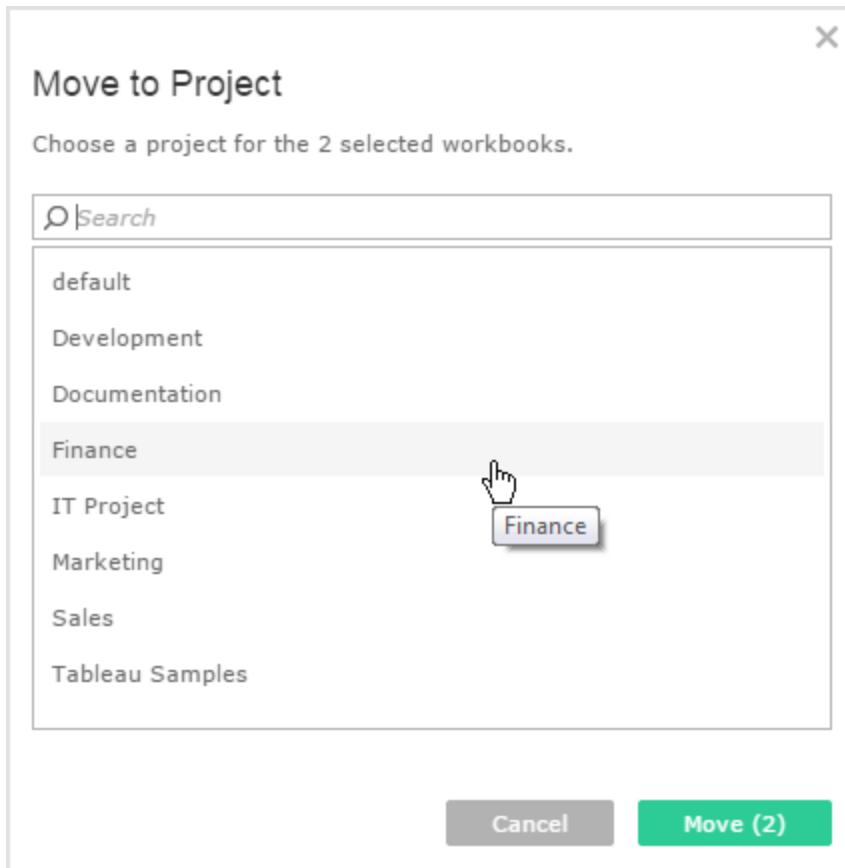
- You have been given the Move permission for the project.
- You have been given Project Leader permission for the project.

To move a workbook into a project

1. In the Workbooks page, select one or more workbooks, and then select **Actions > Move**.

The screenshot shows the Workbooks page with two workbooks selected. A context menu is open over the second workbook, with the 'Move' option highlighted. The background shows two tabs: 'Finance' and 'Sales'. The Sales tab displays a bar chart titled 'Commission Model' showing estimated results with various assumptions.

2. Select a different project for the workbook, and then click **Move**.



Because all workbooks must be part of a project, you can remove a workbook from a project by moving it to the Default project. Each workbook can only be contained in a single project.

Delete Projects

Only administrators can delete projects. When you delete a project, all of the workbooks and views that are part of the project are also deleted from the server.

1. In the Projects page, select a project, and then select **Actions > Delete**.

The screenshot shows the Tableau Content page with the 'Projects' tab selected. There are 8 projects listed. A context menu is open over the second project, which has a checked checkbox. The menu includes 'Permissions', 'Rename', 'Change Owner', and 'Delete'. The 'Delete' option is highlighted with a cursor. A tooltip message 'The Default project cannot be deleted.' is displayed.

Name	Workbooks	Views
Documentation	0	0
Finance	0	0
IT Project	1	5
Marketing	0	0
Sales	0	0
Tableau Samples	5	29

2. Click **Delete** in the confirmation dialog box.

The **Default** project cannot be deleted.

Set Permissions for a Project

Every project includes permissions that can be set for the project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project, and each project can have its own set of default permissions. For more information, see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

Administrators and users with the Project Leader permission can lock content permissions to a project. For more information, see [Quick Start: Lock Project Permissions](#), [Lock Content Permissions to the Project](#) on page 369.

For more information on working with permissions, see [Manage Permissions](#) on page 336 and [Projects](#) on page 136.

Note: When you create a new project, it initially will have the same permissions as the **Default** project in the site, which are the default permissions for the project, and its workbooks and data sources.

Permissions

Edit permissions for the project "Default".

Search for a user to view their permissions

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None	View Save Project Leader	Managed by the owner Managed by the owner	None
Finance (13)	Publisher	✓ ✓	Custom	Connector
Adam Davis	Publisher	✓ ✓	Editor	Editor
Jane Johnson	Project Leader	✓	None	None

The three capabilities you can set specifically for a project are: **View**, **Save**, and **Project Leader**.

Capability	Description
 View	<p>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</p>
 Save	<p>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project.</p> <p>When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining all permissions.</p> <p>Subsequently, the original owner's access to the workbook is determined by that user's group permissions and any further permissions the new owner might set.</p> <p>This permission also determines the user's or group's ability to overwrite a workbook after editing it on the server. For related information, see Grant Web Edit, Save, and Download Permissions on page 377.</p>
 Project Leader	Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions.

To set permissions for the project

1. On the Projects page, select a project, and then select **Actions > Permissions**.

The screenshot shows the Project Management interface. At the top, there are tabs for 'Projects' (10), 'Workbooks' (16), 'Views' (70), and 'Data Sources' (3). Below these are buttons for '+ New Project' and 'Actions'. A search bar and general filters are also present. On the right, a table lists projects: 'Finance' (selected, indicated by a checked checkbox) and 'General Purpose'. A context menu is open over the 'Finance' project, with 'Permissions...' highlighted and a cursor pointing at it. Other options in the menu include 'Rename...', 'Change Owner...', and 'Delete...'. The table below shows metrics for each project: Workbooks and Views.

Project	Workbooks	Views
Finance	8	38
General Purpose	3	7

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

The screenshot shows the 'User / Group' permissions configuration. It lists four entries: 'All Users (58)', 'Finance (13)', 'Adam Davis', and 'Jane Johnson'. Each entry has a 'Project' column showing its current role ('None', 'Custom', 'Custom', 'Project Leader') and a 'Details' column with checkboxes for various permissions. The 'Workbooks' column indicates if the group is managed by the owner ('None', 'Custom', 'Editor', 'None'). At the bottom, there is a button '+ Add a user or group rule' (highlighted with a red box) and a dropdown menu set to 'Group'. A scrollable list of available groups is shown, including 'All Users', 'Development', 'Finance', 'General Purpose' (which is currently selected, indicated by a cursor icon), and 'IT'.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.

The screenshot shows the 'User / Group' section of a project's permission settings. A red box highlights the 'Publisher' dropdown menu, which lists five roles: Viewer, Publisher (selected), Project Leader, None, and Denied. The 'Publisher' role is currently being selected.

The available permission role templates for projects are:

Template	Description
Viewer	Allows the user or group to view the workbooks and views in the project.
Publisher	Allows the user or group to publish workbooks and data sources to the server.
Project Leader	Allows the user or group to set permissions for all items in a project.
None	Sets all capabilities for the permission rule to Unspecified .
Denied	Sets all capabilities for the permission rule to Denied .
Data Source Connector	Allows the user or group to connect to data sources in the project.
Data Source Editor	Allows the user or group to connect to data sources in the project. Also to publish, edit, download, delete, and set permissions for a data source, and schedule refreshes for data sources you publish. This permission is relevant for views when accessing a view that connects to a data source.

- To further customize the rule, click the actions menu (.) next to the permission rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

The screenshot shows two overlapping windows of a software application for managing project permissions.

Top Window (Initial State):

User / Group	Project	Details
All Users (58)	None	(grid of 3x3 checkboxes: 1st row 3 green checkmarks, 2nd row 1 green, 1 red X, 1 grey, 3rd row 1 green, 1 grey, 1 green)
Finance (13)	Custom	(grid of 3x3 checkboxes: 1st row 1 green, 1 red X, 1 grey, 2nd row 1 green, 1 red X, 1 grey, 3rd row 1 green, 1 red X, 1 grey)
General Purpose (6)	Publisher	(grid of 3x3 checkboxes: 1st row 1 green, 1 red X, 1 grey, 2nd row 1 green, 1 green, 1 grey, 3rd row 1 green, 1 green, 1 grey)
Adam Davis	Edit	(grid of 3x3 checkboxes: 1st row 1 green, 1 green, 1 green, 2nd row 1 green, 1 green, 1 green, 3rd row 1 green, 1 green, 1 green)
Jane Johnson	Delete	(grid of 3x3 checkboxes: 1st row 1 green, 1 green, 1 green, 2nd row 1 green, 1 green, 1 green, 3rd row 1 green, 1 green, 1 green)

Bottom Window (Modified State):

User / Group	Project	Details	Workbooks
All Users (58)	None	(grid of 3x3 checkboxes: 1st row 3 green checkmarks, 2nd row 1 green, 1 red X, 1 grey, 3rd row 1 green, 1 grey, 1 green)	None
Finance (13)	Custom	(grid of 3x3 checkboxes: 1st row 1 green, 1 red X, 1 grey, 2nd row 1 green, 1 red X, 1 grey, 3rd row 1 green, 1 red X, 1 grey)	Custom
General Purpose (6)	Custom	(grid of 3x3 checkboxes: 1st row 1 green, 1 red X, 1 grey, 2nd row 1 green, 1 red X, 1 grey, 3rd row 1 green, 1 red X, 1 grey)	None
Jane Johnson	Project Leader	(grid of 3x3 checkboxes: 1st row 1 green, 1 green, 1 green, 2nd row 1 green, 1 green, 1 green, 3rd row 1 green, 1 green, 1 green)	None

A red box highlights the "Custom" permission rule for the "General Purpose (6)" group in the bottom window. A red arrow points from the "Edit" button in the top window to the "Save" button in the bottom window, indicating the save operation.

5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

Permissions

Edit permissions for the project "Finance".

Search for a user to view their permissions

User / Group	Project	Details	Workbooks
All Users (58)	None		Managed by the owner
Finance (13)	Custom		Custom
General Purpose (...	Custom		None
Harold Pawlan	Custom		Editor
Jane Johnson	Project Leader		None

+ Add a user or group rule

User Permissions General Purpose (6)

User	Role	Actions	Notes
Harold Pawlan	Viewer		None
Henry MacAllister	Viewer		Save: Denied (by group rule)
Henry Wilson	Administrator		Administrator
Irene Maddox	Viewer		None
Janet Molinari	Viewer		None
Karen Daniels	Viewer		None

- Follow the same steps to configure additional permission rules on the content for more users or groups.

Set Default Permissions for a Project, and its Workbooks and Data Sources

As an administrator or project leader, you can set a project's permissions and the default permissions for its workbooks and data sources.

Each project can have its own set of default permissions. The permissions that you set are the default permissions for all content in the project, including content that is being published to the project from Tableau Desktop.

Note: New projects are always created with the default permissions set for the **Default** project.

For additional information on working with permissions, see **Manage Permissions** on page 336 and **Projects** on page 136.

Notes on default permissions in locked projects

You can choose to have the default permissions apply to all workbooks and data sources in a project, and ensure that no one can change those settings, by locking content permissions to the project. For more information, see [Lock Content Permissions to the Project on page 369](#).

- Workbooks and data sources in a locked project always use the default permissions set for content in that project. Views in a locked project always use the workbook permissions. This applies to workbooks and data sources when they are being published from desktop.
- Administrators and users with the Project Leader permission can always edit default permissions, even when a project is locked.
- Users, including content owners, cannot edit individual workbook, view, and data source permissions when content is locked to the project.

To set default permissions in a project

1. In the Content page of a site, click a project, and then click **Permissions** in the project place page.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Editor	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

+ Add a user or group rule		Group
<input type="button" value="All Users"/> <input type="button" value="Development"/> <input type="button" value="Finance"/> <input type="button" value="General Purpose"/>		<input type="button" value="Group"/>
or select a permission rule above to view use		

For an existing user or group, click the actions menu (. . .), and then click **Edit**.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (6)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

3. Select a permission role template for **Project**, **Workbooks**, or **Data Sources**, and then click **Save**.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (6)	Viewer	Viewer	Connector
Jane Johnson	Viewer	Editor	Editor
+ Add a user or group rule	None	None	None
	Viewer	Editor	Editor
	None	None	None
	Denied		

Or, to create a custom set of capabilities, click the **Project**, **Workbooks**, or **Data Sources** labels to expand the permissions view. Click capabilities to set them to **Allowed**, **Denied**, or **Unspecified**. Click **Save**.

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None	None	None	None
Finance (13)	Publisher	Custom	Custom	Connector
Adam Davis	Custom	Editor	Editor	Editor
Jane Johnson	Project Leader	None	None	None
		Project Leader - Allowed		

This example shows how to set project permissions. The same general steps apply for workbooks and data sources.

Note: To change the settings after saving, click the actions menu (...), and then click **Edit**.

- View the user permissions, which are the effective permissions.

Click a group name or user name in the permission rules to see the resulting user permissions.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (...)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None
+ Add a user or group rule			
User Permissions General Purpose (6)			
Harold Pawlan	Viewer	Viewer	Connector
Henry MacAllister	Viewer	Viewer	Custom
Henry Wilson	Administrator	Administrator	Administrator
Irene Maddox	Viewer	Viewer	Connector
Janet Molinari	Viewer	Viewer	Connector
Karen Daniels	Viewer	Viewer	Custom

Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None		None	None
Finance (13)	Publisher		Custom	Connector
General Purpose (...)	Viewer		Viewer	Connector
Adam Davis	Custom		Editor	Editor
Jane Johnson	Project Leader		None	None
+ Add a user or group rule				
User Permissions General Purpose (6)				
Harold Pawlan	Viewer		Viewer	Connector
Henry MacAllister	Viewer		Viewer	Custom
Henry Wilson	Administrator		Administrator	Administrator
Irene Maddox	Viewer		Viewer	Connector
Janet Molinari	Viewer		Viewer	Connector
Karen Daniels	Viewer		Viewer	Custom

Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None		Managed by the o...	Managed by the o...
Finance (13)	Publisher	✓ ✓	Custom	Connector
General Purpose (...	Viewer	✓	Viewer	Connector
Adam Davis	Custom	✓ ✓ ✓	Editor	Editor
Jane Johnson	Project Leader		None	None

User Permissions General Purpose (6)				
Harold Pawlan	Viewer		Viewer	Connector
Henry MacAllister	Viewer	✓	Viewer	Custom
Henry Wilson	Administrator	✓	Save: Denied (not granted by any rule)	
Irene Maddox	Viewer		Viewer	Connector
Janet Molinari	Viewer		Viewer	Connector
Karen Daniels	Viewer		Viewer	Custom

- Follow the same steps to configure additional permission rules for more users or groups.

Create Project-Based Permissions

As an administrator, you can organize a collection of related workbooks and data sources in a project. You can then control access to that content by creating permission rules for groups of users who need similar access levels to publish or interact with that content.

Note: For this scenario, you set the permission rule for the All Users group for the project to **None**, which means that permissions are **Unspecified** for the All Users group.

Preparation

Before you begin creating projects and project-based permissions, document the projects and permission levels that you want users to have in each project.

This roadmap exercise helps you organize permissions to be most efficient to manage over time, and can help you identify any user or permission gaps in your solution.

Also read the following topics in the Tableau Server Help:

- [Manage Permissions](#) on page 336 and permissions-related topics
- [Projects](#) on page 136 and projects-related topics
- [Grant Web Edit, Save, and Download Permissions](#) on page 377

Step 1: Create projects and user groups

- Sign in to Tableau Server with your administrator user name and password.
- On the Projects page, click **New Project**.

3. Click **Groups**, and then click **New Group**.

Create groups that correspond to each project and access level. For example, for a project that allows users only to access the views, you might use a name similar to Project1_Viewer. For a project that allows interaction with the views, Project1_Interactor.

4. Click **Users**, and then click **Add Users**. Select one or more users in the list, select **Actions > Group Membership**, and then select a group for the users. Click **Save** to confirm the group membership.

Repeat this step to add users to other groups.

Step 2: Assign permissions at the project level

After you set up your projects and user groups, you can start assigning permissions. Repeat these steps for each project. Also see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

1. On the Projects page, select a project, and then select **Actions > Permissions**.
 2. For the **All Users** group permission rule, set the permission role template to **None**.
Click the actions menu (...) next to **All Users**, and then click **Edit**. Select **None** for **Project, Workbooks, and Data Sources**, and then click **Delete**. This means that all capabilities will be set to **Unspecified**.
 3. Click **Add a user or group rule**, select **Group**, and then select the group name in the list.
To edit an existing rule, click the actions menu (...) next to the permission rule name, and then click **Edit**.
 4. Select a permission role template for **Project, Workbooks, and Data Sources** to specify a predefined set of capabilities for the group or user.
 5. To further change capabilities included in the rule, click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**.
Click **Save** when you are done.
- Repeat steps 3-5 for each group or user requiring project permissions.

Note: You can optionally lock content permissions to the project to enforce the default permissions for all content in the project. This overwrites any previous permissions assigned to workbooks and views in the project. For more information, see [Lock Content Permissions to the Project](#) on page 369.

Step 3: Check project permissions

- View the resulting user permissions.

Click a group name or user name in the permission rules list to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

When you publish workbooks to the project, the permissions are updated accordingly.

For information on granting Save permissions to users, see [Grant Web Edit, Save, and Download Permissions on page 377](#).

Quick Start: Lock Content Permissions to a Project

As an administrator or project leader, you can lock content permissions in a project to prevent users from changing the permissions of any content in the project. When permissions are locked to the project, the default permissions are applied to all workbooks and data sources in a project and cannot be modified by users (including the content owners).

Note: Content owners always get full access to the content they've published, but cannot change permissions for their workbooks and data sources when the parent project permissions are locked.

For information on setting permissions, see [Manage permissions](#) and [Permission Rules and User Permissions on page 344](#). For more information on setting default permissions and locking content permissions to the project, see [Set Default Permissions for a Project, and its Workbooks and Data Sources on page 364](#) and [Lock Content Permissions to the Project on page 369](#).

1 Set Default Permissions for the Project

Because the content inside locked projects always uses the default permissions, first verify that your default permissions are set appropriately. In a site, click **Content > Projects**. Open a project, and then click **Permissions**. Add a user or group and select a permission role template for that content type, or click **Edit**, and then set capabilities to **Allowed**, **Denied**, or **Unspecified**.

Administrators and Project Leaders can edit default permissions at any time.

2 Lock Content Permissions to the Project

In a project's permissions, click the **Managed by the owner** button. The button label indicates whether content permissions are currently locked to the project or managed by the content owner. Select **Locked to the project**, and then click **Save**.

When permissions are locked to the project, all content in the project uses the default permissions. No users can change permissions for individual workbooks (including views) or data sources in the project.

3 View Locked Permissions

Open a project, select a workbook or data source in the project, and then click **Actions > Permissions**. When permissions are locked to the project, users can view workbook or data

source permissions in the project, but they cannot modify them.

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	[Greyed Out]	[Greyed Out]	[Greyed Out]
Finance (13)	Custom	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	X ✓ ✓ ✓ ✓
General Purpose (6)	Viewer	✓ ✓ ✓ ✓ ✓	[Greyed Out]	[Greyed Out]
Adam Davis	Editor	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓

User	Role	View	Interact	Edit
Adam Davis	Administrator	• • • • •	• • • • •	• • • • •
Andrew Allen	Custom	• • • • •	• • • • •	• • • • •
Andrew Smith	Custom	• • • • •	• • • • •	• • • • •
Ashley Garcia	Administrator	• • • • •	• • • • •	• • • • •
Claire Gute	Custom	• • • • •	• • • • •	• • • • •
Darren Powers	Custom	• • • • •	• • • • •	• • • • •

In this example, the workbook owner has full permissions for the workbook, but cannot change the workbook permissions while they are locked to the project.

4Unlock Content Permissions for the Project

In a site, click **Content > Projects**. Select a project, and then click **Actions > Permissions**. Click the **Locked to the project** button. Select **Managed by the owner**, and then click **Save**.

Content Permissions in Project

Permissions for workbooks and data sources in the project "Finance" are:

Locked to the project
Workbooks and data sources in this project always use the default permissions. Permissions for individual workbooks and data sources in this project cannot be modified.

Managed by the owner
Workbooks and data sources in this project start with the default permissions. Permissions for individual workbooks and data sources in this project can be modified.

Locked to the project

When a project's content permissions are **Managed by the owner**, individual workbooks, views, and data sources in the project start with the default permissions and can be modified by users.

Notes on project permissions:

- Only administrators and project leaders can lock content permissions, and set and edit default permissions in a project.
- Administrators and project leaders can edit default permissions for the project, its workbooks, and its data sources at any time, at the project level.
- Individual workbook, view, and data source permissions cannot be edited by users (including content owners) when a project is locked.
- Workbooks and data sources in a locked project always use the default permissions. Views in a locked project always use the workbook permissions.

L-

Lock Content Permissions to the Project

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.

When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

Note: Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For information on default permissions, see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

Note: Administrators and project leaders can set and edit default permissions for the project, and its workbooks and data sources when it is locked.

1. In the Content page of a site, open a project, and then click **Permissions** in the project place page.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (...)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

2. Click the **Managed by the owner** button.

The padlock icon on the button label indicates whether content permissions are currently locked to the project or managed by the content owner.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.

Content Permissions in Project

Permissions for workbooks and data sources in the project "Finance" are:

Locked to the project

Workbooks and data sources in this project always use the default permissions. Permissions for individual workbooks and data sources in this project cannot be modified.

Default permissions will be applied to all workbooks and data sources in the project when you save.

Managed by the owner

Workbooks and data sources in this project start with the default permissions. Permissions for individual workbooks and data sources in this project can be modified.

When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

Permissions

See permissions for the workbook "Finance".

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	[Grey]	[Grey]	[Grey]
Finance (13)	Custom	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Red] ✘ ✓ ✓ ✓ ✓
General Purpose (6)	Viewer	[Green] ✓ ✓ ✓ ✓ ✓	[Grey]	[Grey]
Adam Davis	Editor	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓

User Permissions Finance (13)	
Adam Davis	Administrator
Andrew Allen	Custom
Andrew Smith	Custom
Ashley Garcia	Administrator
Claire Gute	Custom
Darren Powers	Custom

- To unlock content permissions for the projects, open the project permissions again. Click the **Locked to the project** button. In the **Content Permissions in Project** dialog box, select **Managed by the owner**, and then click **Save**.

Home > Finance

Finance

PROJECT This project is for all workbooks, views, and data sources used by the Finance group.

Workbooks 4 Views 20 Data Sources 0 Permissions Details

Content Permissions in Project

Permissions for workbooks and data sources in the project "Finance" are:

Locked to the project Workbooks and data sources in this project always use the default permissions. Permissions for individual workbooks and data sources in this project cannot be modified.

Managed by the owner Workbooks and data sources in this project start with the default permissions. Permissions for individual workbooks and data sources in this project can be modified.

The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

Groups

You can organize Tableau Server users into groups to make it easier to manage multiple users. You can either create groups locally on the server or import groups from Active Directory.

To keep Active Directory group membership up-to-date:

- Site administrators can synchronize selected groups on demand in a site. For more information, see [Synchronize Active Directory Groups on a Site](#).

- Server administrators can synchronize all Active Directory groups on the server based on a schedule or on-demand. For more information, see [Synchronize All Active Directory Groups on the Server](#).

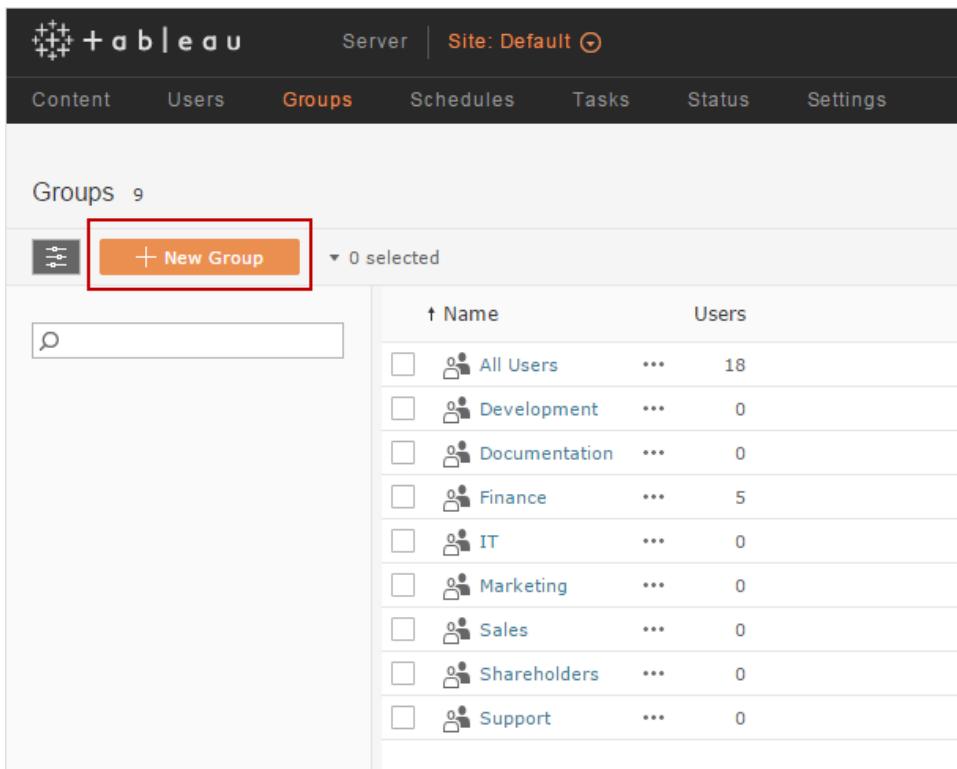
You can also assign permissions to a group for a project, workbook, view, or data source. For details, see [Manage Permissions](#) on page 336.

The All Users group exists in every site by default. Every user added to the server becomes a member of the All Users group automatically. You cannot delete this group, but you can set permissions for it.

Create a Local Group

Local groups are created using the Tableau Server internal user management system. After you create a group you can add and remove users.

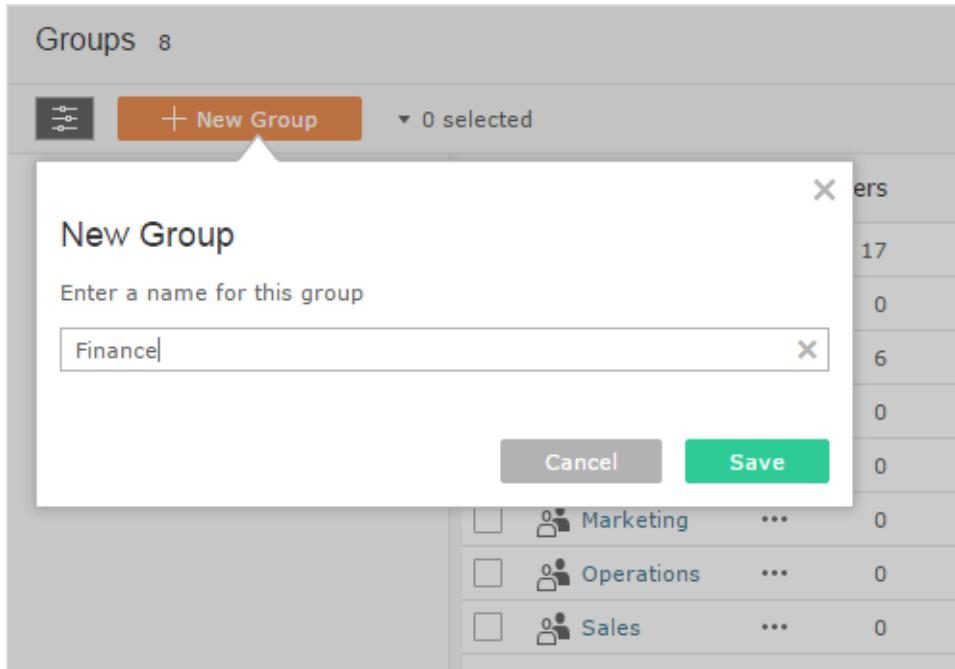
1. In a site, click **Groups**, and then click **New Group**.



The screenshot shows the Tableau Server interface for managing groups. The top navigation bar includes icons for Home, Groups, and Settings, along with links for Server and Site: Default. Below the navigation is a menu bar with Content, Users, Groups (which is highlighted in orange), Schedules, Tasks, Status, and Settings. The main content area is titled "Groups 9". It features a search bar and a "New Group" button, which is highlighted with a red box. A list of existing groups is displayed in a table:

	Name	Users
<input type="checkbox"/>	All Users	18
<input type="checkbox"/>	Development	0
<input type="checkbox"/>	Documentation	0
<input type="checkbox"/>	Finance	5
<input type="checkbox"/>	IT	0
<input type="checkbox"/>	Marketing	0
<input type="checkbox"/>	Sales	0
<input type="checkbox"/>	Shareholders	0
<input type="checkbox"/>	Support	0

- Type a name for the group and click **Save**.



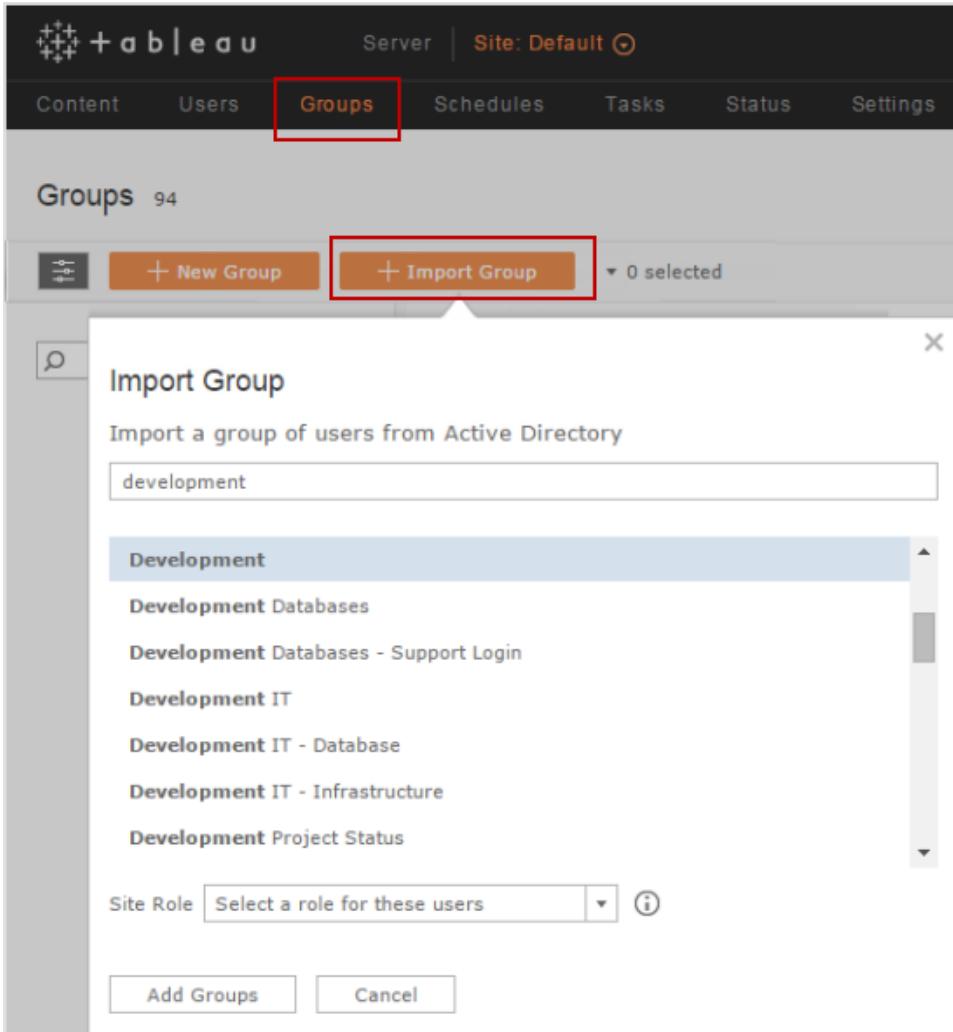
Create a Group via Active Directory

When you import Active Directory groups, a matching group is created on the server and a user is created on the server for each member of the group that is not already on the server.

Each user is assigned a site role as part of the import process. If a user already exists on the site without a group affiliation, the user is added to the group with the assigned site role, and the same permissions in the site.

Note: Importing users and groups will promote a user's site role, but never demote a user's site role. If any of the users to be imported already exist in Tableau Server, the site role assigned during the import process will be applied only if it gives the user more access to the server. For more information, see [Site Roles for Users on page 176](#).

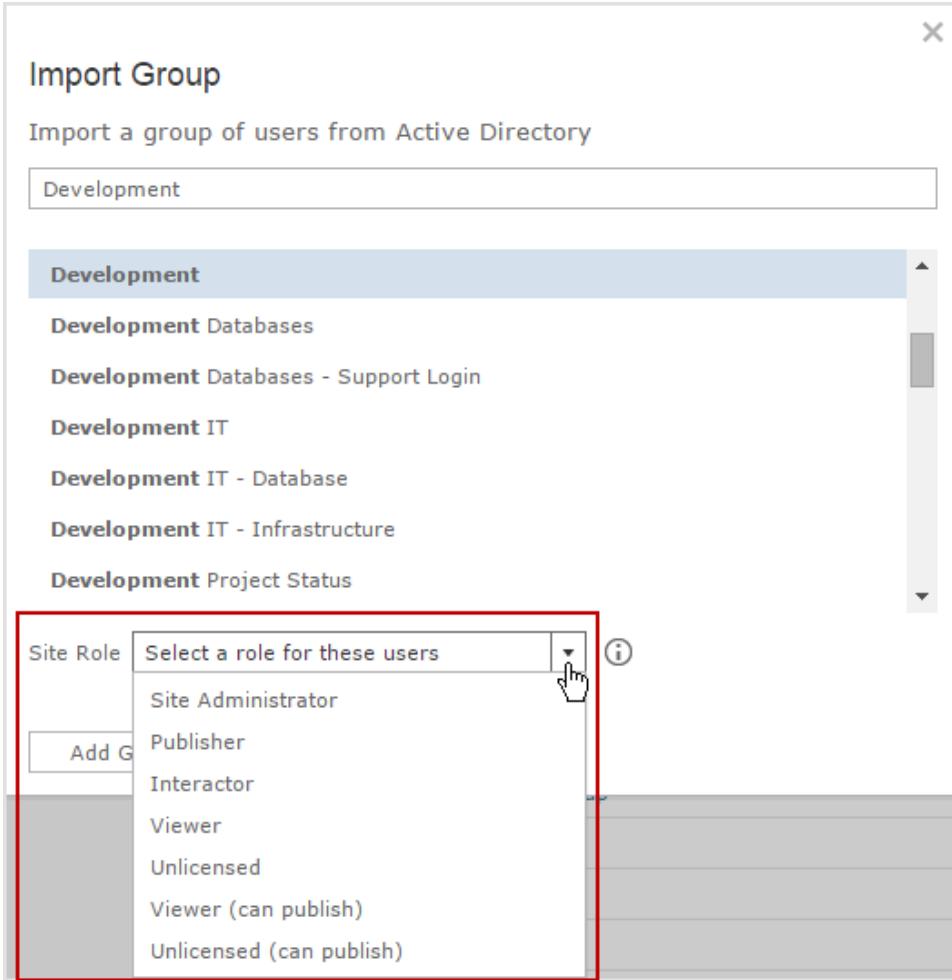
- In a site, click **Groups**, and then click **Import Group**
- Type the name of the Active Directory group you want to import, and then select the group name in the resulting list.



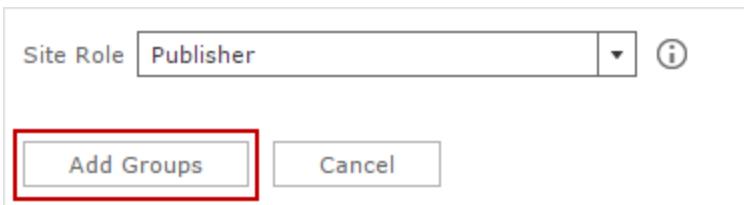
If you are adding a group that is from the same Active Directory domain that the server is running on, you can enter the group name. Also, if there is a two way trust set up between the domain the server is using and another domain, you can add groups from both domains.

The first time you add a group from a different domain than the one the server is using, you must include the fully qualified domain name with the group name. For example, `domain.lan\group` or `group@domain.lan`. Any subsequent groups can be added using the domain's nickname. See `tabcmd editdomain` to learn more about managing domain names.

3. Select the site role for the users.



4. Click **Add Groups**.



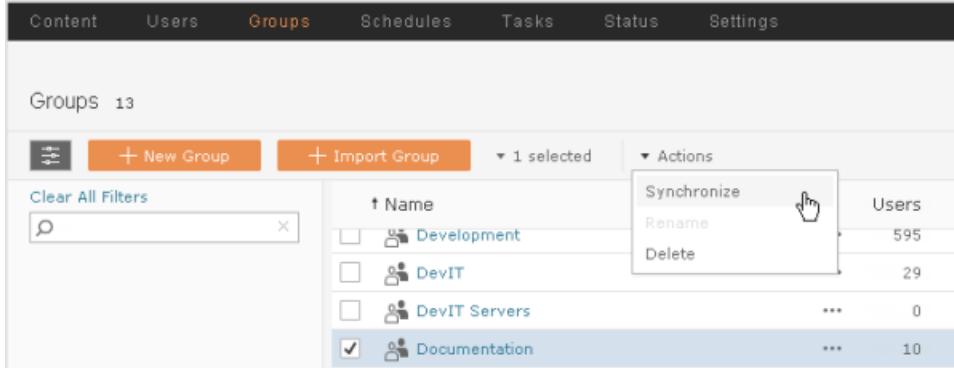
Note: You cannot change the name of groups imported from Active Directory. The group name can only be changed in Active Directory.

Synchronize Active Directory Groups in a Site

At any time, you can synchronize an Active Directory group with Tableau Server to ensure new users in Active Directory are also added in Tableau Server. You can synchronize individual

groups or multiple groups at once.

1. In a site, click **Groups**. On the Groups page, select one or more groups.
2. Select **Actions > Synchronize**.



The screenshot shows the 'Groups' page with 13 items. A context menu is open over the 'Documentation' group, which has a checked checkbox. The menu options are 'Synchronize' (highlighted with a cursor icon), 'Rename', and 'Delete'. Other groups listed include 'Development', 'DevIT', 'DevIT Servers', and 'Documentation' (which is selected).

Name	Users
Development	595
DevIT	29
DevIT Servers	0
Documentation	10

Set the minimum site role for users in an Active Directory group

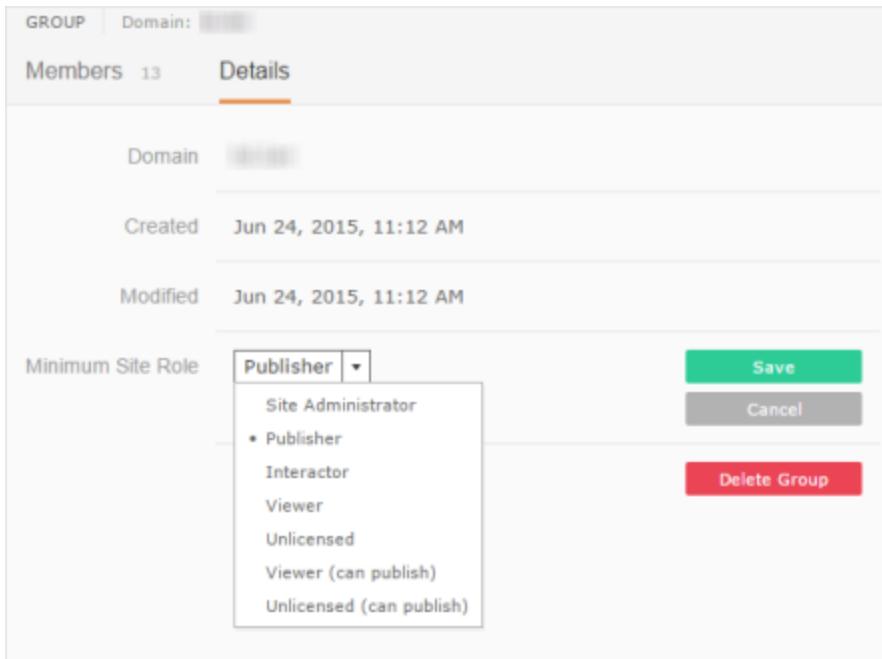
In the **Groups - Details** page, administrators can set the minimum site role for group users to be applied during synchronization.

This setting does not run synchronization; it sets the minimum site role to applied to the group every time synchronization runs. When you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role will be applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see [Site roles and Active Directory import and synchronization on page 181](#).

1. In a site, click **Groups**.
2. Click the group name link and then click the **Details** tab.

3. Select the **Minimum site role**, and then click **Save**.



What happens when users are removed as the result of the synchronization process?

When you remove a user from Active Directory, and then synchronize with that user's group on Tableau Server, the user is:

- Removed from the Tableau Server group you synchronized.
- Unable to sign in to Tableau Server.

Because the user still remains on the server, administrators can audit and reassign the user's content before removing their account completely.

For users that also exist on the server locally, the site role is set to **Unlicensed** in the site. The user will still belong to the **All Users** group with a site role of **Unlicensed**.

To fully remove the user from Tableau Server, the server administrator will need to delete the user from the Server Users page in Tableau Server.

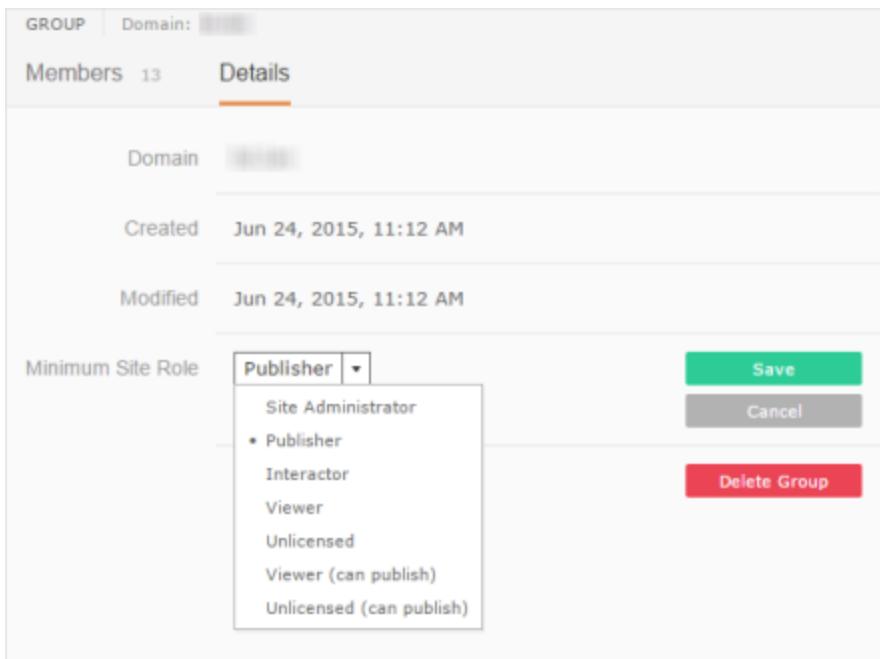
Quick Start: Synchronize All Active Directory Groups on a Schedule

After you import Active Directory groups in Tableau Server, you can make sure they stay synchronized in Tableau Server by setting up a schedule. You can also synchronize all Active Directory groups on the server on-demand, at any time. The minimum site role setting for the group is applied when users are synchronized.

Note: To use this feature, your Tableau Server installation must be set up for Active Directory.

1 Set a minimum site role for synchronization

In a site's **Groups** page, click the **Details** tab, select the minimum site role, and then click **Save**. Server and site administrators can set the minimum site role for group users to be applied during Active Directory synchronization. If you don't set a minimum site role, new users are added as **Unlicensed**.



Synchronizing can promote a user's site role, but will never demote a user's site role.

2 Set the schedule

Server administrators can enable synchronization for all Active Directory groups on the **General** tab of the **Server - Settings** page. Enable synchronization, select the frequency settings, and then click **Save**.

The screenshot shows the Tableau Server - Settings page with the General tab selected. Under Active Directory Synchronization, there is a note about managing synchronization of all Active Directory groups. It shows the last synchronization time as June 25, 2015, at 4:01 PM (Server time) and a link to view synchronization activity. Below this is a button labeled "Synchronize All Groups". A red box highlights this button and the scheduling configuration area. The scheduling area includes a checked checkbox for "Synchronize Active Directory groups on a regular schedule", frequency options (Hourly, Daily, Weekly, Monthly) with "Daily" selected, and a time and AM/PM selector set to 12:00 AM. At the bottom of the page are "Reset to Default Settings", "Revert", and "Save" buttons.

All Active Directory groups on the server are synchronized according to the same schedule.

3 Run synchronization on-demand (optional)

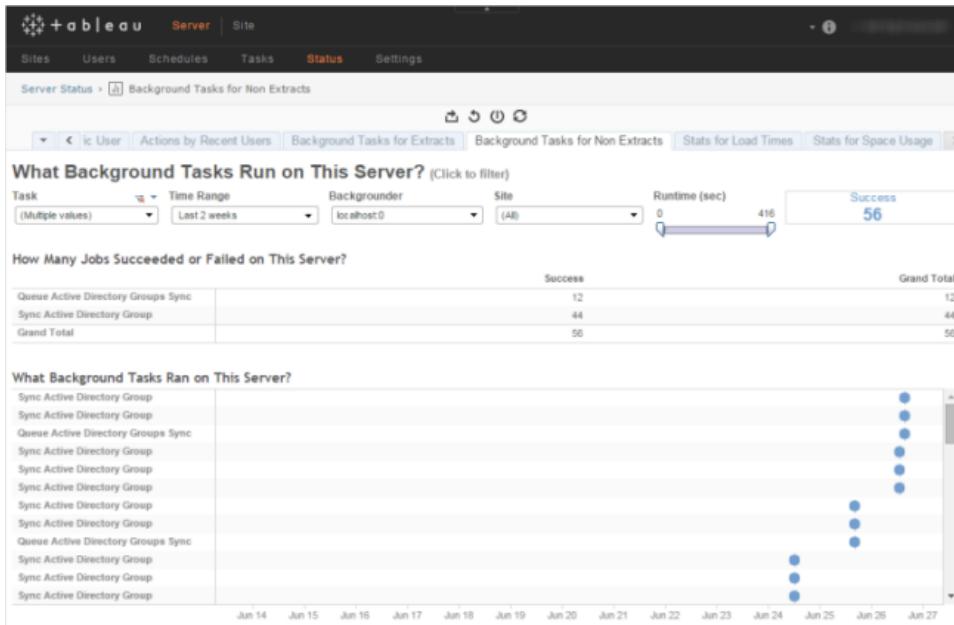
On the **General** tab of the **Server - Settings** page, click **Synchronize All Groups** to synchronize all Active Directory groups on Tableau Server immediately. Click this button at any time to ensure new users and changes are reflected in all Active Directory groups on the server.

This screenshot is identical to the one above, showing the Tableau Server - Settings page with the General tab selected. It displays the Active Directory synchronization settings, including the "Synchronize All Groups" button which is highlighted with a red box. The scheduling section also contains a red box around the "Synchronize Active Directory groups on a regular schedule" checkbox and its associated controls.

Click **Synchronize All Groups** to synchronize all Active Directory groups on the server outside of a schedule.

4 View the status of synchronization tasks

Server and site administrators can view the results of Active Directory synchronization jobs in the **Background Tasks for Non Extracts** administrative view. On the server or in a site, click **Status**. Under **Analysis**, click **Background Tasks for Non Extracts** and filter on the **Queue Active Directory Groups Sync** and **Sync Active Directory Group** tasks.



Queue Active Directory Groups Sync queues the **Sync Active Directory Group** tasks to be run.

Synchronize All Active Directory Groups on the Server

As a server administrator, you can synchronize all Active Directory groups on a regular schedule or on-demand on the **General** tab of the **Server - Settings** page.

The screenshot shows the 'General' tab selected in the 'Settings' section of a web-based management interface. Under the 'Active Directory Synchronization' heading, it says 'Manage the synchronization of all Active Directory groups. Learn more'. Below this, it shows 'Last synchronized: Jun 25, 2015, 4:01 PM (Server time)' and a link to 'View synchronization activity'. A large red box highlights the 'Synchronize All Groups' button and the scheduling options. The scheduling section includes a checked checkbox for 'Synchronize Active Directory groups on a regular schedule', frequency options (Hourly, Daily, Weekly, Monthly) with 'Daily' selected, and a time setting at 12:00 AM. At the bottom are 'Reset to Default Settings', 'Revert', and 'Save' buttons.

The **Last synchronized** time indicates the time that synchronization most recently began.

Synchronize Active Directory groups on a schedule

1. Select **Server > Settings > General**. Under **Active Directory Synchronization**, select **Synchronize Active Directory groups on a regular schedule**.

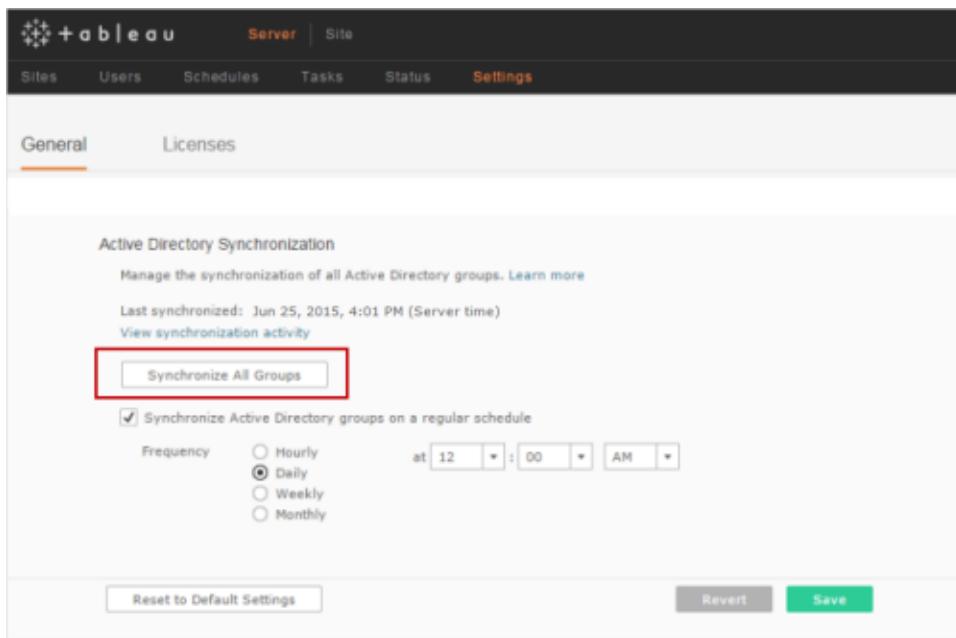
This screenshot is identical to the one above, showing the 'General' tab in the 'Settings' section. The 'Active Directory Synchronization' settings are displayed, including the 'Synchronize All Groups' button and the scheduling configuration. A red box surrounds the entire scheduling section, which includes the checked checkbox for regular synchronization, the frequency selection (Daily), and the specific time set to 12:00 AM. The bottom buttons ('Reset to Default Settings', 'Revert', 'Save') are also visible.

2. Select the frequency and time of synchronization.
3. Click **Save**.

Synchronize all Active Directory groups on demand

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. Select **Server > Settings > General**.



2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. Select **Server > Status**.
2. Click the **Background Tasks for Non Extracts** link.
3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Server - Settings** page.

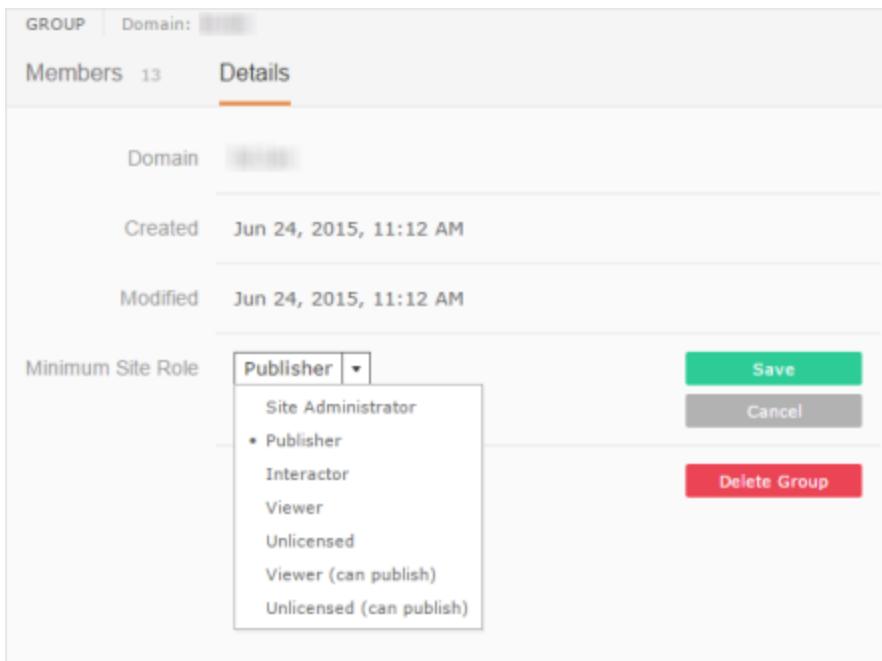
Set the minimum site role for users in an Active Directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see [Site roles and Active Directory import and synchronization](#) on page 181.

1. In a site, click **Groups**.
2. Click the group name link and then click the **Details** tab.
3. Select a site role from the **Minimum site role** list, and then click **Save**.



Users removed during synchronization

When you remove a user from Active Directory, and then synchronize that user's group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user is unable to sign in to Tableau Server.

Because the user remains on the server, administrators can audit and reassign the user's content before removing the user's account completely.

For users who also exist on the server locally, the site role is set to **Unlicensed** in the site as the result of the synchronization. The user continues to belong to the **All Users** group with a site role of **Unlicensed**.

To fully remove the user from Tableau Server, you (server administrator) must delete the user from the Server Users page in Tableau Server.

Delete Groups

You can delete any group from the server (with the exception of the All Users group). When you delete a group, the users are removed from the group but they are not deleted from the server.

1. In a site, click **Groups**.
2. On the Groups page, select one or more groups to delete.
3. Select **Actions > Delete**.

The screenshot shows the Tableau Server Groups page. The top navigation bar includes links for Content, Users, Groups, Schedules, Tasks, Status, and Settings. The Groups link is highlighted. Below the navigation is a search bar and a 'New Group' button. A table lists eight groups: Marketing (selected), Development, Finance, HR, IT, Operations, and Sales. The Marketing group has a checked checkbox in the first column and a 'Delete' option in the Actions dropdown menu. The other groups have unchecked checkboxes and standard Actions menus. The right side of the table shows the number of users in each group.

	Name	Action	Users
<input type="checkbox"/>	Development	Synchronize Rename Delete	17
<input type="checkbox"/>	Finance	...	0
<input type="checkbox"/>	HR	...	0
<input type="checkbox"/>	IT	...	0
<input checked="" type="checkbox"/>	Marketing	...	0
<input type="checkbox"/>	Operations	...	0
<input type="checkbox"/>	Sales	...	0

Users

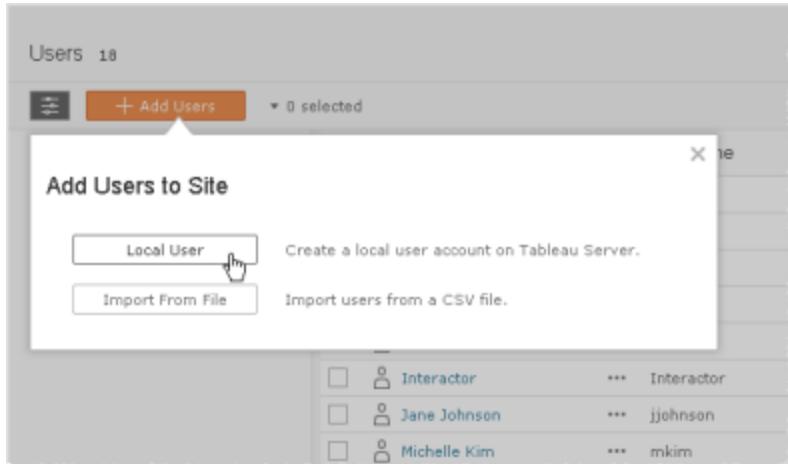
Guest user

A Guest user is available in Tableau Server (core-based licenses only) in each site to allow users without an account on the server see and interact with an embedded view. When enabled, the user can load a webpage containing an embedded visualization without signing in. For more information, see [Guest User on page 182](#).

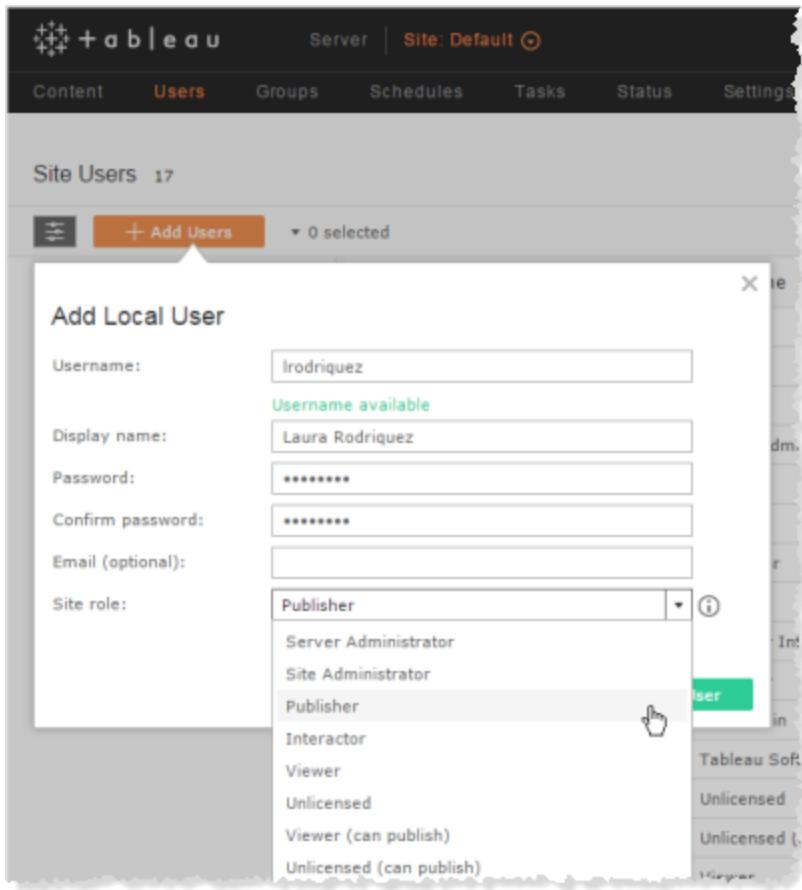
Server users and site users

Server administrators can add users to the server, and server and site administrators (if allowed) can add users to individual sites. For details on allowing site administrators to add users to sites, see step 4 in [Add or Edit Sites on page 117](#).

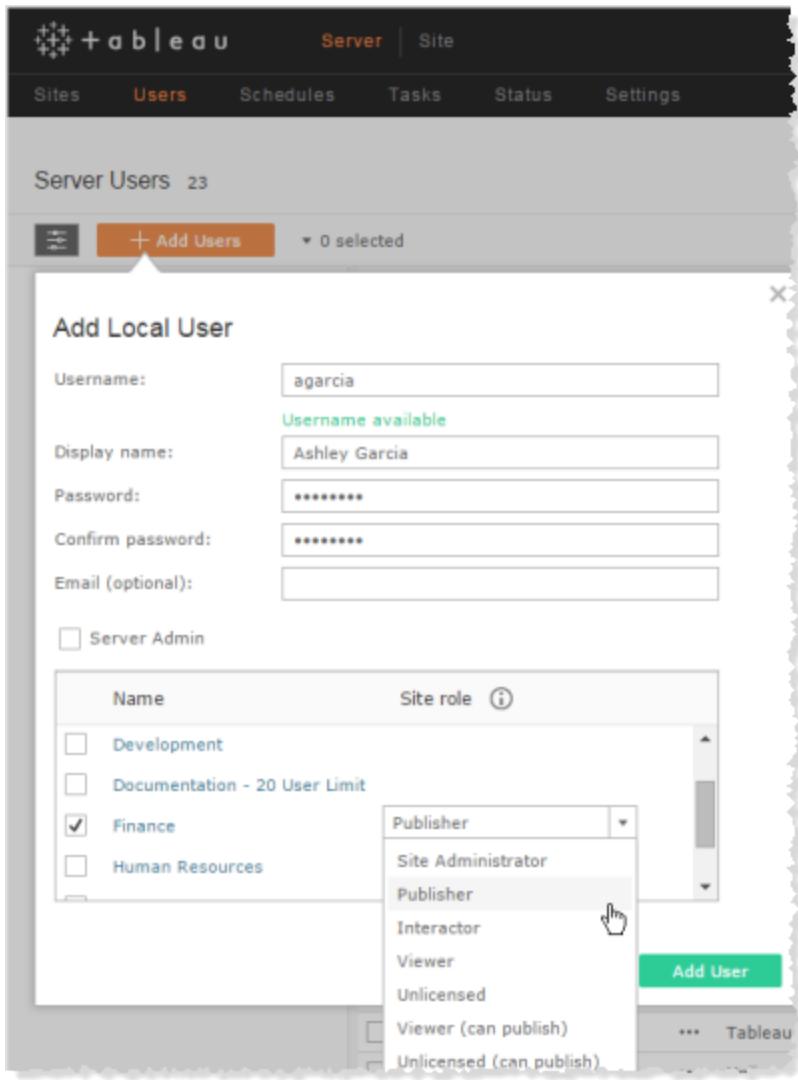
In a single-site environment, server and site administrators can add users on the **Users** page.



In a multi-site environment, server and site administrators can add users in the **Site Users** page.



Server administrators can add users in the **Server Users** page. When you add a user to the server, you can assign the site membership and site roles per site for the user.



If you add a user without assigning site membership and role, the user is assigned the Unlicensed role and won't use a server license (user-based licensing only). The user will exist in Server Users, but will not be a member of any site until you add that user to the site.

Note: Every user who is added to a site is also automatically added to the server. Site administrators can remove users from their sites, but they cannot delete users from the server. Server administrators can delete users from the server.

When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn't own any content.

Site Roles for Users

Every user added to Tableau Server must have an associated site role. The site role is assigned by the administrator. The site role determines the levels of permissions allowed for a user, including whether a user can publish, interact with, or only view content published to the server. Administrators are also defined based on the site role.

Note: Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.

Users are accounts on the server that can be associated with one or more sites, and with groups in those sites. Any user that is added to Tableau Server or to a site becomes member of the All Users group. The All Users group is present in every site and cannot be deleted.

Who can publish content

Users with the following site roles can publish to Tableau Server:

- Server Administrator
- Site Administrator
- Publisher
- Viewer (can publish)
- Unlicensed (can publish)

Users with a site role of **Interactor**, **Viewer**, and **Unlicensed** cannot publish content to the server.

Site roles and permissions

Effective user permissions for a resource are determined by:

- The maximum capabilities allowed for a user's site role. The site role acts as the "ceiling" for what permissions are allowed.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

When you select a site role for a user, help is available to remind you of the general level of permissions for that site role.

Site role	Web access	Interact	Publish	Manage
Server Administrator	✓	✓	✓	✓
Site Administrator	✓	✓	✓	✓
Publisher	✓	✓	✓	
Interactor	✓	✓		
Viewer		✓		
Unlicensed				
Viewer (can publish)	✓		✓	
Unlicensed (can publish)			✓	

When you set permissions for a content item, the User Permissions section in the Permissions window indicates when a permission capability is not allowed for that site role.

The User Permissions area of the Permissions window shows the effective permissions for each user. These are the actual permissions for each user, after the user's site role and permission rule has been evaluated.

For details, see [Permission Rules and User Permissions](#) on page 344.

General capabilities

- **Server Administrator:** The server administrator can access all server features and settings on the server and all sites. Server administrators can create sites, add users of any site role type, control whether site administrators can add users, create additional server administrators, and they can administer the server itself. This includes handling maintenance, settings, schedules, and the search index.

Server administrators can perform operations on all content anywhere on the server, regardless of what permissions have been assigned to the content. Server administrators can also manage other users on the server.
- **Site Administrator:** Site administrators can manage groups, projects, workbooks, and data connections. By default, site administrators can also add users and assign site roles and site membership. This setting can be enabled or disabled by the server administrator (see step 4 in [Add or Edit Sites](#) on page 117).

Site administrators have unrestricted access to content on a specific site. A user can be specified as a site administrator on multiple sites.
- **Publisher:** Publishers can sign in, browse the server, and interact with the published views. They also can connect to Tableau Server from Tableau Desktop in order to publish and download workbooks and data sources.

Publishers can publish (upload) workbooks and data sources to the server. Publishers aren't allowed to manage other users.

- **Interactor:** Interactors can sign in, browse the server, and interact with the published views. It's important to note that specific views, workbooks, and projects may have been published with permissions that restrict a user's capabilities. Permission settings can be edited by the workbook author or an administrator.

Interactors can view workbooks and can interact with views. They are not allowed to publish to the server.

- **Viewer:** Viewers can sign in and see published views on the server but cannot interact with the views. Users with this site role can only be given permission to view, add comments, and view comments. They cannot interact with filters in the view or sort data in a view.
- **Unlicensed:** Unlicensed users cannot sign in to the server. When you import server users from a CSV file, all are assigned a site role of Unlicensed.

If an insufficient number of licenses are available when an administrator creates a user (though CSV import of a site user, or import from Active Directory, or when a local user is created) the user will be assigned the Unlicensed site role.

Attempting to remove a user who owns content from a site will demote the user to Unlicensed. The user will still own the content.

- **Viewer (can publish).** The user can connect to Tableau Server from Tableau Desktop to publish and download workbooks and data sources, but cannot interact with content on the server.
- **Unlicensed (can publish).** This user cannot sign in to Tableau Server, but can connect to the server from Tableau Desktop to publish workbooks to the server.

Maximum permissions allowed for each site role (by content type)

For each content type, the site role determines the capabilities allowed for a user. For example, a user with the site role of **Viewer** can be assigned all capabilities in a permission rule for a workbook, but when the permissions are evaluated for the user, the resulting permissions capabilities will be limited to View, Export Image, Summary Data, View Comments, Add Comments, and Save.

Project

Site Role	Description
Server Administrator	View, Save, Project Leader
Site Admin-	View, Save, Project Leader

istrator	
Publisher	View, Save, Project Leader
Interactor	View, Save, Project Leader
Viewer (can publish)	View, Save
Viewer	View, Save
Unlicensed (can publish)	View, Save
Unlicensed	View, Save
Guest	View

Workbook

Site Role	Description
Server Administrator	All capabilities
Site Administrator	All capabilities
Publisher	All capabilities
Interactor	All capabilities (no ability to publish)
Viewer (can publish)	View, Download Image/PDF, Download Summary Data, View Comments, Add Comments, Download Workbook/Save As
Viewer	View, Download Image/PDF, Download Summary Data, View Comments, Add Comments, Download Workbook/Save As (no ability to publish)
Unlicensed (can publish)	View, Save, Download Workbook/Save As
Unlicensed	View, Save, Download Workbook/Save As (no ability to publish)
Guest	View, Download Image/PDF, Download Summary Data, View Comments, Filter, Download Full Data, Web Edit, Download Workbook/Save As

[View](#)

Site Role	Description
Server Administrator	All capabilities
Site Administrator	All capabilities
Publisher	All capabilities
Interactor	All capabilities (no ability to publish)
Viewer (can publish)	View, Download Image/PDF, Download Summary Data, View Comments, Add Comments
Viewer	View, Download Image/PDF, Download Summary Data, View Comments, Add Comments (no ability to publish)
Unlicensed (can publish)	View
Unlicensed	View (no ability to publish)
Guest	View, Download Image/PDF, Download Summary Data, View Comments, Filter, Download Full Data, Web Edit

[Data Source](#)

Site Role	Description
Server Administrator	All capabilities
Site Administrator	All capabilities
Publisher	All capabilities
Interactor	All capabilities (no ability to publish)
Viewer (can publish)	View, Save

Viewer	View, Save (no ability to publish)
Unlicensed (can publish)	View, Save, Download Data Source
Unlicensed	View, Save, Download (no ability to publish)
Guest	View, Download Data Source

Site roles and Active Directory import and synchronization

When you import Active Directory users to a site, either as a single user or as member of a group, you can specify a site role for the user. If a user is not yet a member of any site on the server, the user is added to the site with the assigned role. When you synchronize Active Directory groups, the site role is applied through the **Minimum Site Role** setting on the **Groups - Details** page.

If a user already exists in a Tableau Server site, the site role assigned during the import or sync process will be applied if it gives the user more access in a site. Importing or synchronizing users and groups will promote a user's site role, but not demote a user's site role.

If a user already has the ability to publish, that ability will always be maintained. For example, if a user with the current site role of **Unlicensed (can publish)** is imported with the new site role of **Interactor**, that user's site role will be promoted to **Publisher** on import.

To guarantee a user maintains a site role with equal or greater capabilities in server after an import, the following matrix shows the rules applied for site roles on import. Bold indicates that a site role was promoted to preserve the user's ability to publish.

Note: The **Import Site Role** row headers indicate the site role specified for import. The **Current Site Role** column headers represent the current user site role. The table values represent the resulting site role. A bold site role in the table indicates a site role promotion that preserves the ability to publish.

	Current Site Role						
Import Site Role	Site Administrator	Pub-lisher	Inter-actor	Viewer	Viewer (can publish)	Unli-censed	Unli-censed (can publish)
Site Administrator	Site Administrator	Site Administrator	Site Administrator				
Publisher	Site Admin-	Publisher	Publisher	Publisher	Publisher	Publisher	Publisher

	Current Site Role							
Import Site Role	Site Administrator	Pub-lisher	Inter-actor	Viewer	Viewer (can publish)	Unli-censed	Unli-censed (can publish)	
	istrator							
Inter-actor	Site Admin-istrator	Publisher	Inter-actor	Inter-actor	Pub-lisher	Inter-actor	Pub-lisher	
Viewer (can pub-lish)	Site Admin-istrator	Publisher	Pub-lisher	Viewer (can pub-lish)	Viewer (can pub-lish)	Viewer (can pub-lish)	Viewer (can pub-lish)	
Viewer	Site Admin-istrator	Publisher	Inter-actor	Viewer	Viewer (can pub-lish)	Viewer	Viewer (can publish)	
Unli-censed (can pub-lish)	Site Admin-istrator	Publisher	Pub-lisher	Viewer (can publish)	Viewer (can pub-lish)	Unli-censed (can pub-lish)	Unli-censed (can pub-lish)	
Unli-censed	Site Admin-istrator	Publisher	Inter-actor	Viewer	Viewer (can pub-lish)	Unli-censed	Unli-censed (can pub-lish)	

Guest User

A Guest user is available in Tableau Server to allow unauthenticated users without an account on the server see and interact with an embedded view. When enabled, the user can load a webpage containing an embedded visualization without signing in.

Note: The Guest user option is only available with a core-based license.

When you embed a Tableau Server view into an internal website page, every person who views that page will need a Tableau Server account (they'll be asked for a user name and password) unless you have purchased a core-based (hardware) license. In that case you can have as many accounts as you want, as well as the ability to enable Guest user access that requires no login or authentication.

Guest is a special account and can only be used to see views. The Guest user cannot browse the Tableau Server interface and won't see server interface commands the view (user name, account settings, comments, and so on).

Note: Enabling the Guest user for a site can increase the number of potential simultaneous viewers beyond the user list you might be expecting. The administrative view **Status > Traffic to Views** can help you gauge the activity.

A Guest user can have the following permissions

Projects, Workbooks, and Views: View, Export Image, Summary Data, View Comments, Filter, Full Data, Web Edit, Download (to save a local copy)

Data Sources: View and Download

When a Guest user is included in a group that has a permission rule set on a content item, Guest user permissions do not affect the permission-levels of other users in that group.

To enable Guest access

1. Select **Server > Settings > General** (multi-site server), or click **Settings > General** (single-site server).
2. For Guest Access, select **Enable Guest account** to allow people who are not signed into a Tableau Server account to see views with Guest access permissions.
3. Click **Save**.

The Guest user is unique in the following ways:

- The Guest user represents all unauthenticated users accessing content on the server.
- Tableau Server must use a core license for Guest to be available.
- Server administrators can enable/disable Guest across the server; it is not controllable per site.
- The Guest user cannot be edited and can never own content.
- The Guest user can be made a member of one or more groups in a site.
- Only the server administrator can enable or disable Guest access (in **Server > Settings > General**).
- The Guest user, when enabled, is a member of the All Users group.
- The Guest user cannot be deleted; it must be disabled by the server administrator in **Server > Settings > General**.
- If the Guest user needs to be able to access a workbook that uses an extract data

source, make sure Guest has the View permission for the data source. The Guest user is not allowed to connect to published data sources, unless the publisher embedded their credentials when publishing the content.

- The Guest user is not allowed to save customized views.

Add Users to a Site

Administrators can add users to sites in the following ways:

- By adding a local user account or a user account from Active Directory, as described in this topic. You can also add users by importing an Active Directory group. For details, see [Create a Group via Active Directory on page 161](#).
- By importing a CSV file that contains user information. For details, see [Import Users on page 194](#) and [CSV Import File Guidelines on page 200](#).

In a single-site environment, administrators can add users to a site on the Users page. In a multi-site environment, you will use the Site Users page. Server administrators must give site administrators the ability to add users to sites. This setting can be enabled or disabled by the server administrator (see step 4 in [Add or Edit Sites on page 117](#)).

Note: Users can be added to sites, or to the server. To add users to the server, see [Add Users to the Server on page 188](#). The options available for adding users depends on the authentication method that you select when you first configure Tableau Server. If you are using local authentication, you cannot add Active Directory users. If you are using Active Directory, you cannot add local users.

On the **Users** (single-site) or **Site Users** (multi-site) page you can see the users on the site you're currently signed into. You can add users to (or remove them from) the current site only. If a user belongs to more than one site, you can remove that user from the current site.

Note: When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn't own any content.

Display name	Username	Site role	Groups	Last signed in
Adam Davis	adavis	Publisher	1	
Admin	***	Server Administrator	2	Feb 10, 2015, 12:25 AM
Alan Wang	awang	Viewer (can publish)	1	
Alejandro Grove	agrove	Publisher	1	
Andrew Allen	aallen	Publisher	1	
Andrew Smith	asmith	Publisher	1	
Ashley Garcia	agarcia	Publisher	1	
Brendan Sweed	bsweed	Viewer	1	
Brosina Hoffman	bhoffman	Publisher	1	
Claire Gute	cgute	Server Administrator	1	
Darren Powers	dpowers	Interactor	1	
Darrin Van Huff	dvhuff	Site Administrator	1	
Emily Burns	eburns	Publisher	1	

Note: This screenshot is from a multi-site environment. In a single-site environment, this would be the Users page.

To add local users to a site

1. In a site, click **Users**, click **Add Users**, and then click **Local User**.

The dialog has two tabs: "Local User" (selected) and "Import From File". Below the tabs is a note: "Create a local user account on Tableau Server." At the bottom right is a "Cancel" button.

Henry Wilson	hwilson
Interactor	Interactor
Jane Johnson	jjohnson
Laura Rodriguez	lrodriquez
Michelle Kim	Publisher Interactor

Note: This screenshot is from a multi-site environment. In a single-site environment, this would be the Users page.

2. Enter a user name. If the server is configured for local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, *jsmith@example.com* instead of *jsmith*).

The screenshot shows a modal dialog titled "Add Local User". It contains the following fields:

Username:	jsmith
Username available	
Display name:	John Smith
Password:	*****
Confirm password:	*****
Email (optional):	
Site role:	Publisher

At the bottom right are two buttons: "Cancel" and "Add User".

Also enter information in the following fields:

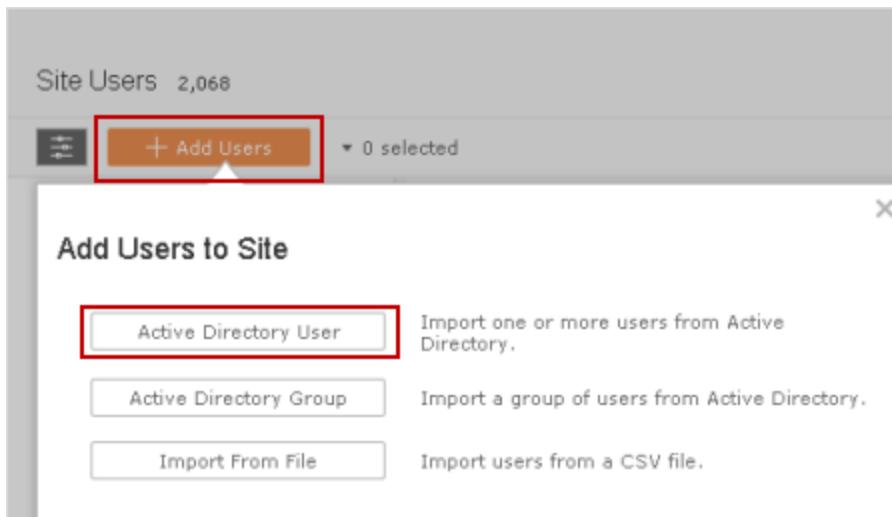
- **Display Name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.

3. Select a site role. For details on site roles, see [Site Roles for Users](#) on page 176.
4. Click **Add User**.

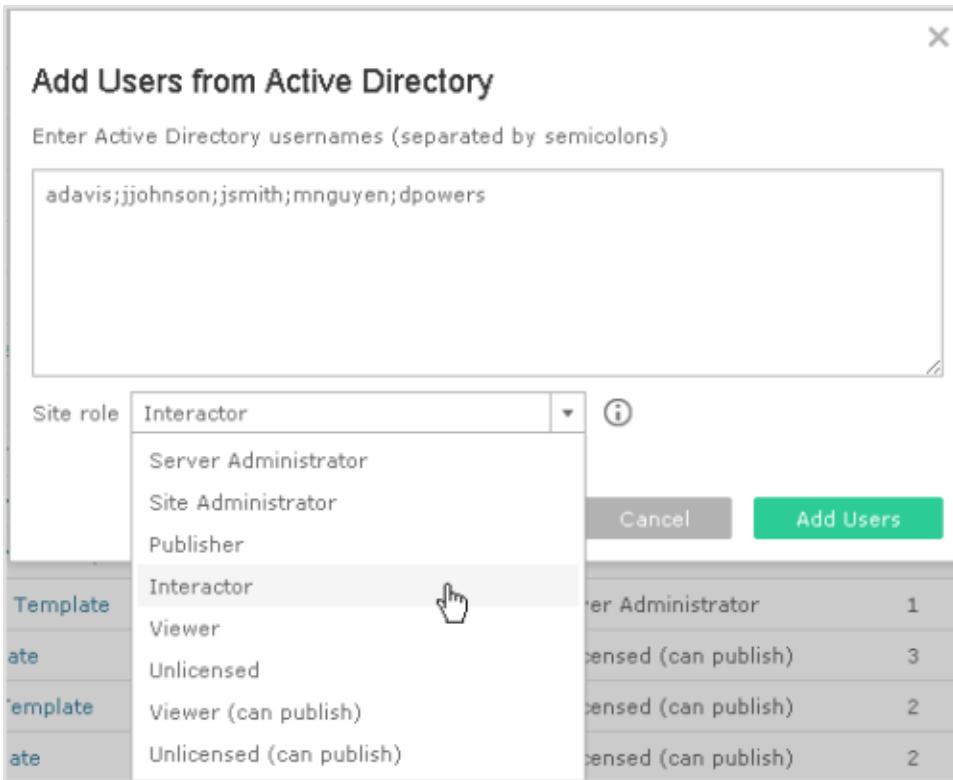
Note for multi-site servers: A site administrator can edit an existing local user account only if the administrator has control over all of the sites the user is a member of. For example, if User1 is a member of sites A and B, an administrator of site B only cannot edit User1's full name or reset the password.

To add Active Directory users to a site

1. In a site, click **Users**, and then click **Add Users**, and then click **Active Directory User**.



1. Enter one or more user names (separated by semicolons). If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server's domain will be assumed.



If there is a two-way trust set up between the server's domain and another domain, you can add users from both domains. The first time you add a user from the “non-server domain,” use the fully-qualified domain name with the username. Subsequent users can be added using [the domain's nickname](#). For example, assuming a “non-server domain” of *mybiz.lan*, enter the first user from that domain as *user1@mybiz.lan* or *mybiz.lan\user1*. The next user can be entered using the domain’s nickname, such as *user2@mybiz* or *mybiz\user2*.

Note: Do not enter the user’s full name in this field; it can cause errors during the importing process.

2. Select a site role. For details on site roles, see [Site Roles for Users](#) on page 176.
3. Click **Add Users**.

Add Users to the Server

In a single-site environment, server administrators can add users on the **Users** page.

The screenshot shows the 'Site' tab selected in the top navigation bar. Below it, the 'Users' tab is also selected. The main content area displays a table titled 'Users 18'. The table has columns for 'Display name', 'Username', and 'Site role'. The data includes:

Display name	Username	Site role
Adam Davis	adavis	Publisher
Admin	Admin	Server Administrator
Alan Smith	asmith	Publisher
Guest	guest	Guest
Henry Wilson	hwilson	Interactor

After you add a site to Tableau Server, it becomes a multi-site server with a **Server Users** page (all server users from every site appear here) and a **Site Users** page. Only server administrators can access the **Server Users** page.

The **Server Users** page is the only place where you can assign users to multiple sites, delete users from the server, and if the server is using local authentication, reset user passwords.

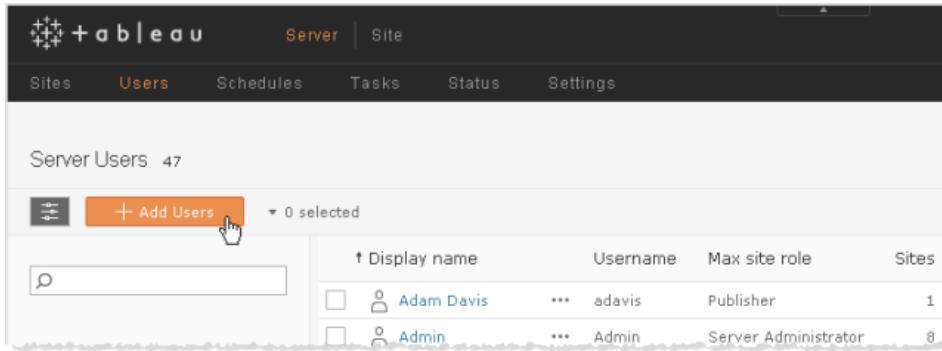
The screenshot shows the 'Server' tab selected in the top navigation bar. Below it, the 'Users' tab is selected. The main content area displays a table titled 'Server Users 47'. The table has columns for 'Display name', 'Username', 'Max site role', 'Sites', and 'Last signed in'. The data includes:

Display name	Username	Max site role	Sites	Last signed in
Adam Davis	adavis	Publisher	1	
Admin	Admin	Server Administrator	8	Feb 10, 2015, 12:25 AM
Alan Wang	awang	Viewer (can publish)	1	
Alejandro Grove	agrove	Publisher	1	
Andrew Allen	aallen	Publisher	1	
Andrew Smith	asmith	Publisher	1	
Ashley Garcia	agarcia	Publisher	1	
Brendan Sweed	bsweed	Viewer	1	
Brosina Hoffman	bhoffman	Publisher	1	
Claire Gute	cgute	Server Administrator	8	
Darren Powers	dpowers	Interactor	1	
Darrin Van Huff	dvhuff	Site Administrator	1	
Emily Burns	eburns	Publisher	1	

The following procedure describes how to add users to the server. There are two approaches you can take: One at a time (described below) or in batches using the **Import** command, which relies on a CSV file (described in [Import Users on page 194](#) and [CSV Import File Guidelines on page 200](#)).

To add a user to the server

1. Click **Server > Users**, and then click **Add Users**.



The screenshot shows the Tableau Server interface with the 'Users' tab selected. The title bar says '+ tableau'. Below it is a navigation bar with 'Sites', 'Users' (highlighted in orange), 'Schedules', 'Tasks', 'Status', and 'Settings'. The main area is titled 'Server Users 47'. It features a search bar and a 'Add Users' button with a plus sign. A table lists users with columns: Display name, Username, Max site role, and Sites. Two users are listed: 'Adam Davis' (Publisher, 1 site) and 'Admin' (Server Administrator, 8 sites). The 'Add Users' button has a cursor icon pointing to it.

2. If you are using local authentication, click **Local User**. If you are using Active Directory, click **Active Directory User**.

Enter a user name.

- **Local authentication:** If the server is using local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, *jsmith@example.com* instead of *jsmith*).
- **Active Directory:** If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server domain will be assumed.

If there is a two-way trust set up between the server's domain and another domain, you can add users from both domains. The first time you add a user from the non-server domain, use the fully-qualified domain name with the user name. Subsequent users can be added using [the domain's nickname](#). For example, assuming a non-server domain of *mybiz.lan*, enter the first user from that domain as *user1@myco.lan* or *mybiz.lan\user1*. The next user can be entered using the domain's nickname, such as *user2@mybiz* or *mybiz\user2*.

Note: Do not enter the user's full name in this field; it can cause errors during the importing process.

3. If the server is using local authentication, provide the following:

- **Display Name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.
- **Server Administrator**: Specify whether the user should be a server

administrator.

- **Name (Site Membership) / Site Role:** If the user is not a server administrator, you can assign a user to zero or more sites, along with a site role for each site. You do not have to choose site membership and site role at this time. If you don't specify site membership and site role for a new server user, the user will be added as a Server User only, with a site role of Unlicensed. For details on site roles, see [Site Roles for Users on page 176](#).

The screenshot shows the 'Add Local User' dialog box. It includes fields for Username (jsmith), Display name (John Smith), Password and Confirm password (both masked as *****), and Email (optional). Below these fields is a search bar and a list of site roles. The 'Finance' role is selected, and its Site role is set to 'Publisher'. At the bottom are 'Cancel' and 'Add User' buttons.

Name	Site role
<input type="checkbox"/> Documentation - 20 User Limit	
<input checked="" type="checkbox"/> Finance	Publisher
<input type="checkbox"/> Human Resources	
<input type="checkbox"/> Server Admin	

4. Click **Add User**.

Add Users to a Group

One way to simplify user management is to assign users to groups. For example, you can assign permissions to a group to apply them to all users in the group.

To add a user to a group, the group must already exist. For information, see [Groups on page 159](#).

Add users to a group (Users page)

1. In a site, click **Users**.
2. Select the users you want to add to a group, and then select **Actions > Group Membership**.

The screenshot shows a modal dialog titled "Group Membership". The instruction "Assign groups to the 4 selected users." is displayed above a search bar. A table lists groups with their names and member counts. The "Marketing" group has a checked checkbox and a count of 0. Other groups listed are All Users (18), Development (0), Finance (6), HR (0), IT (0), Operations (0), Sales (0), and Site Role Max Permissions (9). At the bottom are "Cancel" and "Save" buttons.

Name	Members
All Users	18
<input type="checkbox"/> Development	0
<input type="checkbox"/> Finance	6
<input type="checkbox"/> HR	0
<input type="checkbox"/> IT	0
<input checked="" type="checkbox"/> Marketing	0
<input type="checkbox"/> Operations	0
<input type="checkbox"/> Sales	0
<input type="checkbox"/> Site Role Max Permissions	9

3. Select the groups and then click **Save**.

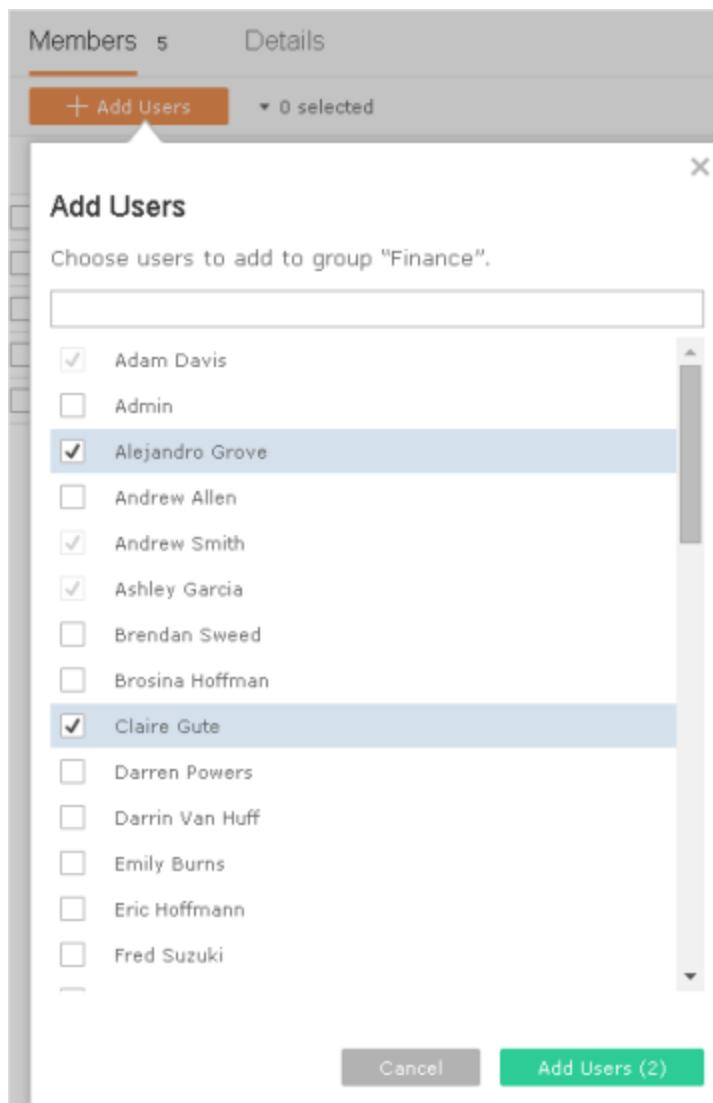
Add users to a group (Groups page)

1. In a site, click **Groups**, and then click the name of the group.
2. In the group's page, click **Add Users**.

The screenshot shows the Tableau Server interface. At the top, there's a navigation bar with the Tableau logo, 'Server', and 'Site: Default'. Below the navigation bar is a menu bar with 'Content', 'Users', 'Groups', 'Schedules', and 'Tasks'. The 'Groups' option is highlighted. Underneath the menu bar, the path 'All Groups > Finance' is displayed. The main content area shows a group named 'Finance' (GROUP) with a 'Domain: local'. There are two tabs: 'Members' (selected) and 'Details'. The 'Members' tab shows 5 users: Adam Davis, Andrew Smith, Ashley Garcia, Jane Johnson, and Laura Rodriguez. Each user has a checkbox next to their name. Below the table is a button labeled '+ Add Users'.

	Display name	Username	Site role
<input type="checkbox"/>	Adam Davis	adavis	Publisher
<input type="checkbox"/>	Andrew Smith	asmith	Publisher
<input type="checkbox"/>	Ashley Garcia	agarcia	Publisher
<input type="checkbox"/>	Jane Johnson	jjohnson	Publisher
<input type="checkbox"/>	Laura Rodriguez	lrodriguez	Viewer (can publish)

3. Select the users to be added, and then click **Add Users**.



Import Users

To automate the process of adding users to a site, you can create a CSV file that contains user information, and then import the file. You can import users to a site, or, to the server (if you are a server administrator).

Note: This topic contains the steps for importing, assuming that you have already created the CSV file. If you have not created the file yet, see [CSV Import File Guidelines](#) on page 200 for a list of file format requirements and import options.

Add users from a CSV file to a site

1. In a site, click **Users**, click **Add Users**.

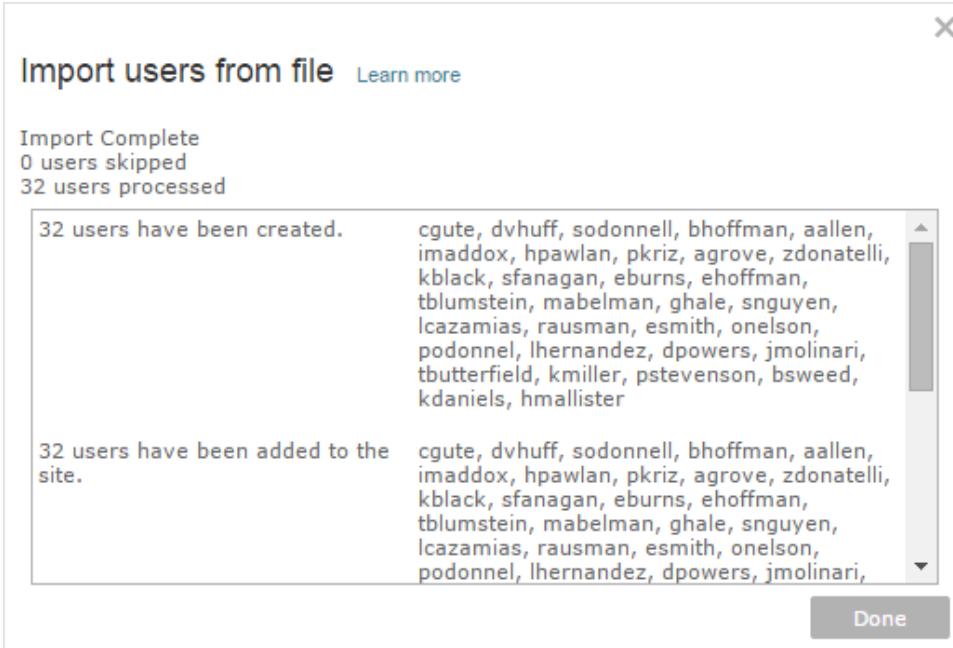
The screenshot shows the 'Add Users to Site' dialog box. At the top, there are two options: 'Local User' and 'Import From File'. The 'Import From File' option is highlighted with a red box. Below this, a list of users is displayed, including Henry Wilson, Interactor, Jane Johnson, Laura Rodriguez, and Michelle Kim. Each user has a checkbox next to their name.

User	Role
Henry Wilson	hwilson
Interactor	Interactor
Jane Johnson	jjohnson
Laura Rodriguez	lrodriquez
Michelle Kim	Publisher Interactor

2. Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

The screenshot shows the 'Import users from file' dialog box. It has a 'File Name:' input field containing 'localUsers.csv', a 'Browse...' button, and two buttons at the bottom: 'Cancel' and 'Import Users'. The 'Import Users' button is highlighted with a red box.

The results of the import are displayed.

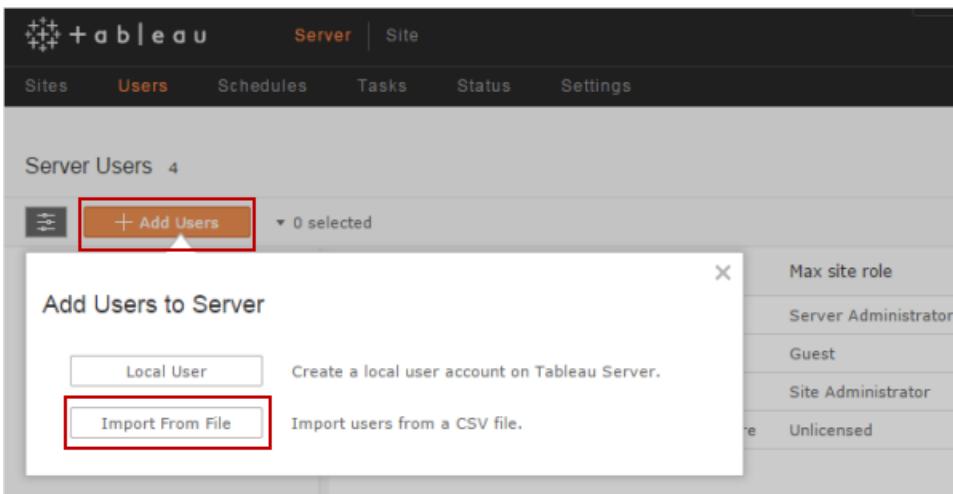


For a single-site server, the site roles assigned to the users during the import process will be imported with the users. If a user already exists in the Tableau Server site, the site role assigned during the import process will be applied only if it gives the user more access to the server. Importing users and groups will promote a user's site role, but not demote a user's site role.

3. Click **Done**.

Add users from a CSV file to a server

1. Click **Server > Users**, and then click **Add Users**.



- Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

The results of the import are displayed.

Max site role	Sites	Last signed in
Unlicensed	0	
Server Administrator	8	Feb 9, 2015, 12:24 PM
Unlicensed	0	
Server Administrator	8	
Unlicensed	0	
Guest	8	

For a multi-site server, when you import users in the Server Users page, you are creating server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a server user can have is either Unlicensed or Server Administrator. When you assign site membership to a server user, you can specify the site role for that user per site. For details, see [Assign Site Membership on page 210](#). If you import the users in the Site Users page, the users will be assigned the site roles you specify in the CSV file, for that site.

- Click **Done**.

Multi-site environments

If the server is running multiple sites and you are a server administrator, you can import a CSV file from two different locations. Where existing user accounts are concerned, each location has different capabilities.

- The **Server Users** page appears in a multi-site environment. Only server administrators can access this page.

Display name	Username	Max site role	Sites	Last signed in
Adam Davis	adavis	Publisher	1	
Admin	Admin	Server Administrator	8	Feb 9, 2015, 1:59 PM
Alan Wang	awang	Viewer (can publish)	1	
Alejandro Grove	agrove	Publisher	1	
Andrew Allen	aallen	Publisher	1	
Andrew Smith	asmith	Publisher	1	
Ashley Garcia	agarcia	Publisher	1	
Brendan Sweed	bsweed	Viewer	1	
Brosina Hoffman	bhoffman	Publisher	1	
Claire Gute	cgute	Server Administrator	8	

You can import the CSV file from here if you want to update existing user accounts in addition to adding new ones. For example, if you import a file that has a new password for each existing user, their passwords will be reset.

- The **Site Users** page.

Display name	Username	Site role	Groups	Last signed in
Adam Davis	adavis	Publisher	1	
Admin	Admin	Server Administrator	2	Feb 9, 2015, 1:59 PM
Alan Wang	awang	Viewer (can publish)	1	
Alejandro Grove	agrove	Publisher	1	
Andrew Allen	aallen	Publisher	1	
Andrew Smith	asmith	Publisher	1	
Ashley Garcia	agarcia	Publisher	1	
Brendan Sweed	bsweed	Viewer	1	
Brosina Hoffman	bhoffman	Publisher	1	
Claire Gute	cgute	Server Administrator	1	
Darren Powers	dpowers	Interactor	1	

Server administrators can add new user accounts with CSV imports and, if existing users are part of the import, the **Password** and **Display Name** fields must either match or be left blank. If new passwords or full names are used, the import will fail.

Single-site environments

Server and site administrators on a single-site server perform CSV user imports from the **Users** page in a site.

	Display name	Username	Site role
<input type="checkbox"/>	Adam Davis	adavis	Publisher
<input type="checkbox"/>	Admin	Admin	Server Administrator
<input type="checkbox"/>	Alan Smith	asmith	Publisher
<input type="checkbox"/>	Guest	guest	Guest
<input type="checkbox"/>	Henry Wilson	hwilson	Interactor
<input type="checkbox"/>	Interactor	Interactor	Interactor
<input type="checkbox"/>	Jane Johnson	jjohnson	Interactor
<input type="checkbox"/>	Michelle Kim	mkim	Publisher
<input type="checkbox"/>	NoSiteRole	NoSiteRole	Unlicensed
<input type="checkbox"/>	Project Owner	Project Owner	Publisher
<input type="checkbox"/>	Publisher	Publisher	Publisher
<input type="checkbox"/>	Sam Jones	Sam	Interactor
<input type="checkbox"/>	SiteAdmin	SiteAdmin	Site Administrator

Multi-site versus single-site import

Users can belong to more than one site on the same server, but they must use the same credentials for each site. This becomes important when you're adding users to a site and those users might already be members of a different site. If you try to import a user who already exists, and if the user's credentials in the CSV file don't match the existing credentials, the import fails for that user.

Note: The issue of credentials mismatch during import doesn't apply if the server is configured to use Active Directory for authentication. In that case, the CSV file should never contain a password, because user passwords are managed by Active Directory.

If you're importing users into a site and you think that the users might already exist on the server, you can try leaving the `Password` column in the CSV file blank. When you import the users, if a user who is defined in the CSV already exists in another site, the user is added to the site where you're importing. However, if the user *doesn't* already exist on the server, the user is created, and the CSV import window alerts you that the new user doesn't have a password. You can then use the server environment to assign a password to any user who doesn't have one.

Multi-site

For a multi-site server, when you import users in the Server Users page, you are creating server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a server user can have is either Unlicensed or Server Administrator.

Single site

For a single-site server, the site roles assigned to the users during the import process will be imported with the user. If a user you are importing already exists in Tableau Server, the site role assigned during the import process will be applied only if it gives the user more access to the server. Importing users and groups will promote a user's site role, but not demote a user's site role.

CSV Import File Guidelines

You can automate adding users by creating a comma-separated values (CSV) file with user information and then importing the file. You can include attributes in the CSV file, such as site role and the ability to publish, to apply to the users at the same time you import them.

To import users, you can use the server administration pages or the `tabcmd` utility. For details, see [Import Users on page 194](#) or [createsiteusers filename.csv on page 559](#).

Note: If you use the `tabcmd` utility to import users, you can pass options on the command line that can specify default values for the users' site roles. For more information, see the [createsiteusers filename.csv on page 559](#) documentation.

You can import users into a site or into the server. If you import users into a site, site roles are applied to the user. If you specify site roles, but importing users would exceed your license limits, users are imported as Unlicensed. If you import users into the server (not into a specific site), the user isn't assigned to a site, and site roles in the CSV file like Publisher and Interactor are treated as Unlicensed.

CSV File Format Requirements

When you create the CSV file for importing users, make sure that the file meets the following formatting requirements:

- The file does not include column headings. Tableau Server assumes that every line in the file represents a user.
- The file is in UTF-8 format, and includes the byte-order mark (BOM).
- Character encodings such as BIG-5 have been converted to UTF-8. You can do this by opening the file in a text editor and using the **Save As** command.

Required Columns in the CSV File

The following values are required for each user:

- Username
- Password: If Tableau Server is configured to use Active Directory authentication, there

must be a Password column, but the column itself should be empty. If the server is using local authentication, you must provide passwords for new users.

Additional Import File Options

The CSV file can contain the following fields, in the order shown here:

- User name: The user name. If the server is configured to use Active Directory, this value must match a user defined in Active Directory. If the user name is not unique across domains, you must include the domain as part of the user name (for example, example\Adam or adam@example). This is the only required field.
- Password: A password for the user. If the server is configured to use Active Directory, this value is not used.
- Full name. The full name (display name) is part of the information that's used to identify a user on the server. If the user's full name is already in use, Tableau Server updates the existing user information with the settings in the CSV file. If the server is configured using Active Directory, this value is not used.
- License level (Interactor, Viewer, or Unlicensed). This setting determines the role for a non-administrator user. If you are using the server administration pages to import users, the license level is set only if you are importing into an individual site. If you are using the server administration pages to import users while managing the server (not a specific site), and if the user is not set to be an administrator, the site role is set to Unlicensed. (You can change the site role later.)

Note: In Tableau Server 9.0, license levels have been replaced with site roles. If you create a user using the server UI, you select a site role like Site Administrator, Publisher, Interactor, and View (can publish). For information about site roles, see [Site Roles for Users on page 176](#). For more information about how the license levels and other values in the CSV file are converted to site roles, see [Settings and Site Roles](#) later in this topic.

- Administrator level (System, Site, or None). This setting determines whether the user is imported as an administrator. If you are using the site administration pages, you can set the administrator role to System only if you are importing while managing the server. If you are using the server administration pages to import users while you are managing a site, and if the administrator role for a user in the CSV file is set to System, Tableau Server imports the user as a site administrator.
- Publisher permissions (yes/true/1 or no/false/0). This setting determines whether the user has publisher permissions. If you are using the site administration pages, the publisher setting is used only if you are importing into an individual site. If you are importing users while managing a server, this value isn't used.

- Email address. The email address is part of the information that's used to identify a user on the server. If the email address is already in use, Tableau Server updates the existing user information with the settings in the CSV file.

The order of the columns is significant. The first column is treated as the user name, the second as the password, the third as full name, etc., regardless of the content in the columns.

Settings and Site Roles

The license level, administrator, and publisher settings for a user are used during the import process to set a user's site role. The following table shows how the settings are converted to site roles.

CSV settings	Site role
License level=(any) Administrator=System Publisher=(any)	System (server) administrator. This setting is valid only if you are importing users while managing the server. If you set a user to be a system administrator, the other values are ignored.
License level=(any) Administrator=Site Publisher=(any)	Site administrator. This setting is valid only if you are importing users while managing a specific site. If you set a user to be a site administrator, the other values are ignored.
License level=Interactor Administrator=None Publisher=true	Publisher
License level=Interactor Administrator=None Publisher=false	Interactor
License level=Viewer Administrator=None Publisher=true	Viewer (can publish)
License level=Viewer Administrator=None	Viewer

CSV settings	Site role
Publisher=false	
License level=Unlicensed Administrator=None Publisher=true	Unlicensed (can publish)
License level=Unlicensed Administrator=None Publisher=false	Unlicensed

Notes

- If you are importing users while managing the server, you can create users with only two site roles: system (server) administrator and Unlicensed. All other settings are site specific. In that case, if the administrator level for a user in the CSV file is not System, the user's site role is set to Unlicensed.
- If you have a user-based server installation, and if adding users would exceed the number of users allowed by your license, the users are added as unlicensed users.

Example

The following example shows a CSV file that contains information for several users.

```
Henry W,passw0rd,Henry,Interactor,None,yes,henryw@example.com
Fred S,pa$$word,Fred,Viewer,None,no,freds@example.com
Alan W,p@ssword,Alan,Interactor,Site,yes,alanw@example.com
Michelle K,my-
password,Michelle,Interactor,System,yes,michellek@example.com
```

If you import this file while managing a site, four users are added to that site. The Administrator mode for user Michelle is set to System. However, because you are importing the users into a site, Tableau Server sets user Michelle to be a site administrator, not a system administrator. Three of the users are allowed to publish.

If you import this file while managing the server, four users are added to the server, but they are not added to any site. The site roles in the CSV file (Interactor and Viewer) must be associated with site users, so the site role for the users who are not administrators is set to Unlicensed.

View, Edit, and Delete Users

View and edit site users

Sign in to a site as an administrator, and then click **Users**. In this page you can set group membership, set site role, or remove the user from the site.

The screenshot shows the 'Site Users' page with 46 users listed. A user named 'Andrew Smith' is selected, indicated by a checked checkbox in the 'Actions' column. A red box highlights the 'Actions' dropdown menu, which includes options like 'Group Membership', 'Site Role', and 'Remove'. The table columns are 'Username', 'Site role', 'Groups', and 'Last signed in'.

	Username	Site role	Groups	Last signed in
<input type="checkbox"/>	adavis	Publisher	1	
<input type="checkbox"/>	Admin	Server Administrator	2	Feb 10, 2015, 2:09 PM
<input type="checkbox"/>	Alan Wang	Viewer (can publish)	1	
<input type="checkbox"/>	Alejandro Grove	Publisher	1	
<input type="checkbox"/>	Andrew Allen	Publisher	1	
<input checked="" type="checkbox"/>	Andrew Smith	Publisher	1	
<input type="checkbox"/>	Ashley Garcia	Publisher	1	

Click a user name to see the content they own.

The screenshot shows the user profile for 'Andrew Smith'. It displays sections for 'Workbooks', 'Views', 'Data Sources', and 'Subscriptions'. Below these are filters for 'Project', 'Tag', and date ranges. On the right, there are three visualizations: 'Tale of 100 Start-ups', 'Economic Indicators', and 'Investing in the Dow'.

Click **Settings** in a user page to view their account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.

The screenshot shows the Tableau Server interface with the navigation bar 'Server | Site: Default'. Below it, the 'Content', 'Users', 'Groups', 'Schedules', 'Tasks', 'Status', and 'Settings' tabs are visible, with 'Users' being the active tab. Under 'All Site Users', a user named 'Andrew Smith' is selected. The user details show 'local\asmith' as the login, 'Publisher' as the site role, and 'Never' as the last sign-in date. The 'Settings' tab is highlighted with a red box. The user's profile information includes fields for 'Username' (asmith), 'Display name' (Andrew Smith), 'Email' (empty), and 'Start page' (/). There are dropdown menus for 'Language' (Unspecified) and 'Locale' (Unspecified). Buttons for 'Save Changes' and 'Reset to Default' are present.

If Tableau Server is running multiple sites, **Server Users** lists all users on the server system, and **Site Users** displays all users for the current site.

If the server is configured to use the internal user management system (Local Authentication), you can edit the **Display Name**, **Email**, and **Password** for users after they have been added. If you are making many changes, you may find it easier to import the changes from a CSV file. For details, see [Import Users](#) on page 194 and [CSV Import File Guidelines](#) on page 200.

For multi-site servers: Site administrators can edit an existing user's account as long as the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings. For example, if User Joe is a member of Site A and Site B and the site administrator is only an administrator of Site B, the site administrator cannot edit Joe's Full Name or reset his password.

View and edit server users

Sign into Tableau Server as a server administrator. Click **Server > Users**. In this page you can set site membership or delete the user from the server.

		Actions	Username	Max site role	Sites	Last signed in
<input type="checkbox"/>	<input type="checkbox"/>	Edit site membership	adavis	Publisher	1	
<input type="checkbox"/>	<input type="checkbox"/>	Delete	Admin	Admin	8	Feb 10, 2015, 2:09 PM
<input type="checkbox"/>	<input type="checkbox"/>		Alan Wang	Viewer (can publish)	1	
<input type="checkbox"/>	<input type="checkbox"/>		Alejandro Grove	Publisher	1	
<input type="checkbox"/>	<input type="checkbox"/>		Andrew Allen	Publisher	1	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Andrew Smith	asmith	Publisher	1	

Click a user name to view account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.

USER local\asmith • Max site role: Publisher • Last sign in: Never

Settings

Username	asmith
Display name	Andrew Smith
Email	
Change Password	
Start page	/
Reset to Default	
Language	Unspecified
Locale	Unspecified

Search for users

To search for a specific user, in the **Search** box on the left, type all or part of the user's name, and then press **Enter**.

The search operation checks the display name and user name attributes.

You can use the asterisk (*) character as a search wildcard. For example, searching for *John** will return all user names that start with *John*.

The screenshot shows the Tableau Server interface with the 'Site: Default' selected. The top navigation bar includes 'Content', 'Users' (which is highlighted in orange), 'Groups', 'Schedules', 'Tasks', 'Status', and 'Settings'. Below the navigation is a section titled 'Site Users 2'. It features a search bar with a placeholder 'Clear All Filters' and a search input field containing 'An', which is also highlighted with a red box. To the right of the search bar are buttons for 'Add Users' and '0 selected'. The main area displays a table with columns for 'Display name', 'Username', and 'Site role'. Two users are listed: 'Andrew Allen' (aallen, Publisher) and 'Andrew Smith' (asmith, Publisher).

Remove users from a site

You can remove a user from a site only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but not removed.

Note: When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn't own any content.

1. In a site, click **Users**. Select one or more users to delete, and then select **Actions > Remove**.

The screenshot shows the 'Site Users' page with 46 users listed. A user named 'Alan Wang' is selected, indicated by a checked checkbox in the first column. An 'Actions' dropdown menu is open over the selected user, showing options like 'Group Membership' and 'Site Role'. The 'Remove' option is highlighted with a red box. The table columns include 'Display name', 'Username', and 'Site role'. Other users listed include 'adavis' (Publisher), 'Admin' (Server Administrator), 'Andrew Allen' (aallen, Publisher), 'Andrew Smith' (asmith, Publisher), and 'Alejandro Grove' (agrove, Publisher).

2. Click **Remove** in the confirmation dialog.

Delete users from the server

You can delete a user from Tableau Server only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to delete a user who owns content, the user

site role will be set to Unlicensed, but the user will not be deleted.

If a user is a member of multiple sites, and owns content in one or more of those sites, the user will be removed from the sites in which they don't own content. The user will remain a member in sites where they do own content, but demoted to the Unlicensed site role.

1. In a site, click **Users**. Select one or more users to delete, and then select **Actions > Delete**.

The screenshot shows the 'Server' tab selected in the top navigation bar. Below it, the 'Users' tab is active. A search bar and a 'Add Users' button are visible. A table lists 'Server Users' with 58 entries. The columns are 'Display Name', 'Site Membership', 'Username', and 'Max site role'. A context menu is open over the user 'Andrew Smith', with the 'Delete' option highlighted. The table data is as follows:

Display Name	Site Membership	Username	Max site role
Adam Davis	...	adavis	Publisher
Admin	...	Admin	Server Administrator
Alejandro Grove	...	agrove	Publisher
Andrew Allen	...	aallen	Publisher
Andrew Smith	...	asmith	Publisher
Ashley Garcia	...	agarcia	Publisher

2. Click **Delete** in the confirmation dialog box.

Change passwords for users of a single site

To change the password for a user with membership to a single site, sign in to Tableau Server as a site administrator or a server administrator.

1. Ensure that the correct site is selected in the menu.
2. Click **Users**.
3. Click the display name of a user.
4. Click **Settings** in the menu below the display name of the user.
5. Click the **Change Password** link.

The screenshot shows the Tableau Server interface. At the top, there's a navigation bar with icons for Home, Content, Users (which is selected), Groups, Schedules, Tasks, Status, and Settings. Below this, a breadcrumb trail shows 'All Site Users > Jane Johnson'. The main area displays user details: 'USER local\jjohnson' with 'Site role: Interactor', 'Last sign in: Never', and 'Email: jjohnson@example.com'. Below these details are tabs for Workbooks, Views, Data Sources, Subscriptions, and Settings (which is currently selected). Under the Settings tab, there are three input fields: 'Username' (jjohnson), 'Display name' (Jane Johnson), and 'Email' (jjohnson@example.com). A red box highlights the 'Change Password' button below these fields. To the right of the input fields is a 'Save Changes' button.

Change passwords for users of multiple sites

To change the password of a user with membership to multiple sites, sign in to Tableau Server as a server administrator.

1. Click **Server**.
2. Click **Users**.
3. Click the display name of a user.
4. Click the **Change Password** link.

Change Site Roles

Server administrators and site administrators with the ability to add site users can change the site role of a user at any time. For details on site roles, see [Site Roles for Users on page 176](#).

Only server administrators can change the site membership of users. For details, see [Assign Site Membership on the next page](#).

1. In a site, click **Users**.
2. Select one or more users, and then select **Actions > Site Role**.

The screenshot shows the 'Site Users' page with 46 users listed. A user named 'Andrew Smith' has a checkmark next to their name, indicating they are selected. A red box highlights the 'Actions' dropdown menu, which is open and shows options: 'Group Membership', 'Site Role', and 'Remove'. The 'Site Role' option is being pointed at by a cursor.

	Username	Site role	Groups
<input type="checkbox"/>	adavis	Publisher	1
<input type="checkbox"/>	Admin	Server Administrator	2
<input type="checkbox"/>	Alan Wang	Viewer (can publish)	1
<input type="checkbox"/>	Alejandro Grove	Publisher	1
<input type="checkbox"/>	Andrew Allen	Publisher	1
<input checked="" type="checkbox"/>	Andrew Smith	Publisher	1
<input type="checkbox"/>	Ashley Garcia	Publisher	1

3. Select a site role, and then click **Change Site Role**.

The screenshot shows a modal dialog box titled 'Site Role'. It asks to choose a site role for the user 'Andrew Smith'. A dropdown menu lists several roles: Interactor, Server Administrator, Site Administrator, Publisher, Interactor, Viewer, Unlicensed, Viewer (can publish), and Unlicensed (can publish). The 'Interactor' role is currently selected. A red box highlights the 'Change Site Role' button at the bottom right of the dialog. The background shows the same Site Users list as the previous screenshot, with 'Andrew Smith' now listed under the 'Publisher' role.

Assign Site Membership

Server administrators and site administrators with the ability to add site users can change the site role of a user at any time. For details on site roles, see [Site Roles for Users on page 176](#).

Only server administrators can change the site membership of users.

1. Click **Server > Users**.
2. Select one or more users, and then select **Actions > Site Membership**.

The screenshot shows a user interface for managing server users. At the top, there's a navigation bar with tabs: Sites, Users (which is selected), Schedules, Tasks, Status, and Settings. Below the navigation is a search bar and a button labeled '+ Add Users'. A message 'Server Users 47' is displayed. The main area is a table with columns: Actions, Site Membership, Username, Max site role, and Sites. The 'Actions' column contains icons for edit and delete. The 'Site Membership' column has a dropdown menu with options 'Edit' and 'Delete', both of which are highlighted with a red box. The 'Username' column lists user names like Adam Davis, Admin, Alan Wang, Alejandro Grove, Andrew Allen, and Andrew Smith. The 'Max site role' column shows roles such as Publisher, Server Administrator, Viewer (can publish), and Interactor. The 'Sites' column indicates the number of sites each user is associated with. A red box highlights the row for Andrew Smith.

Actions	Site Membership	Username	Max site role	Sites
	Edit	Adam Davis	Publisher	1
	Edit	Admin	Server Administrator	8
	Edit	Alan Wang	Viewer (can publish)	1
	Edit	Alejandro Grove	Publisher	1
	Edit	Andrew Allen	Publisher	1
	Edit	Andrew Smith	Publisher	1

3. Select one or more sites, and a role for each site, and then click **Save**.

The screenshot shows a modal dialog titled 'Site Membership' with the sub-instruction 'Edit site membership for the 5 selected users.' The dialog contains a table with two columns: 'Name' and 'Site role'. There are five rows, each corresponding to a selected user from the previous screen. The first row has a checked checkbox next to 'Customer Support' and a dropdown menu set to 'Publisher'. The second row has a checked checkbox next to 'Default' and a dropdown menu set to 'Publisher'. The third row has an unchecked checkbox next to 'Development'. The fourth row has a checked checkbox next to 'Finance' and a dropdown menu set to 'Interactor'. The fifth row has an unchecked checkbox next to 'Documentation - 20 User Limit'. The bottom of the dialog has 'Cancel' and 'Save' buttons.

Name	Site role
Customer Support	Publisher
Default	Publisher
Development	
Documentation - 20 User Limit	
Finance	Interactor
Human Resources	
Operations	
Sales	

Cancel Save

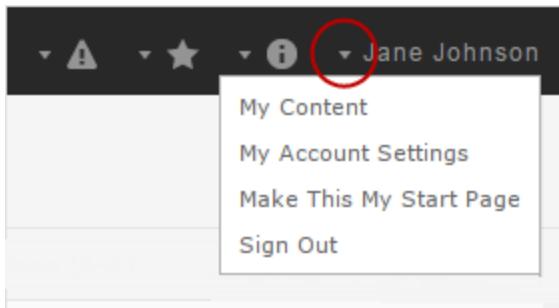
Manage Your Content and Account Settings

Use your Content page to quickly browse items that you've published and your subscriptions.

Use your Accounts Settings page to change your display name and password (local users only), add or change your email address, manage your subscription settings, change your start page, change the language and locale you see in Tableau Server, or clear cookies for data connection passwords.

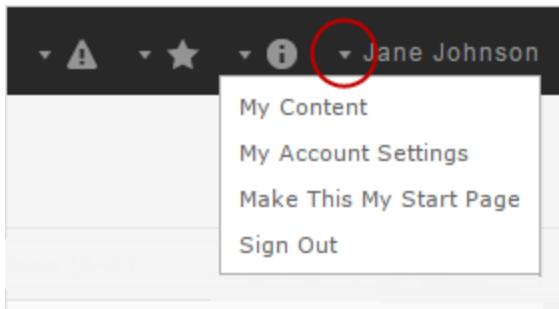
Access your Account Settings page

- Click your name at the top of the page, and then click **My Account Settings**. Or, click your name at the top of the page, click **My Content**, and then click **Settings**.



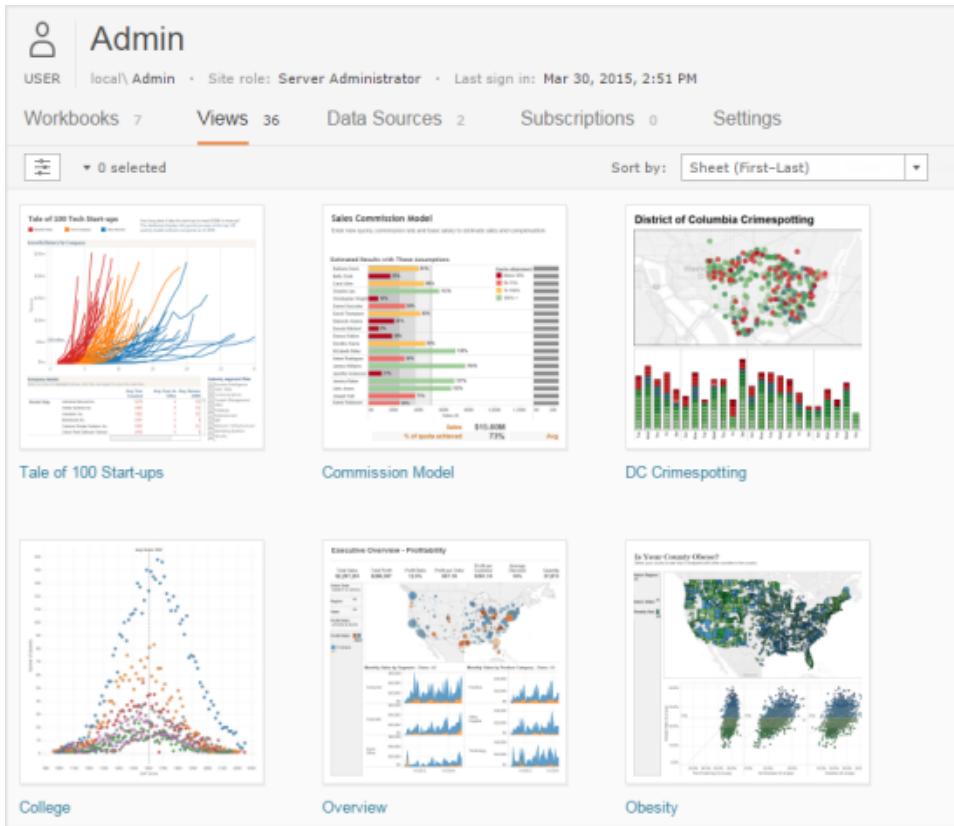
Access your Content page

- Click your name at the top of the page, and then click **My Content**.



Quickly Access Your Content

To access any content that you have published to the server, click your name at the top of the page, and then click **My Content**.



Change Your Display Name

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your display name. Select the display name text and enter the new display name, and then click **Save Changes**.

The screenshot shows a user profile edit form with the following fields:

- Username:** jjohnson
- Display name:** Jane Johnson (Lead) (highlighted with a red border)
- Email:** jjohnson@myco.com
- Change Password:** Link at the bottom left
- Save Changes:** Button at the bottom right

Change Your Password

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your Tableau Server password by clicking **Change Password**. When you click this link you are asked to enter your **Current Password** and the **New Password** (twice). After you've typed in the required information, click **Save Password** to save the changes.

Username	jjohnson
Display name	Jane Johnson (Lead)
Email	jjohnson@myco.com
Save Changes	
Change Password	

Username	jjohnson
Display name	Jane Johnson (Lead)
Email	jjohnson@myco.com
Current password	*****
New password	*****
Confirm password	*****
Save Changes	
Save Password	

Change Your Email Address

If you have a subscription for a Tableau Server view or workbook, the email account that receives the subscription is listed on the Account Settings page.

To enter or change the email address that Tableau Server sends subscriptions to, enter the new email address in the **Email** text box, and then click **Save Changes**.

Username	jjohnson
Display name	Jane Johnson (Lead)
Email	jjohnson@myco.com
Save Changes	
Change Password	

Manage Your Credentials and Passwords

If you access a view or workbook that has a live database connection and requires you to authenticate, Tableau offers to save your password for you. If you accept, it stores your credentials in a cookie.

- Under **Saved Credentials**, click **Clear All Saved Credentials** to remove the cookie from Tableau Server.

Clear Credentials for Connected Devices

When you sign in to Tableau Server from a device, your credentials can be stored for that device after it is authenticated for the first time. At any time, you can clear the credentials for all devices that you have used to connect to Tableau Server. You will need to sign in the next time you connect to Tableau Server from any device.

- Under **Connected Devices**, click **Clear All Connected Devices**.

Manage Your Subscription Settings

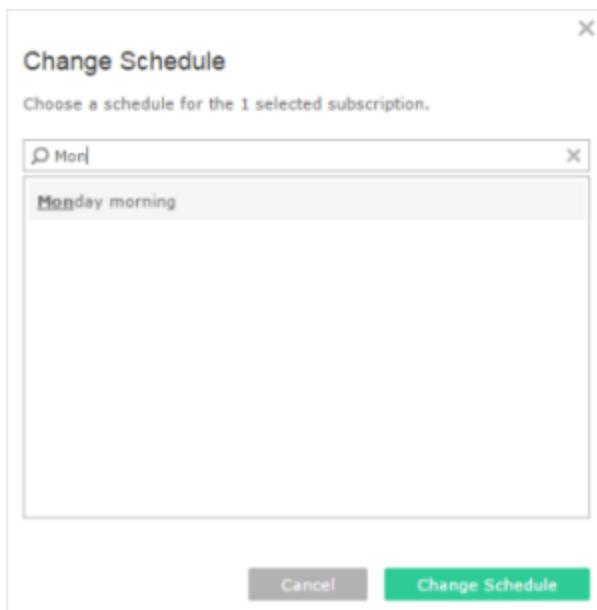
Go to **Subscriptions** in your content to change the schedule for any subscriptions you're

receiving.

1. Click **Subscriptions**, and then select the workbook or view.
2. Select **Actions > Change Schedule**.

Subject	Schedule
... Puget Sound Home Sales by Index	Weekday mornings – Weekly at 6:00 AM on Monday
... Sales Actual vs. Planned	Monday morning – Weekly at 6:00 AM on Monday
... SAT Performance	Monday morning – Weekly at 6:00 AM on Monday

3. Select the new schedule from the list of available schedules, and then click **Change Schedule**.



To change the subject name of the subscription, click **Change Subject**. To unsubscribe from a view or workbook, by click **Delete**.

Change Your Start Page

Tableau Server installs with **Views** as the default start page for all users. Administrators can also specify a different default start page.

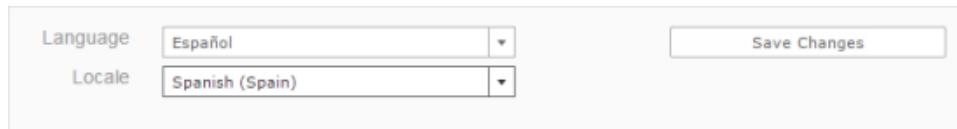
The URL for your current start page is displayed here. Click the link to go to the page.

To designate a different start page for yourself, navigate to the server page you want (such as **Workbooks**), click your name in the upper right corner of the page, and then click **Make This My Start Page**.

To return to using the start page designated by your administrator, click your user name, and then click **My Account Settings**. For start page, click **Reset to Default**.

Language and Locale

The **Language** setting controls the language you see for the Tableau Server user interface and **Locale** affects views, such as how numbers are formatted, or which currency is used. Your administrator can configure these settings for all server users, but you can change them here, just for yourself. If you do change the settings, note that they will only take effect if they are a supported language. See [Language and Locale on page 288](#) to learn more.



Change the **Language** and **Locale**, and then click **Save Changes**. The language and locale update immediately and will continue to be used for your server sessions the next time you sign in.

Scheduled Refresh Tasks and Subscriptions

Server administrators can configure the server to allow end users to subscribe to published views, and they can set the schedules for data extract refresh tasks and the subscription email deliveries.

Administrators can also specify which other users are allowed to set schedules. Otherwise, non-administrator users can work with schedules in the following ways:

- Tableau Desktop publishers can set scheduled refresh tasks when they publish a data source or a workbook with a data extract.
- Tableau Server users can subscribe to views that are delivered by email on a schedule.

Changes to a schedule on the server are reflected in the Tableau Desktop Schedule dialog box the next time an author publishes content. Similarly, changes to a subscription schedule are reflected in the choices a server user has when subsequently subscribing to a view.

About Extracts and Schedules

Tableau Desktop authors can create data extracts, which are copies or subsets of data from the original data sources. Workbooks that use data extracts are generally faster than those that use live database connections because the extracted data is imported into the Tableau data engine. Extracts can also increase functionality. After an author publishes a workbook or a data source with an extract, the extract resides on Tableau Server.

Refreshing extracts on Tableau Server

Administrators can change or reassign extract refresh schedules. Any scheduling changes the site administrator makes in Tableau Server are reflected in the Schedule dialog box in Tableau Desktop when the workbook or data source is published again.

You can also refresh an extract immediately, using the **Run Now** option. Before you can create refresh schedules, you must enable scheduling on the server.

tabcmd command line utility: The tabcmd command line utility provides a `refreshextracts` command which you can use from the command line or incorporate into your own script.

Refreshing extracts from Tableau Desktop

- **At publish time:** When an author publishes a workbook or data source that uses an extract, that author can assign it to a recurring refresh schedule on Tableau Server. The refresh can be a full refresh or an incremental refresh. Incremental refreshes reference a column in the extract that has a data type of date, date/time, or integer; such as a timestamp. Tableau uses this column to identify new rows that need to be added to your extract. See [Refreshing Extracts](#) and [Schedules](#) in the Tableau Desktop help for more information.
- **User interface:** You can use the **Refresh from Source**, **Add Data From File**, and **Add Data From Data Source** options in Tableau Desktop to upload an addition to or refresh an extract on Tableau Server. You may want to do this if Tableau Server doesn't have sufficient credentials to refresh data from the original data source. See [Updating Extracts on Tableau Server](#) in the Tableau Desktop online help for details on how to upload.
- **Data Extract command line utility:** The Data Extract command line utility installs with Tableau Desktop. You can use it to upload an addition to an extract on Tableau Server or refresh it. See [Tableau Data Extract Command Line Utility](#) in the Tableau Desktop online help for more information on how to upload.

See also

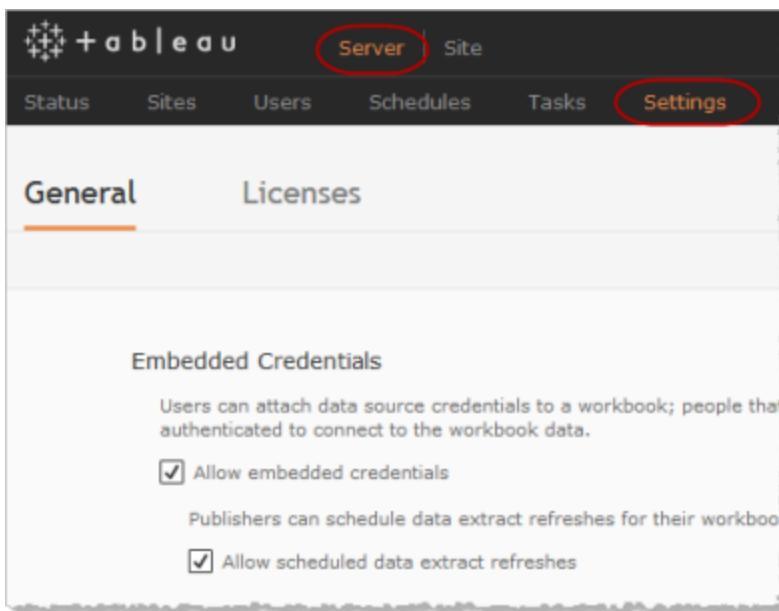
[Enable Scheduling](#) on the next page

[Automate Refresh Tasks](#) on page 229.

Enable Scheduling

Before you can schedule an extract refresh, scheduling must be enabled on the server.

1. In the top navigation area of the server web editing environment, select **Server** > **Settings**.
2. On the **General** page, under **Embedded Credentials**, select the following settings:
 - **Allow embedded credentials**
For some data source types, scheduled refresh tasks are possible only when credentials are embedded.
 - **Allow scheduled data extract refreshes**



Create or Modify a Schedule

The Schedules page shows a list of schedules, including their name, type, what they're for (scope), number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

1. To create a new schedule, click **New Schedule**:

The screenshot shows a user interface for managing schedules. At the top, there is a navigation bar with tabs: Content, Users, Groups, Schedules (which is highlighted in orange), Tasks, and Stats. Below the navigation bar, the title "Schedules 6" is displayed. A prominent orange button labeled "+ New Schedule" is located at the top left of the main content area. To its right, a dropdown menu shows "0 selected". The main content area lists six existing schedules in a table format:

Name	Frequency	Task type
<input type="checkbox"/> End of the month	Monthly	Extract Refresh
<input type="checkbox"/> Monday morning	Weekly	Subscription

2. To modify an existing schedule, select it then click **Edit**:

The screenshot shows the same Schedules page as above, but with one schedule selected: "Weekly Recap". A context menu is open over this selected row, with the "Edit Settings" option highlighted. The menu also includes other options: Run Now, Enable, Disable, Rename, Delete, and Extract Refresh.

Name	Frequency	Task type
<input type="checkbox"/> End of the month	Monthly	Extract Refresh
<input type="checkbox"/> Monday morning	Weekly	Subscription
<input type="checkbox"/> Saturday night	Weekly	Extract Refresh
<input type="checkbox"/> Weekday early mornings	Weekly	Extract Refresh
<input type="checkbox"/> Weekday mornings	Weekly	Subscription
<input checked="" type="checkbox"/> Weekly Recap	Weekly	Subscription

3. Specify a descriptive **Name** for the schedule (for example, Every Saturday Morning, End of the Month).
4. Choose a **Task Type** the schedule will handle—either refreshing extracts or delivering subscriptions.



5. Optionally define a **Default Priority** from 0 to 100. This is the priority that will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the higher priority runs first. See [Manage Refresh Tasks](#) on page 225 to learn more about modifying a task's priority.
6. Choose whether the jobs in the schedule will run at the same time (Parallel, the default) or one after the other (Serially).
7. Finish defining or editing the schedule. You can define an hourly, daily, weekly, or monthly schedule.

New Schedule

Name	<input type="text" value="End of week"/>
Task type	<input type="text" value="Subscription"/>
Default priority	Tasks in a schedule are run in priority order from 1 to 100. <input type="text" value="50"/> 1 is the highest priority 100 is the lowest
Execution	Tasks in this schedule will be run by a background process in priority order. <input checked="" type="radio"/> Parallel: Use all available background processes for this schedule <input type="radio"/> Serial: Limit this schedule to one background process
<input checked="" type="radio"/> Hourly <input type="radio"/> Daily <input checked="" type="radio"/> Weekly <input type="radio"/> Monthly	at <input type="text" value="5"/> : <input type="text" value="00"/> <input type="text" value="PM"/> <ul style="list-style-type: none"> <input type="checkbox"/> Sunday <input type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input type="checkbox"/> Wednesday <input type="checkbox"/> Thursday <input checked="" type="checkbox"/> Friday <input type="checkbox"/> Saturday

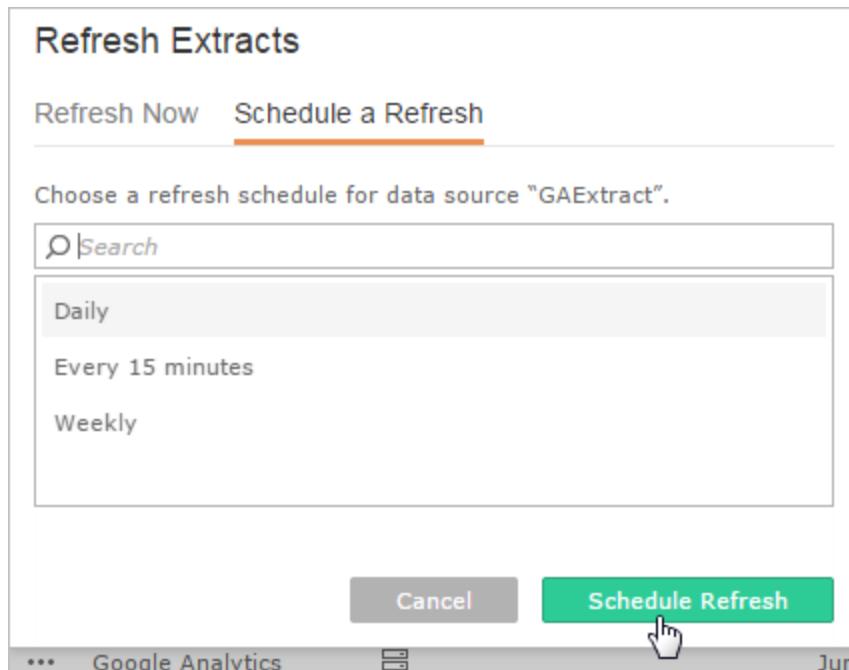
Cancel **Save**

8. Click **Save**.

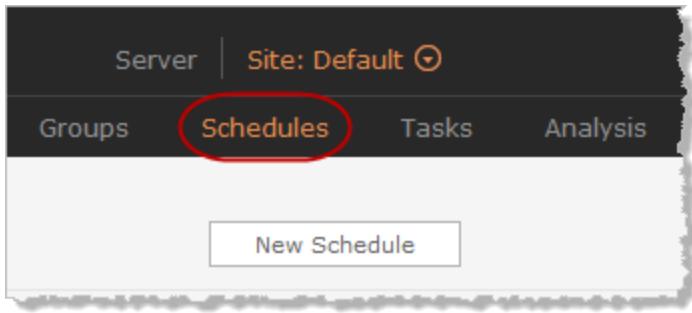
Create a Scheduled Extract Refresh Task

You can set scheduled refresh tasks for published data source extracts and published workbooks that connect to data extracts.

1. When you're signed in to Tableau Server, display **Content > Data Sources** or **Content > Workbooks**, depending on the type of content you want to refresh.
2. Select the check box for the data source or workbook you want to refresh, and then select **Actions > Extract Refresh**.
3. In the Refresh Extracts dialog, select **Schedule a Refresh**, and complete the following steps:
 - Select the schedule you want.
 - If available, specify whether you want a full or incremental refresh.
By default, and if this option is not shown, a full refresh is run. Incremental refresh is available only if you configured it in Tableau Desktop before publishing the extract. For information, see [Refreshing Extracts](#) in the Tableau Desktop Help.
 - Click the **Schedule Refresh** button.



Note: If you want to add a new schedule, you can do so on the **Schedules** page.



Quick Start: Refresh Extracts on a Schedule

For published workbooks that connect to a database extracts, you can set up the server to automatically refresh the data on a recurring schedule. Refreshing extracts on a regular schedule improves performance by extracting just the data you need, and helps to always show recent data.

1 Set up a Schedule on the Server

Sign in to the server as an administrator and select **Server > Schedules**. Click **New Schedule** to create a new schedule.

Content	Users	Groups	Schedules	Tasks	Status	Settings
Schedules 5						
			+ New Schedule		▼ 0 selected	
Name	Frequency	Task type	Tasks	Execution	Next run at	
<input type="checkbox"/> End of the month	Monthly	Extract Refresh	0	Parallel	Feb 28, 2015, 11:00 PM	
<input type="checkbox"/> Monday morning	Weekly	Subscription		Parallel	Feb 9, 2015, 6:00 AM	
<input type="checkbox"/> Saturday night	Weekly	Extract Refresh	0	Parallel	Feb 7, 2015, 11:00 PM	
<input type="checkbox"/> Weekday early mornings	Weekly	Extract Refresh	0	Parallel	Feb 6, 2015, 4:00 AM	
<input type="checkbox"/> Weekday mornings	Weekly	Subscription		Parallel	Feb 6, 2015, 6:00 AM	

Tableau Server provides several extract schedules by default. You can add a new schedule if needed.

2 Enable Scheduled Extract Refreshes

Log into the server as an administrator and select **Server > Settings > General**.

General Licenses Add a Site

Allow publishers to embed data source credentials in a workbook

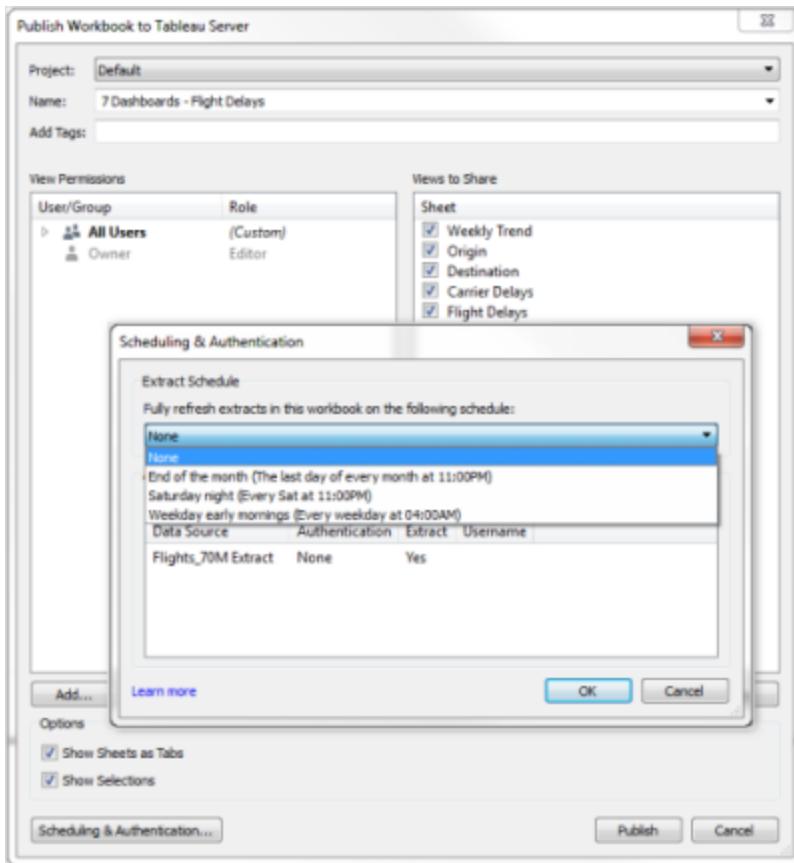
Allow publishers to schedule data extract refreshes

Revert Save

Select **Allow embedded credentials** and **Allow scheduled data extract refreshes** and then click **Save**.

3 Publish a Workbook with an Extract

In Tableau Desktop, select **Server > Publish Workbook**. Sign in to the server if needed. In the **Publish Workbook to Tableau Server** dialog box, click **Schedules & Authentication**. Under **Extract Schedule**, select the schedule from the list.



If the data source requires authentication, you will also need to select the authentication type.

4 Monitor the Scheduled Tasks

As an administrator you can monitor scheduled tasks by viewing **Background Tasks for Extracts** on the **Status** page.

The screenshot shows the 'Site Status' page with a navigation bar at the top: Content, Users, Groups, Schedules, Tasks, and Status. The 'Status' tab is selected. Below the navigation bar, there's a section titled 'Analysis' which includes 'Dashboards that monitor site activity.' A table follows, listing various metrics:

Views	Analysis
Traffic to Views	View count, viewers, and viewer behavior for published views.
Traffic to Data Sources	Data source usage, users, and user behavior for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.

Background Task Prioritization

Note: This topic only covers prioritization of background tasks for extract refreshes and schedules.

Scheduled extract refreshes and subscriptions are run in this order:

1. Any task that is already in process is completed first.
2. Any task that is manually **Run Now** will start when the next backgrounder process becomes available.
3. Tasks with the highest priority (the lowest number) start next, independent of how long they have been in the queue. For example, a task with a priority of 20 will run before a task with a priority of 50, even if the second task has been waiting longer.
4. Tasks with the same priority are executed in the order they were added to the queue. The first task added to the queue will be started first and the second task added will be started next.
5. When multiple tasks with the same priority are scheduled to run at the same time, they are started in the order they were created or enabled. There is no distinction between extract refreshes and email subscriptions.

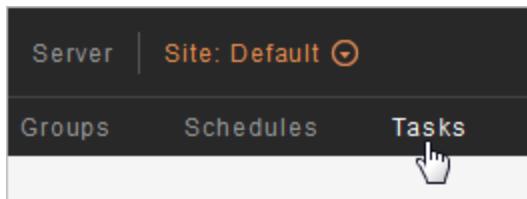
The following limitations also impact when scheduled tasks run:

- Tableau Server can only run as many concurrent tasks as there are backgrounder processes configured.
- Separate extract refreshes for the same data cannot run at the same time.
- Tasks associated with a schedule that uses serial execution must run one at a time.

Manage Refresh Tasks

The Tasks page displays the extract refresh tasks that are scheduled to run on Tableau Server. Administrators can change a task's priority or its schedule, run tasks, or delete them.

To display the Tasks page, sign in to the site you want to work with, and then click **Tasks**.



Change a task's schedule

1. On the Tasks page select one or more tasks to modify.
2. Click **Change Schedule**. Select a new schedule from the list of schedules:

A screenshot of the Tableau Server Tasks page. The top navigation bar includes 'Content', 'Users', 'Groups', 'Schedules', 'Tasks', 'Status', and 'Settings'. Below, it shows 'Extract Refreshes 84' and 'Subscriptions 127'. A table lists tasks: 'Workbook / Sales over quarter' (selected), 'Sports', and 'Sports' (checked). A context menu is open over the first 'Sports' entry, with 'Change Schedule' highlighted. The table columns are 'Actions', 'Refresh type', and 'Schedule'.

Actions	Refresh type	Schedule
Run Now	Full refresh	Every hour – Every 1 hour fro...
Change Schedule	Full refresh	End of the month – On the last...
Change Priority	Incremental refresh	Saturday night – Weekly at 11...
Delete		

Changes the administrator makes in Tableau Server are reflected in the Schedule dialog box in Tableau Desktop when the workbook or data source is published again.

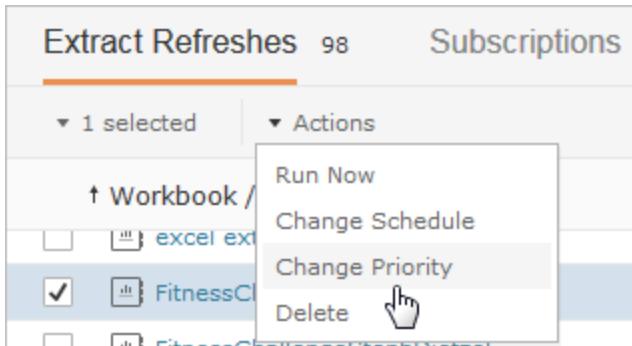
Perform a refresh task on demand

1. On the Tasks page, select a task to run.
2. Select **Actions > Run Now**.

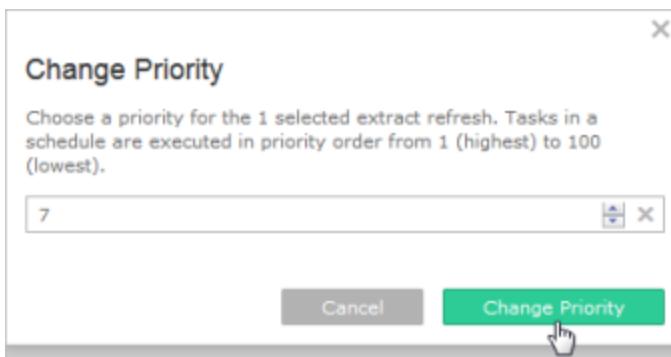
Note: If a scheduled task is not set for an extract, you can refresh it on demand from the Data Connections page.

Change a task's priority

1. On the Tasks page select one or more tasks to modify.
2. Select **Actions > Change Priority**.



3. Type a new priority from 0 to 100 and click **Change Priority**.

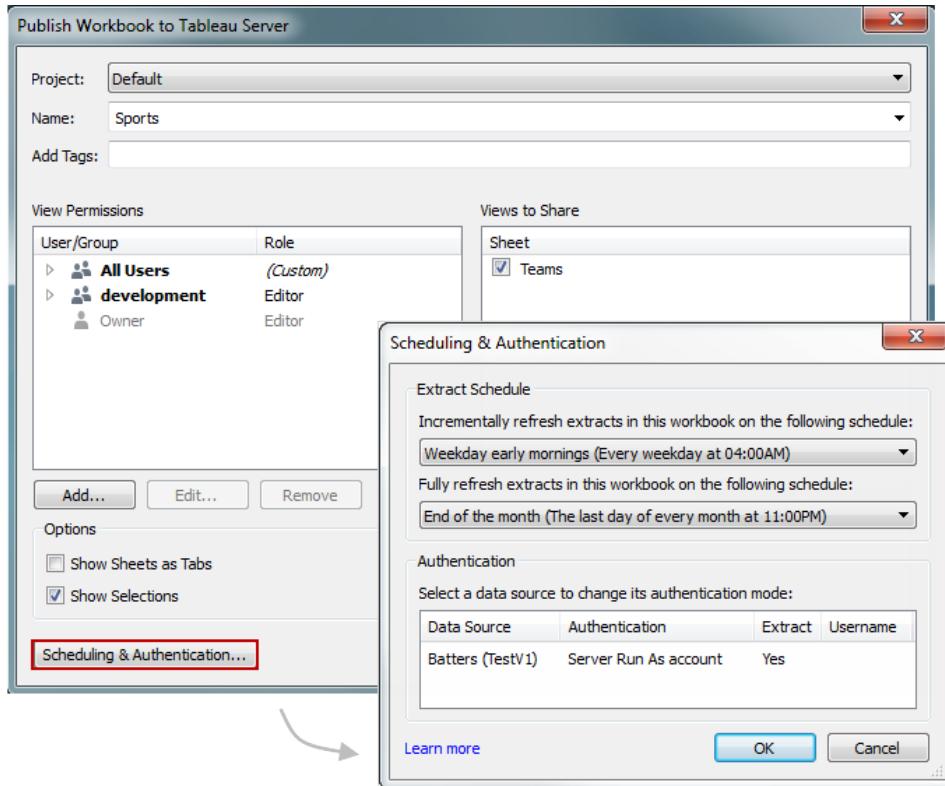


Quick Start: Manage Incremental Extracts

When you publish a workbook that has an incremental extract, you can associate it with up to two refresh tasks that Tableau Server will handle for you: An incremental refresh of the extract and a full refresh. After you publish the workbook, you or a Tableau Server administrator can modify any tasks that are associated with the workbook. You can also delete tasks or add more.

1 Publish and Assign a Schedule

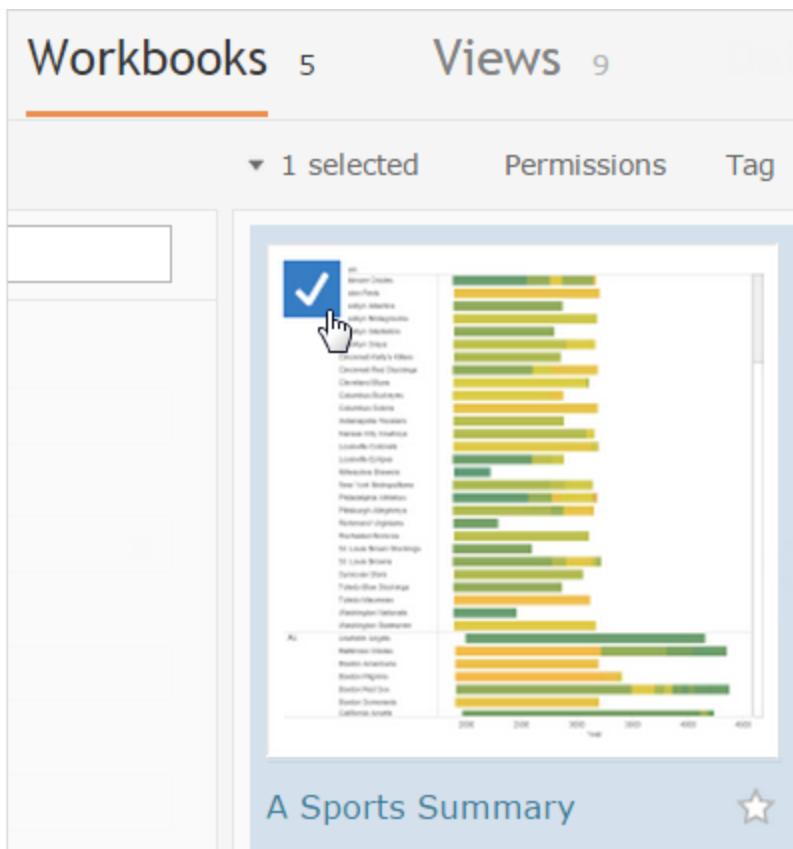
In Tableau Desktop, after you create a workbook that uses an extract, go to **Server > Publish Workbook**, and click **Scheduling & Authentication**. Next, choose schedules for your refreshes and click OK.



After you publish in Tableau Desktop and choose your refresh schedules, Tableau Server handles the refresh tasks for you.

2 Select the Workbook

To modify a workbook's scheduled task, sign in to Tableau Server and on the **Workbooks** page, select the workbook:



3 Access the Refresh Schedule

Click **Refresh Schedule**.

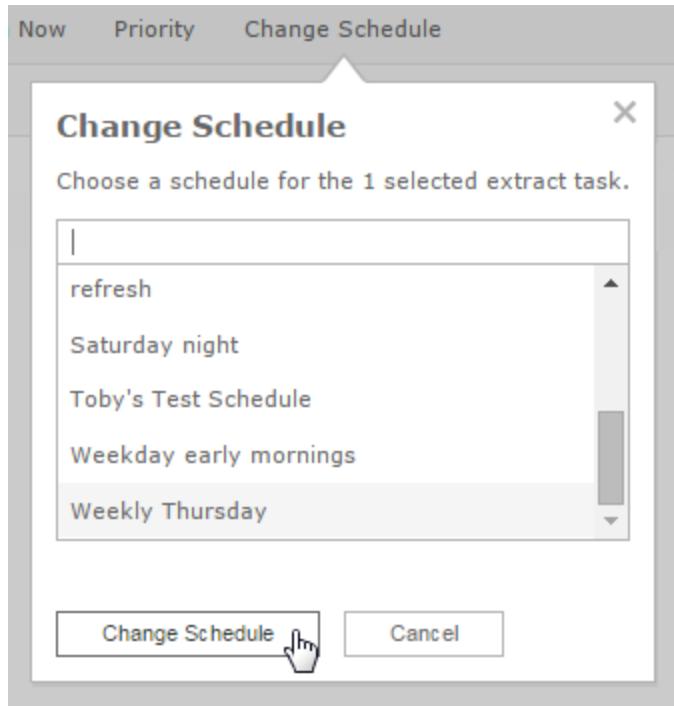


Select the check box for the refresh task you want to modify:

Refresh Type	Schedule
<input type="checkbox"/> Full refresh	End of the month - ⏲
<input checked="" type="checkbox"/> Incremental refresh	Saturday night - Weekends

4 Edit, Delete, or Add More Tasks

Select the action you want to take—for example, **Change Schedule**—and make your selection. You can also delete the task, change its priority, or add more refresh tasks.



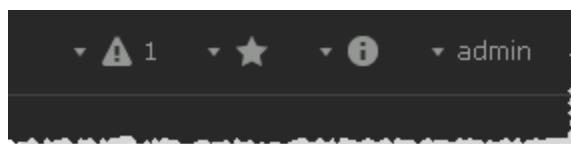
Automate Refresh Tasks

You can associate extract refresh tasks with schedules in Tableau Server to automate refreshing data extracts. You can also automate extract refreshes using tabcmd, a command line utility that comes with Tableau Server and can be installed on a separate computer from Tableau Server. In particular, you can use the `refreshextracts` command in combination with other commands in your own script. For example:

```
tabcmd login - http://mytabserver -u jsmith -p P@ssw0rd!
refreshextracts --datasource salesq4
```

Handle Extract Refresh Alerts

If scheduled extract refreshes did not succeed, Tableau displays an Alerts menu in the upper right corner:



You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn't be refreshed

- The author of a workbook that connects to a data source that couldn't be refreshed

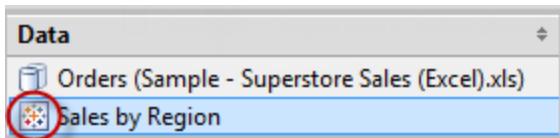
When you open the Alerts menu you can see more information about the refresh failure(s):



When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the **Data source** (for example, **Data source: sales_data**), it means that the data source is a **Tableau Server data source**. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon :



Resolving Extract Refresh Problems

You can resolve some extract refresh problems by clicking the **Edit connection info** link in the alert, and then entering the missing information, and clicking **Save**:

Edit Data Connection

Server:	SQL2008r2	undo x
Server Port:	6060	undo x
Database Username:	jsmith	undo x
Password:	*****	
Confirm Password:	*****	
Test Connection		
Save		Cancel

If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

Tip: Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections

Manage Subscriptions

A subscription is a regularly scheduled email delivery of a Tableau Server view or workbook to subscribed users. When subscribers click the snapshot of the view or workbook in their email, it opens on Tableau Server.

To view information about each subscription, such as the subscriber's email address and name, the name of the view, and the delivery schedule, click **Tasks > Subscriptions**.

Requirements

For Tableau Server users to receive subscriptions, the following things need to be in place:

- **Email settings configuration:** As the system administrator, you configure the basic SMTP server settings for subscriptions on the **Alerts and Subscriptions** tab in the Configuration dialog box, which displays during Setup. This is the "from account" Tableau Server uses to email subscriptions to server users. You can access this tab after Setup as well. See [Reconfigure the Server](#) on page 35 and [Configure SMTP for Email Subscriptions](#) on page 18 for steps.
- **Credentials embedded or not required:** From Tableau Server's perspective, a subscription includes a workbook, data, and a schedule. To deliver the data piece,

Tableau Server needs to be able to access the data with no end-user involvement. This can be accomplished by using either a workbook with embedded database credentials, a Tableau Server data source, or by using data that doesn't require credentials, such as a file that's included with the workbook at publish time. Workbooks that prompt for credentials for live database connections can't be subscribed to.

- **User requirements:** If a user can see a view or workbook on Tableau Server and it has the subscription icon () in the upper right corner, he or she can [subscribe to it](#). The ability to see a view or workbook is controlled by the **View** permission. A user must also have an email address. If Tableau Server doesn't already have an email address for a subscribing user, it prompts for one at subscription sign-up time. Users can change their delivery options, unsubscribe, or update their email address on their [User Preferences page](#).
- **Trusted authentication:** If you are using a restricted ticket (the default) to render an embedded view, subscriptions are disabled.

Additional Subscription Settings

As long as subscriptions are configured on the **Alerts and Subscriptions** tab and Tableau Server is using its default settings, server users can subscribe to the views and workbooks they see. To prevent users from subscribing or to customize their subscription experience, here's where to go:

- **Sites page:** By default, subscriptions are enabled for every site, but you can use the [Sites page](#) to disable subscriptions on a per-site basis or to customize it. For example you can enter a custom **From address** for subscriptions instead of the one you specified in the Configuration dialog box. You can also create your own footer for the subscription emails your users receive.
- **Schedules page:** Your users will need at least one subscription schedule to choose when they subscribe. Tableau provides two by default. As the system administrator, you can create additional schedules or remove the default ones. See [Create or Modify a Schedule on page 218](#) for details.
- **Subscriptions page:** This page lists all the subscriptions on the server or, if you're a site administrator, on the site. System administrators can use this page to change a server user's subscription schedule or delete their subscription. See the topics below for details.

For steps on how to test whether you've configured subscriptions correctly, see [Test Your Subscription Configuration on the next page](#). If you're experiencing an issue with subscriptions, see [Troubleshoot Subscriptions on page 658](#).

Delete a Subscription

To delete a subscription, select the subscription you want to remove and click **Delete**:

Subscriptions					
	User	Address	Subject	Schedule	
<input type="checkbox"/>	Henry Wilson	hwilson@myco.com	Area Sales Performance	Monday morning	
<input checked="" type="checkbox"/>	Albert Singh	asingh@myco.com	Commission Model	Weekday mornings	
<input type="checkbox"/>	Keith Harding	kharding@myco.com	Daily Sales Report - UK	Weekday mornings	
<input checked="" type="checkbox"/>	Sheila Kim	skim@myco.com	Baseball Statistics	Weekday mornings	
<input type="checkbox"/>	Leigh Winters	lwinters@myco.com	Education Around the World	Monday morning	
<input type="checkbox"/>	Rosa Garcia	rgarcia@myco.com	Green Living	Monday morning	

Edit a Subscription Schedule

To change the schedule for a subscription, select the subscription, click **Edit Schedule** and select a schedule:

Subscriptions					
Delete	Edit Schedule				
<input type="checkbox"/>	Monday morning	<input type="checkbox"/>	Address	Subject	Schedule
<input type="checkbox"/>	Henry Wilson	hwilson@myco.com	Area Sales Performance	Monday morning	
<input type="checkbox"/>	Albert Singh	asingh@myco.com	Commission Model	Weekday mornings	
<input type="checkbox"/>	Sheila Kim	skim@myco.com	Baseball Statistics	Weekday mornings	
<input type="checkbox"/>	Leigh Winters	lwinters@myco.com	Education Around the World	Monday morning	
<input checked="" type="checkbox"/>	Rosa Garcia	rgarcia@myco.com	Green Living	Weekday mornings	
<input type="checkbox"/>	Keith Harding	kharding@myco.com	Daily Sales Report - UK	Weekday mornings	

Test Your Subscription Configuration

As the administrator, you can test whether you've correctly configured subscriptions by doing the following:

1. **Subscribe to a view.**
2. On the Schedules page, select the schedule that contains your subscription.
3. Click **Run Now:**

Schedules				
	New	Modify	Delete	Enable
	Disable	Run Now		
<input type="checkbox"/>	End of the month	Monthly	Extract	
<input type="checkbox"/>	End of week	Weekly	Subscription	
<input checked="" type="checkbox"/>	Monday morning	Weekly	Subscription	
<input type="checkbox"/>	Saturday night	Weekly	Extract	

4. In a few moments, your subscription should appear in your email inbox.

Troubleshoot Subscriptions

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials:** Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.
- **Database temporarily down:** If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.
- **Background process timeout:** By default, the background process that handles subscriptions times out after 30 minutes. In the majority of cases, this is plenty of time. However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the [Background Tasks for Non Extracts](#) on page 297 admin view to see if that's the case. To increase the timeout threshold, use the tabadmin option `subscriptions.timeout`.

Can't subscribe

If you can see a view on Tableau Server and it has a subscription icon () in the upper right corner, you can subscribe to it.

Two things need to be in place for you to subscribe to a view: Tableau Server needs to be correctly configured (described in [Manage Subscriptions](#) on page 231) and the view you're subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn't being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the [Tableau Desktop help](#) for details).

No subscription icon

It's possible to see a view on Tableau Server but be unable to subscribe to it. This happens for views with live database connections, where you're prompted for your database credentials when you first click the view. A subscription includes a view (or workbook), data, and a schedule. To deliver the data piece, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don't require credentials or have them embedded.

You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is configured for trusted authentication. See [Subscription Requirements](#) for more information.

Receiving invalid or "broken" subscriptions

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don't work, or receiving subscriptions even though they've unsubscribed from the view or workbook.

Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")

You may see the above error in Windows Event Viewer if subscriptions appear to be sent (according to the [Background Tasks for Extracts](#) on page 295 admin view), yet subscriptions aren't arriving, and your SMTP server is using encrypted (SSL) sessions. Subscriptions are only supported for unencrypted SMTP connections. The solution is to use an unencrypted SMTP server.

Custom scripts not working after upgrade to 8.1

To support better session management, starting with version 8.1, a hash tag (#) was added to the end of view URLs. If you had custom subscriptions scripting that generated views as PDFs or PNGs you may need to update your scripts to allow for the hash tag.

For example, prior to version 8.1, view URLs use this syntax:

`http://tableauserver/views/SuperStore/sheet1`. To generate a view as a PNG, `.png` could be added to the end of the URL. For example,
`http://tableauserver/views/SuperStore/sheet1.png`.

In versions 8.1, 8.2, or 8.3, view URLs use this syntax:

`http://tableauserver/views/SuperStore/sheet1#1`. To generate a PNG, add `.png` before the hash tag. For example:
`http://tableauserver/views/SuperStore/sheet1.png#1`

Custom scripts not working after upgrade to 9.0

In version 9.0, the session ID at the end of server URLs is indicated by an "iid" parameter, :`iid=<n>`. For example,
`http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1`. This parameter replaces the hash tag "#<n>" used for the session ID in 8.x versions of Tableau Server.

If you use custom subscriptions scripts that generate views as PDFs or PNGs, you may need to update your scripts by removing the hash tag and number (#<n>), and inserting the ?:`iid=` session ID parameter before the number.

Starting in version 9.0, view URLs use this syntax:

`http://tableauserver/views/SuperStore/sheet1?:iid=2`.

To generate a PNG in version 9.0 and later, add .png before the session ID:

`http://tableauserver/views/SuperStore/sheet1.png?:iid=2`

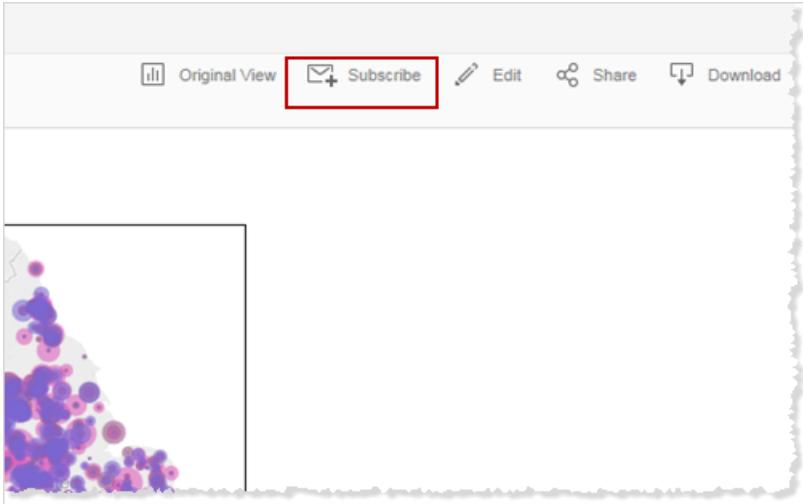
Subscribe to Views

When you open a view on Tableau Server, if it has a subscription icon (✉) in the upper-right corner, your administrator has configured subscriptions for your site. You can click the Subscribe button to select options for subscribing to the view. This means that, at regular intervals, you can have a snapshot of the view automatically delivered to your email account—without having to sign in to Tableau Server.

You can also choose to receive each view in a workbook in a single email, or unsubscribe to views you no longer want to receive.

Subscribe to a view

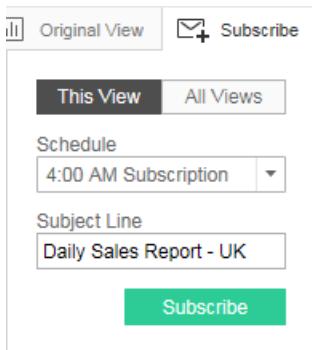
1. Click **Views or Workbooks**.
2. Open a view, or open a workbook, and then open a view in it.
3. Click **Subscribe**.



4. If your Tableau Server account hasn't already been associated with an email address, you are prompted to provide one. Enter your email address, and then click **Next**.

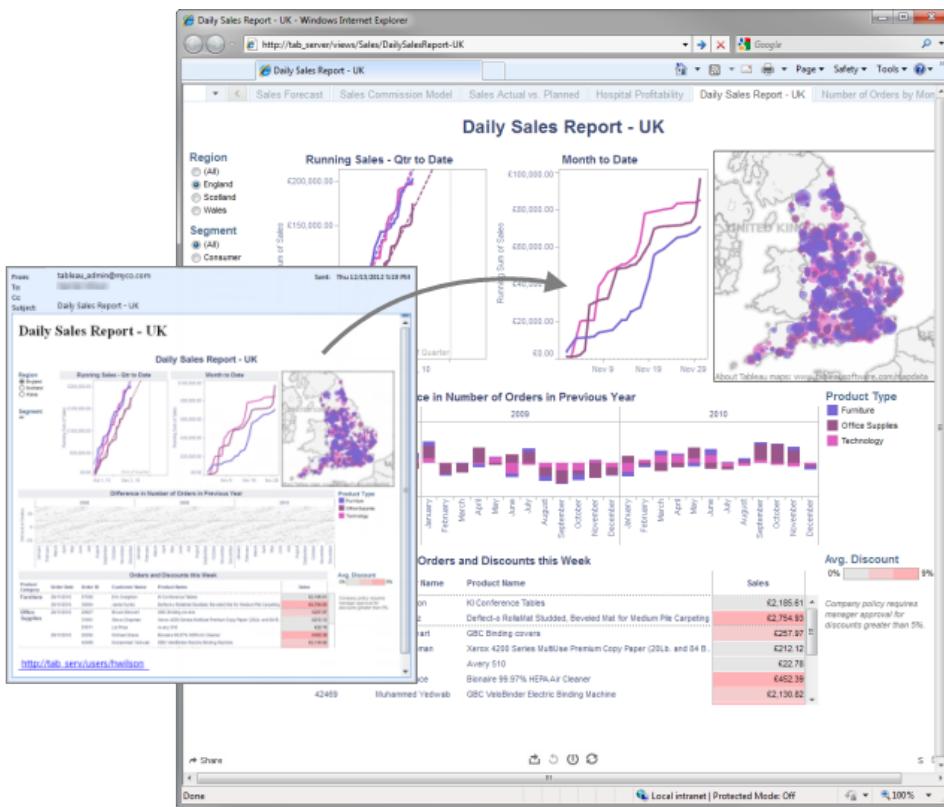
You can change the email address a subscription is sent to. For details, see [Change Your Email Address](#) on page 214.

5. Select the current view (**This View**) or, if the workbook includes multiple views, **All Views**. Pick a schedule, enter a subject line, and then click **Subscribe**.



By default, Tableau Server provides a weekday morning schedule and a Monday morning schedule. The Tableau Server administrator can also create custom subscription schedules.

When you receive the subscription by email, click the snapshot of the view to open it in Tableau Server.



Note: If a dashboard size is set to **Automatic**, the image included in the subscription email is fixed at 800 pixels by 600 pixels.

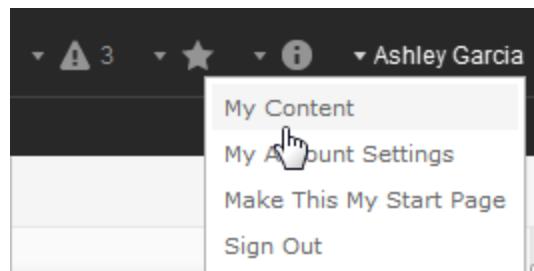
Unsubscribe from a view

1. Open your account settings on Tableau Server by using one of the following ways:

- Click the link at the bottom of a subscription email.

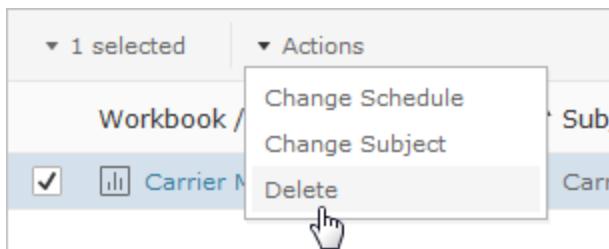


- Sign in to Tableau Server, select your name, and then on the drop-down list, select **My Content**.



2. Click **Subscriptions**.

3. Select the check box next to the view you want to unsubscribe from, and then select **Actions > Delete**.



4. Click **Delete** in the confirmation message that appears.

You can also manage your subscriptions on this page. For example, select a different schedule or change the email subject line. For more information, see [Manage Your Subscription Settings on page 214](#).

Server Maintenance

As an administrator, you will want to check the status of the server, analyze and monitor the activity on the server, manage scheduled tasks, or perform certain maintenance activities such as clearing saved data connection passwords. In addition, there are several settings that you may want to specify to customize the user experience for people using the server. You can do some of these tasks from the General page of the Status page and others from the Settings page.

View Server Process Status

You can use the Process Status table on the Server Status page to view the state of Tableau processes on each Tableau server:

The screenshot shows the 'Server Status' section of the Tableau Server interface. At the top, there's a navigation bar with tabs: Content, Users, Groups, Schedules, Tasks, Status, and Settings. The 'Status' tab is selected. Below the navigation bar, the title 'Server Status' is displayed. Underneath, a section titled 'Process Status' is shown with the sub-instruction 'The real-time status of processes running in Tableau Server.' A table lists ten processes along with their IP address (10.32.139.32) and status indicators:

Process	IP Address
Gateway	✓
Application Server	✓
API Server	✓
VizQL Server	✓
Cache Server	✓
Search & Browse	✓
Backgrounder	✓
Data Server	✓
Data Engine	✓
File Store	✓
Repository	✓

At the bottom of the table, there are several status icons with their corresponding labels: Refresh Status, Active (green checkmark), Busy (green circle with arrow), Passive (grey checkmark), Unlicensed (orange exclamation mark), Down (red X), and Status unavailable (grey square).

Possible status indicators are listed at the bottom of the table:

Active
 Busy
 Passive
 Unlicensed
 Down
 Status unavailable

When Tableau Server is functioning properly, most processes will show as Active, Busy or Passive (Repository):

- **Active**—The process is functioning as intended. See File Store below for details on possible active states.
- **Busy**—The process is completing some task. See File Store and Repository below for more information.
- **Passive**—The repository is in passive mode
- **Unlicensed**—The process is unlicensed.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

If there is additional information, a message appears below the status icon:



Ready for removal

For more information about troubleshooting process status, see [Troubleshoot Server Processes](#) on the next page.

Troubleshoot Server Processes

When Tableau Server is functioning properly, processes will show as Active, Busy or Passive (Repository). If there is additional information, a message appears below the status icon:



Ready for removal

Possible status indicators are:

Active Busy Passive Unlicensed Down Status unavailable

Use this table to help troubleshoot issues with your Tableau Server installation.

Process	Status (Icon)	Message	Implications	Actions
Cluster Controller (displays only if you have two or more nodes)		"Node degraded"	<ul style="list-style-type: none">Repository on the node is stopped.Node cannot respond to fail-over elsewhere in the cluster.If Tableau Server is configured for high availability and this is the active repository, fail-over to the second repository occurs.No status available for repository or file store on this node.	<p>No action is necessary unless the cluster controller is regularly down or is down for an extended period of time.</p> <p>If that occurs, take the following actions, in order, until the problem is resolved:</p> <ol style="list-style-type: none">1. Check disk space. If disk space is limited, save the log files (use <code>tabadmin ziplogs</code>) in case you need them for Support, then remove unnecessary files (<code>tabadmin cleanup</code>).2. In Windows Task Manager, stop the cluster-

Process	Status (Icon)	Message	Implications	Actions
				<p>controller.exe process tree and let it restart automatically.</p> <ol style="list-style-type: none"> 3. Restart Tableau Server. 4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (<code>tabadmin stop</code>), clean up files (<code>tabadmin cleanup --reset-coordination</code>), and then start the cluster (<code>tabadmin start</code>). 5. If Cluster Controller continues to show as down, save the log files (<code>tabadmin zip-logs</code>) and contact Support.
File Store File Store status only reflects the state of the file store when the page was loaded.		none	<ul style="list-style-type: none"> • No extracts were being synchronized when the page was loaded. (It is possible that the recurring "catch-all" job is running and synchronizing extracts.) 	None.

Process	Status (Icon)	Message	Implications	Actions
		"Synchronizing"	<ul style="list-style-type: none"> Extracts were being synchronized across file store nodes when the page was loaded. Initial status following installation (both single-node and multi-node). Should disappear within 15 or 20 minutes. 	None.
		"Data Extracts unavailable"	<ul style="list-style-type: none"> Single-node installation: existing extracts may be available but publish/refresh will fail. Multi-node installation: extract synchronization will fail for this node. 	<p>No action is necessary unless the file store is regularly down or is down for an extended period of time.</p> <p>If that occurs, take the following actions, in order, until the problem is resolved:</p> <ol style="list-style-type: none"> 1. Check disk space. If disk space is limited, save the log files (tabadmin ziplogs) in case you need them for Support, and then remove unnecessary files (tabadmin

Process	Status (Icon)	Message	Implications	Actions
				<p>cleanup).</p> <ol style="list-style-type: none"> 2. Stop the filestore.exe process using Windows Task Manager and let it restart automatically. 3. Restart Tableau Server. 4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (tabadmin stop), clean up files (tabadmin cleanup --reset-coordination), and then start the cluster (tabadmin start). 5. If the file store continues to be down, save the log files (tabadmin zip-logs) and contact Support.
		"Decommissioning"	<ul style="list-style-type: none"> • File store is in read-only mode. • Any unique files on this node are being replicated to other file store 	Wait until the status message changes to "Ready for removal".

Process	Status (Icon)	Message	Implications	Actions
			nodes.	
	✓	"Ready for removal"	<ul style="list-style-type: none"> File store is in read-only mode. Ready for user to stop cluster and remove data engine-/file store or remove entire node. 	<p>Stop Tableau Server (<code>tabadmin stop</code>) and then run the Configuration utility to remove Data Engine and File Store or the entire node.</p>
	✓	"Decommission failed"	<ul style="list-style-type: none"> File store is in read-only mode. At least one unique file failed to replicate to another file store node. 	<p>Take the following actions in order until the problem is resolved:</p> <ol style="list-style-type: none"> Run the <code>tabadmin decommission</code> command again. Check disk space on other file store nodes. Decommissioning will fail if another file store node does not have enough space to store all the extracts. Check the <code>tabadmin.log</code> file on the primary node and workers for errors. Stop Tableau Server (<code>tabadmin stop</code>) and then try running the <code>tabadmin</code>

Process	Status (Icon)	Message	Implications	Actions
				<p>decommission command again.</p> <ol style="list-style-type: none"> 5. Put the file store node back into read/write mode (tabadmin recommission), collect logs, and then contact Support. 6. With Support: copy and merge extracts directory from this file store node to the same directory on another file store node.
Repository		"Setting up"	<ul style="list-style-type: none"> • Passive repository is being synchronized with active repository. • Repository is not ready to handle fail-over. • Repository may have gotten more than two minutes behind active repository and is being setup again (this is faster than waiting for a sync). 	<p>Wait until the repository status message changes to "Passive".</p> <p>If this message does not appear, or if it is taking a long time:</p> <ol style="list-style-type: none"> 1. Check disk space and free space if possible. 2. Check cluster controller logs for errors. 3. Restart node.

Process	Status (Icon)	Message	Implications	Actions
			<ul style="list-style-type: none"> Failover occurred and this former active repository is rejoining the cluster. 	
		none	<ul style="list-style-type: none"> If the installation is configured for high availability, failover of the repository occurred. Processes are restarting with updated database connection configurations after failover. If another active repository is not available, Tableau Server is down. 	<p>Take these actions in order until the problem is resolved:</p> <ol style="list-style-type: none"> Wait several minutes for cluster controller to attempt to restart. Restart Tableau Server (<code>tabadmin restart</code>). Check disk space to make sure there is free space. Collect logs (<code>tabadmin ziplogs</code>) in case you need them for Support, and then cleanup files (<code>tabadmin cleanup</code>). Restart Tableau Server. Stop Tableau Server, collect logs and cleanup coordination service files (<code>tabadmin cleanup --reset-coordination</code>)

Process	Status (Icon)	Message	Implications	Actions
				6. Start Tableau Server. 7. Collect logs (<code>tabadmin zip-logs</code>) and contact Support.
		none	<ul style="list-style-type: none"> Working as intended. Node is ready if needed for failover. 	None.
VizQL Server		none		
		none		For information about unlicensed status for a VizQL Server process, see Handle an Unlicensed VizQL Server Process on page 655.

Get Process Status as XML

To get a machine-readable version of the server process status, that is, a version of the status formatted in XML, use the following URL:

`http://my_tableau_server/admin/systeminfo.xml`

You must be signed in to Tableau Server to view the machine-readable process status, or have [enabled remote access](#).

The server returns a status report similar to the following:

```

<systeminfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <machines>
    <machine name="my_tableau_server">
      <repository worker="my_tableau_server:8060" status="Active">

```

```

preferred="false"/>
    <dataengine worker="my_tableau_server:27042" status-
s="Active"/>
    <applicationserver worker="my_tableau_server:8600"
status="Active"/>
    <apiserver worker="my_tableau_server:8000" status="Active"/-
<vizqlserver worker="my_tableau_server:9100" status-
s="Active"/>
    <dataserver worker="my_tableau_server:9700" status="Active"/>
    <backgrounder worker="my_tableau_server:8250" status-
s="Active"/>
    <gateway worker="my_tableau_server:80" status="Active"/>
    <searchandbrowse worker="my_tableau_server:11000" status-
s="Active"/>
    <cacheserver worker="my_tableau_server:6379"
status="Active"/>
    <filestore worker="my_tableau_server:9345" status="Active"
pendingTransfers="0" failedTransfers="0" syncTimestamp="2015-02-
27T20:30:48.564Z"/>
    <clustercontroller worker="my_tableau_server:12012" status-
s="Active"/>
    <coordination worker="my_tableau_server:12000"
status="Active"/>
  </machine>
</machines>
<service status="Active"/>
</systeminfo>

```

Status values in the XML

- <process> **worker** - The name of the node running the process and the port the process is using.
- **status** - The status of the process on the node. Possible values are: Active, Passive, Unlicensed, Busy, Down, ReadOnly, ActiveSyncing, StatusNotAvailable, StatusNotAvailableSyncing, DecommissionedReadOnly, DecomisioningReadOnly, and DecommissionFailedReadOnly
- **pendingTransfers** - A count of the workbook or data source extracts the node needs to get to be fully synced. These represent items that were published to this file store node, and items that were published to other file store nodes and need to be copied to this node.
- **failedTransfers** - A count of the workbooks or data sources that did not transfer

successfully to this file store node during the last automated job. The automated job normally runs about every 15 to 30 minutes, but may take longer when transferring a large number of extracts or large extracts.

Failed transfers do not necessarily indicate a problem with Tableau Server. The recurring automated job will normally transfer files that failed during the previous sync. Reasons for failed file transfers are listed in the logs.

- **syncTimestamp** - The time in UTC of the last automated job that ran and synchronized files.

Access Status Remotely

As the Tableau administrator, only you can see the Status table, but you can grant remote access to make the machine-readable version of the Status table available to non-admin users and to computers other than the one that's hosting Tableau Server. You might do this as part of a remote monitoring process.

To grant remote access to Tableau Server status:

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Enable remote access by typing the following:

```
tabadmin set wgserver.systeminfo.allow_referrer_ips <ip  
address>
```

In the above command, <ip address> is the IPv4 address of the computer for which you want to enable remote access to the Tableau Server status XML. If you are enabling remote access for more than one computer, use commas to separate each IP address.

For example:

```
tabadmin set wgserver.systeminfo.allow_referrer_ips  
10.32.139.31
```

3. After you make the configuration change, restart Tableau Server by typing the following:

```
tabadmin restart
```

Now, users of computers with the IP addresses that have been added can view Tableau process status by entering the URL `http://<server>/admin/systeminfo.xml` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo.xml`).

This functionality can also be used as part of an automated remote monitoring process.

Archive Logs on Status Page (Snapshot)

You can generate and download a snapshot (archive) of the Tableau Server log files from a web browser, without opening a command prompt. This zipped snapshot contains a copy of up to seven days of log file data from Tableau Server and any worker servers (if you have a distributed environment). The snapshot process does not change or remove either the Tableau Server log files or the log archives created with tabadmin.

Note To specify the amount of data you want to collect or the name of the zip file you are creating, use tabadmin to create an archive of server logs. For more information, see [Archive Logs on Command Line \(tabadmin\) on page 644](#).

To generate a snapshot of server log files:

1. Open the Status page:
 - Multi-site: Select **Server > Status** .
 - Single-site: Select **Status**.
2. Click **Generate Snapshot** to create a snapshot of the Tableau Server logs. The Generate Snapshot button is available only if there is no existing snapshot.

Note: This option is available whether or not you have created log archives with tabadmin.

Log Files		
Date Generated	Size	Status
None	None	Generate a new snapshot of log files
Generate Snapshot	Download Snapshot	Delete Snapshot

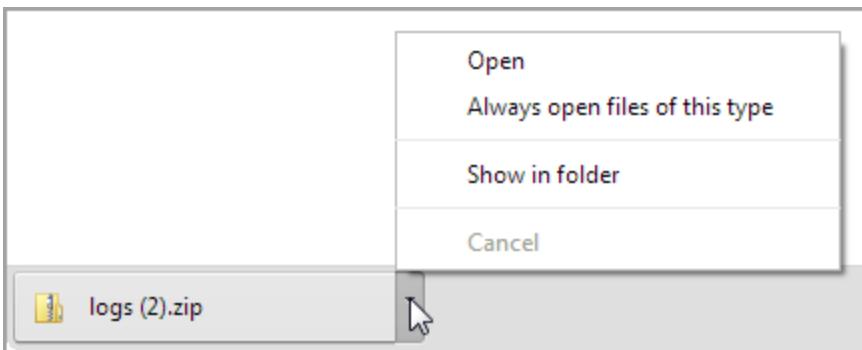
3. Select the number of days of logs you want to include. The default is **Last 7 days**, but you might want to select fewer if you want to reduce the size of the zip file. For example, if you just reproduced an issue and are collecting logs related to the issue, you may want to select **Today** to create the smallest zip file necessary.
4. Click **Download Snapshot** to download the log snapshot to your web browser's default download location. This option is available after you create a snapshot.

Google Chrome shows you the download in the bottom of the window:

The screenshot shows a user interface for managing log files. At the top, there's a section titled "Log Files" with a table header for "Date Generated", "Size", and "Status". A single row is listed: "Dec 22, 2014, 3:07 PM", "49.8 MB", and "Snapshot ready to download. Contains logs from previous seven days.". Below the table are three buttons: "Generate Snapshot", "Download Snapshot", and "Delete Snapshot".

Below this, there's a section titled "Rebuild Search Index" with a note: "You may need to rebuild the search index if the Search & Browse process is down." A "Rebuild Search Index" button is present. At the bottom, there's a file list window showing "logs (1).zip".

5. Click the arrow and then click **Open** to unzip the snapshot or **Show in folder** to see where it was downloaded:



6. (Optional) Click **Delete Snapshot** to delete a log snapshot. This option is available after you create a snapshot. You need to delete the existing snapshot before you can create a new one.

The screenshot shows the same "Log Files" interface as the first one. The "Delete Snapshot" button at the bottom of the main panel is highlighted with a red box and a cursor arrow pointing to it.

For example, you might want to delete the snapshot that you created before an event that you want to investigate.

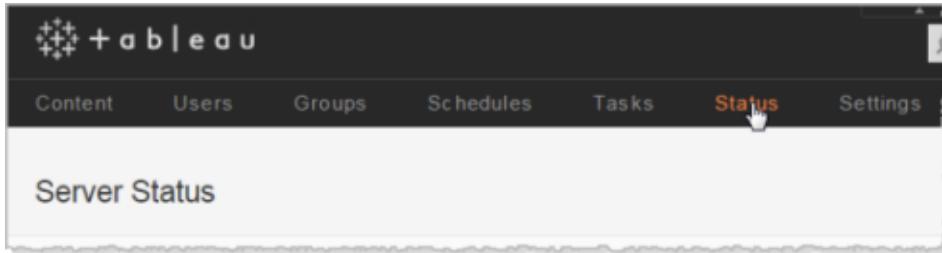
Rebuild the Search Index

If search is returning incomplete or incorrect results, or if the Search & Browse process is down for an extended period of time, you may need to rebuild the search index.

Important: The recommended way to reindex search is to use the `tabadmin reindex` command while Tableau Server is stopped. Reindexing while the server is running can result in content, including sites and projects, temporarily disappearing from server pages.

The search index is built or rebuilt at key points during installation or upgrade of Tableau Server, when you restore a backup, and when you add the Search & Browse process to a new or existing node. The index is kept updated by a background task when content changes. If necessary you can force a rebuild of the index using the `tabadmin reindex` command.

1. To rebuild the search index, click **Status**.



In a multi-site environment, select **Server > Status**.

2. At the bottom of the page, click **Rebuild Search Index**.



Note: You might not see all available server content while the index is rebuilding, and larger search indexes can take longer times to finish rebuilding. Reindexing first removes all content from the index, and then re-adds the content to the index. If you do this while Tableau Server is running, users who are logged into the server

will see content disappear, and then slowly start to reappear in server pages. Reindexing while Tableau Server is stopped provides a better user experience.

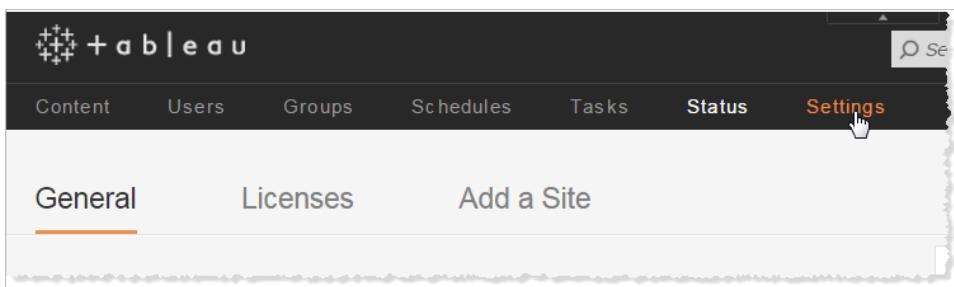
Clear Saved Data Connection Passwords

As the administrator, if you enable the [Allow users to save data source passwords](#) setting, server users can save data source passwords across multiple visits and browsers so they are no prompted for their credentials each time they connect to a data source.

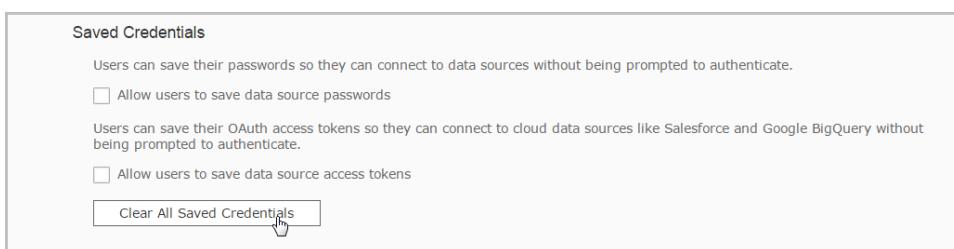
You can reset the data source passwords for all Tableau Server users. Doing this forces them to sign in to the data sources the next time they visit a view that requires database authentication. Server users can also clear their saved data connection passwords on an individual basis using their [User Preferences](#) page.

To clear saved data connection passwords for all server users:

1. Click Settings and General:



2. Under Saved Credentials, click **Clear All Saved Credentials**.



View Licenses

Server administrators can view the license and product key information for Tableau Server.

Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.

To open the Licenses page, select **Server > Settings > Licenses**.

Note: The **Server** option only appears on multi-site installations of Tableau Server. If you have a single site, click **Settings > Licenses**.

Product Key	Cores	Guest	Maintenance Ends	Expires	Valid
[REDACTED]	16	✓	Apr 30, 2015	Apr 30, 2015	✓

Machine	Cores	Licenses Used
[REDACTED]	4	4

If you have a user-based Tableau Server license, you can review how these levels have been distributed.

If you have a core-based Tableau Server license, the Licenses page shows how many cores are allowed, how many have been licensed, and how many are in use (and on what server computers).

User-based versus core-based licensing

Tableau Server can be licensed under two models: user-based or core-based. User-based licenses lets you deploy Tableau Server on a single computer or on multiple computers in a cluster. The license restricts how many users can work on your installation of Tableau Server.

For core-based licensing, there are two license options: Tableau Server—Single-Machine Core and Tableau Server—Multi-Machine Core. The Tableau Server—Single-Machine Core option restricts you to installing Tableau Server on a single computer. The Tableau Server—Multi-Machine Core option lets you install Tableau Server on a multi-node cluster, as long as the total number of cores for all of the computers does not exceed the number of cores that you have licensed. In either case, core-based licensing imposes no constraints on the number of user accounts in the system.

Also see [Handle an Unlicensed Server](#) on page 654.

Server Settings (General)

The following settings are available on the **General** page in **Server - Settings**.

Setting	Description
Embedded Credentials - Allow publishers to embed data source credentials in a workbook	Allows publishers to attach passwords to published workbooks that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks and are only accessible on server. That is, when the workbook is opened in Tableau Desktop, users will still need to enter a user name and password to connect to the data source. When this setting is turned off, all existing embedded passwords are saved but are not used for authentication. If you turn the setting back on, users don't have to re-embed the passwords.
Embedded Credentials - Allow publishers to schedule data extract refreshes	Allows publishers to assign workbooks to schedules. This option is only available if Allow publishers to embed data source credentials in a workbook is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.
Saved Credentials - Allow users to save data source passwords	Allows users to save data source passwords across multiple visits and browsers. By default users can choose to "Remember my password until I sign out," which lets them save their password during a single browser session. When the Saved Passwords setting is selected a user can instead choose to Remember my password , which saves the password across multiple visits and browsers so users will be automatically authenticated regardless of the computer they are using. You, as an administrator, can clear all saved passwords at any time. In addition, users can clear their own saved passwords.
Saved Credentials - Allow users to save data source access tokens	Allows users to store access tokens with their user preferences. Access tokens are provided by cloud data sources that support OAuth connections, and they are used instead of user names and passwords to grant access to the data. For more information, see OAuth Connections on page 461.

Connected Devices - Allow devices to automatically connect to Tableau Server	Controls whether mobile users must sign in and provide their credentials every time they connect to Tableau Server, or if users can connect with their devices to Tableau Server without providing credentials after they authenticate their device successfully the first time. For more information, see Authentication for Connected Devices on the next page.
Guest Access - Enable Guest account	Allows users to view and interact with embedded views without having to sign in to a Tableau Server account. Permission can be assigned to the Guest User account to control the interactivity allowed for each view. This option is only available if you have a core-based server license. This option can be used with Enable automatic logon , an option you can select during Setup .
Default Start Page	Takes you to the server's current default start page for all users. For more information on how to change the default start page, see Set the Default Start Page for All Users on the next page. Individual users will be able to override this setting (see Manage Your Content and Account Settings on page 212 for details).
Language and Locale	Controls the language used for the server user interface and the locale used for views. Individual users can override this setting on their Account Settings page. Also, web browser settings are evaluated first to determine which language and locale should be used. For more information, see Language and Locale on page 288.
Active Directory Synchronization - Synchronize Active Directory groups on a regular schedule	Controls the synchronization of all Active Directory groups in Tableau Server based on a schedule that you specify after you select the option Synchronize Active Directory groups on a regular schedule . For more information, see Synchronize All Active Directory Groups on the Server on page 260.
Reset to Default Settings	Any server settings that have been changed since setup are returned to their original state.

Authentication for Connected Devices

Tableau Server administrators can control when users of the Tableau Mobile app are required to sign in to connect to Tableau Server.

As an administrator, you can choose between the following options:

- Require Tableau Mobile users to sign in every time they connect to Tableau Server.
- Allow users to connect from their mobile devices to Tableau Server without signing in after they sign in, provide credentials, and authenticate their device successfully the first time.

The server setting that controls this behavior is **Allow devices to automatically connect to Tableau Server** in the **Server - Settings** page on the **General** tab. This setting is turned on by default.

Note: This functionality is not yet available in the Tableau Mobile app, but it is expected to be in an app update released in the near future. In addition, the functionality does not apply to signing in to a server from a mobile web browser.

Turn off this option to require users to sign in every time they connect to Tableau Server from Tableau Mobile.

Disable automatic authentication of connected devices

To immediately disconnect all devices from Tableau Server and require users to sign in every time they connect to Tableau Server in the future from the Tableau Mobile app:

1. Select **Server > Settings > General**.
2. Under **Connected Devices**, clear the option **Allow devices to automatically connect to Tableau Server**.
3. Click **Save**.

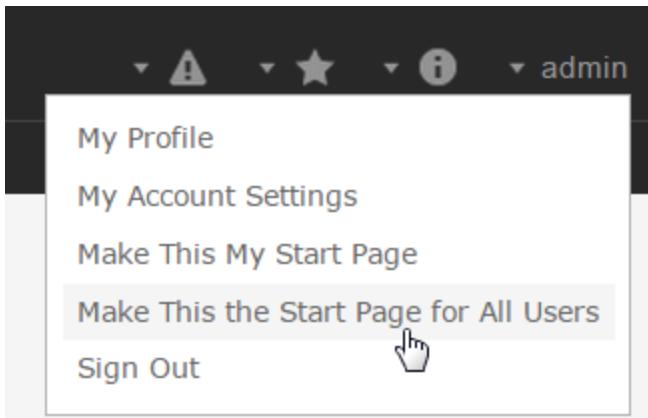
Set the Default Start Page for All Users

By default, Tableau Server installs with the Views page as the default start page for all users. As the administrator, you can change this to another page that all users have access to, such as the Workbooks page. Individual users will be able to override your setting (see [Manage Your Content and Account Settings](#) on page 212 for details).

To set the default start page for all users:

1. Navigate to the page you want to be the default page.
2. Click your name on the upper right corner of the page.

3. Select **Make This the Start Page for All Users**.



Synchronize All Active Directory Groups on the Server

As a server administrator, you can synchronize all Active Directory groups on a regular schedule or on-demand on the **General** tab of the **Server - Settings** page.

A screenshot of the 'Server - Settings' interface, specifically the 'General' tab. Under 'Active Directory Synchronization', there is a note about managing synchronization of all Active Directory groups. It shows the last synchronization time as 'Jun 25, 2015, 4:01 PM (Server time)' and a link to 'View synchronization activity'. A large 'Synchronize All Groups' button is present. Below it, a checkbox labeled 'Synchronize Active Directory groups on a regular schedule' is checked, with 'Daily' selected as the frequency. The time is set to 'at 12 : 00 AM'. At the bottom are 'Reset to Default Settings', 'Revert', and 'Save' buttons.

The **Last synchronized** time indicates the time that synchronization most recently began.

Synchronize Active Directory groups on a schedule

1. Select **Server > Settings > General**. Under **Active Directory Synchronization**, select **Synchronize Active Directory groups on a regular schedule**.

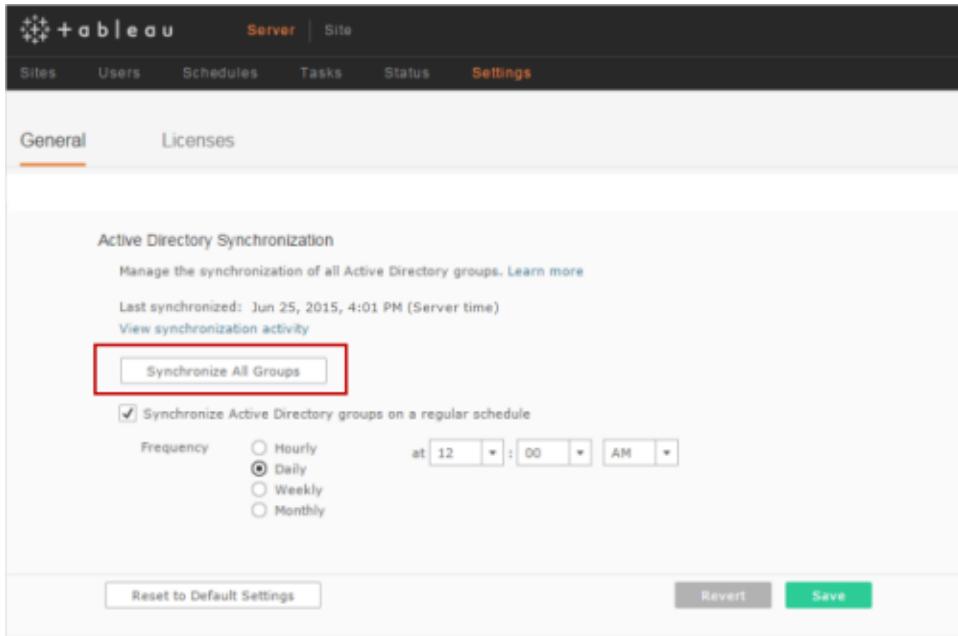
The screenshot shows the Tableau Server interface with the 'Server' tab selected. The 'Settings' tab is active. Under the 'General' tab, there is a section titled 'Active Directory Synchronization'. It displays a message to manage synchronization of all Active Directory groups, with a link to learn more. Below this, it shows the last synchronization time as 'Jun 25, 2015, 4:01 PM (Server time)' and a link to 'View synchronization activity'. A red box highlights the 'Synchronize All Groups' button and the frequency scheduling options. The frequency is set to 'Daily' at 12:00 AM. At the bottom of the section are three buttons: 'Reset to Default Settings', 'Revert', and a green 'Save' button.

2. Select the frequency and time of synchronization.
3. Click **Save**.

Synchronize all Active Directory groups on demand

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. Select **Server > Settings > General**.



2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

[View synchronization activity](#)

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. Select **Server > Status**.
2. Click the **Background Tasks for Non Extracts** link.
3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Server - Settings** page.

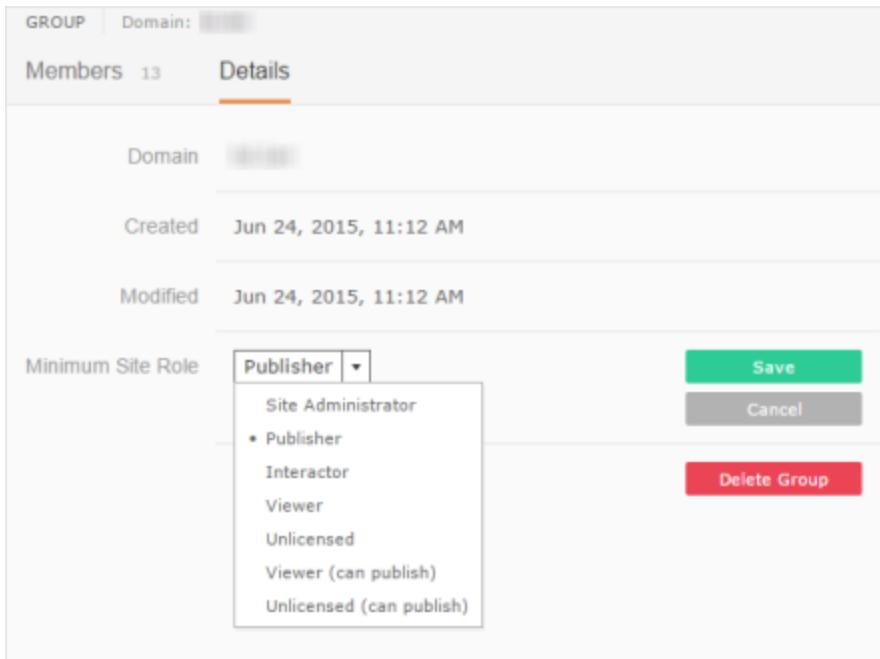
[Set the minimum site role for users in an Active Directory group](#)

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

This setting does not run synchronization; instead, it sets the minimum site role to be applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see [Site roles and Active Directory import and synchronization](#) on page 181.

1. In a site, click **Groups**.
2. Click the group name link and then click the **Details** tab.
3. Select a site role from the **Minimum site role** list, and then click **Save**.



Users removed during synchronization

When you remove a user from Active Directory, and then synchronize that user's group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user is unable to sign in to Tableau Server.

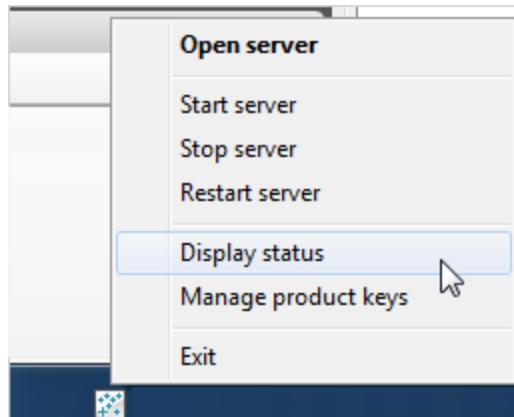
Because the user remains on the server, administrators can audit and reassign the user's content before removing the user's account completely.

For users who also exist on the server locally, the site role is set to **Unlicensed** in the site as the result of the synchronization. The user continues to belong to the **All Users** group with a site role of **Unlicensed**.

To fully remove the user from Tableau Server, you (server administrator) must delete the user from the Server Users page in Tableau Server.

Tableau Server Monitor

Tableau Server Monitor is installed as part of Tableau Server and can be accessed in the Windows system tray.



Using this tool you can start and stop the server, open Tableau Server, and display server status.

Open the Server

This command launches Tableau Server in your web browser. This is an easy way to access the web application and the associated maintenance tools.

Start/Stop the Server

You can start and stop the server using these commands. When you stop the server you make it unavailable to all of your users and terminate any sessions that are currently in progress. If someone is publishing a workbook when the server is stopped, the process is abandoned. As a result, only some of the worksheets in the workbook may be published to the server. Because stopping the server can be very disruptive to your users, be sure to warn them prior to this operation or plan maintenance during non-business hours.

Restart the Server

This command restarts the server. While the server is restarting it will be unavailable to all users. Be sure to warn your users of the outage prior to this operation. You will need to restart the server if you make changes to the Tableau Server configuration.

Display Status

This command opens a screen tip containing the status of each process. For more detailed status, use the [Maintenance page](#).

Manage Product Keys

This command opens the product key manager where you can add and remove product keys.

Exit

This command closes Tableau Server Monitor. It does not stop Tableau Server. You can re-open the application by selecting **All Programs > Tableau Server 9.2 > Tableau Server Monitor** on the Windows Start menu.

Data Sources

A data source is reusable connection to data, including connections to relational databases, cloud-hosted databases, spreadsheets, and more. To share a data source with other users, connect to the data source in Tableau Desktop, then either publish the data source to Tableau Server or embed the data source in a published workbook. Data sources can include data extracts or information for a pass-through connection to a live database. A data source can also include a layer of customizations, such as calculations, groups, or sets.

Publish a data source when you want users to connect to a data source from multiple workbooks. Workbook authors can also replace a local data source in an existing workbook with a published data source. When the published data source is refreshed, workbooks that connect to it show the updates as well.

Embed a data source in a workbook if you only want users to connect to the data source from a single workbook. Every published workbook has at least one embedded data source.

Managing data sources

As a best practice, only administrators should manage data sources. However, both administrators and data source owners can perform the following management tasks for published data sources:

- **Edit and view permissions:** Permissions can specify which users or groups can connect to, modify, or download data sources. For information, see [Set Permissions for a Data Source on page 354](#).
- **Schedule data refresh tasks:** If a data source includes an extract, you can assign the extract to a refresh schedule. For information, see [Scheduled Refresh Tasks and Subscriptions on page 216..](#)
- **Add or remove keyword tags:** Tags can contain a single word or multiple words, delimited by a comma.

- **Delete:** Deleting a data source affects workbooks that connect to the data source. Before you delete a data source, ensure that there are no workbooks that connect to the data source or edit the workbooks to use another data source.

Only administrators can perform the following tasks:

- **Move:** You can move a data source to another project.
- **Monitor security and updates:** For data sources that are proxy connections, you can stay aware of how users authenticate to the database, and whether you have the appropriate drivers installed on Tableau Server. For information, see [Database Drivers](#) on page 78 and [Data Security](#) on page 417.

About Tableau Data Sources

The Tableau Server data server is a server component that lets you centrally manage and store Tableau Server data sources. A data source is a reusable connection to data. The data can be located either in Tableau's data engine, as an extract, or in a live relational database. For relational database connections, the information stored in the data source is used for a pass-through connection to the database. The data source can also include customizations you've made at the field-level in Tableau Desktop, such as calculations, dimension aliases, groups, or sets.

For administrators, there are many advantages to using Tableau Server data sources. Because one data source can be used by many workbooks, a data source that includes an extract means you save on server space and processing time. Extract refreshes can be scheduled per-extract instead of per-workbook, and when a workbook using a Tableau Server data source is downloaded, the data extract stays on the server, resulting in less network traffic. Finally, if a database driver is required for a connection, you only have to install the driver once, on Tableau Server, instead of multiple times, on all your users' desktops.

To use the data server, authors connect to data in Tableau Desktop, either by creating an extract or using a connection to a live relational database, and publish the data source to Tableau Server. Once published, these data sources and the server contain everything workbook authors need to quickly connect to data and start authoring. To change a published data source, you download it to Tableau Desktop, make your changes, then republish, overwriting your original. Note that any new members you add to a parameter or any changes you make to the default sort order are not part of the data source (they are part of the workbook).

If you are running a distributed installation of Tableau Server and expect data sources to be heavily used, there are several ways you can optimize your server deployment. See [Distributed Environments](#) on page 73 for more information.

Note: To use published multidimensional (cube) data sources, you must download them to Tableau Desktop, so many of the above advantages do not apply. For more information, see [Cube Data Sources](#) on page 275.

The Difference Between Published Data Sources and Embedded Data Sources

Published data sources contain connection information that is independent of any workbook and can be used by multiple workbooks. An embedded data source contains connection information and is associated with a workbook. Every workbook has one or more embedded data sources. If a workbook uses a published data source, an embedded data source is listed for the workbook.

Identifying Types of Data Sources

The list of data sources gives you information about the data sources you are looking at:

Name	Connection type	Connects to
 2009 tech recession - Finance	...	Tableau Data Extract  2009 tech recession.tde
 COUNT	...	Tableau Server  COUNT
 Airline_schedule_records - Airport by I...	...	Tableau Server  Airline_schedule_records
 Airline_schedule_records (island-only)	...	Microsoft SQL Server  mssql2008.test.lan
 Airline_schedule_records (island-only)...	...	Tableau Server  Airline_schedule_records (is...
 Baseball Stats - Variety	...	Tableau Data Extract  Baseball Stats.tde

Data sources are distinguished by a number of characteristics in the list:

- **Icon/Name**—The data source icon next to the Name lets you know whether the data source is published () or embedded in a workbook ()
 - Published data source names are links. Clicking the name of a published data source opens the data source workbooks page, showing the workbooks connected to the data source.
 - Embedded data source names include the name of the workbook associated with the data source (**Variety** in the Baseball Stats - Variety data source above). The workbook name is a link, and clicking it opens the workbook on its Data Sources page.
- **Connection Type**—The connection type gives you information about the type of connection the data source is making. A connection type of **Tableau Server** indicates that the connection is to a published data source. A Tableau Data Extract connection type means that the data source has an extract which is stored in Tableau Server.

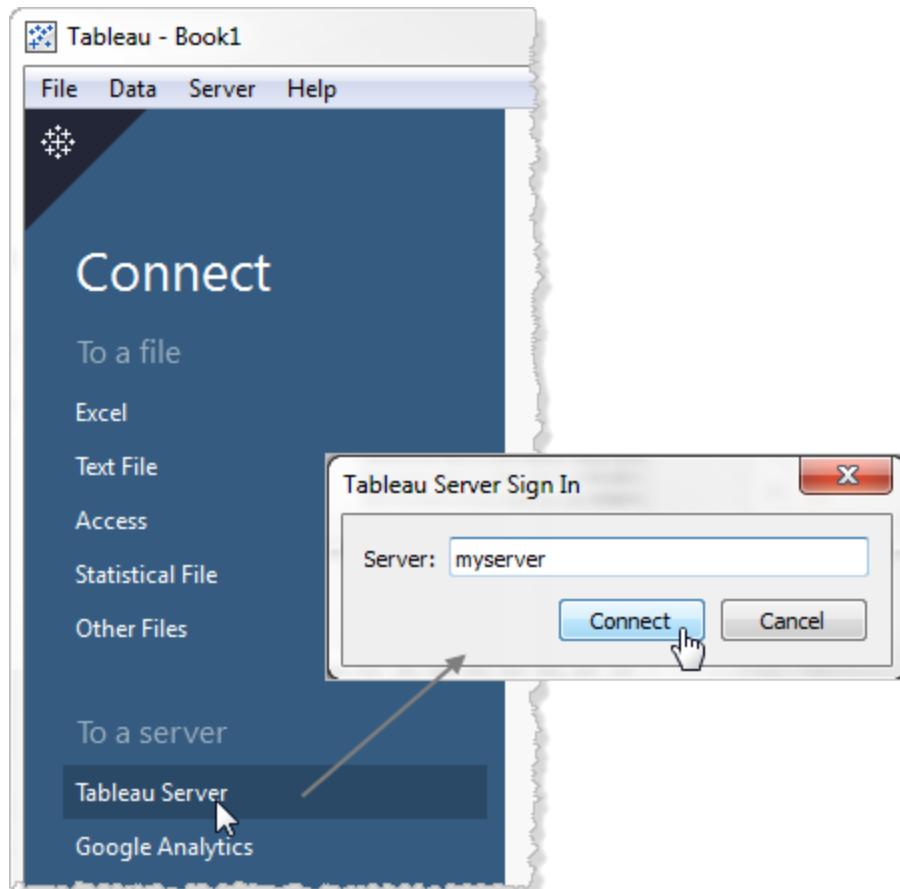
- **Connects To**—The Connects To list tells you what the data source is connecting to. This could be a database outside of Tableau Server (mssql2008.test.lan for example), an extract (2009 tech recession.tde), or a published data source (Airline_schedule_records).
- **Live or Last Extract**—This column tells you whether the connection to the data is live, or, if it is a connection to an extract, when the extract was last updated.

Connect to Published Data Sources

You can use published data sources to create new workbooks or edit existing ones. You can access published data sources from Tableau Desktop or the Tableau Server web authoring environment.

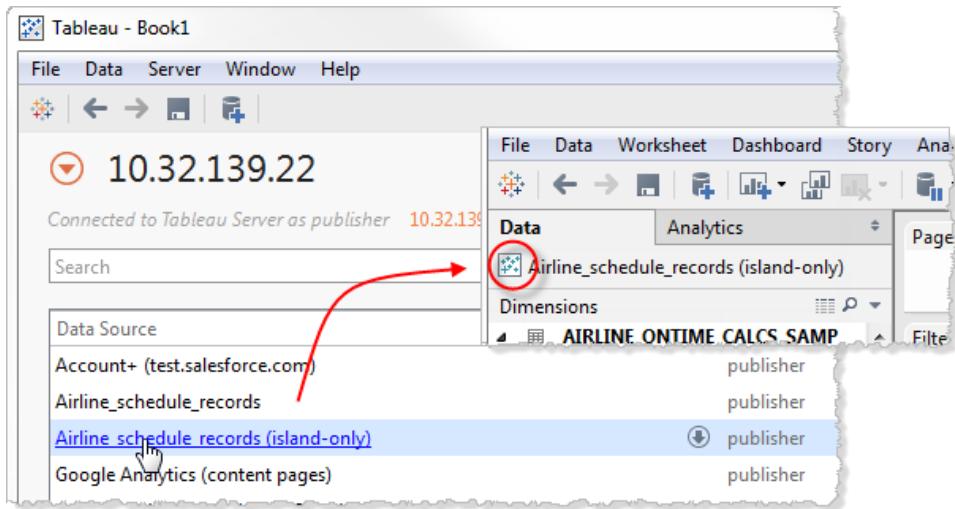
Connect to a Tableau Server data source from Tableau Desktop

1. On the Connect to Data page in Tableau Desktop, click **Tableau Server**, and then provide the server name and your credentials.



2. Select a data source you want to use.

The data source opens in the Data pane in the workbook. Tableau Server data sources show a Tableau icon instead of a database icon.



Connect to a data source in the web authoring environment

1. After you sign in to Tableau Server, display the **Content** page, and select **Data Sources**.
2. In the list of data sources, select the check box next to the one you want to use, and then click **Actions** and **New Workbook**.

Note: By default the list of data sources is filtered to only display published data sources.

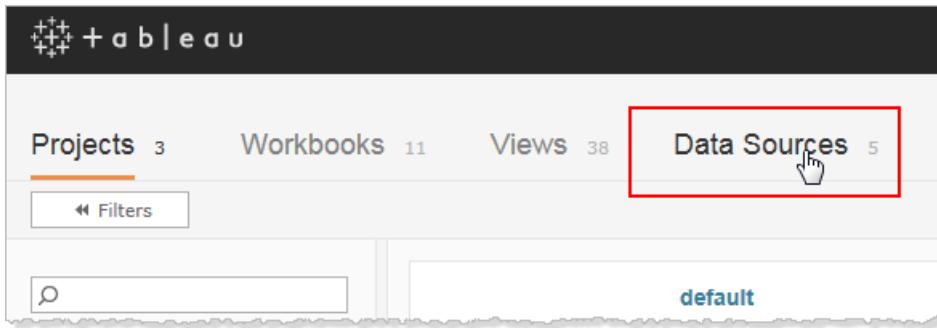
See also

[Build a View on page 673](#)

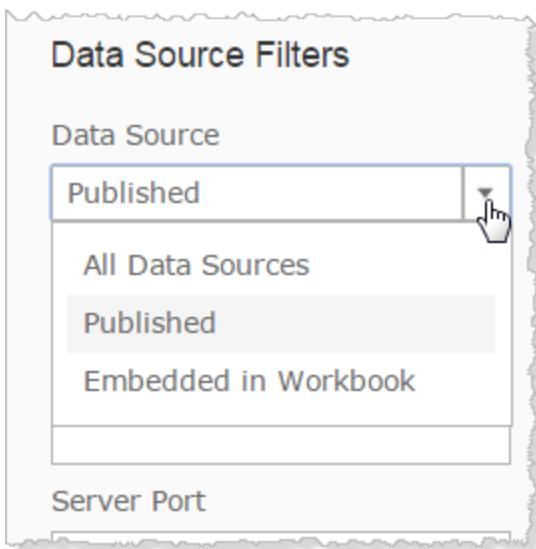
[Publish Data Sources \(Tableau Desktop Help\)](#)

Embedded Data Sources

Every workbook that is published to Tableau Server contains at least one embedded data source. These embedded data sources contain the connection information for the workbook and are listed on the Data Sources page:



By default the list of data sources is filtered to only display published data sources. To view embedded data sources, change the filter:



The Difference Between Published Data Sources and Embedded Data Sources

Embedded data sources are different from published data sources in that each embedded data source is associated with a single workbook and describes the attributes required for connecting to a data source (e.g., server name, database name, etc.). That means if you have three workbooks that connect to the same data source, you will still have three embedded data sources listed on the Data Sources page.

Searching for Embedded Data Sources

The Filter area on the left side of the Data Sources page helps you find embedded data sources by connection type, database server name, port, username, password status (whether or not the database password is embedded) and whether or not there is an extract:

Data Source Filters

Data Source
Embedded in workbook ▾

Connection type
[empty dropdown]

Server
[empty input field]

Server port
Any port

Database username
[empty input field]

Password status
Any status ▾

Has a data extract

Which Connections Can I Edit?

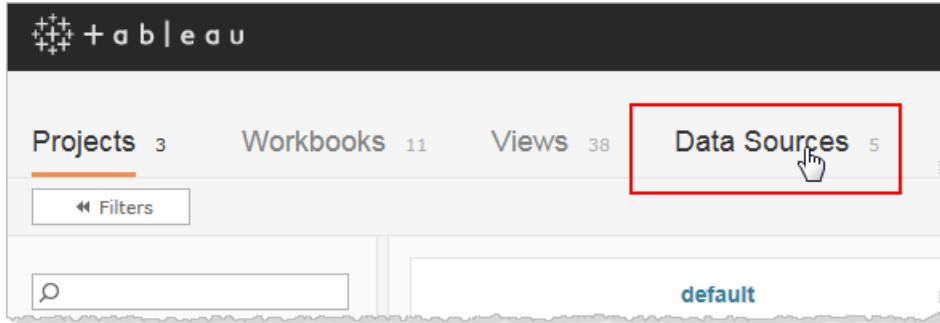
You can edit connection information for live database connections and for extracts that need to be refreshed by Tableau Server. For example, you may have a large number of workbooks that connect to a database on a specific database server. If the name of the server changes, you can update all of the workbooks at once so they reference the new server name. Another example is if a workbook connects to a database using a specific user name and password. You can quickly update all of the workbooks to use a different set of credentials.

For details on how to edit data connections, see [Edit Data Source Connections](#) below.

Edit Data Source Connections

On the Data Sources page, server administrators and data source owners can manage connection information for data sources that connect to live databases or that include extracts. You can change the database server, the server port, the username, and whether or not the password is embedded in the data source.

1. Sign in to the site that has the data sources you want to modify, and open the Data Sources page.



2. Select the data source or data sources with the connection you want to update, and on the Actions menu, select **Edit Connection**.

Name	Connection type	Connects to	Live or last extracted
<input checked="" type="checkbox"/> Airline_schedule_records	Microsoft SQL Server	mssql2008.test.lan	Live
<input checked="" type="checkbox"/> Airline_schedule_records (l...)	Microsoft SQL Server	mssql2008.test.lan	Live
<input type="checkbox"/> Location+ (Sample - Coffee ...)	Access	Sample - Coffee Chain.mdb	Jan 26, 2015, 1:30 PM

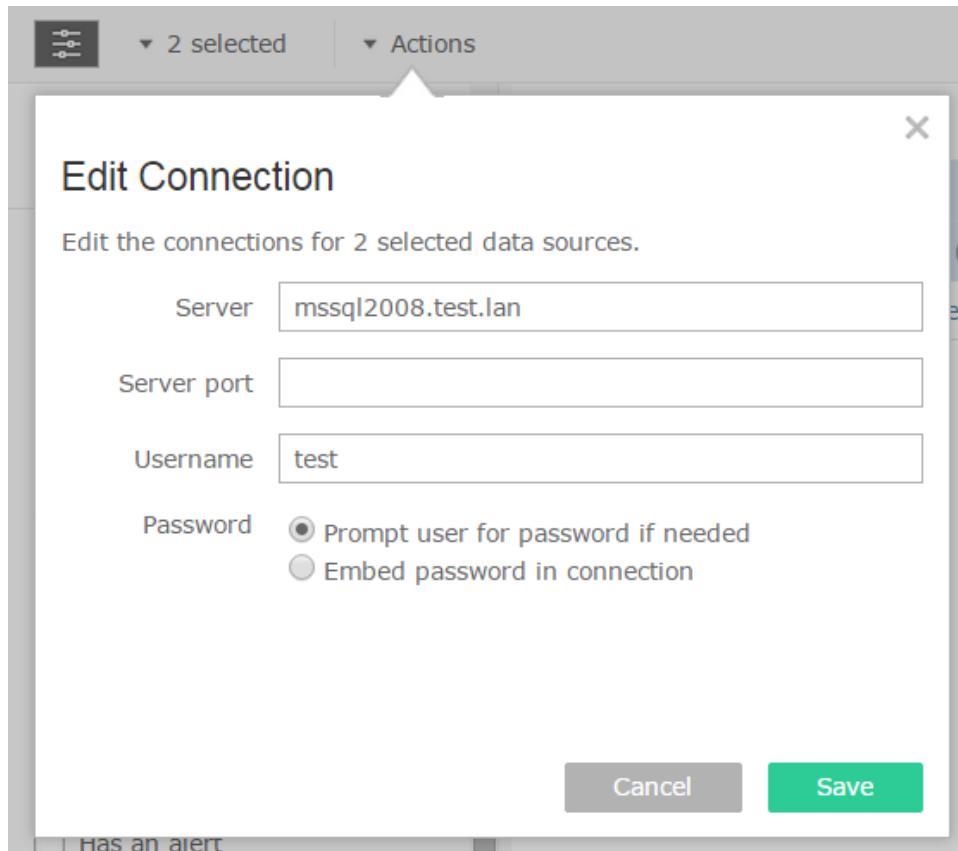
The screenshot shows the Data Sources page with two data sources selected. On the left, there are filters and a search bar. In the center, a context menu is open over the second data source, listing actions: New Workbook, Download, Tag, Permissions, Move, Change Owner, Extract Refresh, Edit Connection (which is highlighted with a red box and has a hand cursor icon), and Delete.

Use the search box or filters on the left to narrow the data source list. The values you type into the **Server** and **Database Username** fields are treated as regular expressions.

3. Update the connection options you want to change.

For connection options for Google and Salesforce.com data sources, see [Authentication Options for Google and Salesforce.com](#) on the next page later in this topic.

If a database or database driver doesn't support connecting by using an IP address, you must enter the database name as the value for **Server**. If you leave field blank, the attribute will be empty.



4. Click **Save**.
5. Refresh the Data Connections page (press F5 or Ctrl+R) for your changes to take effect.

Authentication Options for Google and Salesforce.com

Google BigQuery, Google Analytics, and Salesforce.com provide a protected authentication option. When you select this option, the connection is created through an OAuth access token. Database credentials do not need to be stored in Tableau, and all users connect through this access token, including Tableau Desktop users who want to create or edit workbooks using this connection.

For an overview, see [OAuth Connections](#) on page 461.

Google Authentication Options

When you edit Google BigQuery or Google Analytics connections, select either of the following options in the Edit Data Connection dialog box:

- Select **Embed Google BigQuery credentials in the connection** to authenticate through a designated account, and then select an existing account from the list or select **authenticate account now...** to add a new one.

When you add a new account, the Google sign-in page appears. After you provide your database credentials, Google prompts you to confirm Tableau access to the data. When you click **Accept**, Google returns an access token to use for connecting to the data.

Note: If you create extracts of your Google data source, select this first option, so that you can schedule refresh tasks.

- Select **Prompt user for Google BigQuery credentials** to require users to connect through their own individual access tokens or sign in each time they connect.

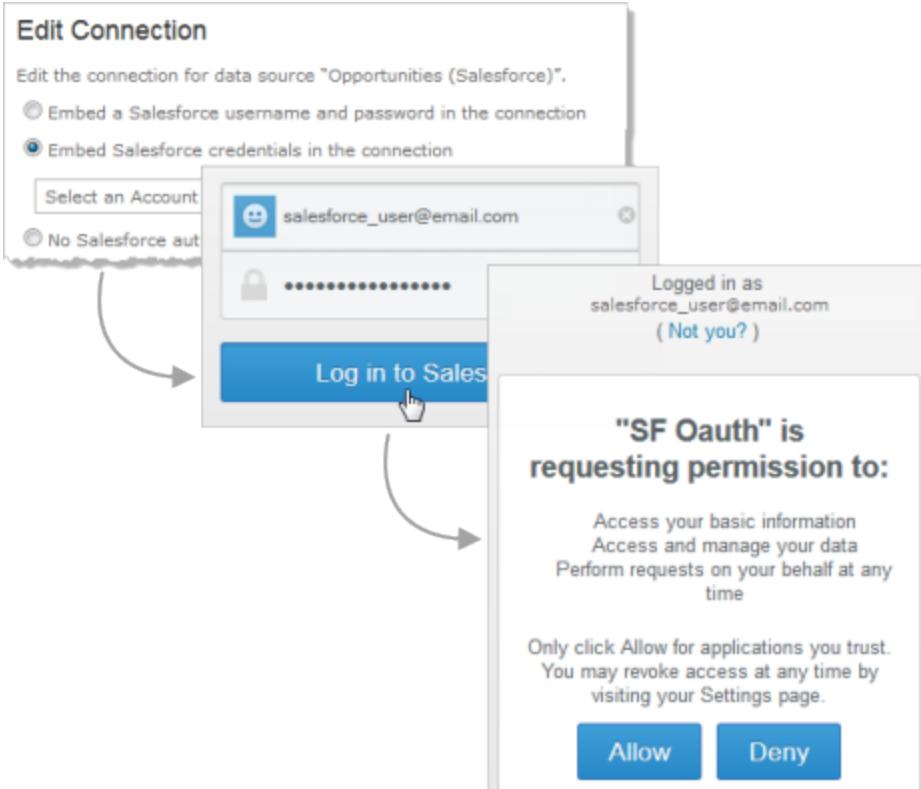
Salesforce.com Authentication Options

When you edit Salesforce.com connections, you can select any of the following options in the Edit Data Connection dialog box:

- Select **Embed a Salesforce username and password** to use a traditional authentication method.

Select **Embed Salesforce credentials in the connection** to use a protected OAuth connection and schedule refresh tasks, and then select an existing account from the list or click **Add a Salesforce Account** to add a new one.

When you add a new account, the Salesforce.com sign-in page appears. After you provide your database credentials, Salesforce.com prompts you to confirm Tableau access to the data. When you allow Tableau access, Salesforce.com creates an access token through which it connects to the data.



- Select **No Salesforce authentication** to require users to sign in to Salesforce.com each time they connect.

Monitor Progress

When you save your changes in the Edit Data Connection dialog box, the dialog displays the progress. If you close the dialog box, the modifications continue to run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped, but they will not impede other changes. For example, if you try to change the server name and add a password to several connections, the server names will be changed, and the passwords on workbooks will be changed. However, because you cannot add a password to a data source, the passwords for the data sources will not be changed.

For information about checking the progress of these tasks, see [Background Tasks for Extracts](#) on page 295.

Cube Data Sources

Cube (multidimensional) data sources have certain characteristics that make them unique in Tableau.

Cube data sources do not support pass-through connections. This means that when a cube data source is published, you cannot make a connection from Tableau Server using the data source. It also means you cannot create a workbook using the data source in Tableau Server.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally. To download a published data source you need:

- The **Download/Web Save As** permission for the data source. For more information, see [Manage Permissions](#) on page 336 and [Set Permissions for a Data Source](#) on page 354.
- Correct drivers installed and ports opened on computer running Tableau Desktop.

Web Data Connectors in Tableau Server

Web data connectors are web pages that provide a data connection that is accessible over HTTP for data sources that don't already have a connector in Tableau. Web data connectors allow users to connect to almost any data that is accessible over the web and to create extracts for their workbooks. Data sources for a web data connector can include internal web services, JSON data, REST APIs, and other sources that are available over HTTP or HTTPS. Users can create their own web data connectors or use connectors that were created by others.

This topic discusses how to import (copy) a web data connector to Tableau Server so that users can use the connector for their workbooks.

For information about how to use a web data connector as a data source, see [Web Data Connector](#) in the Tableau Desktop documentation.

For information about how to create a web data connector, see the [Web Data Connector SDK](#) documentation.

- [Why import a web data connector?](#)
- [Importing a web data connector](#)
- [Importing external files](#)
- [Listing web data connectors on the server](#)
- [Refreshing an extract using a web data connector](#)
- [Deleting web data connectors from the server](#)
- [Importing and deleting web data connectors in a distributed environment](#)
- [Performing site import and site export with web data connectors](#)
- [Managing web data connectors for failover in a cluster](#)
- [Troubleshooting](#)

Why import a web data connector?

Web data connectors contain executable code and typically make requests to third-party websites. As a security measure, Tableau Server won't load a web data connector that is hosted on the user's local computer or on a third-party site. Therefore, any data extracts created by that connector can't be refreshed on the server. (Refreshing the extract would require loading the web data connector.)

If users want to create data extracts using a web data connector and want to refresh those extracts on the server, you can *import* that connector to Tableau Server. This puts a copy of the web data connector on the server. Before you import a web data connector, you can vet and test a connector so that you know what the connector does and what sites it connects to. After you import the web data connector, users who want to use the web data connector can point to the connector on your server.

Importing a web data connector

The process of importing the web data connector gives you a chance to review the connector for security and performance issues. We recommend the following steps:

1. Obtain the HTML file for the web data connector and any supporting files, such as .css files or .js files.
2. Vet the code and HTML in the file and test the web data connector thoroughly. For more information, see [Testing and Vetting Web Data Connectors on page 282](#).
3. On the server, run the [import_webdataconnector on page 603](#) command, as in this example:

```
tabadmin import_webdataconnector connector1.html
```

Note: The connector name (connector1.html in this example) can contain only these characters: a-zA-Z0-9()~.-_.

You can import a web data connector as a local file on the server or from a network share (for example, \\myshare\connector1.html), as in these examples:

```
tabadmin import_webdataconnector  
c:\webdataconnectors\connector1.html
```

```
tabadmin import_webdataconnector  
\\myshare\webdataconnectors\connector2.html
```

If you want to re-import a web data connector that's already been imported (for example, you want to import an updated version of the connector), use the `import_webdataconnector` command with the `overwrite` option, as in this example:

```
tabadmin import_webdataconnector  
\\myshare\webdataconnectors\connector2.html --overwrite
```

When the command finishes, it displays a URL, as in this example:

```
===== Importing web data connector to server...  
-- The web data connector with the following URL  
was imported to the server:  
http://myserver/webdataconnectors/connector1.html
```

4. Give the URL of the imported web data connector to any users who want to use that connector.

Note: If you re-import a web data connector, the older version of the connector might still be available in the server's cache, and users who work with the connector might still see the older version. By default, the maximum lifetime for an item in the cache is eight hours. To force a cache reset, restart the server.

Importing external files

If a web data connector .html file references external files, you must make sure that those are available on the server. For example, a web data connector might reference an external .css file in a `<link>` element or a .js file in a `<script>` element.

If the external files are referenced using a URL (`http://`), Tableau Server can access the external files as long as the files are on a server that is accessible to Tableau Server.

If the external files are referenced as local files, you can import them into Tableau Server using the `import_webdataconnector` command. For example, if a web data connector that you are importing references the `myconnectors.css` file, you import the connector and the .css file using this sequence of commands:

```
tabadmin import_webdataconnector connector1.html  
tabadmin import_webdataconnector myconnectors.css
```

An important point is that all files imported using the `import_webdataconnector` command are stored in the same directory on the server—Tableau Server does not let you import external files into a subdirectory. Therefore, you must make sure that any local files referenced in `<link>` or `<script>` elements in the connector's .html file do not include paths, only file names.

Listing web data connectors on the server

As the server administrator, you can see a list of web data connectors by running the following command:

```
tabadmin list_webdataconnectors
```

In order to reference a web data connector in a workbook, users need to know the URL for the connector. To get a list of connector URLs, use this command:

```
tabadmin list_webdataconnectors --urls
```

Refreshing an extract using a web data connector

When a user creates a workbook that uses a web data connector, Tableau creates an extract from the data returned by the connector. If the user then publishes the workbook, the publish process sends the workbook and the data extract to the server.

Tableau can refresh an extract that was created by a web data connector, the same as it can refresh any extract. However, you must have previously imported the connector as described earlier. Tableau Server *won't* invoke a web data connector to refresh an extract if the connector is not on the server. This is a security measure, so that Tableau does not invoke code in the connector unless you have had a chance to review the code and then import the connector.

Tableau Server also cannot invoke a web data connector to refresh an extract if the connector requires credentials to sign in to the web-based data source. This is because the refresh can occur on a schedule or in some other background context, and the server cannot prompt for credentials.

If the background process that performs the refresh operation fails, it creates an alert and a log entry that indicates this issue. (Users will be able to see that the timestamp on the extract does not change.)

If you want, you can disable refresh for all web data connectors, even those that were previously imported. To disable refresh, use the `tabadmin set` command to change the `webdataconnector.refresh.enabled` setting to `false`, as in the following example:

```
tabadmin set webdataconnector.refresh.enabled false
```

Deleting web data connectors from the server

If you no longer need a web data connector, you should delete it from the server. Use the following command to remove an individual web data connector, where `connector_name` is the name of the connector file to delete:

```
tabadmin delete_webdataconnector connector_name
```

(To see a list of web data connectors on the server, use the `tabadmin list_webdataconnectors` command).

To remove all web data connectors from the server, use the following command:

```
tabadmin delete_webdataconnector --all
```

Note: When you delete a web data connector, a version of the connector might still be available in the server's cache, and users might still be able to work with the connector. By default, the maximum lifetime for an item in the cache is eight hours. To force a cache reset, restart the server.

Importing and removing web data connectors in a distributed environment

If your server is configured as a cluster, web data connectors are imported to each computer where a gateway process is running. This makes the web data connector available for distributed access across your cluster. Deleting a connector in a distributed environment removes the connector from all the computers where the gateway process is running.

In a distributed environment, the process of importing or deleting a web data connector might complete only partially. If you're importing a connector, the connector might be copied to some of the computers where the gateway process is running, but not to all of them. In that case, the `tabadmin import_webdataconnector` command reports the error using text like this:

The web data connector with the following URL has been imported to some gateways on the server, but not all.

Similarly, if you're deleting a web data connector, the connector might be removed from some computers but not all of them. The `tabadmin delete_webdataconnector` command reports the error using text like this:

The web data connector was deleted from some gateways on the server, but not all.

Note: If the delete process is partially successful, users can still access the connector.

If the import or delete process reports partial success, you can try either of the following solutions:

- Run the import or delete process again. If you're importing, run the `tabadmin import_webdataconnector` command again, and use the `--overwrite` option to overwrite any instances of the connector that were successfully installed. If you're deleting, run the `tabadmin delete_webdataconnector` command again. Tableau Server will remove any remaining instances of the connector.
- Stop the server, run `tabadmin configure`, and then restart the server. The configuration process makes sure that any web data connectors are correctly distributed (imported or deleted) in all nodes where the gateway process is running. Since this option requires you to stop the server, you would choose it if it's practical to stop the server, or if you have some other reason to stop and restart the server.

Performing site import and site export with web data connectors

Web data connectors are imported as server-wide resources; they are not associated with a specific site on your server. Therefore, if you export a site using the `tabadmin exportsite` command, the resulting .zip file does not include web data connectors that might be referenced by workbooks on the site.

Managing web data connectors for failover in a cluster

If your server is configured as a cluster with a backup primary server, you must make sure that web data connectors that you have imported to the primary are available if you need to failover to your backup primary. If the web data connectors are not available on the new primary after a failover, running the configuration process on the primary server can end up removing the connectors from other computers where a gateway process is running.

To make sure that web data connectors are available after a failover, follow these steps:

1. Make sure that you keep an up-to-date backup of the web data connectors that have been imported to your server.
2. After the primary fails, and before you start the backup primary, copy the web data connectors from the backup location to the following folder on the backup primary:

```
C:\ProgramData\Tableau\Tableau  
Server\data\tabsvc\httpd\htdocs\webdataconnectors
```

If you have created a backup of the primary server using the `tabadmin backup` command, the .tsbak file created by the backup file contains the web data connectors. You can extract the contents of a .tsbak file and get the web data connectors.

If you installed Tableau Server on a different drive, substitute that drive letter for C:.

3. Overwrite the tabsvc.yml file on the backup primary.
4. Run the `tabadmin failoverprimary` command. For more information, see [Quick Start: Creating a Backup Primary on page 86](#)

If necessary, you can also reimport the web data connectors, as described earlier in this topic.

Troubleshooting

If the server experiences problems with importing or deleting web data connectors, you can examine the `tabadmin.log` files. Be sure to check the log files on both the primary server and on the other servers that are running the gateway process. For more information about log files, see [Server Log File Locations on page 645](#).

If the issue is that Tableau Server will not refresh an extract that was created by a web data connector, make sure that the web data connector that created the extract has been imported to the server. In addition, make sure that the `webdataconnector.refresh.enabled` configuration setting has been set to `true`.

If you have re-imported a changed web data connector on the server (overwriting an existing one), but users who work with the web data connector are not seeing the changes, the users might be getting a cached version of the older version. By default, the cache is reset after eight hours; after a cache reset, older versions of the web data connector will no longer be used. If you want to force the cache to reset, you can restart the server.

If you have deleted a web data connector from the server but users are still able to work with the connector, the connector is probably still in the server's cache. A web data connector can stay available in the cache for up to eight hours. To clear the cache, restart the server. If you delete a web data connector from a server in a distributed environment, make sure that the connector has been successfully deleted from all computers where a gateway process is running.

Testing and Vetting Web Data Connectors

Web Data Connectors contain JavaScript that typically connects to data on another site. Because of this, you should test and vet web data connectors before users use them as data sources for a workbook, and before you import them into Tableau Server.

This topic includes some suggestions for testing and vetting web data connectors.

- Examine the source
- Test the web data connector in an isolated environment
- Monitor the traffic created by the connector
- Test the performance and resource usage of the connector

Examine the source

The code in a web data connector is in JavaScript, so you can open the file (and any external files that the connector uses) and examine the source code.

Many connectors reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe. If it is practical for you to change the source code of the connector, use HTTPS protocol (`https://`) to reference external libraries (if the source site supports HTTPS) to help verify the site's authenticity.

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request.

Note: Experienced JavaScript programmers often compress (minify) their code to reduce the size of the code for download. Dense blocks of code that use cryptic function and variable names are not uncommon. While this can make it more difficult to examine the code, it is not a sign that the code was written to be deliberately difficult to understand.

Test the web data connector in an isolated environment

If possible, test the web data connector in an environment that is isolated from your production environment and from user computers. For example, import a web data connector onto a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor the traffic created by the web data connector

When you test a web data connector, use a tool like [Fiddler](#), [Charles HTTP proxy](#), or [Wireshark](#) to examine the requests and responses that the connector makes. Make sure that you understand what sites the connector makes requests to and what content the connector is requesting. Similarly, examine the responses and their content to be sure that the connector is not reading data or code that is not directly related to the connector's purpose.

Test the performance and resource usage of the web data connector

When you test a web data connector, use tools to monitor its CPU and memory usage. Remember that the web data connector will run on Tableau Server, which is an environment in which many processes are already running. You want to make sure that when the connector fetches data, the connector does not have an undue impact on server performance.

Check whether the connector writes to disk. If it does, check how much disk space it occupies, and examine the output to make sure you understand what it's writing and why.

Troubleshoot Data Sources

For users to work with Tableau Server data sources, up to three things need to be in place:

- **Permissions for the data source:** Anyone connecting to a data source must have the **Connect** and **View** permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be licensed to Publish and also have the **Write/Save As** and **Download/Web Save As** permissions. See [Manage Permissions on page 336](#) and [Set Permissions for a Data Source on page 354](#) for more information.

Multidimensional (cube) data sources have to be downloaded and used in Tableau Desktop, so they require **Download/Web Save As** permission. For more information about cubes in Tableau, see [Cube Data Sources on page 275](#).

- **Ability to authenticate to the database:** There are several ways you can connect to data in Tableau and control who has access to what. Basically, whichever entity is connecting to the database must be able to authenticate. The entity could be Tableau Server performing an extract refresh. It could be a Tableau Desktop user connecting to a data source that then connects to a live database. It could also be a Tableau Server user who's accessing a view that connects to a live database. Refer to [Data Security on page 417](#) to learn more about your options.
- **Database drivers:** If the person who created and published the data source in Tableau

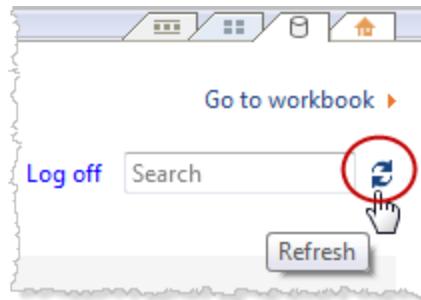
Desktop needed to install additional database drivers, you may need to install them on Tableau Server as well. If you are running a distributed installation of Tableau Server where, for example, the data server process is running on a worker server, any required database drivers must be installed there as well as on the primary server. Other processes require drivers as well. See [Database Drivers](#) on page 78 for more information.

Data Source Error Messages

Here are some errors that workbook authors and other users may encounter as they work with data sources and views:

Permission to access this Tableau Server data source denied: Connecting to a data source requires the Connect permission. See [Manage Permissions](#) on page 336 and [Set Permissions for a Data Source](#) on page 354 for more information.

Data source not found: Someone working with a view may see this error if a data source is removed from Tableau Server or if their Connect to Data page needs to be updated. To update the Connect to Data page in Tableau Desktop, click the Refresh icon:



Unable to connect to this Tableau Server data source: This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct settings.

Unable to list Tableau Server data sources: This error may occur if a user is trying to access Tableau Server data sources and there are connectivity issues between Tableau Server and Tableau Desktop.

Can't connect with a cube data source: To use a published multidimensional (cube) data source, you must download the data source and use it in Tableau Desktop. Verify that you have the [Download/Web Save As](#) permission for the data source. For more information about cubes in Tableau, see [Cube Data Sources](#) on page 275.

Customize the Server

You can customize how Tableau Server looks to personalize it for your company or group. For example, you can change the name that appears in screen tips and messages, and you can change the logo that appears on most server pages.

You can also customize how users can interact with the server. For example, you can allow workbook publishers to embed their data source credentials so that when people click a published view with a connection to a live data source they get immediate access to the view and don't have to supply their database credentials first.

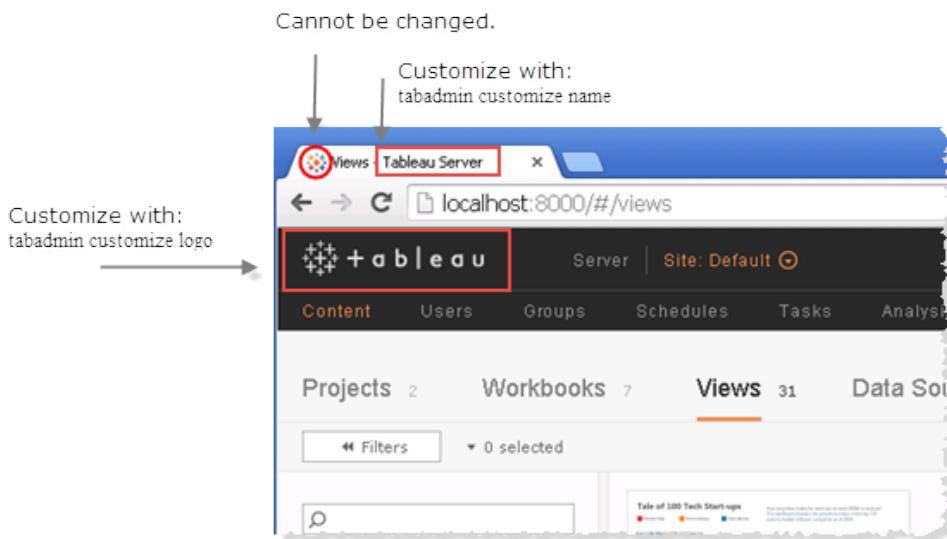
You can also control which language is used for the server user interface and which locale is used for views.

See the following topics for more information on customizing Tableau Server:

Change the Name or Logo

You can customize the Tableau Server look and feel by changing the displayed name, the logo, or both.

The customizable name appears on browser tabs and in a tooltip when you hover over the home logo in the upper left. The customizable logo appears in the upper left and on the sign-in screen. Some logos and references to Tableau Server cannot be changed, for example the logo on browser tabs and the phrase "Tableau Server" in the copyright notice.



Change the Name

By default Tableau Server displays a tooltip "Tableau Server" when you hover over the Tableau logo:



Note: The copyright information in the About Server dialog box will still list Tableau (for example, ©2015, Tableau Software, Incorporated and its licensors. All rights reserved.)

To change the name that appears in the tooltip:

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Type:

```
tabadmin customize name "new_name"
```

Replace "new_name" with the text that you want to appear as the name on the server.

Example: tabadmin customize name "My Company"

Note: To change to a name that includes Unicode characters, identify the hex encoding for each Unicode character and add "\u" before each hex value. For example for the two-character string 测试, you would type the command
tabadmin customize name "\u6D4B\u8BD5".

3. Restart the server for the change to take effect by typing:

```
tabadmin restart
```

Change the Logo

You can customize the logo that appears on the Tableau Server sign-in page and in the upper left of the server pages. The name "Tableau" is part of this logo. It cannot be changed independently of the logo.

Note: The background colors differ in the two locations, so your logo may look different depending on which location you are viewing.



If an image is larger than 160 x 160 px (large logo), it will be clipped. The image file you use should be in GIF, JPEG, or PNG format.

To change the logo:

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Type (for an image up to 160 x 160 px, but not smaller than 32 x 32 px):

```
tabadmin customize logo "C:\My Pictures\logo.png"
```

3. Restart the server for the change to take effect by typing:

```
tabadmin restart
```

Restore the Default Name or Logo

You can restore Tableau Server's default look and feel by doing the following:

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Restore the default logo or name by typing the following:

```
tabadmin customize <parameter> -d
```

In the above line, replace `<parameter>` with either `name` or `logo`.

3. Restart the server for the change to take effect by typing:

```
tabadmin restart
```

Language and Locale

Tableau Server is localized into several languages and has language and locale settings. The **Language** setting controls user interface (UI) items such as menus and messages. The **Locale** setting controls items in views such as number formatting and currency.

Administrators can configure language and locale on a server-wide basis (see [Server Settings \(General\) on page 257](#), and individual users can configure their own settings (see [Manage Your Content and Account Settings on page 212](#)). If a user configures their own language and locale, their settings override the server settings.

Default Settings

The default language for Tableau Server is set during Setup. If the host computer is configured for a language Tableau Server supports, it installs with that language. If it's not a supported language, Tableau Server installs in English.

How Language and Locale are Determined

Another influence on which language and locale display when a user clicks a view is the user's web browser. If a server user has not specified a **Language** setting on their User Account page, and their web browser is set to a language that Tableau Server supports, the browser's language will be used—even if Tableau Server itself is set to a different language.

Here's an example: Assume that Tableau Server has a system-wide setting of English as the **Language** for all users. Server user Claude does not have a language specified on his Tableau Server User Account page. Claude's browser uses German (Germany) for its language/locale.

When Claude signs in to Tableau Server, the server UI displays in German and when he clicks View A, it's using the Germany locale for numbers and currency. If Claude had set his user account **Language** and **Locale** to French (France), the UI and view would have been

displayed in French. His user account setting supercedes those of his web browser, and both of those have precedence over Tableau Server's system-wide setting.

Another setting to be aware of is the **Locale** setting in Tableau Desktop (**File > Workbook Locale**). This setting determines the locale of the data in the view, such as which currency is listed or how numbers are formatted. By default, **Locale** in Tableau Desktop is set to **Automatic**. However, an author can override that by selecting a specific locale. Using the above example, if the author of View A set **Locale** to **Greek (Greece)**, certain aspects of the data in View A would display using the Greek (Greece) locale.

Tableau Server uses these settings, in this order of precedence, to determine language and locale:

1. Workbook locale (set in Tableau Desktop)
2. Tableau Server User Account language/locale settings
3. Web browser language/locale
4. Tableau Server Maintenance page language/locale settings
5. Host computer's language/locale settings

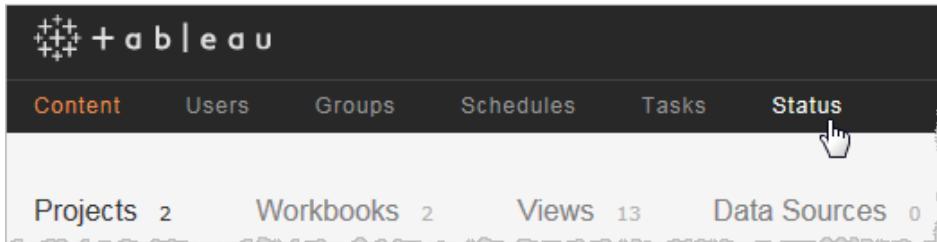
Administrative Views

The Status page contains an embedded Tableau workbook with various administrative views for your Tableau Server or site. These views help you to monitor different types of activity on the server or site.

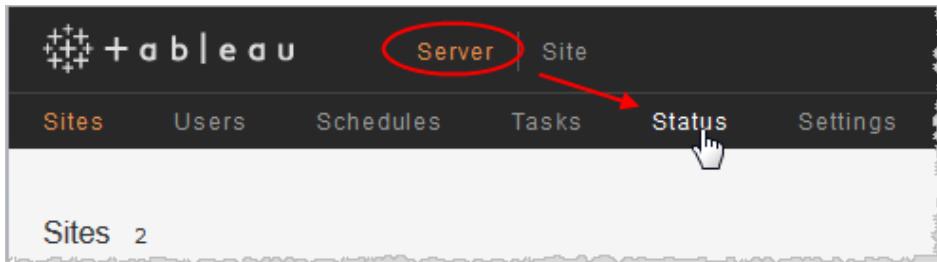
Analysis	
Dashboards that monitor Tableau Server activity.	
Views	Analysis
Traffic to Views	View count, viewers, and viewer behavior for published views.
Traffic to Data Sources	Data source usage, users, and user behavior for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.

Navigating to administrative views

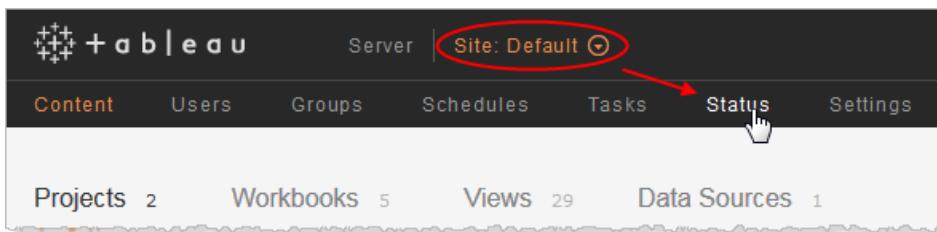
Site administrators can see administrative views for their site. Administrators of multiple sites can see views for the current site.



Server administrators can see views for the entire server:



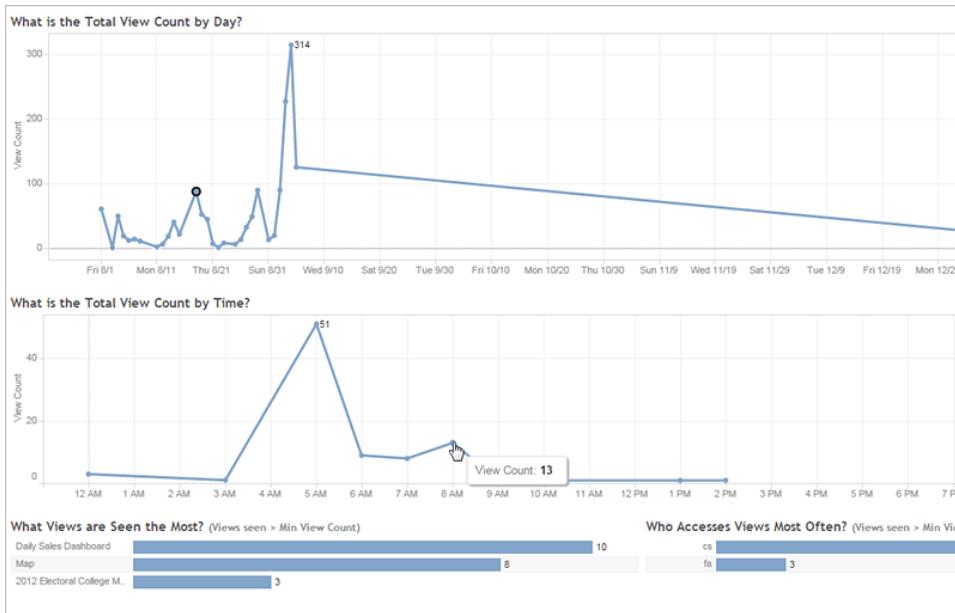
Or for individual sites:



Traffic to Views

The Traffic to Views view gives you the ability to see how much of your user traffic goes to views.

You can filter what information is displayed and the time frame it comes from by selecting the view, the workbook, and the time range. Server administrators can specify the site.



Two time lines at the top of the view show you how views are being used over a time range you specify (the default is the last 7 days):

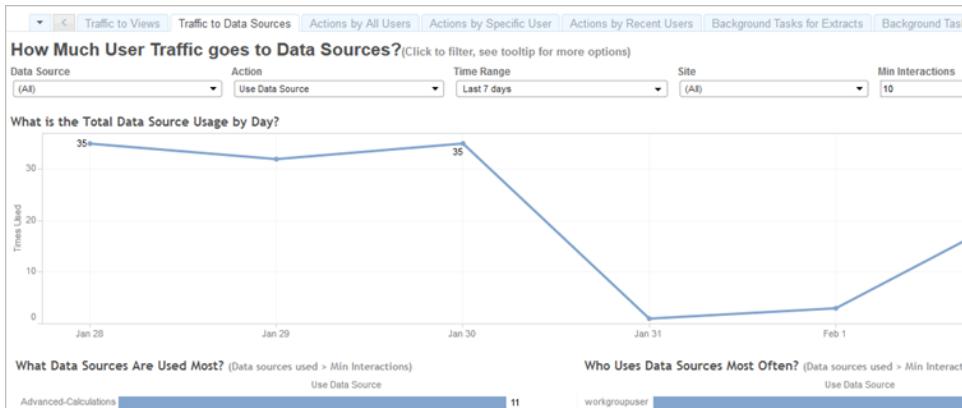
- **What is the Total View Count by Day**—This shows total view count by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count of views. Select the point to update the other sections of the view based on your selection.
- **What is the Total View Count by Time**—This shows the view count by time of day. The filters and any selection impact this graph.

Two bar graphs at the bottom of the view show results that are filtered by the **Min View Count** filter at the top of the view. These show you the views that are most often accessed, and the users who most frequently access views Only those views and users with counts greater than or equal to the minimum view count value are displayed:

- **What Views are Seen the Most**—This is a list of the most visited views. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Accesses Views Most Often**—This shows the users who most often access the views and is limited by filters and any selection you make.

Traffic to Data Sources

The Traffic to Data Sources view gives you the ability to see usage of data sources on your Tableau Server installation. This can help you determine which data sources are most heavily used and those that are less often used. You can filter the information you see by selecting the data source, the action taken on that data source, and the time range. Server administrators can specify the site.



A time line at the top of the view shows you how data sources are being used over a time range you specify (the default is the last 7 days):

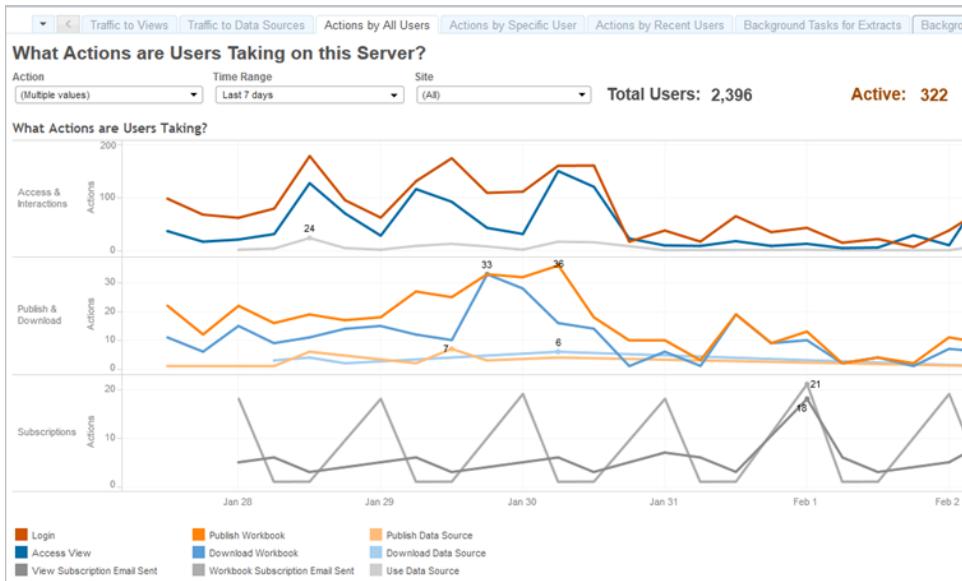
- **What is the Total Data Source Usage by Day**—This shows total data source usage by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count. Select the point to update the other sections of the view based on your selection.

Two bar graphs at the bottom of the view show results that are filtered by the **Min Interactions** filter at the top of the view. These show you which data sources are most used, and who uses data sources most often. Only those data sources and users with interaction counts greater than or equal to the minimum interactions value are displayed:

- **What Data Sources are Used Most**—This is a list of the most used data sources. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Uses Data Sources Most Often**—This shows the users who most often use the data sources. This is impacted by filters and any selection you make.

Actions by All Users

The Actions by All Users view gives you insight into how your Tableau Server installation is being used. You can filter the view by actions and by time range. Server administrators can filter by site. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of active users who have performed one of the selected actions.



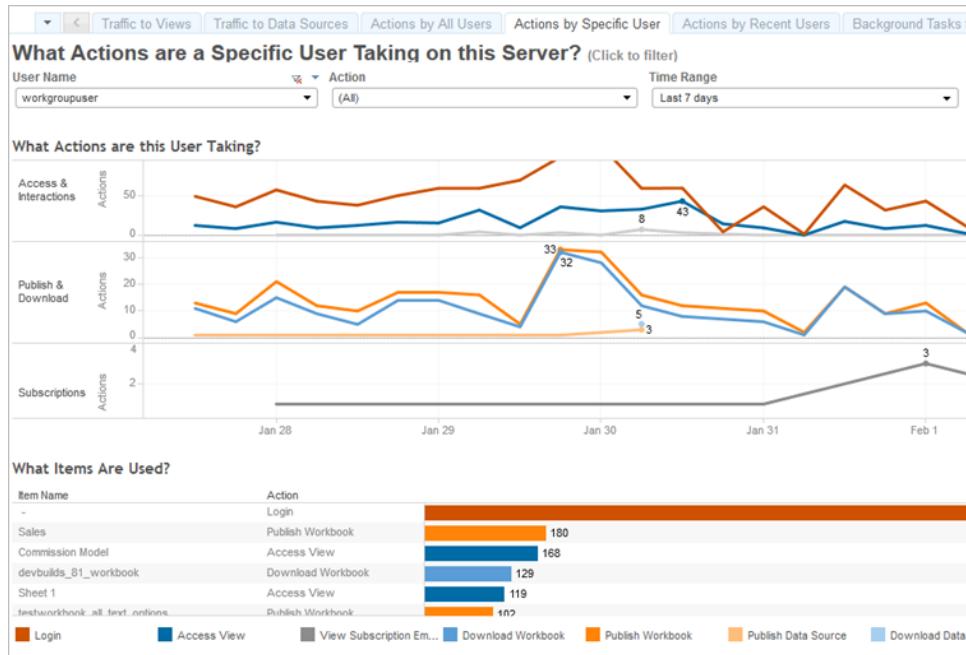
Up to three separate groups of time lines show you how users are using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

Actions by Specific User

The Actions by Specific User view gives you insight into how individual users are working in your Tableau Server installation. You can filter the view by user name, actions, and time range. Server administrators on multi-site installations can filter by site.



Up to three separate groups of time lines show you how a selected user is using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, or if no actions were taken, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

A bar graph at the bottom of the view shows which items the selected user is using.

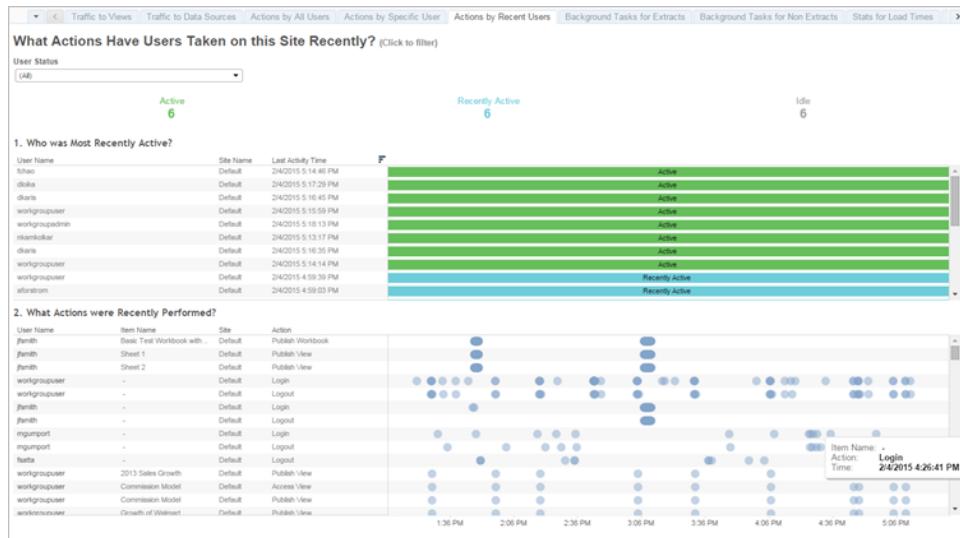
Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

Actions by Recent Users

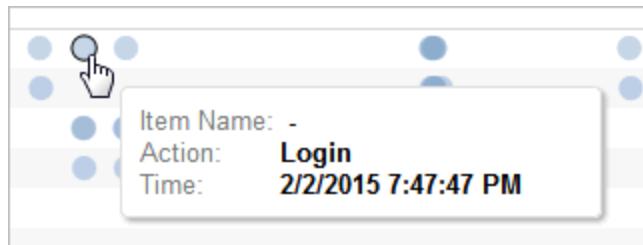
The Actions by Recent Users view shows you which signed-in users have been active on Tableau Server recently. This can be useful if you need to perform some maintenance activity and want to know how many and which users this will affect, and what they are doing on Tableau Server.

The view **Active**, **Recently Active**, and **Idle** users that are currently signed in to Tableau Server. For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last

took an action more than 30 minutes ago. The actions are displayed in the lower section of the view.

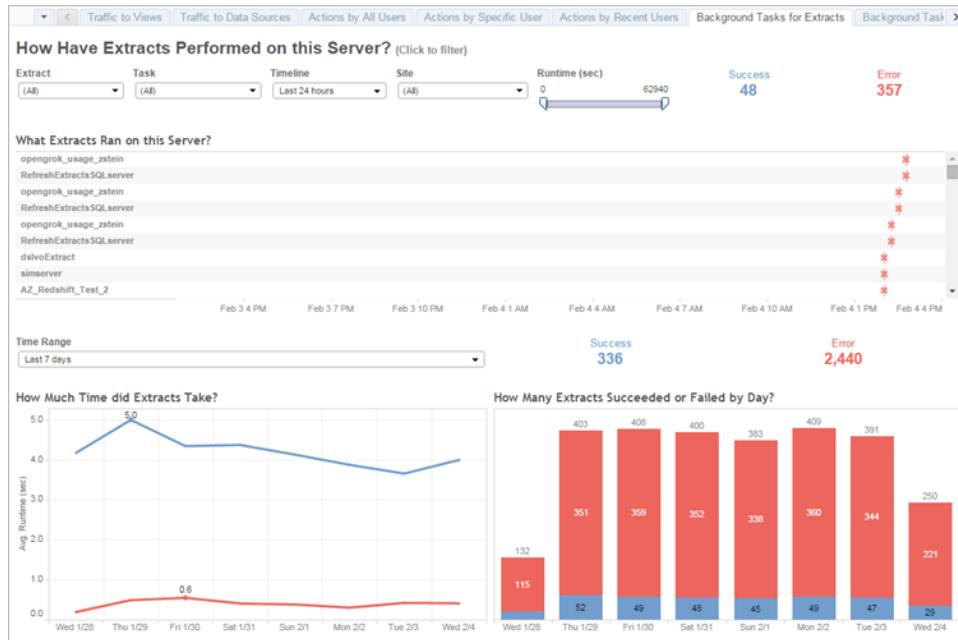


Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.



Background Tasks for Extracts

The Background Tasks for Extracts view displays extract-specific tasks that run on the server.



A table lists the extracts that ran in the time period specified in Timeline. Click **Success** or **Error** to filter the table based on status. Click a specific task to update the **How Much Time did Extracts Take** graph for the selected task. The **How Many Extracts Succeeded or Failed** table updates for the status (success or failure) of the task, but the count of extracts that succeeded or failed does not change.

Tasks can have a status of successful or error:

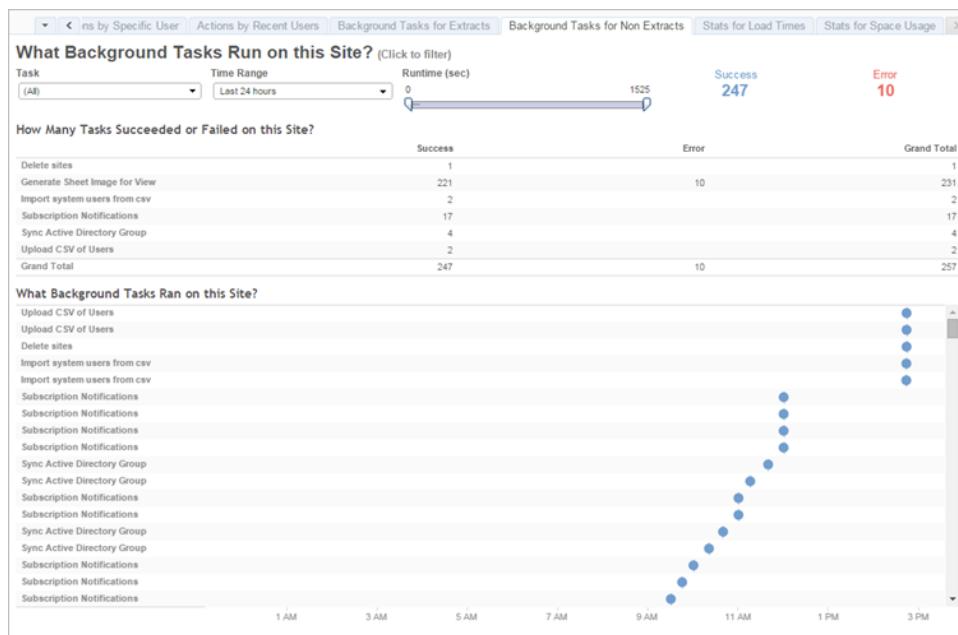
Icon	Description
✗	Error —Server was unable to complete the task.
✓	Success —Server completed the task.

For details on a task, hover over its icon:



Background Tasks for Non Extracts

The Background Tasks for Non Extracts view displays tasks that the server runs that are not related to refreshing extracts. For example, edited OAuth connections, subscription notifications, and so on.



A table lists the tasks that ran in the time range specified. Click **Success** or **Error** to filter the table based on status. Select a specific task in the **How Many Tasks Succeeded or Failed on this Site** table to update the **What Background Tasks Ran on this Site** graph for the selected task.

Tasks can have a status of successful or error.

Icon	Description
✗	Error —Server was unable to complete the task.
●	Success —Server completed the task.

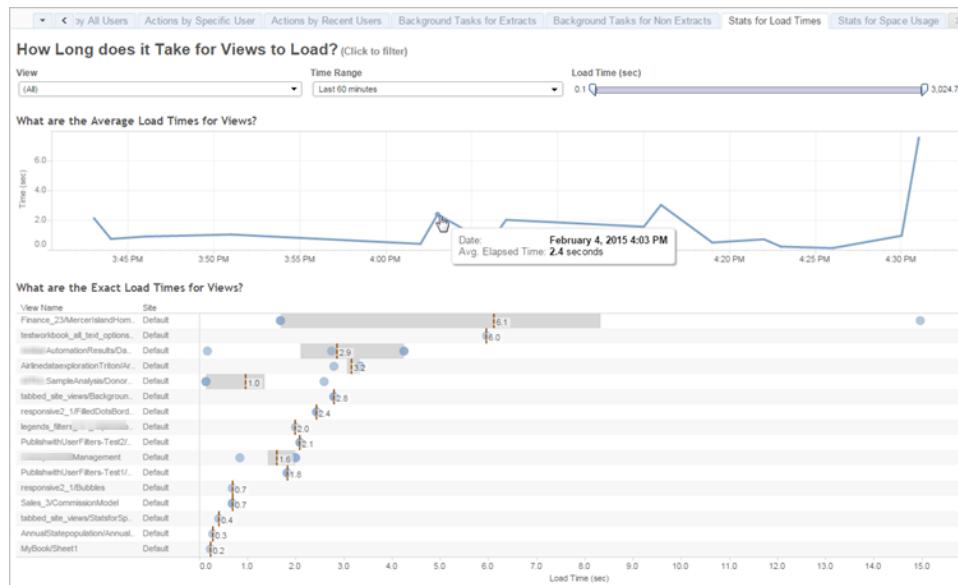
For details on a task, hover over its icon.

Stats for Load Times

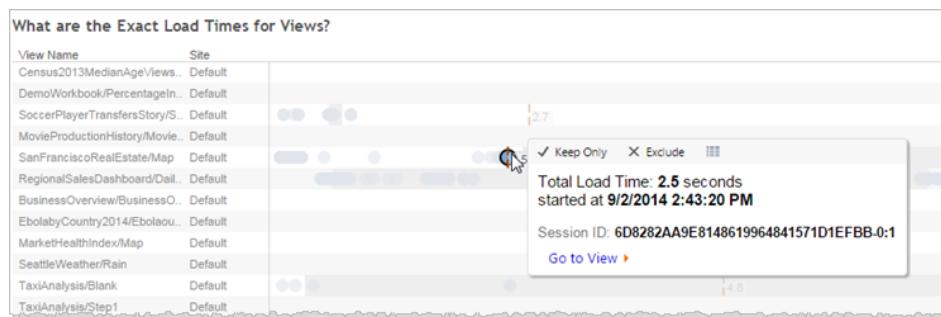
The Stats for Load Times view shows you which views are the most expensive in terms of server performance. You can filter by view and time range. Server administrators can filter by site. You can also limit the view based on load time in seconds, using the sliding Load Time

filter. Load times are for the server. Depending on your client browser and networking, actual load time may vary slightly.

The **Average Load Times** graph shows average load times for views based on the filters you set. Hover over a point to see details. Select a point on the line to update the rest of the view for the selection:

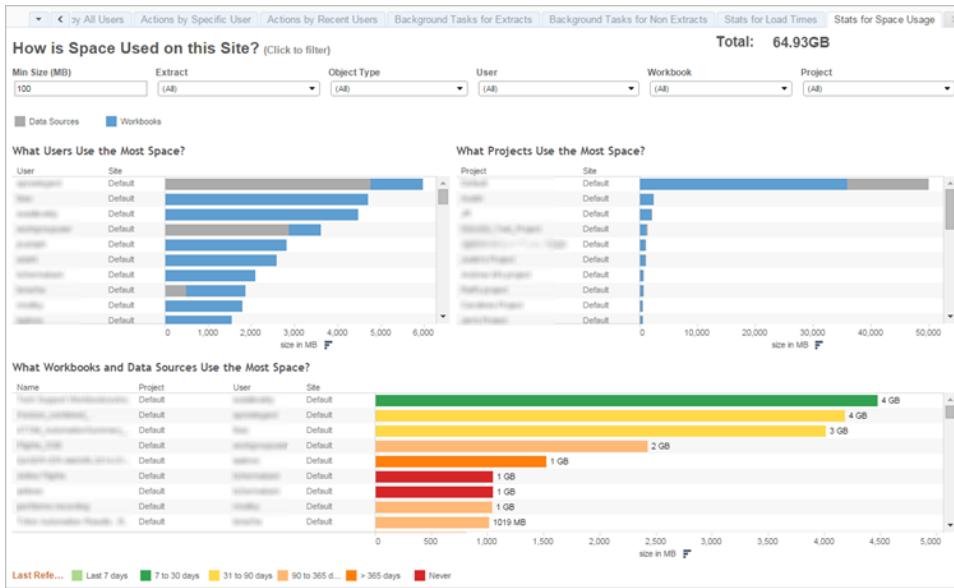


The **Exact Load Times** view shows exact time to load the listed views. A vertical line shows the average load time for each view. Select a mark to see details of a specific instance of the view loading:



Stats for Space Usage

The Stats for Space Usage view can help you identify which workbooks and data sources are taking up the most disk space on the server. Disk space usage is displayed by user, project, and by the size of the workbook or data source and is rounded down to the nearest number:

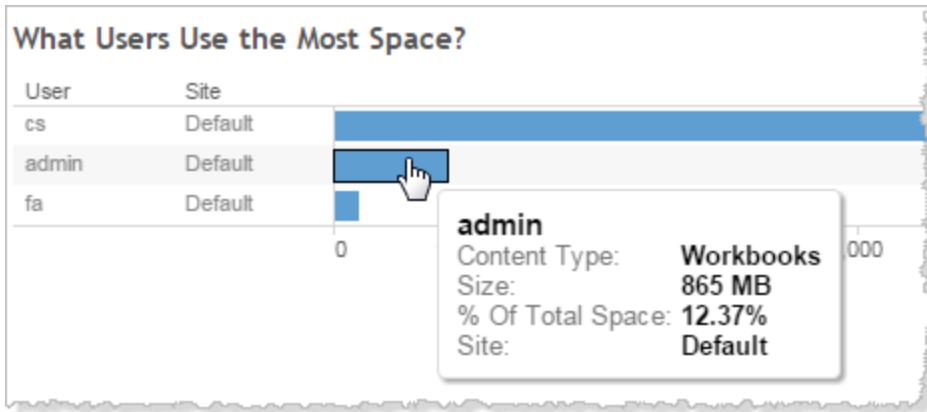


Use the **Min Size** filter to control which data sources and workbooks are displayed, based on the amount of space they take up.

Three bar graphs give you information about space usage on your Tableau Server:

- **What Users Use the Most Space**—This shows the users who own data sources and workbooks that are taking up the most space. Click a user name to filter the next two graphs for that user. Click the data source bar or the workbook bar for a user to filter the next two graphs for that type of object for that user. Click the selected user or bar to clear the selection.
- **What Projects Use the Most Space**—This shows the projects with the data sources and workbooks that are using the most space. If a user or object type is selected in the What Users Use the Most Space graph, this displays information specific to the selection.
- **What Workbooks and Data Sources Use the Most Space**—This shows the workbooks and data sources that are taking the most space. The bars are color-coded based on the length of time since the last refresh.

Move your cursor over any bar to display usage details:



Click on a bar to select it and update the other areas of the view based on that selection.

Create Custom Administrative Views

In addition to the pre-built administrative views available on the Maintenance page on the Server, you can use Tableau Desktop to query and build your own analyses of server activity. To do this, you can connect to and query views in the Tableau Server repository using one of two built-in users: the "tableau" or "readonly" user.

- The **tableau** user—The tableau user has access to special views and a subset of tables in repository database. These views and tables are provided so that administrators can create custom administrative views. Tableau makes an effort to limit changes to these tables and views so that custom views built with them do not break.
- The **readonly** user—The readonly user has access to a large number of the repository tables, providing more data about server usage. Administrators can use these to create custom administrative views too, but many of the tables are intended primarily to support the functioning of Tableau Server and may be changed or removed without warning. This means that views created from these tables can break when the database structure is changed.

Note: The readonly user is available in Tableau Server 8.2.5 and later.

For examples of using the readonly user to connect to the workgroup database, see the following articles in the Tableau Knowledge Base: [Group Membership](#), [Server Access](#), [Server Access \(2\)](#), and [Workgroup Usage](#)

Before you can connect using one of the built-in users, you must enable access to the Tableau Server database. After doing this you can use Tableau Desktop to connect to and query the database as the tableau user or the readonly user.

The tabadmin set option `auditing.enabled` controls whether Tableau Server collects historical user activity and other information in the repository. It is enabled by default. Be aware that

collecting historical events impacts the size of Tableau Server's backup file (.tsbak).

- All hist_tables are controlled by the tabadmin set option `wgserver.audit_history_expiration_days`, which controls how many days of event history are kept in the repository and has default value of 183 days.
- The _http_requests table is cleaned of all data older than 7 days every time tabadmin [cleanup on page 590](#) or tabadmin [backup on page 589](#) is used. For more information, see [Remove Unneeded Files on page 634](#).
- The _background_tasks table is cleaned automatically and keeps data for the last 30 days.
- All other tables with names that begin with a "_" prefix contain current data.

Enabling External Access to the Tableau Server Database

You can use Tableau Desktop to connect to and query the Tableau Server repository using two special, built-in users. The "tableau" user has access to several database views you can use as part of building your own analyses of Tableau Server activity. The "readonly" user has access to additional database tables that you can use to create views for even more in-depth analysis.

To access the Tableau Server repository, you need to use the tabadmin command line utility to enable external access to the database.

1. Open a command prompt as an administrator and type:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. **Version 8.2.4 and earlier:** Enter the following command to enable external access to the database for the tableau user:

- `tabadmin dbpass [password]`

For example, to enable access for the "tableau" user with a password of "p@ssword":

```
tabadmin dbpass p@ssword
```

Version 8.2.5 and later: Enter the following command to enable external access to the database for the tableau user or the readonly user:

- `tabadmin dbpass --username [tableau | readonly] [password]`

For example, to enable access for the "tableau" user with a password of "p@ssword":

```
tabadmin dbpass --username tableau p@ssword
```

or to enable access for the "readonly" user with a password of "p@ssword":

```
tabadmin dbpass --username readonly p@ssword
```

Note: If no user is specified, dbpass enables access for the "tableau" user.

3. Restart Tableau Server:

```
tabadmin restart
```

After you've enabled external access to the database, Tableau allows any IP address access to the database as long as the correct password is provided. Follow the steps in [Connecting to the Tableau Server Database](#) below to connect.

[Disable external access to the Tableau Server Database](#)

If you want to disable access by "tableau" or "readonly" after enabling it, use the tabadmin dbpass again.

- Run the command `tabadmin dbpass --disable --username [user]` then restart the server.

For example:

```
tabadmin dbpass --disable --username readonly  
tabadmin restart
```

Note: If no user is specified, the --disable option disables access for the "tableau" user.

[Connecting to the Tableau Server Database](#)

After you [enable external access](#) to the Tableau Server database, follow the steps below to connect to and query the database. The username you use will depend on which database views and tables you want to use.

- In Tableau Desktop select **Data > Connect to Data**, then select **PostgreSQL** as the database to connect to. You may need to install the PostgreSQL database drivers. You can download drivers from www.tableau.com/drivers.
- In the PostgreSQL connection dialog box, enter the name or URL for Tableau Server in the **Server** box. If you have a distributed server installation and a worker is hosting the repository, enter the name of the worker instead. If you are using an Apache load balancer, enter the actual name or ip address of the database server rather than the Tableau Server name.

You should connect using the port you have set up for the pgsql.port, which is 8060 by default. For more information about ports, see [Tableau Server Ports](#) on page 540.

Note: The dbpass command does not open any ports in the firewall. You may need to manually open the port in any firewall between your external client and the Tableau Server database.

3. Enter `workgroup` as the **Database** to connect to.
4. Connect using one of the following users and the password you specified:

Username: tableau or readonly

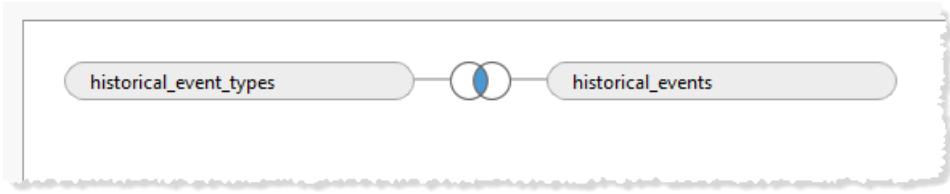
Password: The password you specified when you [enabled access to the Tableau Server](#) database for the specified user

5. Click **Connect**.

The screenshot shows a PostgreSQL connection dialog box. At the top, it says "PostgreSQL". Below that, there are fields for "Server" (set to "tableauserver.myco.com"), "Port" (set to "8060"), and "Database" (set to "workgroup"). Underneath these, a section titled "Enter information to sign in to the database:" contains fields for "Username" (set to "tableau") and "Password" (represented by a series of dots). At the bottom right of the dialog is a large orange "Connect" button.

6. Select one or more tables to connect to.

The "tableau" user has access to all of the tables that start with an underscore or with **hist_**. For example, you can connect to **_background_tasks** and **_datasources**. The tables that begin with **historical_** point to **hist_** tables. The **hist_** tables include information about server users that isn't currently presented in the [Actions by Specific User on page 293](#) view. The "readonly" user has access to additional tables that can be used to query other information about server usage.



7. Click **Go to Worksheet**.

Interact with Views

When you find content that interests you, you can view and interact with the data in many different ways, depending on the content you are allowed access.

With the view open, you can choose from options such as sharing, creating custom views, exporting, downloading, subscribing, and editing the view. You can also interact with the view to explore its data. If you have web editing permissions, you can edit existing workbooks or create new workbooks from data sources.

Content Actions

When you select a project, workbook, view, or data source, the **Actions** menu will offer different commands that apply to that content. Your site role and permissions for content will determine the actions that are available to you (such as setting permissions, adding tags, or deleting views and workbooks).

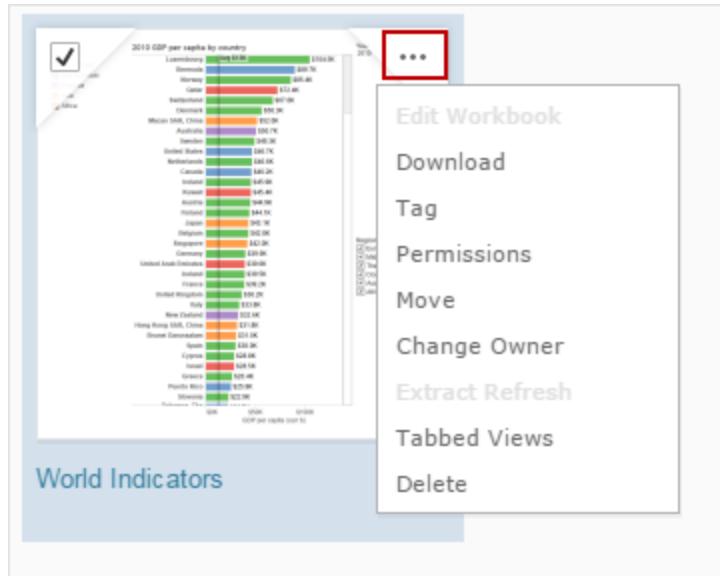
On the Projects, Workbooks, Views, and Data Sources pages, select content to perform various actions, such as to tag content or to assign permissions. Select the **Actions** menu to access commands available for the selected content.

The screenshot shows the Tableau interface with a navigation bar at the top: Content, Users, Groups, Schedules, Tasks, Status, Settings. Below the navigation bar, there are four main categories: Projects (9), Workbooks (10), Views (52), and Data Sources (2). The 'Views' category is currently selected, indicated by an orange border. In the center, a view titled 'Sales' is displayed, showing a bar chart of sales data. To the left of the view, a list of workbooks is shown, with one item highlighted: 'District of Columbia' (selected). A red box highlights the 'Actions' menu, which is open and lists the following options: Edit Workbook, Download, Tag, Permissions, Move, Change Owner, Extract Refresh, Tabbed Views, and Delete. The 'Edit Workbook' option has a checked checkbox next to it. The 'Sales' view itself shows a bar chart with various sales metrics like 'Sales', '% of quota achieved', and 'Avg'. To the right of the 'Sales' view, there are other views: 'Finance' and 'Science'.

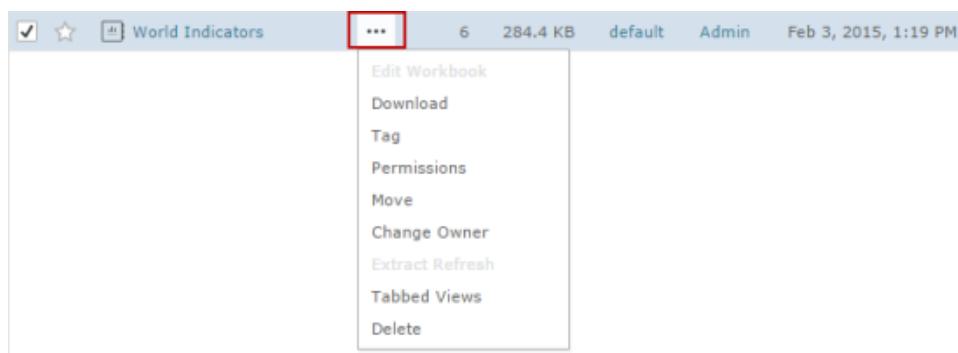
Notes on selecting multiples items: When you select multiple items, the **Actions** menu may display commands that don't apply to one or more items in the set of selected items. When you select multiple items and apply an Actions menu command to those

items, Tableau Server will attempt to perform the action on all of the items and notify you of the results (whether the action succeeds or not) for the various items.

If many items are available on a page, click the ellipses menu to select the item to see the Action commands that are available.



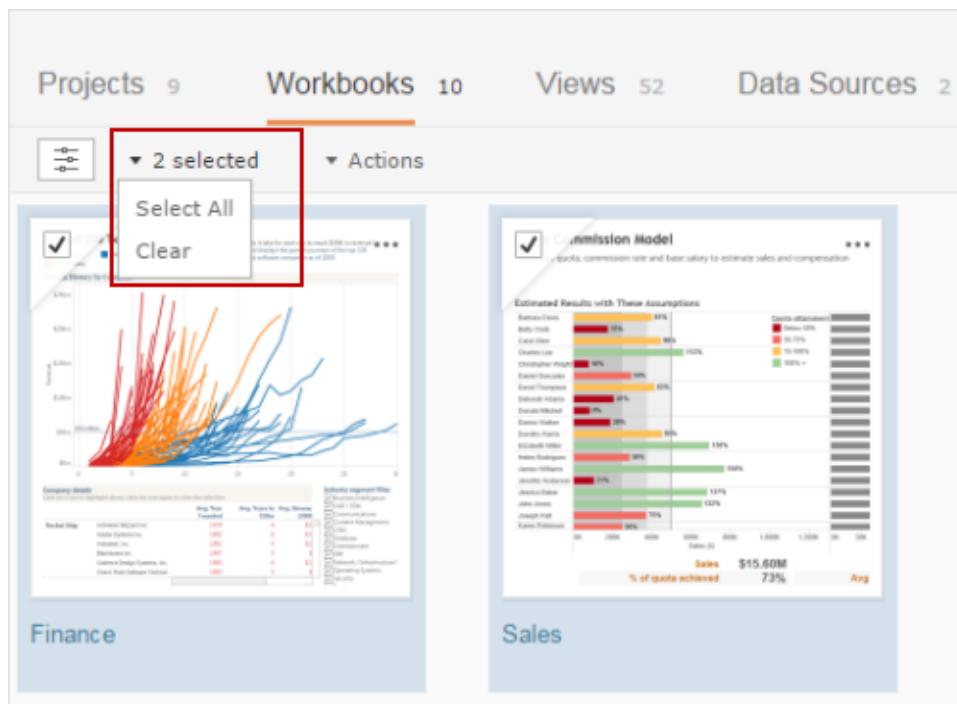
Thumbnail view



List view

Select all or clear the selection

To select all content on the page, click the drop-down arrow, and then click **Select All**. To clear a content selection, click the drop-down arrow, and then click **Clear**.



Comment on Views

You can add comments to any view you have access to on Tableau Server. You can also see any comments associated with a particular view.

Type your text in the **Comment** text box located below the view and click **Post Comment**.

The screenshot shows the comment section for a view:

- Comments (0)**
- A text input box containing the text "Spectacular viz!".
- A "Post comment" button at the bottom left.
- A "Show formatting hints" link at the bottom right.

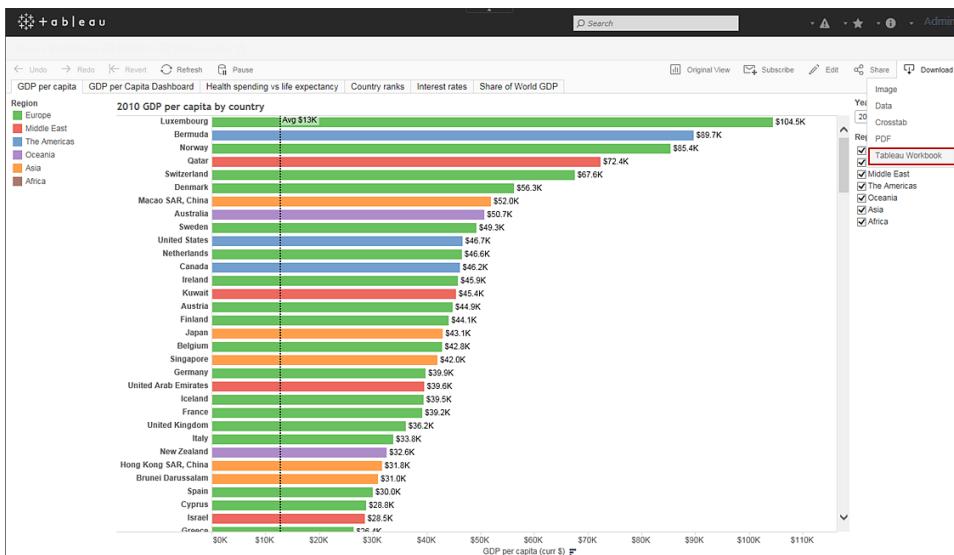
You can add formatting to your comment by inserting hyperlinks, bolding, italics, and underlining. Examples of how to add each of these types of formatting are shown in the table below.

Format	What to Type	Example
Hyperlink	"My Link": http://www.tableau.com	My Link

Format	What to Type	Example
Bold	*Bold Text*	Bold Text
Italics	_Italic Text_	<i>Italic Text</i>
Underline	+Underlined Text+	<u>Underlined Text</u>

Download Workbooks

Workbooks can be downloaded using the **Download** button in the upper-right corner of the view. The downloaded workbook can be opened with a version of Tableau Desktop. Downloading the workbook from the server is the same as selecting **Server > Open Workbook** in the desktop application.



This option is only available if you've been given the **Download/Save** permission by the author of the workbook or an administrator.

Refresh Data

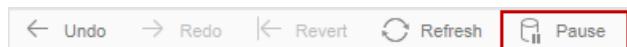
If the data source is changed, such as new fields have been added or data values and field names have been modified, the view will reflect those changes the next time you load the page. However, you may need to manually update the view using the **Refresh Data** button on the toolbar.



When you refresh the data, you clear any cache that may exist and retrieve the latest data from the data source. This option is different than the **Pause Automatic Updates** below option, which still may load the view based on cached data. Depending on the size of your data source and the view, refreshing the data may take longer than other queries that operate on cached data.

Pause Automatic Updates

As you interact with the view on the server, it will sometimes have to send a query to the data source to update the data in the view. If you are working with a dense view with a lot of data or a very large data source, the automatic update may take a long time. To avoid waiting for each update while you make several changes you can click **Pause** on the toolbar to pause automatic updates.



When you **Resume** automatic updates using the same toolbar button you only have to wait for a single query to the data source.

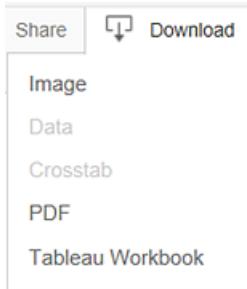


Download Views

You can download a view in the following formats:

- An image.
- A PDF.
- A crosstab. This option opens a file in Microsoft Excel.
- A data file. This option opens a new tab in the browser window and displays the data in summary and in detail. You can also download the data from this view as a comma-separated value (.csv) file.
- A workbook. For information about how to download a workbook, see [Download Workbooks](#).

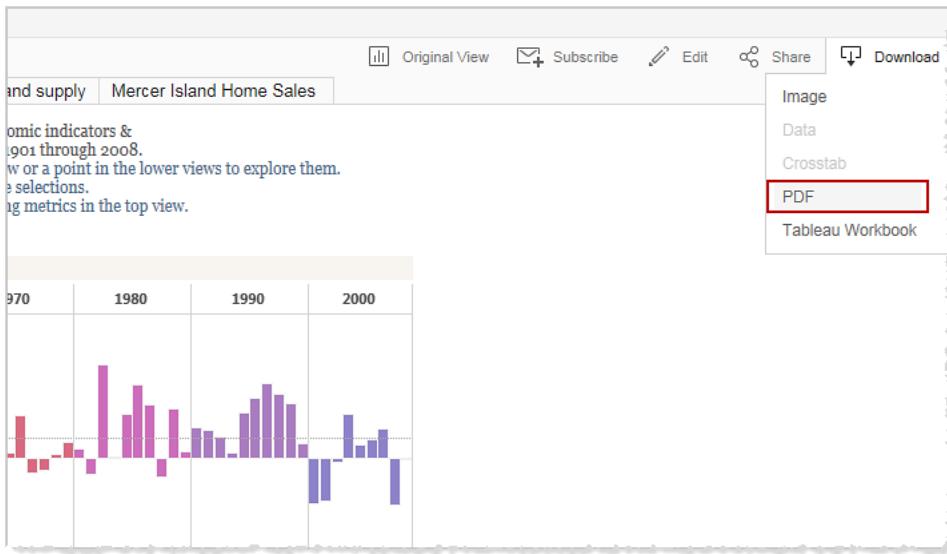
On the toolbar at the top of the view, select an option on the **Download** menu.



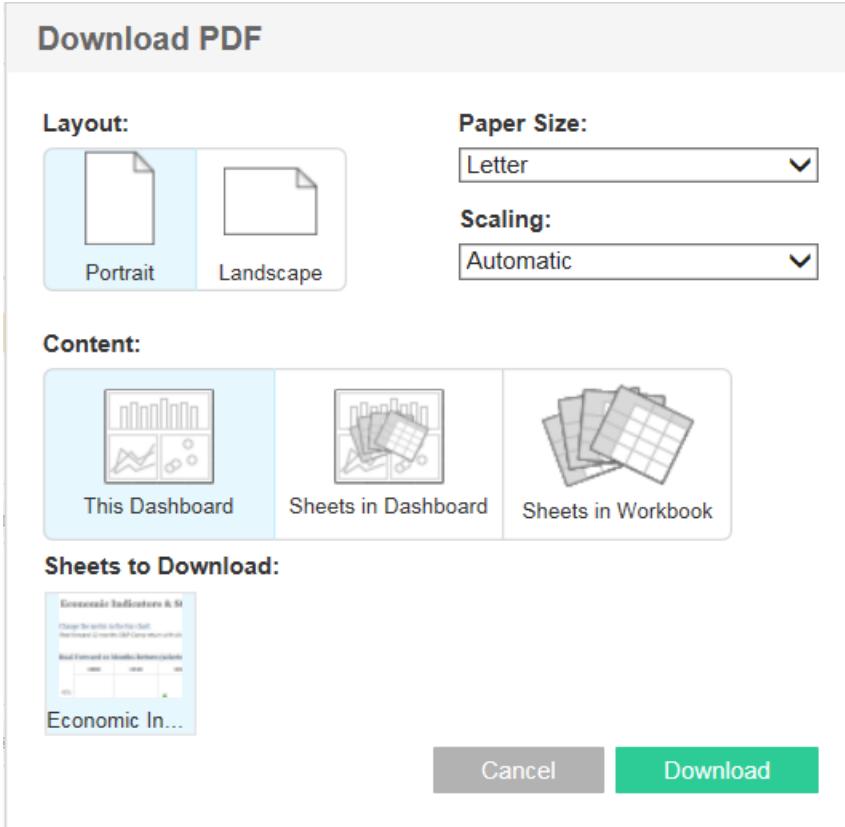
If you are downloading a dashboard to a PDF and the **dashboard includes a web page object**, the web page object is not included. Also, when you select a download option, the image, PDF, or crosstab must be generated. A dialog box opens when it is done generating so you can continue downloading the file.

To download a view as a PDF

1. Open a view, click **Download** on the toolbar, and select **PDF** from the context menu.

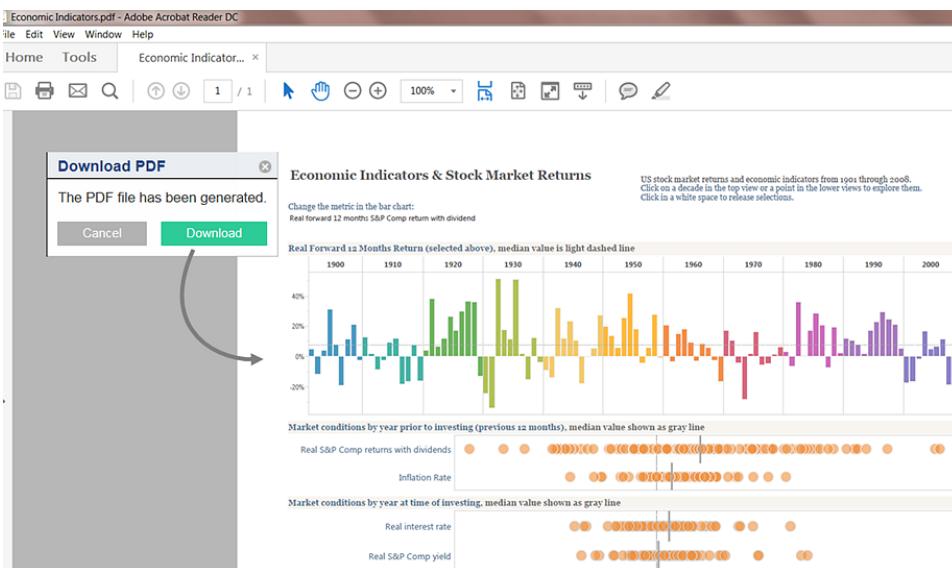


2. Select either a **Portrait** or **Landscape** orientation and a **Paper Size**. Under **Content**, select the part of the workbook to download (current dashboard, selected sheets in the dashboard, or selected sheets in the workbook). Then, under **Sheets to Download**, select the specific sheets to download. Click a sheet thumbnail to select or deselect the sheet or press **Cntrl** and click to select multiple sheets.



Selected sheets are indicated with a blue highlight.

- Click **Download**, then, in the **Download PDF** dialog box, click **Download** again.



Save Passwords

Sometimes a view requires you to enter a database user name and password. If you have access to the database you should enter your user name and password into the appropriate text boxes. If you select the **Remember my password** option you will be automatically signed in each time you look at the view. Your sign in information is stored encrypted on the server so you will be automatically signed in even between browser sessions and when accessing the view from multiple computers. This is convenient when you have a select number of views that you access all the time.

Administrators can restrict whether to allow users to remember database passwords. If you are an administrator, see [Server Settings \(General\)](#) on page 257 to learn more.

To open this view, you must log in to the database that the view uses.

Connection Type: Microsoft SQL Server
Authentication: Database Built-in Security
Server: mssql2012
Database: Election

Log In

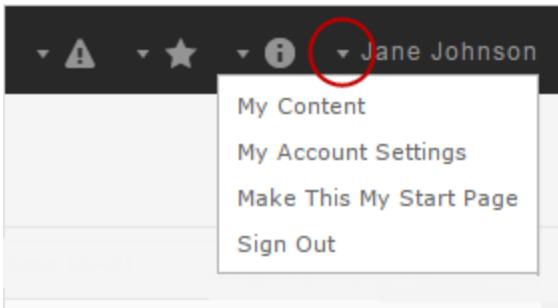
Username: **Log In**

Password: Remember my password until I log out

Clearing and Resetting Saved Passwords

If your passwords are being saved (**Allow users to save data source passwords** is enabled on the General page in **Server > Settings**, you can clear your saved passwords. When you do this, the next time you visit the server, you are prompted to enter your user name and password. You may want to do this if your user name and password change so you can begin using and saving your new credentials.

1. Click your name at the top of the page, and then select **My Account Settings**.

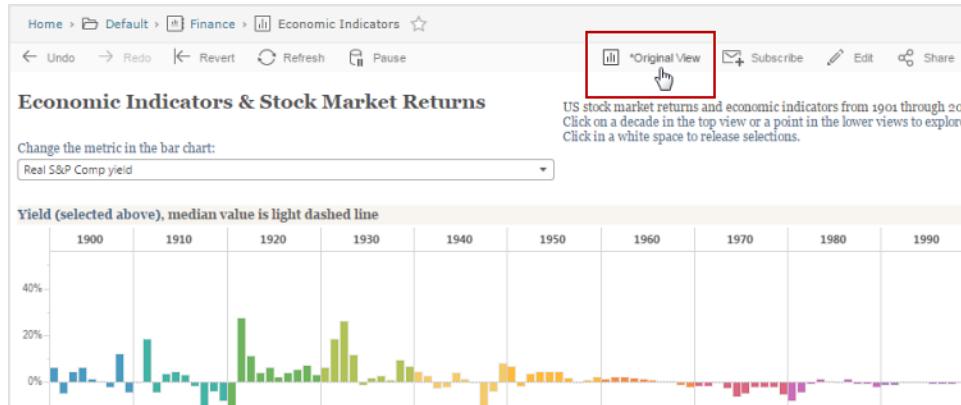


2. In **Manage Credentials**, click **Clear All Saved Credentials**.

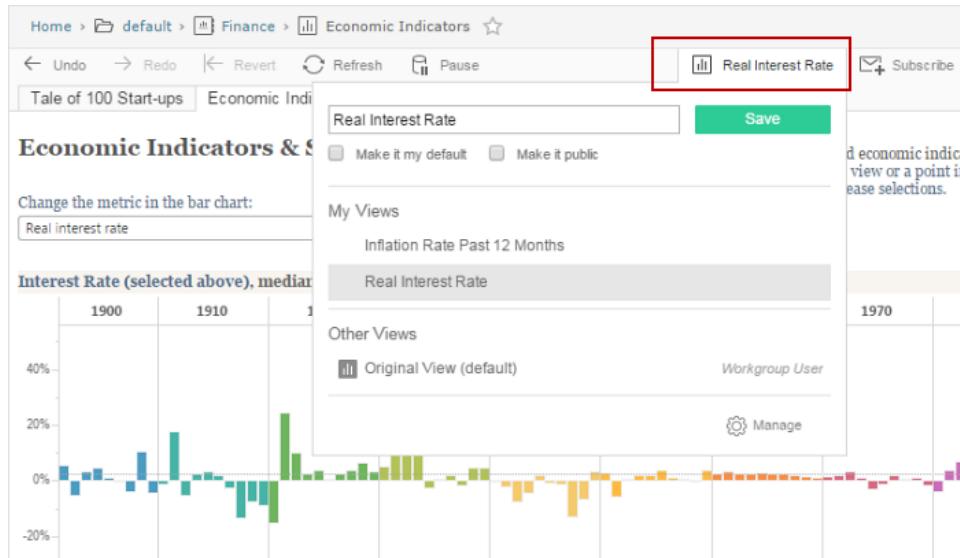
Note: Administrators can also clear all saved passwords on the server using the **Clear All Saved Credentials** for all users link on the Server Settings - General page.

Save Custom Views

When you interact with a view, you have the option to save any changes you've made to the view (sorting, filtering) as a custom view.



If you make a change to the original view, the **Original View** menu in the toolbar indicates the view has changed with an asterisk. You can access custom views by clicking the **Original View** menu, or by clicking the name of the custom view in the same menu location.



Custom views are always associated with the original view. As the original view is updated or republished, customized versions of the view are also updated.

- If the original view is deleted from the server, its associated custom views are also deleted.
- If filters are removed from the original view and it's republished, the filters will be unavailable in customized versions of the view.
- If filters are restored and the view is republished, customized versions of the view include the restored filters.

To save a custom view

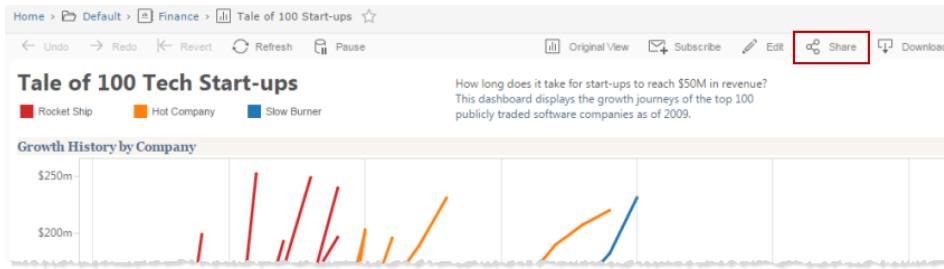
1. Open the individual view that you want to customize.
2. Filter the data, change sort orders, highlight, zoom in or out, or make any other modifications.
3. Click **Original View** or the name of the currently selected custom view as it is shown in the toolbar. Enter a name for the custom view. Select whether you want it to be the default view, or if it should be public, and then click **Save**.

Quick Start: Share Server Views

Easily share your Tableau Server views with others. Click **Share** in a workbook or view to create links that you can embed in a blog or webpage or email to a friend or co-worker.

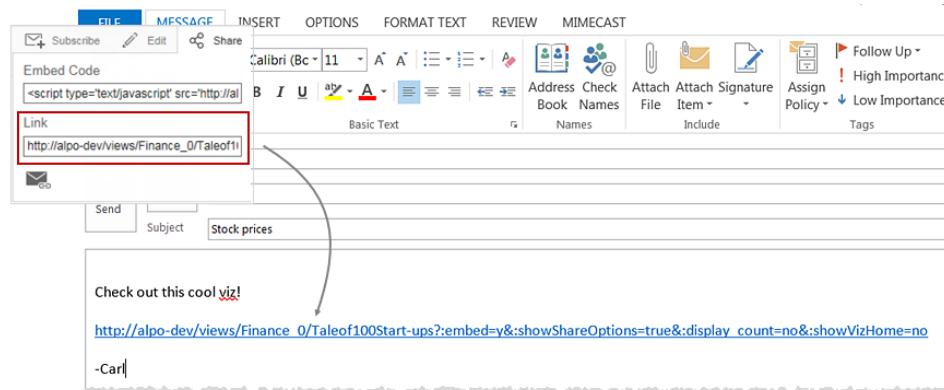
1 Open the share options

Click the **Share** button in the upper right corner of the view.



2 Email the view

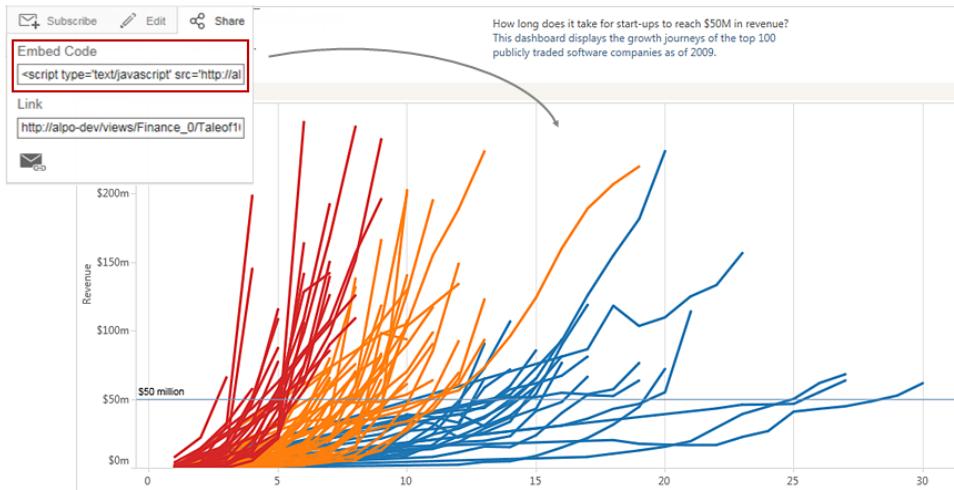
Copy the URL in the **Link** field and paste it into an email or click the envelope icon in the lower left corner of the dialog box to automatically open a blank email that includes the link to the view in the email body.



Make sure that the people you are sharing with have access to the server and the right permissions to see the view.

3 Embed the view

Copy the HTML code in the **Embed Code** field and paste it into your blog or webpage. The interactive view will display inline on the page.

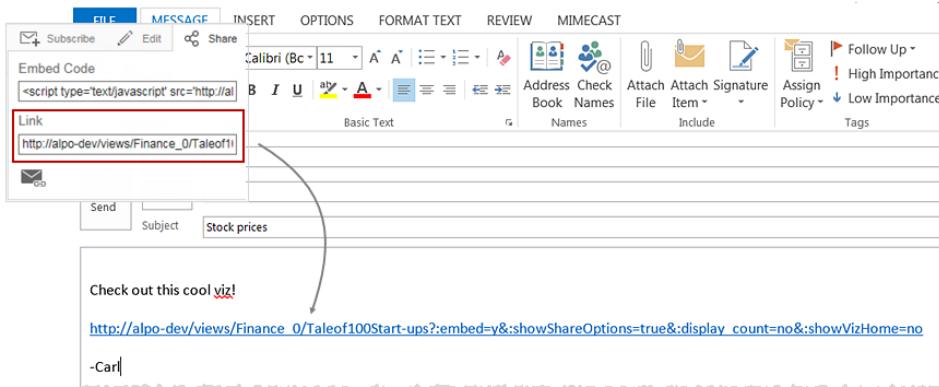


Share Views

Every published view and workbook can be shared via email or embedded into another webpage, wiki, or web application. Anyone viewing a shared view must have an account on Tableau Server and permission to access the view.

Email a view

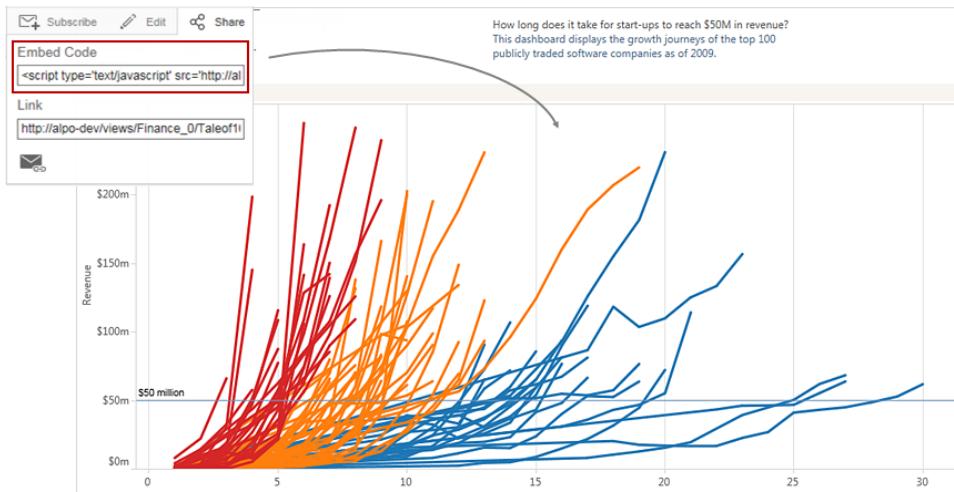
1. Click **Share** in the upper-right corner of the view.
2. Copy and paste the provided link into your email message or click the envelope icon in the lower left corner of the dialog box to open a blank email that includes the link to a view in the email body.



Embed a view

You can share a view by embedding it into another webpage such as your wiki, blog, or web application.

1. Click **Share** in the upper right corner of the view.
2. Copy the provided embed code, and then paste it into the source code of the page in which you want to embed the view.



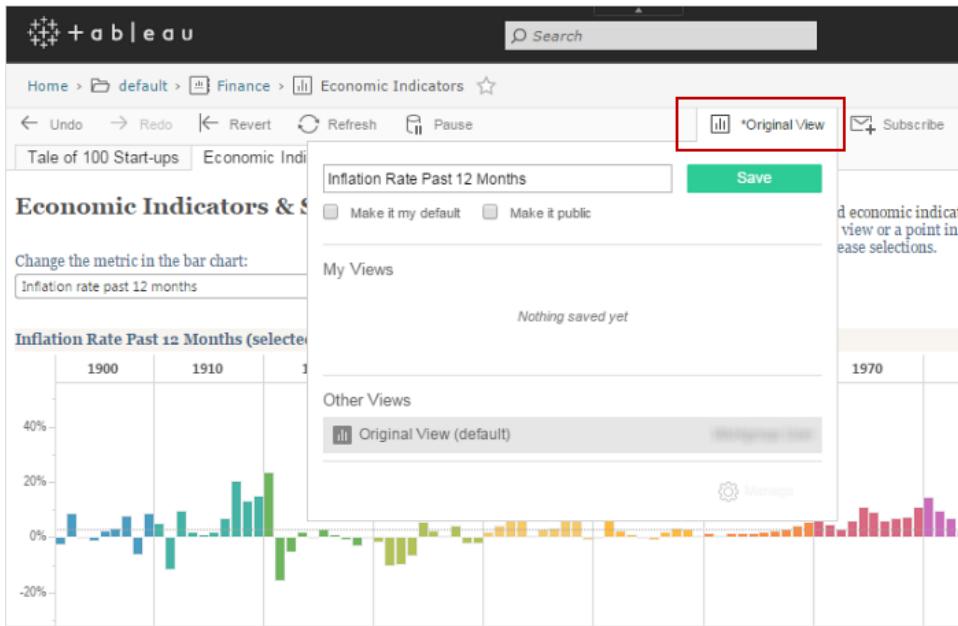
Note: The embed code generated by Tableau will automatically refer to the current view. For information about how embedded custom views are displayed in Tableau, see [Embed Code for Custom Views](#).

Quick Start: Custom Views

If you regularly look at certain views and find you need to make the same changes each time you open the views, you can create custom views that "remember" your changes. Each time you open a custom view, it displays the information you want to see. You can then share custom views with other Tableau Server users who have permission to see the original view.

1 Create a custom view

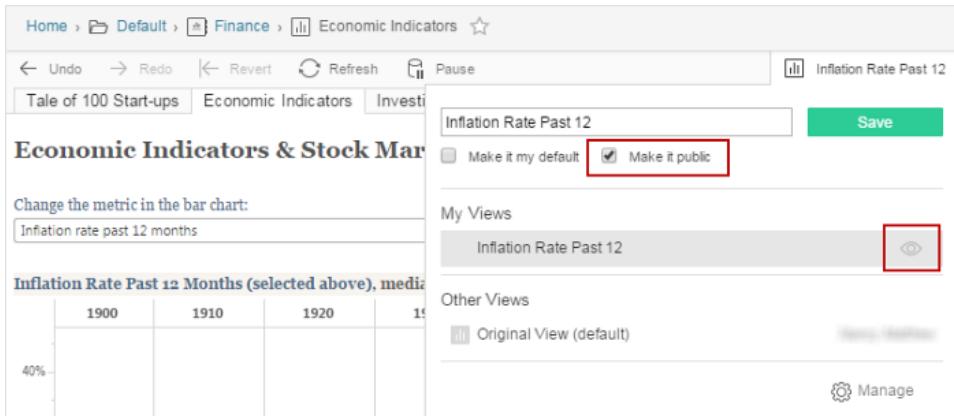
Open a view and make the changes you want to save. For example, modify filters, sort, or set zoom level. Click ***Original View** on the toolbar and enter a name for your custom view. Select **Make it my default** to make the new view your default view. When you are done, click **Save**.



The currently selected view is indicated by a gray background.

3 Share custom views

To make a custom view available to others, select the **Make it public** option when you create the view.

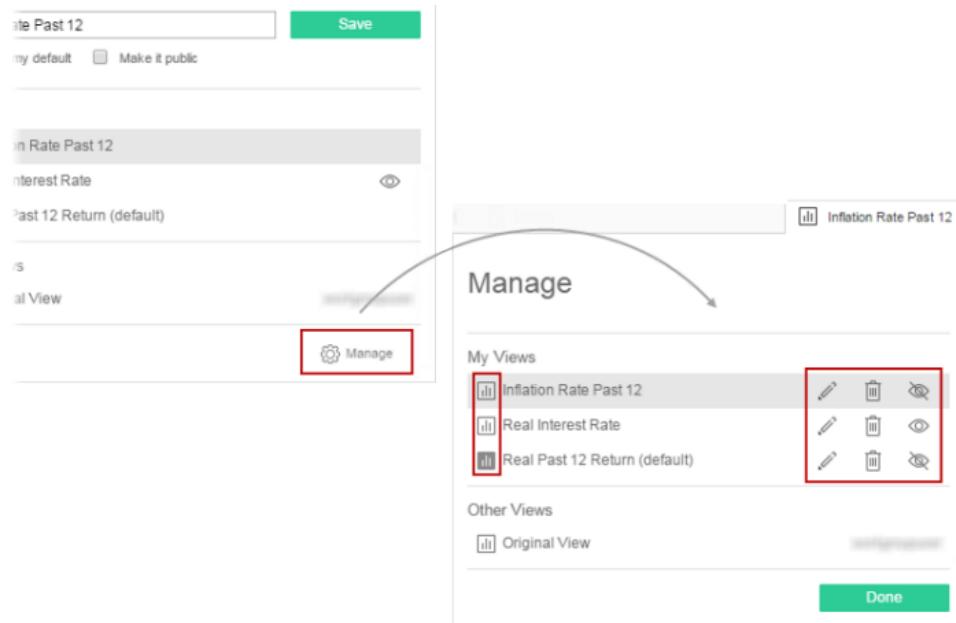


The public view icon (👁) indicates that the view is shared. The private view icon (👁) in the Manage dialog box indicates that the view can only be seen by you.

As you browse views on the server, you can see and use custom views that other people have shared.

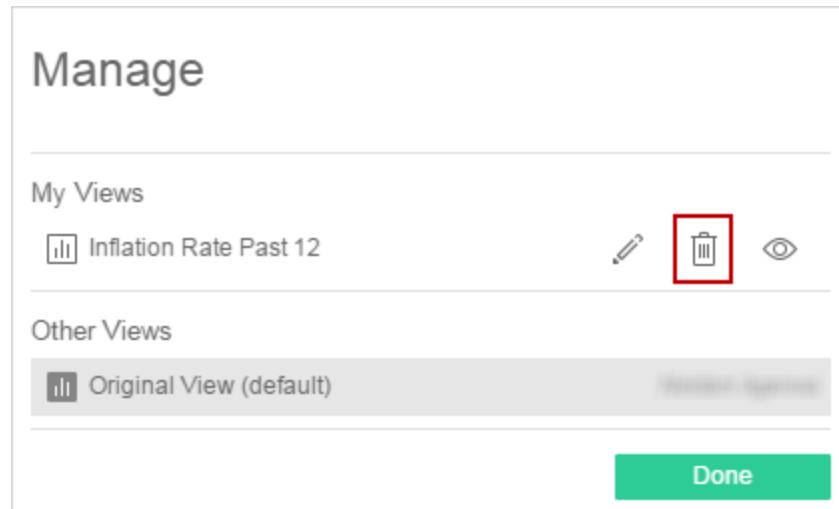
2 Manage custom views

To change the default view, edit a custom view name, delete a custom view, or make it private, click the view name, and then click **Manage**.



The default view is indicated by . To change the default view, click next to the custom view name in the Manage dialog box.

4 Delete custom views



Click the delete icon () in the Manage dialog box to delete the view.

Custom Views

If you notice you are making the same changes to a view every time you open it, you might want to consider saving the changes as a custom view.

For example, in a particular view, you might apply a filter to include only data relevant to you, or you might sort a view differently than how it was published. You might also want to keep different versions of the same view—for example, one with two filters selected, and another with only one selected. Each of these different configurations can be saved as a custom view.

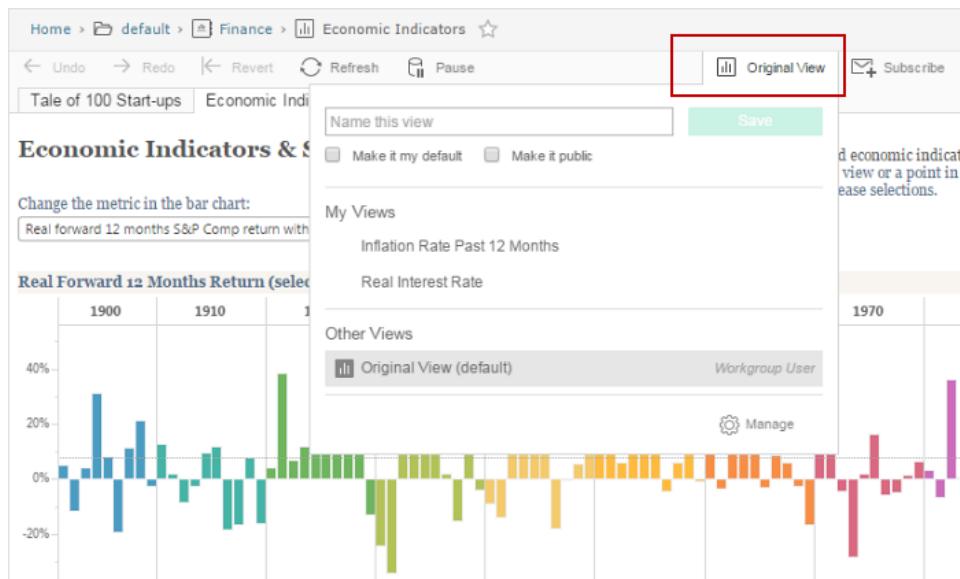
A custom view is always displayed with the latest saved changes. A custom view does not change the original, but is related to it. If the original view is updated or republished, the custom view is also updated. If the original view is deleted from the server, the custom view is also deleted.

You can also choose whether your custom views are visible to other users (public), or only to you (private).

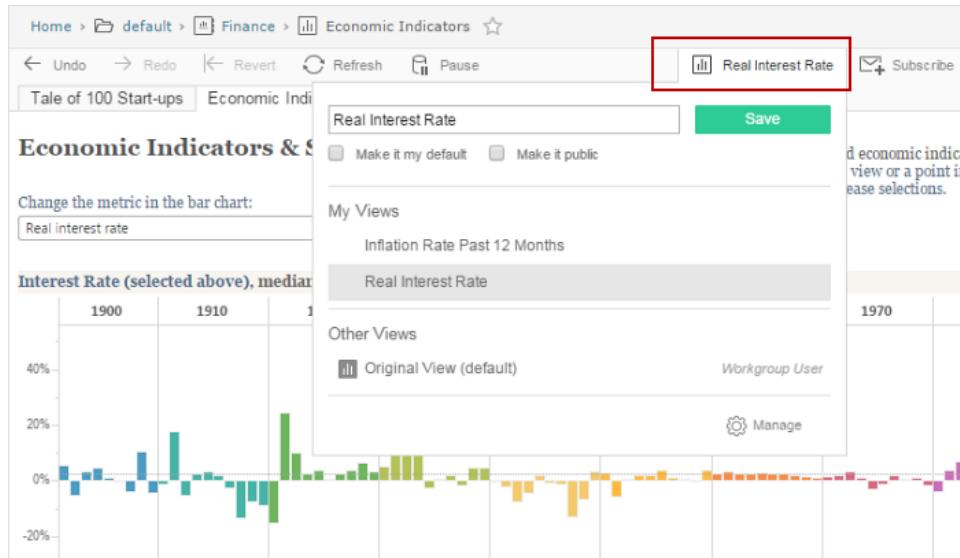
See the following topics for more information:

Access Custom Views

To see the custom views available for a view, click the **Original View** menu, or click the name of the custom view in the same menu location.



This example shows the original view.



This example shows a custom view name in the same location as **Original View**.

The custom views that you have saved are displayed under **My Views**. Views created by other people, including the **Original View**, are listed under **Other Views**.

Select a view

- Click a view name in the list of views to display it. The currently selected view is indicated by a gray background.

Change the default view

The default view is indicated by . To change the default view, click next to the custom view name in the Manage dialog box.

- In a view, click **Original View** or click the name of the custom currently selected view as it is shown in the toolbar.
- Click **Manage**.

The screenshot shows a user interface for managing views. At the top, there is a text input field containing "Inflation Rate Past 12" and a green "Save" button. Below this are two checkboxes: "Make it my default" and "Make it public".

The main area is titled "My Views" and contains a list of views:

- "Inflation Rate Past 12" (highlighted with a gray background)
- "Real Interest Rate" (with an eye icon to its right)
- "Real Past 12 Return (default)" (preceded by a bar chart icon)

Below this is a section titled "Other Views" which lists "Original View".

At the bottom right, there is a red-bordered button labeled "Manage" with a gear icon.

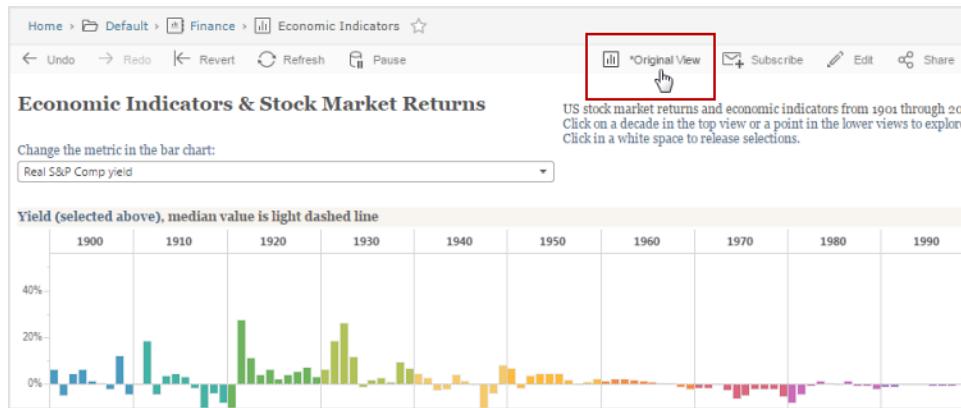
3. Click next to the custom view name to make the view the default view. The icon will change to to indicate the view is now the default view.

The screenshot shows a 'Manage' interface with a header 'Inflation Rate Past 12'. Below it, there's a 'My Views' section containing three items: 'Inflation Rate Past 12', 'Real Interest Rate', and 'Real Past 12 Return (default)'. Each item has edit, delete, and eye icons. Below this is an 'Other Views' section with a single item 'Original View' highlighted with a red box and a hand cursor icon. At the bottom of this section are two buttons: 'Set this view as your default' and a large green 'Done' button.

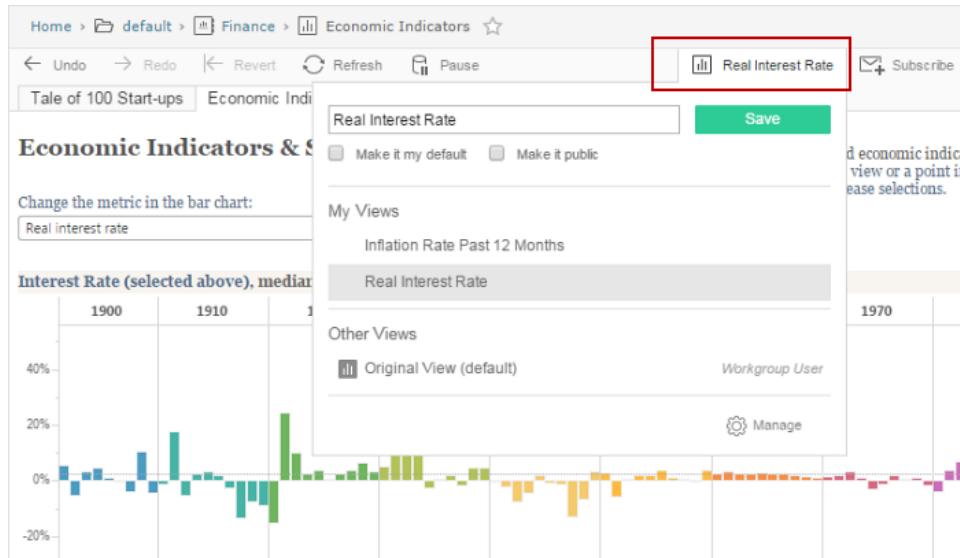
4. Click **Done**.

Save Custom Views

When you interact with a view, you have the option to save any changes you've made to the view (sorting, filtering) as a custom view.



If you make a change to the original view, the **Original View** menu in the toolbar indicates the view has changed with an asterisk. You can access custom views by clicking the **Original View** menu, or by clicking the name of the custom view in the same menu location.



Custom views are always associated with the original view. As the original view is updated or republished, customized versions of the view are also updated.

- If the original view is deleted from the server, its associated custom views are also deleted.
- If filters are removed from the original view and it's republished, the filters will be unavailable in customized versions of the view.
- If filters are restored and the view is republished, customized versions of the view include the restored filters.

To save a custom view

1. Open the individual view that you want to customize.
2. Filter the data, change sort orders, highlight, zoom in or out, or make any other modifications.
3. Click **Original View** or the name of the currently selected custom view as it is shown in the toolbar. Enter a name for the custom view. Select whether you want it to be the default view, or if it should be public, and then click **Save**.

Advertise Custom Views

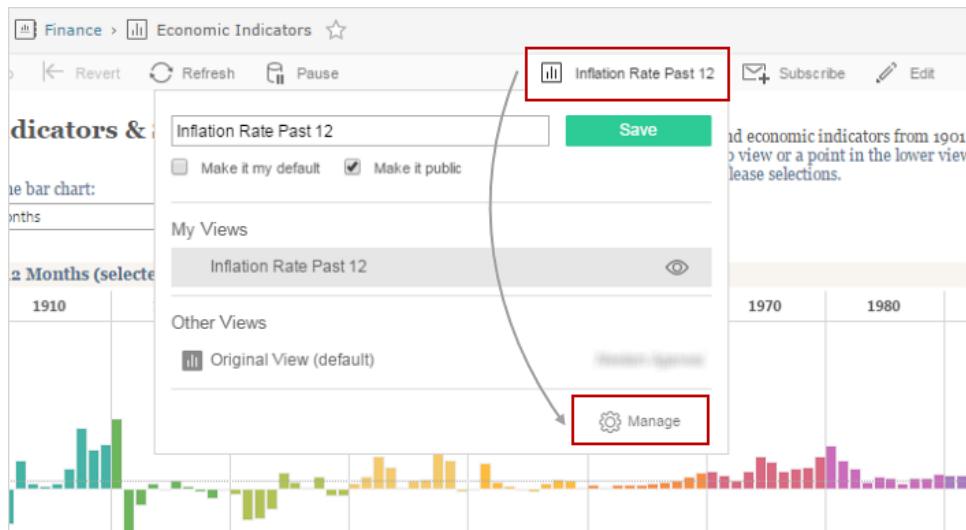
By default, your custom view is private so only you can see it on your list. If your site role is Interactor or Publisher, you can advertise the view to other users. Anyone who has access to the original published view will be able to see your advertised custom view.

To advertise a custom view, click **Make it public** when you first create the view.

Note: Even if you don't advertise your custom view, you can still share it by copying the URL or clicking **Share**.

To change the public or private status of a custom view

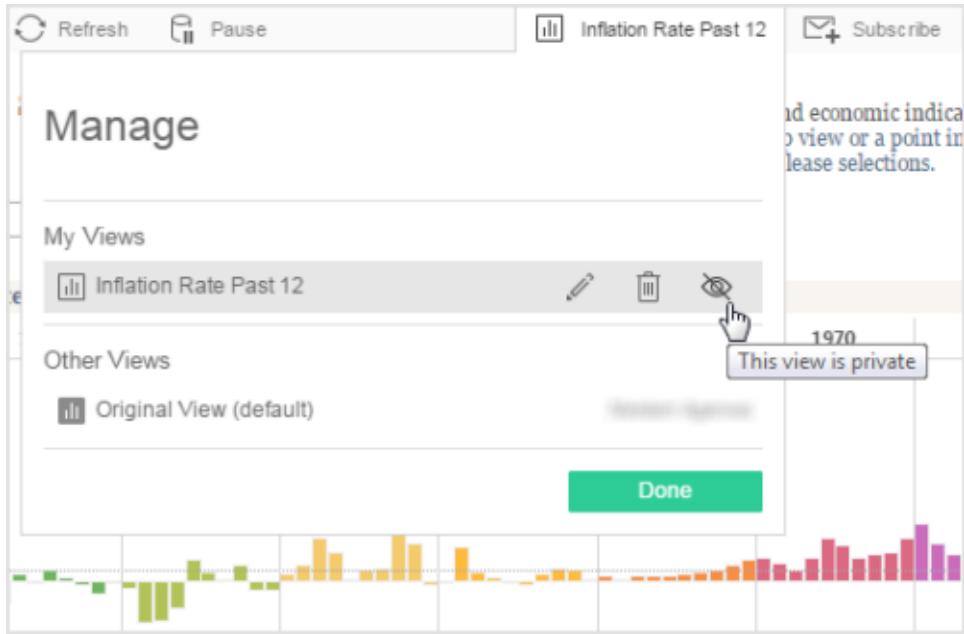
1. Click **Original View** or the name of the current view in the toolbar. Select the view you want to edit, and then click **Manage**.



2. In the **Manage** dialog box, click the public or private view icon to change the view status.



The public view icon () indicates that the view is shared.



The private view icon () indicates that the view can only be seen by you.

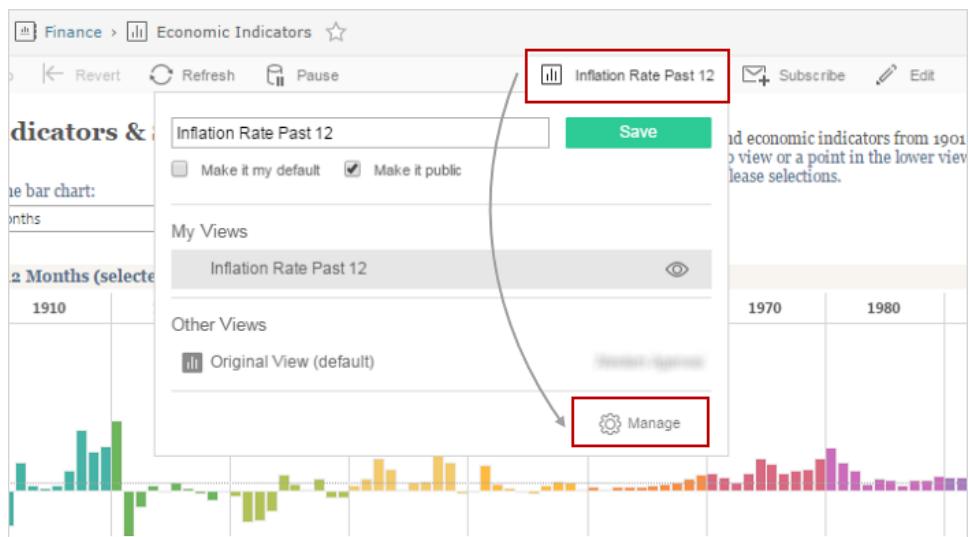
3. Click **Done**.

Make Views Private

You can always make a public view private. When a custom view is private, it no longer shows in the drop-down list for others and only you can see it on your list of custom views.

To change a public custom view to be private

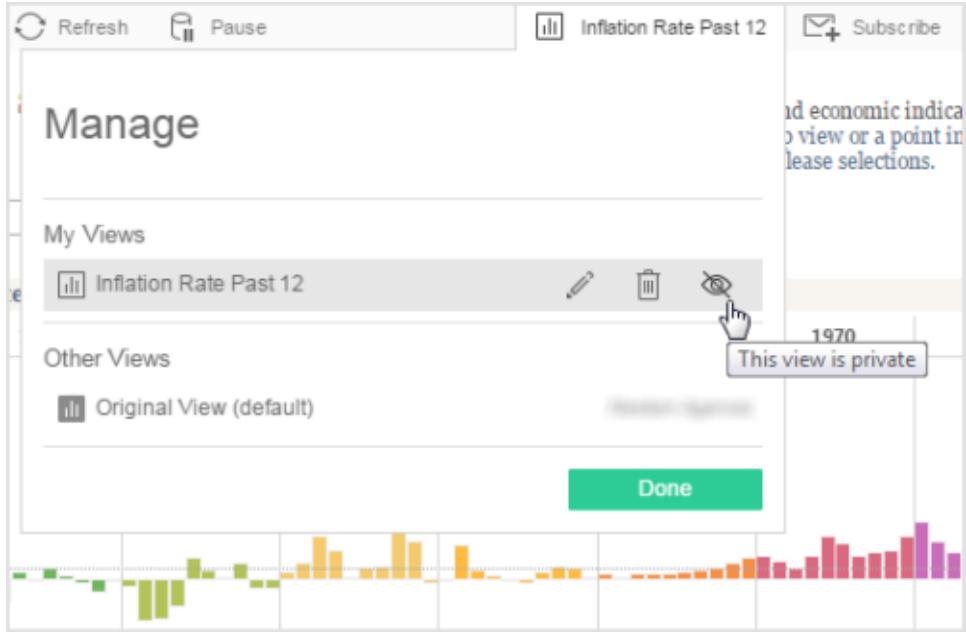
1. Click **Original View** or the name of the current view in the toolbar. Select the view you want to edit, and then click **Manage**.



2. In the Manage dialog box, click the public view icon () next to the view to make it private ().



The public view icon () indicates that the view is shared.



The private view icon (🔗) indicates that the view can only be seen by you.

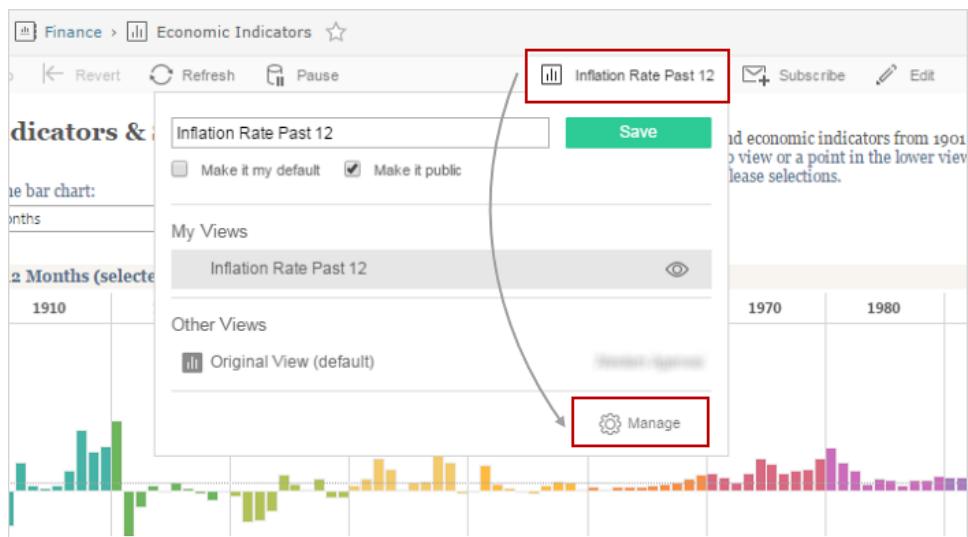
3. Click **Done**.

Delete Custom Views

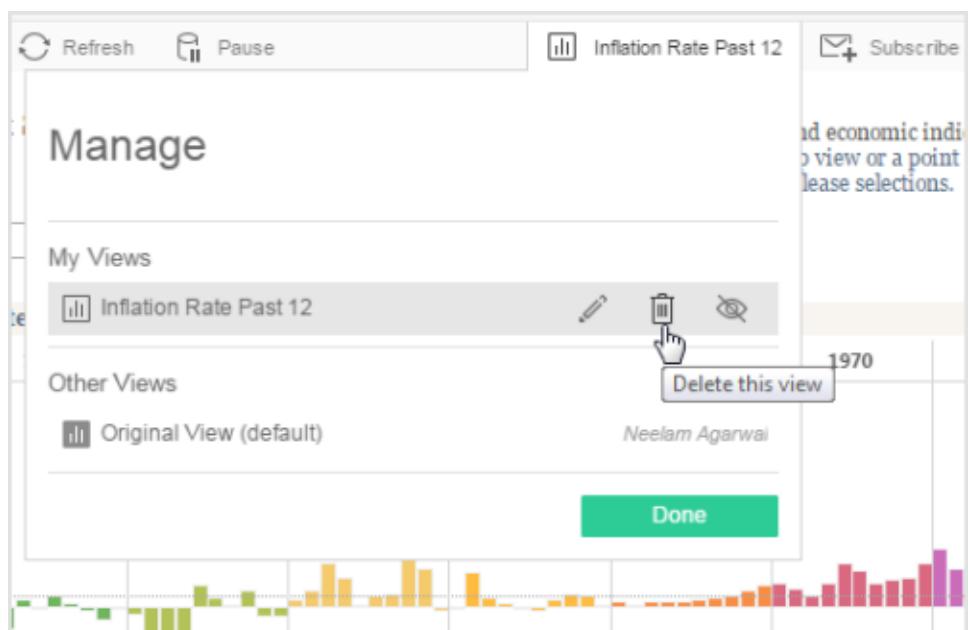
You can delete a custom view you created at any time. Removing your custom view does not affect the original view.

Note: If the original view is deleted from the server, your custom view is also deleted.

1. When you're looking at the custom view you want to delete, click the view name in the toolbar, and then click **Manage**.



- Click the delete icon () next to the custom view name.



- Click **Done**.

Control Access to Published Content

Administrators can control access to Tableau Server content by assigning permissions to projects, workbooks, views, and data sources. They also can specify and change owners for projects, workbooks, and data sources.

Content owners have control over the permissions for the content that they publish to the server.

Manage Ownership

When you publish a data source or workbook on Tableau Server or when you create a project, you become its owner. Ownership can be changed. For example, if an employee who is the original owner leaves, the administrator can reassign ownership to another user. After you change ownership, the original owner has no special connection to the item, and their ability to access it is determined by their Tableau Server permissions.

Note: You cannot delete a Tableau Server user if the user owns any items. When you attempt to delete the user, their site role is set to Unlicensed. You must first change the ownership of the items and then delete the user. For more information, see [Deleting a User from Tableau Server](#).

If you change the ownership of a workbook or data source that has embedded credentials, the embedded credentials will be deleted. You will need to download the workbook or data source, update the embedded credentials for the new owner, and then re-upload the workbook or data source.

Your ability to change or be given ownership depends on your permissions and your relationship to the item, as described in the following table.

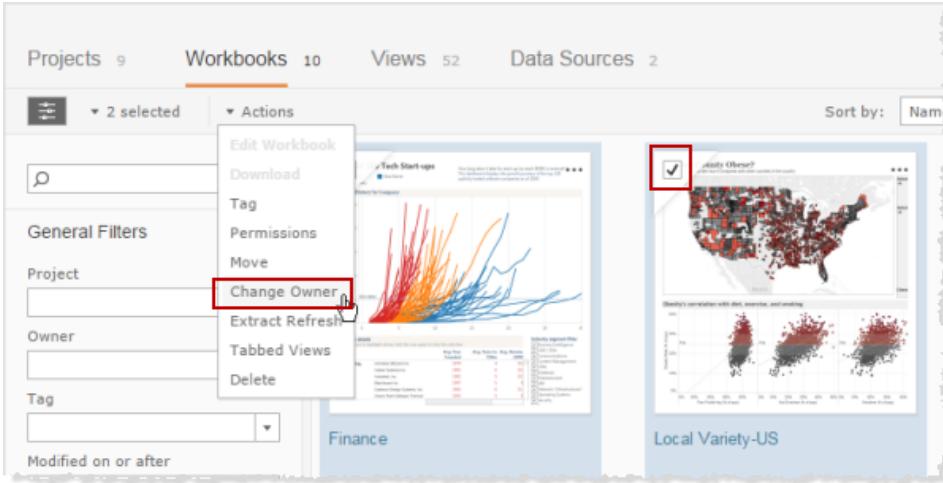
Item type	Who can change ownership	Who can be given ownership
Projects	Server administrator Site administrator	Server administrator Site administrator
Workbooks and Data Sources	Server administrator Site administrator Owner of the item	Server administrator Site administrator Member of the site that contains the item (Guest user excluded).

Change a Workbook Owner

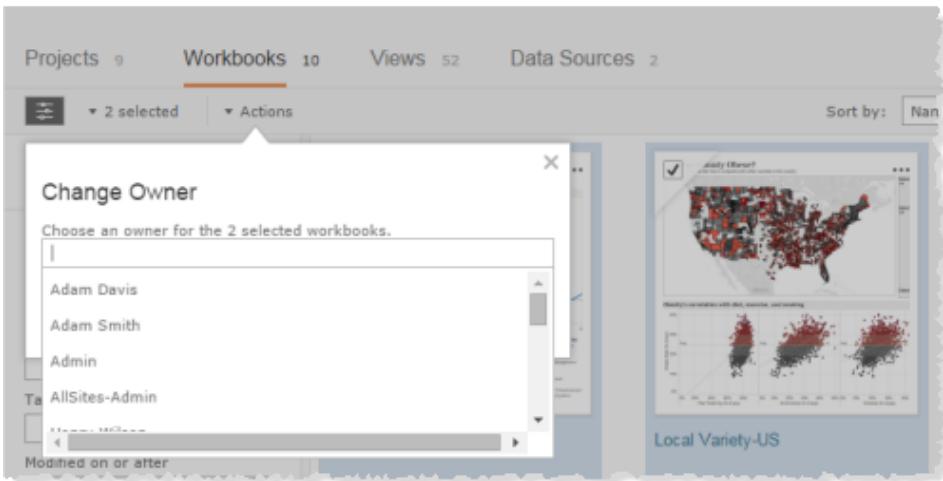
By default, the publisher of a workbook is its owner. Administrators and the current owner of the workbook can change ownership. The new owner must be a server administrator or a site administrator, or be any user other than Guest on the same site as the workbook.

To change the owner for a workbook

1. On the Content page for a site, select **Workbooks**.
2. Select one or more workbooks, and then select **Actions > Change Owner**.



3. Type the name of a user or select a user from the list.



4. Click **Change Owner**.

Change a Data Source Owner

By default, the publisher of a data source is its owner. Administrators and the current data source owner can change ownership. The new owner must be a server or site administrator, or be any user other than Guest on the same site as the data source.

To change the owner for a data source

1. On the Content page for a site, select **Data Sources**.
2. Select one or more data sources, and then select **Actions > Change Owner**.

The screenshot shows the 'Content' page with the 'Data Sources' tab selected. A context menu is open over a selected item named 'Data by country'. The 'Change Owner' option in the menu is highlighted with a red box and a cursor icon.

3. Type the name of a user or select a user from the list.

The screenshot shows the 'Change Owner' dialog box. It displays a list of users: Adam Davis, Adam Smith, Admin, and AllSites-Admin. The 'AllSites-Admin' option is highlighted with a red box and a cursor icon.

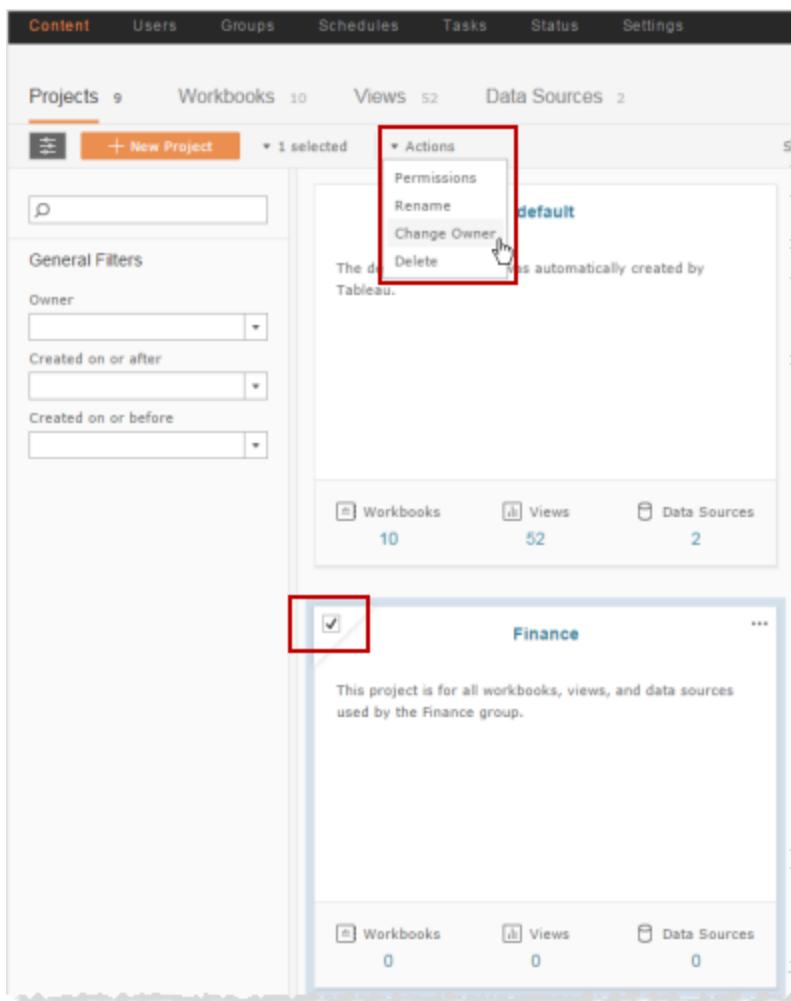
4. Click **Change Owner**.

Change a Project Owner

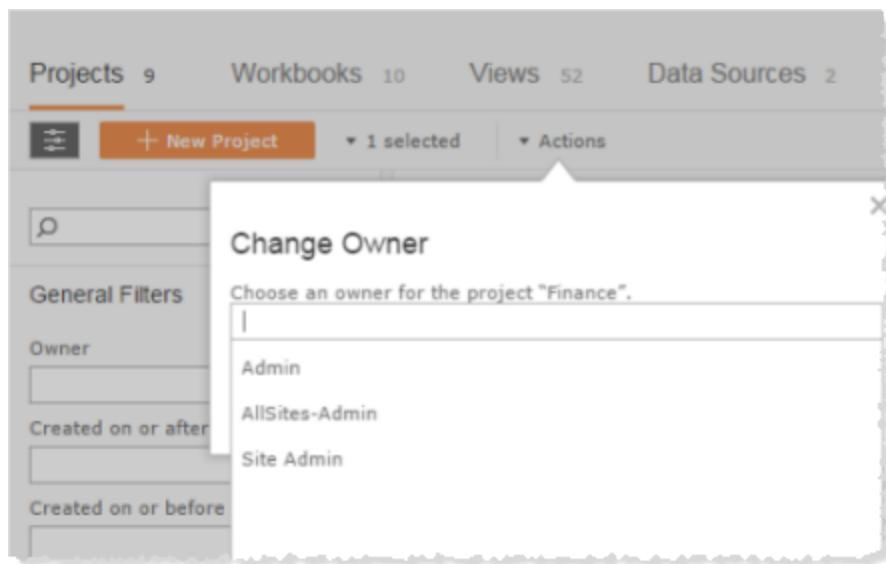
By default, the creator of a project is its owner. Administrators can change project ownership. The new owner must be a server administrator or an administrator for the project's site.

To change the owner for a project

1. On the Content page for a site, select **Projects**.
2. Select one or more projects, and then select **Actions > Change Owner**.



3. Type the name of a user or select a user from the list.



4. Click **Change Owner**.

Manage Permissions

In Tableau Server, you set *content permissions* in order to specify who is allowed to work with what content in a site.

About content permissions

Content permissions ensure that only the right people can see and interact with your content. For example, you can tightly restrict who has access to your company's financial information, but widely share organizational development content.

You assign content permissions to the following items:

- Projects
- Workbooks
- Views
- Data sources

About permission rules, site roles, and user permissions

You assign content permissions by setting *permission rules*. Permissions rules are the explicit capabilities you assign to a user or group for a given content item. A *capability* is a task that you want a user to be able to perform, such as editing a view. Every project, workbook, view, or data source can have a unique set of permission rules.

In addition to content permissions, a user's *site role* and whether the user is a *content owner* also affects what tasks a user can perform and what actions are available to the user for each content type.

User permissions are the effective permissions that determine what a user can actually do with the content. They are the result of how Tableau evaluates each user or group permission rule that applies to a user for a given content item.

For more information, see [Site Roles for Users](#) on page 176 and [How Permissions are Evaluated](#) on page 348.

For more information, see [Permission Rules and User Permissions](#) on page 344.

Permissions		Permissions for views are inherited from the workbook									
User / Group	Permissions	View			Interact			Edit			
		bd	↳	☰	↳	⋮	⋮	✎	⋮	⋮	⋮
All Users (58)	...	None	----	----	----	----	----	----	----	----	----
Finance (13)	...	Custom	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✗	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
General Purpose (...)	...	Viewer	✓ ✓ ✓ ✓ ✓	----	----	----	----	----	----	----	----
Adam Davis	...	Editor	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓

+ Add a user or group rule

User Permissions	Finance (13)			
Adam Davis	Administrator	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
Andrew Allen	Custom	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	----
Andrew Smith	Custom	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	----	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
Ashley Garcia	Administrator	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
Claire Gute	Custom	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	----	----
Jane Johnson	Project Leader	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
Ken Black	Custom	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	----	----
Laura Rodriguez	Viewer	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	----	----
Lena Hernandez	Custom	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	----	----

Example: A permission rule set for the "Finance" group.

Who can set permissions

User who have the **Set Permissions** capability can change permissions for content items in projects that aren't locked. Administrators, content owners, and users with the **Project Leader** capability automatically receive the **Set Permissions** capability.

Note Project Leader is a permissions capability that you can set for a user or group at the project level.

Default permissions and projects

The permissions assigned to content when it is published or created on the server are the item's *default permissions*. Default permissions are set only at the project level, and only by administrators and users with the Project Leader capability.

- New projects get a copy of content permissions from the **Default** project in the site. These permissions include the permissions for the project, and the default permissions for its workbooks and data sources.
- New workbooks and data sources use the default permissions from their project. When content permissions are not locked, the individual workbook and data source permissions can be edited to differ from the defaults.

- New views use the default permissions from their workbook. When content permissions are not locked and the views aren't shown as tabs in the workbook, the individual view permissions can be edited to differ from the defaults. Note that tabbed views always use their workbook permissions.

When the content permissions are locked to the project, workbooks and data sources in the project will always use the default permissions. Views in the workbooks will always use their workbook permissions. The default permissions can only be changed at the project level.

For more information on the **Default** project, see [Projects on page 136](#).

For more information on default permissions, see [Set Default Permissions for a Project, and its Workbooks and Data Sources on page 364](#).

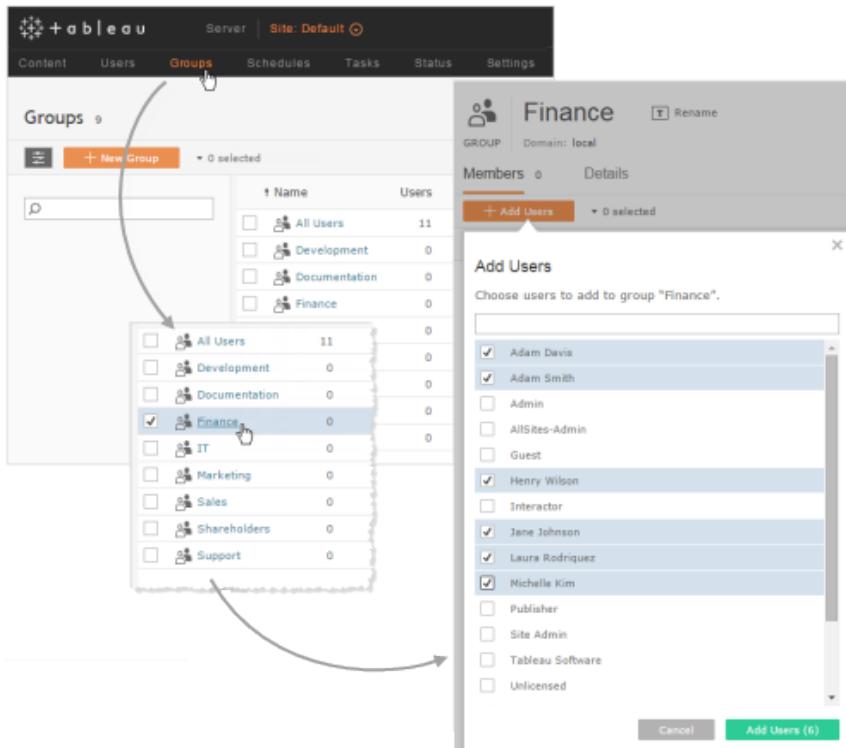
For more information on locking content permissions, see [Lock Content Permissions to the Project on page 369](#).

Quick Start: Permissions

You can use permission rules to control access to specific content on a site. Every user has a set of allowed capabilities based on their site role. Each content type—projects, workbooks, views, and data sources—can have permission rules assigned to groups or to specific users. The easiest and most efficient way to manage permissions is to create permission rules for groups.

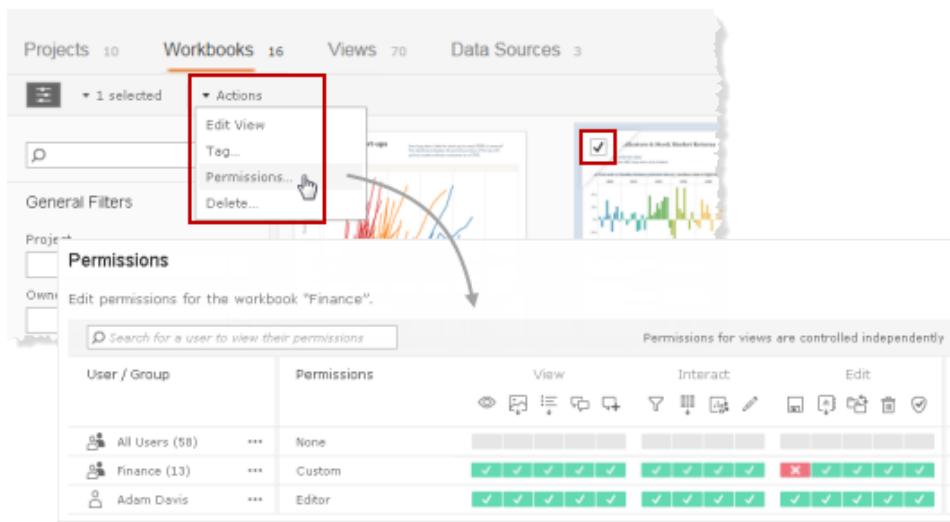
1 Add Users to Groups

Within a site, click **Groups**. Create groups for users who should have the same permissions, and then add the users to these groups. Click a group name, and then click **Add Users** to select the users to be included in the group.



2 Select the Content

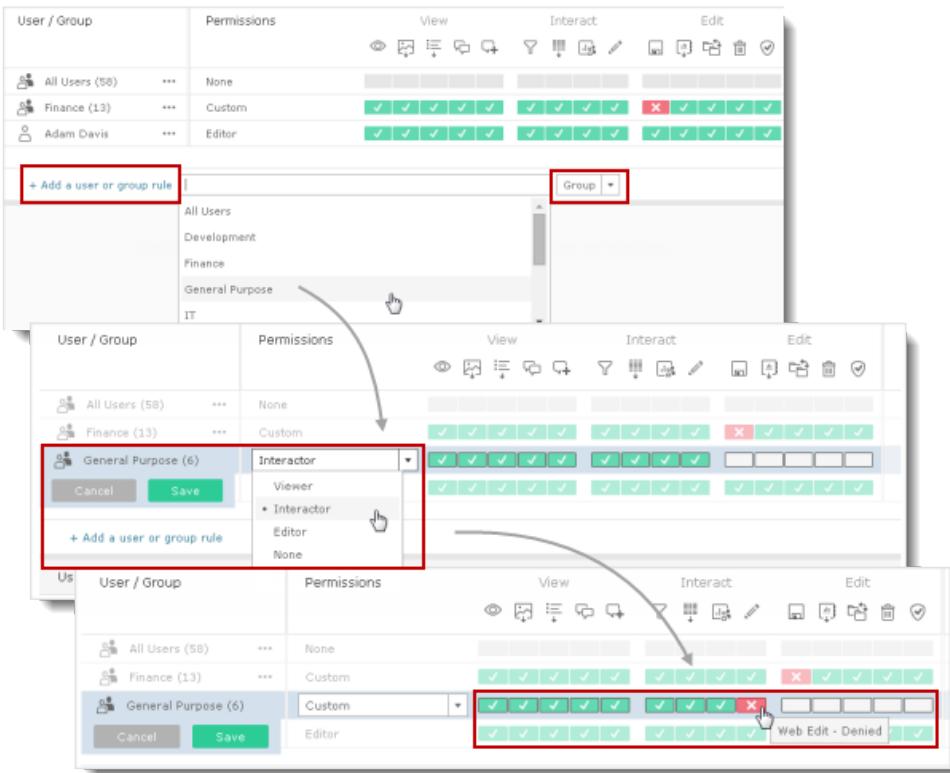
On the Content page for a site, click **Workbooks**, **Views**, **Projects**, or **Data Sources**. Select an item in the page. Select **Actions > Permissions** to view the permission rules for that content.



A permission rule is a set of capabilities (such as the ability to edit a view) that are allowed or denied to a user or group of users. Available capabilities vary depending on the type of content selected.

3 Create a Permission Rule

Click **Add a user or group rule**, select **Group**, enter search text , and then select a name from the list. Select a permission role template to apply an initial set of capabilities for the group. Click a capability to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.



Whether a user can set permissions is based on their site role and how their **Set Permissions** capability is set.

4 View User Permissions

After you save the permission rule for the group, you can view the effective permissions for that content.

Click a group name to see the group's users and their permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

User / Group	Permissions	View	Interact	Edit
All Users (58) ...	None
Finance (13) ...	Custom
General Purpose (...	Custom	X
Adam Davis ...	Editor
+ Add a user or group rule <input type="text" value="Enter a group name"/> Group				
User Permissions General Purpose (6)				
Harold Pawlan	Custom
Henry MacAllister	Viewer
Henry Wilson	Administrator
Irene Maddox	Custom
Janet Molinari	Custom
Karen Daniels	Viewer

Custom indicates a user's capabilities have been changed from the initial settings for their site role or content role.

Site roles

A user's site role determines the maximum permissions allowed for that user. Q-

- Server and site administrators can access all site content with full permissions.
- Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For more information, see [Site Roles for Users](#) on page 176.

Permissions evaluation

- **Denied** takes precedence over **Allowed**.
- **Unspecified** results in **Denied** if no other permissions are specified.
- Specific user permissions on content take precedence over group permissions on content. In other words, user permissions trump group permissions.

For more information on working with permissions, see [Manage Permissions](#) on page 336, [How Permissions are Evaluated](#) on page 348, [Permission Rules and User Permissions](#) on page 344, and [Projects](#) on page 136.

Quick Start: Lock Content Permissions to a Project

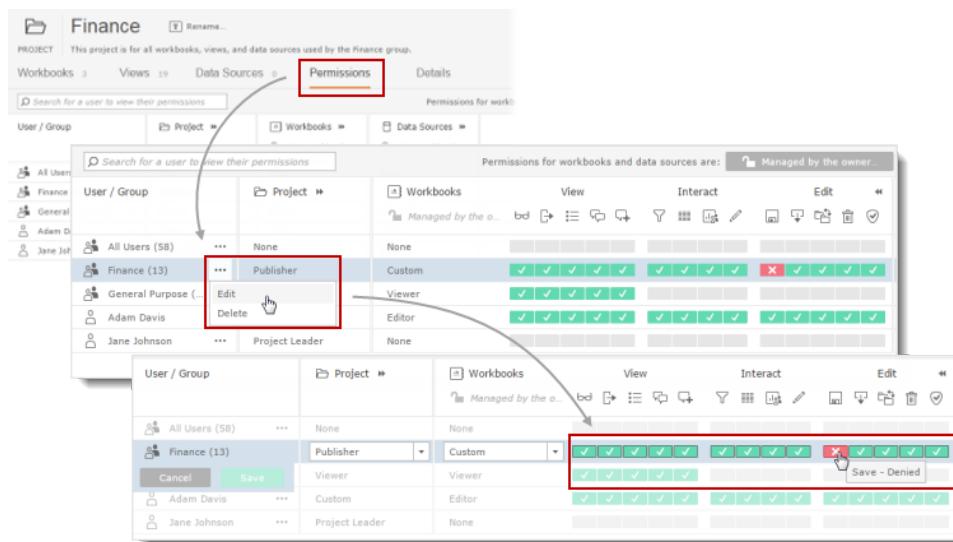
As an administrator or project leader, you can lock content permissions in a project to prevent users from changing the permissions of any content in the project. When permissions are locked to the project, the default permissions are applied to all workbooks and data sources in a project and cannot be modified by users (including the content owners).

Note: Content owners always get full access to the content they've published, but cannot change permissions for their workbooks and data sources when the parent project permissions are locked.

For information on setting permissions, see [Manage permissions](#) and [Permission Rules and User Permissions](#) on page 344. For more information on setting default permissions and locking content permissions to the project, see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364 and [Lock Content Permissions to the Project](#) on page 369.

1 Set Default Permissions for the Project

Because the content inside locked projects always uses the default permissions, first verify that your default permissions are set appropriately. In a site, click **Content > Projects**. Open a project, and then click **Permissions**. Add a user or group and select a permission role template for that content type, or click **Edit**, and then set capabilities to **Allowed**, **Denied**, or **Unspecified**.



Administrators and Project Leaders can edit default permissions at any time.

2 Lock Content Permissions to the Project

In a project's permissions, click the **Managed by the owner** button. The button label indicates whether content permissions are currently locked to the project or managed by the content owner. Select **Locked to the project**, and then click **Save**.

When permissions are locked to the project, all content in the project uses the default permissions. No users can change permissions for individual workbooks (including views) or data sources in the project.

3 View Locked Permissions

Open a project, select a workbook or data source in the project, and then click **Actions > Permissions**. When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

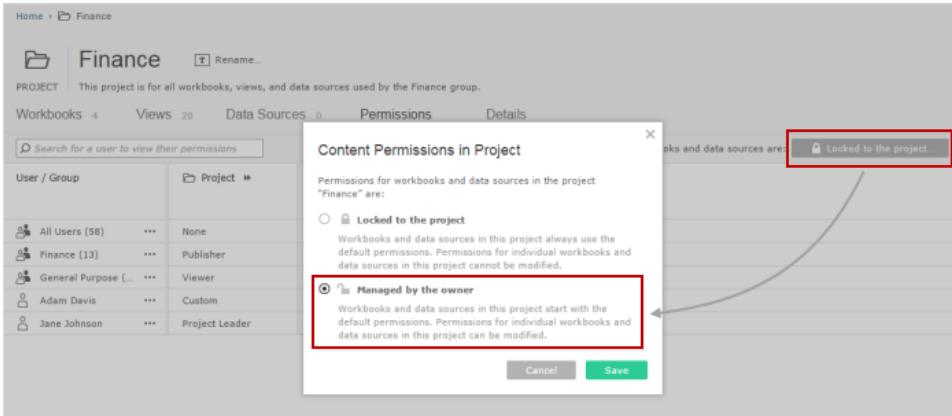
User / Group	Permissions	View	Interact	Edit
All Users (58)	None	[Grey]	[Grey]	[Grey]
Finance (13)	Custom	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Red X] ✓ ✓ ✓ ✓ ✓
General Purpose (6)	Viewer	[Green] ✓ ✓ ✓ ✓ ✓	[Grey]	[Grey]
Adam Davis	Editor	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓

User	Role	View	Interact	Edit
Adam Davis	Administrator	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓
Andrew Allen	Custom	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Grey]
Andrew Smith	Custom	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Grey]
Ashley Garcia	Administrator	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓
Claire Gute	Custom	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Grey]
Darren Powers	Custom	[Green] ✓ ✓ ✓ ✓ ✓	[Green] ✓ ✓ ✓ ✓ ✓	[Grey]

In this example, the workbook owner has full permissions for the workbook, but cannot change the workbook permissions while they are locked to the project.

4Unlock Content Permissions for the Project

In a site, click **Content > Projects**. Select a project, and then click **Actions > Permissions**. Click the **Locked to the project** button. Select **Managed by the owner**, and then click **Save**.



When a project's content permissions are **Managed by the owner**, individual workbooks, views, and data sources in the project start with the default permissions and can be modified by users.

Notes on project permissions:

- Only administrators and project leaders can lock content permissions, and set and edit default permissions in a project.
 - Administrators and project leaders can edit default permissions for the project, its workbooks, and its data sources at any time, at the project level.
 - Individual workbook, view, and data source permissions cannot be edited by users (including content owners) when a project is locked.
 - Workbooks and data sources in a locked project always use the default permissions. Views in a locked project always use the workbook permissions.
- P-

Permission Rules and User Permissions

When you specify permissions for a project, workbook, view, or data source, you specify who is allowed to work with that resource through a permission rule. Permission rules are the explicit capabilities that can be set for an individual user, or for a group—for each resource.

The Permissions window has two sections: **Permission Rules** (upper section) and **User Permissions** (lower section). You set permissions in **Permission Rules**, and you view the effective or resulting permissions in **User Permissions**.

Permissions		Permissions for views are inherited from the workbook									
User / Group	Permissions	View			Interact			Edit			
		bd	↳	☰	💬	➕	✖	⠇	⠇	⠇	⠇
All Users (58)	...	None									
Finance (13)	...	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓
General Purpose (6)	...	Viewer	✓	✓	✓	✓	✓				
Adam Davis	...	Editor	✓	✓	✓	✓	✓	✓	✓	✓	✓

+ Add a user or group rule

User Permissions	General Purpose (6)										
Harold Pawlan	Viewer	• • • • •									
Henry MacAllister	Viewer	• • • • •									
Henry Wilson	Administrator	• • • • •			• • • • •		• • • • •				
Irene Maddox	Viewer	• • • • •									
Janet Molinari	Viewer	• • • • •									
Karen Daniels	Viewer	• • • • •									

Permission Rules

The permission rules you set up include the user or group and the set of capabilities you want users to have for on that content item (such as the ability to edit a view). Available capabilities vary depending on the type of content selected, and can be set to **Allowed**, **Denied**, or **Unspecified**.

For information about setting and viewing permissions, see [Quick Start: Permissions](#) on page 338, [Edit Permission Rules](#) on page 372, and [View Permission Rules and User Permissions](#) on page 371.

User / Group	Permissions	View	Interact	Edit							
		bd	↳	☰	💬	➕	✖	⠇	⠇	⠇	⠇
All Users (58)	...	None									
Finance (13)	...	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓
General Purpose (6)	...	Viewer	✓	✓	✓	✓	✓				
Adam Davis	...	Edit			✓	✓	✓	✓	✓	✓	✓
		Delete	👉								

Click the ellipses next to the permission rule name.

Select a permission role template and edit capabilities (the actions allowed on the content).

- **User / Group:** Lists users or groups of users the rule applies to.
- **Permissions:** Lists available permission role templates for a specific project, workbook, view, or data source. Each permission role template (such as **Editor**, **Interactor**, **Viewer**) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to **Custom**. For more information about permission role templates and capabilities, see [Set Permissions for Workbooks and Views on page 350](#), [Set Permissions for a Project on page 359](#), and [Set Permissions for a Data Source on page 354](#).
- **View / Interact / Edit:** Categories for the sets of capabilities that can be set to **Allowed**, **Denied**, or **Unspecified**. (**Unspecified** results in **Denied** if no other permissions are specified for a user or group on the content.)

User Permissions

The User Permissions area of the Permissions window shows the effective permissions for each user. These are the actual permissions for each user, after the user's site role and permission rules have been evaluated.

To view the user permissions for a group or user, click a user or group name in the permission rules list. The effective permissions for users in the group are displayed in the lower half of the Permissions window.

Effective user permissions for a resource are determined by:

- The maximum capabilities allowed for a user's site role. The site role acts as the "ceiling" for what permissions are allowed. For more information, see [Site Roles for Users on page 176](#).
- Whether the user owns the content item

- The evaluation of each user or group permission rule that applies to that user for that content item

For example, if a user is granted Editor-level permissions for a workbook (which allows all available capabilities), but has the site role of Viewer and does not own the workbook, the user will only be allowed the capabilities of **View, Export Image, Summary Data, View Comments, Add Comments, and Save**.

In the following example, a permission rule has been created for the Finance group. The permission role template of **Editor** was initially applied to the group, which granted all capabilities. The administrator then set the **Save** capability to **Denied**, so the name for the set of permissions applied to the group became **Custom**. The **User Permissions** section for the Finance group shows that most of the users in the group have all capabilities, except for the **Save** capability. One user has even fewer capabilities because that user has a site role of **Viewer**.

Permissions		Permissions for views are inherited from the workbook									
User / Group	Permissions	View			Interact			Edit			
		bd	Up	Down	Filter	Search	Copy	Print	Download	Upload	Delete
All Users (58)	None	grey	grey	grey	grey	grey	grey	grey	grey	grey	grey
Finance (13)	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
General Purpose (...)	Viewer	✓	✓	✓	✓	✓	grey	grey	grey	grey	grey
Adam Davis	Editor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

+ Add a user or group rule
User Permissions Finance (13)
Adam Davis Administrator

Adam Davis	Administrator	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Andrew Allen	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Andrew Smith	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ashley Garcia	Administrator	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Claire Gute	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Jane Johnson	Project Leader	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ken Black	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Laura Rodriguez	Viewer	✓	✓	✓	✓	✓	grey	grey	grey	grey	grey
Lena Hernandez	Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note that the **All Users** group permission rule in this example has been set to **None**, which leaves all of the permissions as **Unspecified** for the **All Users** group. This approach requires the administrator to specifically assign permissions for only the groups or users that should see the content.

How Permissions are Evaluated

Permissions in Tableau Server are assigned to resources, also known as content—projects, workbooks, views, and data sources. After you specify permissions for a resource, you specify who can work with that resource using permission rules.

The views, workbooks, projects, and data sources on Tableau Server that users can access, and the actions available for these different content types, are affected by:

- **Site role.** A user's site role determines whether a user can publish, interact with, or only view resources and the different levels of permission capabilities allowed for a user. The site role acts as the "ceiling" for what permissions are allowed. For more information, see [Site Roles for Users on page 176](#).
- **Content permissions.** Every resource, that is, every project, workbook, view, or data source, can have a unique set of permission rules.

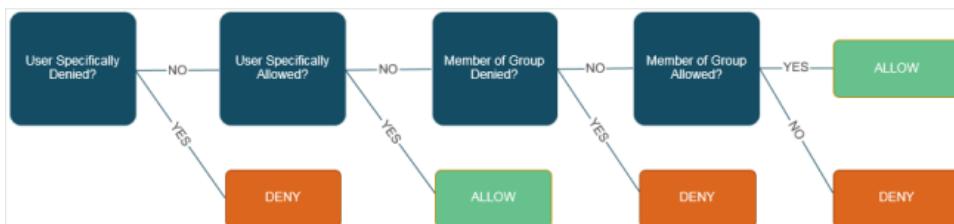
A permission rule includes the user or group, and the set of capabilities you want to grant users for a resource (such as the ability to edit a view). Each permission role template (such as **Editor**, **Interactor**, **Viewer**) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to **Custom**.

Available capabilities vary depending on the resource. Capabilities can be set to **Allowed**, **Denied**, or **Unspecified**. **Denied** always takes precedence over **Allowed**, and **Unspecified** results in **Denied** if no other permission rules allow a capability for a user.

- **Ownership.** Content owners always get full access to the content they've published. In projects with locked permissions, content owners cannot edit permissions for their workbooks and data sources.

Users with the **Set Permissions** capability can change permissions for content items in projects that aren't locked. Administrators, content owners, and users with the **Project Leader** capability automatically have the **Set Permissions** capability.

You can set permission rules for an individual user or group for each resource. This diagram illustrates how permission rules are evaluated in Tableau Server.



Effective user permissions are determined by:

- Maximum permissions allowed for a user's site role. For more information, see [Site Roles for Users](#) on page 176.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

Notes on permissions

- Server and site administrators can access all the resources in a site with full permissions.
- You cannot set permissions at the site level; permissions are assigned to resources only.
- Publishers (content owners) always get full access to their content. Content owners can change permissions on their workbooks and data sources, unless the parent project permissions are locked. For more information, see [Lock Content Permissions to the Project](#) on page 369.
- Individual user permissions on resources take precedence over group permissions on resources. In other words, user permissions trump group permissions.
- Workbook permissions serve as templates for view permissions. When content permissions are locked to the project, and when a workbook uses tabbed views, views inherit their workbook permissions. When permissions are not locked, and when a workbook is saved without tabs, the workbook and view permissions can be edited independently.
- Project default permissions serve as templates for content in a project. When content permissions are locked to the project, the workbooks and data sources always use the default permissions. When permissions are not locked, workbook and data source permissions can be edited independently.
- For each content item, every site user is automatically included in the **All Users** group. As a result, the All Users permission rule affects how permissions are evaluated for users when you create additional group permission rules for that content item.

If you use Tableau Server in an environment where openly sharing knowledge and information across the organization is important, set the permission rule for the **All Users** group in the **Default** project to the **Publisher** permission template. Users can publish to and consume content from new projects.

If you use Tableau Server in an environment where restricting access is important, set the permission rule for the **All Users** group in the **Default** project to the role of **None**. Then, add *explicit permissions* for groups and users to allow them to publish and work with content in new projects.

Tableau Server evaluates permissions in the following order of precedence:

1. **Server and Site Administrator:** Administrators can access all site content with full permissions.
2. **User - Unlicensed, Viewer license, or Guest:** If a user is Unlicensed, has a Viewer license (different than Viewer site role), or is a Guest, there are certain capabilities they are never allowed to perform. If the capability is explicitly denied for the user because of licensing, they are denied.
3. **Project Owner:** If the user owns the project that contains the content, the capability is allowed. Otherwise,
4. **Project Leader:** If the user has the Project Leader capability, or is in a group that has the Project Leader capability, they are allowed. If the user is explicitly denied the Project Leader capability, they are denied. Otherwise,
5. **User - Authorizable Owner:** If the user is the owner of the content, they are allowed. Otherwise,
6. **User - Capability Denied:** If the user has been explicitly denied the capability for the content, they are denied. Otherwise,
7. **User - Capability Allowed:** If the user has been explicitly allowed the capability for the content, they are allowed. Otherwise,
8. **Group - Capability Denied:** If the user belongs to a group that has been explicitly denied the capability for the content, they are denied. Otherwise,
9. **Group - Capability Allowed:** If the user belongs to a group that has been explicitly allowed the capability for the content, they are allowed. Otherwise,
10. The user is denied access to the content.

Set Permissions for Workbooks and Views

As an administrator or user with the **Set Permissions** capability, you can set permission rules for a workbook or a view.

For more details on working with permissions, see [Manage Permissions on page 336](#) and [Projects on page 136](#).

Note: When project content permissions are locked, permissions cannot be changed for individual workbooks and views in the locked project. For more information, see [Lock Content Permissions to the Project on page 369](#) and [Set Default Permissions for a Project, and its Workbooks and Data Sources on page 364](#).

Use permission rules to set these capabilities for workbooks:

User / Group	Permissions	View	Interact	Edit
All Users (58) ***	None	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions
Adam Davis ***	Editor	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions

Use permission rules to set these capabilities for views:

User / Group	Permissions	View	Interact	Edit
All Users (58) ***	None	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions
Adam Davis ***	Editor	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions

To set permissions on a workbook or view

- In the Content page of a site, click **Workbooks** or **Views**. Select a workbook or view, and then click **Permissions** to view the current permission rules.

The screenshot shows the Power BI Content page with the 'Workbooks' tab selected. In the center, a thumbnail of a workbook titled 'Finance' is displayed with a checkmark in its status bar. To the left, a sidebar shows '1 selected' and a 'General Filters' section. A red box highlights the 'Actions' dropdown menu, which contains options like 'Edit View', 'Tag...', 'Permissions...', and 'Delete...'. A callout arrow points from this menu to a detailed view of the 'Permissions' table below. This detailed view shows the same permissions matrix as the previous table, with columns for 'User / Group' and 'Permissions' (View, Interact, Edit), and rows for 'All Users (58)' and 'Adam Davis'.

User / Group	Permissions	View	Interact	Edit
All Users (58) ***	None	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions	<input type="checkbox"/> View <input type="checkbox"/> Download Image/PDF <input type="checkbox"/> Download Summary Data <input type="checkbox"/> View Comments <input type="checkbox"/> Add Comments <input type="checkbox"/> Filter <input type="checkbox"/> Download Full Data <input type="checkbox"/> Share Customized <input type="checkbox"/> Web Edit <input type="checkbox"/> Save <input type="checkbox"/> Download Workbook/Save As <input type="checkbox"/> Move <input type="checkbox"/> Delete <input type="checkbox"/> Set Permissions
Adam Davis ***	Editor	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Download Image/PDF <input checked="" type="checkbox"/> Download Summary Data <input checked="" type="checkbox"/> View Comments <input checked="" type="checkbox"/> Add Comments <input checked="" type="checkbox"/> Filter <input checked="" type="checkbox"/> Download Full Data <input checked="" type="checkbox"/> Share Customized <input checked="" type="checkbox"/> Web Edit <input checked="" type="checkbox"/> Save <input checked="" type="checkbox"/> Download Workbook/Save As <input checked="" type="checkbox"/> Move <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Set Permissions

Note: If you select multiple items and some of the items are read-only, you cannot view the permissions. Instead, select one view at a time.

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

The screenshot shows the 'Permissions' dialog for a workbook named 'Finance'. At the top, there is a search bar labeled 'Search for a user to view their permissions' and a note that 'Permissions for views are controlled independently'. Below this is a table with columns for 'User / Group', 'Permissions', 'View', 'Interact', and 'Edit'. Two rows are visible: 'All Users (58)' with 'None' permissions and 'Adam Davis' with 'Editor' permissions. A modal window titled '+ Add a user or group rule' is open, showing a dropdown menu with options: All Users, Development, Finance, General Purpose, and IT. The 'Finance' option is selected. The 'Save' button in the modal is highlighted with a red box.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.

The screenshot shows the 'Permissions' dialog for the 'Finance' group. The 'Save' button is highlighted with a green box. A dropdown menu is open next to the 'Permissions' column for the 'Finance' group, showing options: Interactor, Viewer, Editor, None, and Denied. The 'Interactor' option is selected. The 'Save' button is also highlighted with a green box.

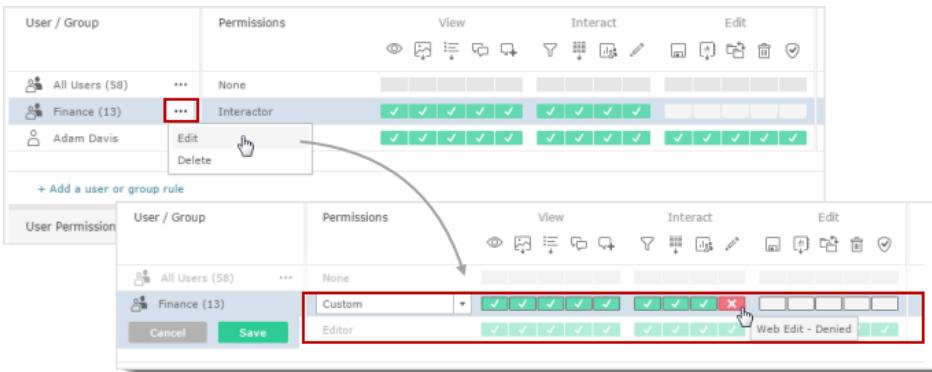
The list of capabilities and the available permission role templates vary depending on whether you are setting permissions for a workbook or a view. For more information on capability definitions, see [Permissions Reference on page 374](#).

Note: For workbooks and views that contain confidential data, it is good practice to set the All Users group permissions to **None** (all permissions **Unspecified**). You can then add other group permission rules to allow access.

The available permission role templates for workbooks and views are:

Template	Applies to...	Description
Viewer	workbooks views	Allows the user or group to view the workbook or view on the server.
Interactor	workbooks views	Allows the user or group to view the workbook or view on the server, edit workbook views, apply filters, view underlying data, export images, and export data. All other permissions are inherited from the user's or group's project permissions.
Editor	workbooks views	Sets all capabilities for the rule to Allowed .
None	workbooks views	Sets all capabilities for the rule to Unspecified .
Denied	workbooks views	Sets all capabilities for the rule to Denied .

4. To further customize the rule, click the actions menu (. . .) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.



5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

User / Group	Permissions	View	Interact	Edit
All Users (58)	None			
Finance (13)	Custom	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✗
Editor		✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
+ Add a user or group rule				
User Permissions Finance (13)				
Adam Davis	Administrator	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
Andrew Allen	Custom	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✗
Andrew Smith	Custom	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✗
Ashley Garcia	Administrator	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
Claire Gute	Viewer	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓

- Follow the same steps to configure additional permission rules on the content for more users or groups.

Note: Tabbed views are views in a workbook that is published to the server with **Show Sheets as Tabs** enabled. Tabbed views use the workbook permissions instead of the view permissions. When you view the permissions for a tabbed view in a workbook, you see the workbook's permission rules in the Permissions window, not the view's permission rules.

To edit tabbed view permissions, you must open the tabbed view's workbook permissions. The changes that you make to the workbook permissions affect all tabbed views in that workbook. When the workbook is saved again without tabs (or tabs are hidden), the default permissions are again applied to the workbook and views, and view permissions can then be edited.

Views in a workbook in a project with locked permissions will also use the workbook permissions. For more information, see [Lock Content Permissions to the Project](#) on page 369.

Set Permissions for a Data Source

As an administrator or user with the **Set Permissions** capability, you can change permissions for a data source.

For more details on working with permissions, see [Manage Permissions](#) on page 336 and [Projects](#) on page 136.

Note: When project content permissions are locked, permissions cannot be changed for data sources in the locked project. For more information, see [Lock Content](#)

[Permissions to the Project](#) on page 369 and [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

Permissions

Edit permissions for the data source "Data by country".

User / Group	Permissions	Use	Edit
		View Connect Save Download Data Source Delete Set Permissions	View Connect Save Download Data Source Delete Set Permissions
All Users (58)	None		
Finance (13)	Connector	✓ ✓	
Adam Davis	Editor	✓ ✓	✓ ✓ ✓ ✓

Use permission rules to set the following capabilities for a data source:

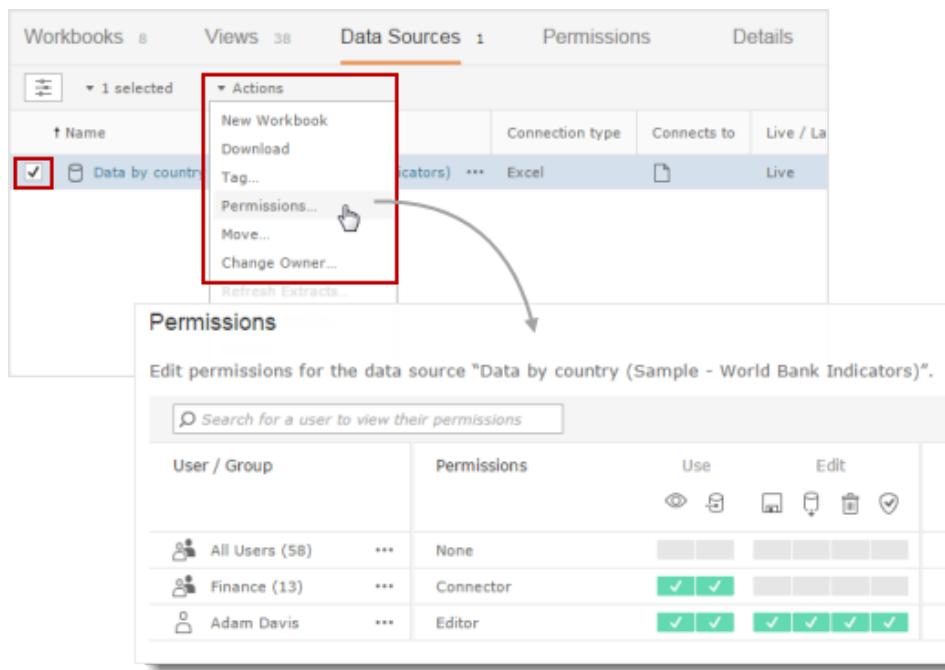
Capability	Description
View	Allows the user or group to view the workbooks and views in the project.
Connect	Connect to the data source. A user who accesses a view (in a project) that connects to a data source must have both View permission for the view and Connect permission for the data source.
Save	Allows the user or group to publish data sources to the server and overwrite data source connections on the server.
Download Data Source	Download the data source from the server.

Delete	Delete the data source.
	

Set Permissions	Specify permissions for the data source.
	

To set permissions for a data source

1. In the Data Sources page, select one or more data sources, and then select **Actions > Permissions**.



2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

The screenshot shows a table with columns for 'User / Group' and 'Permissions'. Under 'Permissions', there are 'Use' and 'Edit' sections with checkboxes. A dropdown menu is open over the 'Edit' section, listing 'None', 'Connector', 'Editor', 'None', and 'Denied'. The 'Connector' option is highlighted with a cursor icon.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.

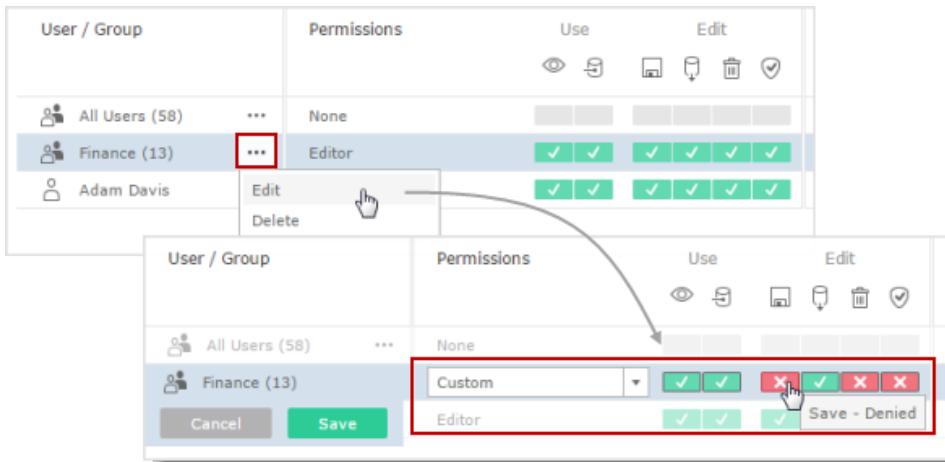
The screenshot shows a table with columns for 'User / Group' and 'Permissions'. Under 'Permissions', there are 'Use' and 'Edit' sections with checkboxes. A dropdown menu is open over the 'Edit' section, listing 'None', 'Connector', 'Editor', 'None', and 'Denied'. The 'Connector' option is highlighted with a cursor icon.

The permission role templates for data sources are:

Template	Description
Connector	Allows the user or group to connect to the data source on the server.
Editor	Allows the user or group to connect to data sources on the server. Also to publish, edit, download, delete, and set permissions for a data source, and schedule refreshes for data sources you publish.
None	Sets all capabilities for the permission rule to Unspecified .
Denied	Sets all capabilities for the permission rule to Denied .

Note: Cube data sources, like those for Microsoft Analysis Services or Oracle Essbase connections, must be used locally. To download the published data source to Tableau Desktop, you need the **Download** permissions. You must explicitly grant the **Download** permissions because the Data Source Connector role does not provide these. For more information, see [Cube Data Sources](#) on page 275.

4. To further customize the rule, click the actions menu (.) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.



5. Follow the same steps to configure additional permission rules on the content for more users or groups.
6. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

User / Group	Permissions	Use	Edit	
All Users (58) ***	None			
Finance (13) ***	Custom	✓ ✓	X ✓ X X	
Adam Davis ***	Editor	✓ ✓	✓ ✓ ✓ ✓ ✓ ✓	

+ Add a user or group rule

User Permissions Finance (13)	
Adam Davis	Administrator
Andrew Allen	Custom
Andrew Smith	Custom
Ashley Garcia	Administrator
Claire Gute	Custom
Darren Powers	Custom

Set Permissions for a Project

Every project includes permissions that can be set for the project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project, and each project can have its own set of default permissions. For more information, see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

Administrators and users with the Project Leader permission can lock content permissions to a project. For more information, see [Quick Start: Lock Project Permissions](#), [Lock Content Permissions to the Project](#) on page 369.

For more information on working with permissions, see [Manage Permissions](#) on page 336 and [Projects](#) on page 136.

Note: When you create a new project, it initially will have the same permissions as the **Default** project in the site, which are the default permissions for the project, and its workbooks and data sources.

Permissions

Edit permissions for the project "Default".

Search for a user to view their permissions

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None	View Save Project Leader	Managed by the owner Managed by the owner	None
Finance (13)	Publisher	✓ ✓	Custom	Connector
Adam Davis	Publisher	✓ ✓	Editor	Editor
Jane Johnson	Project Leader	✓	None	None

The three capabilities you can set specifically for a project are: **View**, **Save**, and **Project Leader**.

Capability	Description
 View	<p>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</p>
 Save	<p>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project.</p> <p>When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining all permissions.</p> <p>Subsequently, the original owner's access to the workbook is determined by that user's group permissions and any further permissions the new owner might set.</p> <p>This permission also determines the user's or group's ability to overwrite a workbook after editing it on the server. For related information, see Grant Web Edit, Save, and Download Permissions on page 377.</p>
 Project Leader	Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions.

To set permissions for the project

1. On the Projects page, select a project, and then select **Actions > Permissions**.

The screenshot shows the Project Management interface. At the top, there are tabs for 'Projects' (10), 'Workbooks' (16), 'Views' (70), and 'Data Sources' (3). Below these are buttons for '+ New Project' and 'Actions'. A search bar and general filters are also present. On the right, a table lists projects: 'Finance' (selected, indicated by a checked checkbox) and 'General Purpose'. A context menu is open over the 'Finance' project, with 'Permissions...' highlighted and a cursor pointing at it. Other options in the menu include 'Rename...', 'Change Owner...', and 'Delete...'. The table below shows columns for 'Workbooks' and 'Views'.

Project	Workbooks	Views
Finance	8	38
General Purpose	3	7

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

The screenshot shows the 'User / Group' permissions settings. It lists four entries: 'All Users (58)', 'Finance (13)', 'Adam Davis', and 'Jane Johnson'. Each entry has a 'Details' section with checkboxes for 'View', 'Edit', and 'Delete'. The 'Workbooks' section indicates if they are 'Managed by the owner'. A red box highlights the '+ Add a user or group rule' button at the bottom left. A dropdown menu is open, showing categories like 'All Users', 'Development', 'Finance', 'General Purpose', and 'IT', with 'General Purpose' currently selected. A cursor is pointing at the 'General Purpose' category.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.

The available permission role templates for projects are:

Template	Description
Viewer	Allows the user or group to view the workbooks and views in the project.
Publisher	Allows the user or group to publish workbooks and data sources to the server.
Project Leader	Allows the user or group to set permissions for all items in a project.
None	Sets all capabilities for the permission rule to Unspecified .
Denied	Sets all capabilities for the permission rule to Denied .
Data Source Connector	Allows the user or group to connect to data sources in the project.
Data Source Editor	Allows the user or group to connect to data sources in the project. Also to publish, edit, download, delete, and set permissions for a data source, and schedule refreshes for data sources you publish. This permission is relevant for views when accessing a view that connects to a data source.

- To further customize the rule, click the actions menu (.) next to the permission rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

The screenshot displays two tables side-by-side, illustrating the process of viewing and modifying user permissions.

Top Table (Initial Permissions):

User / Group	Project	Details
All Users (58)	None	(grid of 3x3 checkboxes: 1st row 3 green checkmarks, 2nd row 1 green, 1 red, 1 grey, 3rd row 3 green)
Finance (13)	Custom	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 1 green, 1 red, 1 grey, 3rd row 1 green, 1 red, 1 grey)
General Purpose (6)	Publisher	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 3 green, 3rd row 3 green)
Adam Davis	Edit	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 3 green, 3rd row 3 green)
Jane Johnson	Delete	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 3 green, 3rd row 3 green)

Bottom Table (Resulting Permissions):

User / Group	Project	Details	Workbooks
All Users (58)	None	(grid of 3x3 checkboxes: 1st row 3 green)	Managed by the owner
Finance (13)	Custom	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 1 green, 1 red, 1 grey, 3rd row 1 green, 1 red, 1 grey)	Custom
General Purpose (6)	Custom	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 3 green, 3rd row 3 green)	None
Jane Johnson	Project Leader	(grid of 3x3 checkboxes: 1st row 3 green, 2nd row 3 green, 3rd row 3 green)	None

A red box highlights the 'Custom' row under 'General Purpose' in both tables. A red arrow points from the 'Save' button in the top table to the 'Save - Denied' message in the bottom table.

5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

Permissions

Edit permissions for the project "Finance".

Search for a user to view their permissions

User / Group	Project	Details	Workbooks
All Users (58)	None		Managed by the owner
Finance (13)	Custom		Custom
General Purpose (...	Custom		None
Harold Pawlan	Custom		Editor
Jane Johnson	Project Leader		None

+ Add a user or group rule

User Permissions General Purpose (6)

User	Role	Actions	Notes
Harold Pawlan	Viewer		None
Henry MacAllister	Viewer		Save: Denied (by group rule)
Henry Wilson	Administrator		Administrator
Irene Maddox	Viewer		None
Janet Molinari	Viewer		None
Karen Daniels	Viewer		None

- Follow the same steps to configure additional permission rules on the content for more users or groups.

Set Default Permissions for a Project, and its Workbooks and Data Sources

As an administrator or project leader, you can set a project's permissions and the default permissions for its workbooks and data sources.

Each project can have its own set of default permissions. The permissions that you set are the default permissions for all content in the project, including content that is being published to the project from Tableau Desktop.

Note: New projects are always created with the default permissions set for the **Default** project.

For additional information on working with permissions, see **Manage Permissions** on page 336 and **Projects** on page 136.

Notes on default permissions in locked projects

You can choose to have the default permissions apply to all workbooks and data sources in a project, and ensure that no one can change those settings, by locking content permissions to the project. For more information, see [Lock Content Permissions to the Project on page 369](#).

- Workbooks and data sources in a locked project always use the default permissions set for content in that project. Views in a locked project always use the workbook permissions. This applies to workbooks and data sources when they are being published from desktop.
- Administrators and users with the Project Leader permission can always edit default permissions, even when a project is locked.
- Users, including content owners, cannot edit individual workbook, view, and data source permissions when content is locked to the project.

To set default permissions in a project

1. In the Content page of a site, click a project, and then click **Permissions** in the project place page.

The screenshot shows the 'Finance' project page in the 'Permissions' tab. The page header includes 'Home > Finance' and a 'Rename...' button. Below the header, it says 'This project is for all workbooks, views, and data sources used by the Finance group.' The navigation bar has tabs for 'Workbooks' (3), 'Views' (19), 'Data Sources' (0), 'Permissions' (selected), and 'Details'. A search bar says 'Search for a user to view their permissions'. To the right, it says 'Permissions for workbooks and data sources are: Managed by the owner...'. The main table lists users and groups with their respective permissions:

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Editor	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

+ Add a user or group rule		Group
<input type="button" value="All Users"/> <input type="button" value="Development"/> <input type="button" value="Finance"/> <input type="button" value="General Purpose"/>		<input type="button" value="Group"/>
or select a permission rule above to view use		

For an existing user or group, click the actions menu (. . .), and then click **Edit**.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (6)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (6)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

3. Select a permission role template for **Project**, **Workbooks**, or **Data Sources**, and then click **Save**.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (6)	Viewer	Viewer	Connector
Jane Johnson	Viewer	Editor	Editor
+ Add a user or group rule	None	None	None
	Viewer	Editor	Editor
	None	None	None
	Denied		

Or, to create a custom set of capabilities, click the **Project**, **Workbooks**, or **Data Sources** labels to expand the permissions view. Click capabilities to set them to **Allowed**, **Denied**, or **Unspecified**. Click **Save**.

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None	None	None	None
Finance (13)	Publisher	Custom	Custom	Connector
Adam Davis	Custom	Editor	Editor	Editor
Jane Johnson	Project Leader	None	None	None
		Project Leader - Allowed		

This example shows how to set project permissions. The same general steps apply for workbooks and data sources.

Note: To change the settings after saving, click the actions menu (.), and then click **Edit**.

- View the user permissions, which are the effective permissions.

Click a group name or user name in the permission rules to see the resulting user permissions.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (...)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None
+ Add a user or group rule			
User Permissions General Purpose (6)			
Harold Pawlan	Viewer	Viewer	Connector
Henry MacAllister	Viewer	Viewer	Custom
Henry Wilson	Administrator	Administrator	Administrator
Irene Maddox	Viewer	Viewer	Connector
Janet Molinari	Viewer	Viewer	Connector
Karen Daniels	Viewer	Viewer	Custom

Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.

User / Group	Project	Details	Workbooks	Data Sources
All Users (58)	None		None	None
Finance (13)	Publisher		Custom	Connector
General Purpose (...)	Viewer		Viewer	Connector
Adam Davis	Custom		Editor	Editor
Jane Johnson	Project Leader		None	None
+ Add a user or group rule				
User Permissions General Purpose (6)				
Harold Pawlan	Viewer		Viewer	Connector
Henry MacAllister	Viewer		Viewer	Custom
Henry Wilson	Administrator		Administrator	Administrator
Irene Maddox	Viewer		Viewer	Connector
Janet Molinari	Viewer		Viewer	Connector
Karen Daniels	Viewer		Viewer	Custom

Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

User / Group	Project	Details	Workbooks	Data Sources
	None	Managed by the owner	Managed by the owner	Managed by the owner
All Users (58) ***	None		None	None
Finance (13) ***	Publisher	✓ ✓	Custom	Connector
General Purpose (1) ***	Viewer	✓	Viewer	Connector
Adam Davis ***	Custom	✓ ✓ ✓	Editor	Editor
Jane Johnson ***	Project Leader		None	None

+ Add a user or group rule

User Permissions General Purpose (6)

User	Project	Workbooks	Data Sources
Harold Pawlan	Viewer	Viewer	Connector
Henry MacAllister	Viewer	Viewer	Custom
Henry Wilson	Administrator	Administrator	Administrator
Irene Maddox	Viewer	Viewer	Connector
Janet Molinari	Viewer	Viewer	Connector
Karen Daniels	Viewer	Viewer	Custom

- Follow the same steps to configure additional permission rules for more users or groups.

Lock Content Permissions to the Project

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.

When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

Note: Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For information on default permissions, see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

Note: Administrators and project leaders can set and edit default permissions for the project, and its workbooks and data sources when it is locked.

- In the Content page of a site, open a project, and then click **Permissions** in the project place page.

User / Group	Project	Workbooks	Data Sources
All Users (58)	None	None	None
Finance (13)	Publisher	Custom	Connector
General Purpose (...)	Viewer	Viewer	Connector
Adam Davis	Custom	Editor	Editor
Jane Johnson	Project Leader	None	None

2. Click the **Managed by the owner** button.

The padlock icon on the button label indicates whether content permissions are currently locked to the project or managed by the content owner.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.

Content Permissions in Project

Permissions for workbooks and data sources in the project "Finance" are:

Locked to the project

Workbooks and data sources in this project always use the default permissions. Permissions for individual workbooks and data sources in this project cannot be modified.

Default permissions will be applied to all workbooks and data sources in the project when you save.

Managed by the owner

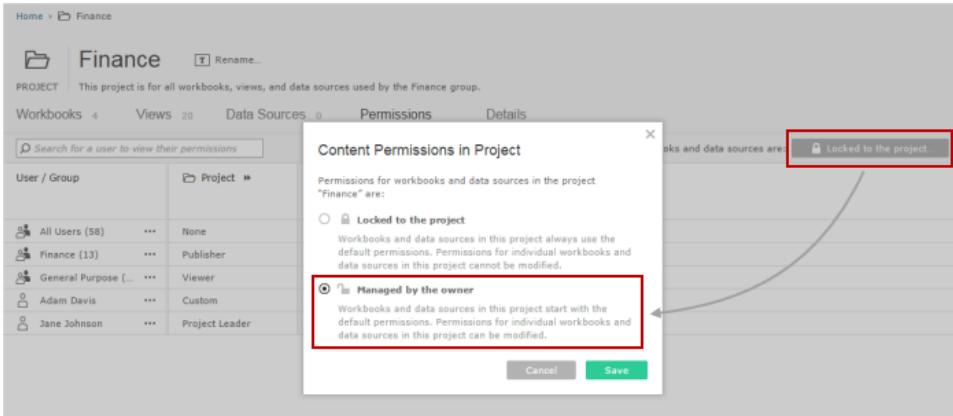
Workbooks and data sources in this project start with the default permissions. Permissions for individual workbooks and data sources in this project can be modified.

When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

Permissions					
See permissions for the workbook "Finance".					
User / Group	Permissions	View	Interact	Edit	Permissions locked
All Users (58)	None	gd	↳	☰	⋮
Finance (13)	Custom	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓	✗	✓ ✓ ✓ ✓
General Purpose (6)	Viewer	✓ ✓ ✓ ✓ ✓	⋮	⋮	⋮
Adam Davis	Editor	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓

User Permissions Finance (13)					
Adam Davis	Administrator	• • • • •	• • • • •	• • • • •	• • • • •
Andrew Allen	Custom	• • • • •	• • • • •	• • • • •	• • • • •
Andrew Smith	Custom	• • • • •	• • • • •	• • • • •	• • • • •
Ashley Garcia	Administrator	• • • • •	• • • • •	• • • • •	• • • • •
Claire Gute	Custom	• • • • •	• • • • •	• • • • •	• • • • •
Darren Powers	Custom	• • • • •	• • • • •	• • • • •	• • • • •

- To unlock content permissions for the projects, open the project permissions again. Click the **Locked to the project** button. In the **Content Permissions in Project** dialog box, select **Managed by the owner**, and then click **Save**.



The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

View Permission Rules and User Permissions

At any time, you can view the permissions for a user or group, for a view, workbook, project, or data source. The permissions shown are specific to the view, workbook, data source, or project you have selected.

- On the Content page for a site, click **Workbooks**, **Views**, **Projects**, or **Data Sources**. To select an item in the page, select the checkbox for the item.

2. Select **Actions > Permissions** to view the current permission rules.

The screenshot shows the Power BI service dashboard with 'Projects 10', 'Workbooks 16', 'Views 70', and 'Data Sources 3'. A specific workbook is selected. In the top navigation bar, the 'Actions' dropdown is open, with 'Permissions...' highlighted and a red box around it. A cursor arrow points from this box to a larger screenshot below. The larger screenshot shows the 'Edit permissions for the workbook "Finance"' dialog. It includes a search bar, a note about independent view permissions, and a table with columns for 'User / Group', 'Permissions', 'View', 'Interact', and 'Edit'. The table data is as follows:

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Finance (13)	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Adam Davis	Editor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

3. Click a group or user name in the permission rules area to see the resulting permissions. Hover over a capability box in User Permissions to see a tooltip with details on whether a capability is allowed or denied.

This screenshot shows the 'User Permissions' section for the 'Finance (13)' group. It lists users and their roles: Adam Davis (Administrator), Andrew Allen (Custom), Andrew Smith (Custom), Ashley Garcia (Administrator), and Claire Gute (Viewer). Below this is a table of user permissions with columns for 'View', 'Interact', and 'Edit'. A tooltip 'Web Edit: Denied (by group rule)' is shown over a 'Edit' checkbox for Andrew Allen, which is highlighted with a red box. Other checkboxes for the same row are also highlighted with red boxes.

User	Role	View	Interact	Edit
Adam Davis	Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Andrew Allen	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Andrew Smith	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ashley Garcia	Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Claire Gute	Viewer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Custom indicates a user's capabilities have been changed from the initial settings for their site role or content role.

Edit Permission Rules

1. In the Content page of a site, select a project, workbook, view, or data source, and then select **Actions > Permissions** to view the current permission rules.

To select an item in the page, select the checkbox for the item.

Projects 10 Workbooks 16 Views 70 Data Sources 3

General Filters

Permissions

Edit permissions for the workbook "Finance".

User / Group Permissions

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Finance (13)	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Adam Davis	Editor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Example of permission rules for a workbook.

2. For the permission rule that you want to change, click the actions menu (. . .) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

User / Group Permissions

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Finance (13)	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Adam Davis	Editor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

User Permission

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Finance (13)	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Editor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

+ Add a user or group rule

Cancel Save

3. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

User / Group	Permissions	View	Interact	Edit
All Users (58)	None	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Finance (13)	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Editor		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

+ Add a user or group rule

User Permissions	Finance (13)			
Adam Davis	Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Andrew Allen	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Andrew Smith	Custom	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ashley Garcia	Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Claire Gute	Viewer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Follow the same steps to configure additional permission rules on the content for more users or groups.

Permissions Reference

Administrators and other authorized users can allow or deny permissions on resources in Tableau Server. Permissions can also be set in Tableau Desktop when publishing a workbook or data source to Tableau Server.

Administrators always have full control of all resources on Tableau Server, and site administrators have full control of all resources on a site. If you publish a workbook or data source to Tableau Server, you are the owner of that resource, and you retain full control over that resource, with the exception of setting permissions on resources in locked projects. For more information, see [Lock Content Permissions to the Project on page 369](#).

The following table shows which permissions apply to which resources in Tableau Server, and describes the capabilities (that is, the actions users can perform) with each permission.

Permission	Applies to...	When allowed, users can...
👁 View	workbooks data sources views projects	View the item on Tableau Server. A user who accesses a view that connects to a data source must have both View permission for the workbook and Connect permission for the data source. Note: When a workbook is configured to show sheets as tabs, all views use the workbook permissions, even if different permissions are specified on an individual view.
📝 Web Edit	workbooks views	Edit views in workbooks. See Grant Web Edit, Save, and Download Permissions on page 377 .

Permission	Applies to...	When allowed, users can...
Save	workbooks data sources views projects	<p>Overwrite the resource on the server. When allowed, the user can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining access to all permissions. Subsequently, the original owner's access to the workbook is determined by that user's group permissions and by any further permissions the new owner might set.</p> <p>This permission also determines the user's or group's ability to overwrite a workbook after editing it on the server. See Grant Web Edit, Save, and Download Permissions on page 377.</p> <p>Special consideration for the All Users group: To help protect an owner's content from being overwritten by another user (via publishing from Tableau Desktop or saving a web-edited workbook on Tableau Server), whenever a user publishes into a project where the All Users group has permissions, the Save permission for the All Users group is changed from Allowed to Unspecified by default. You can then manually modify this permission by following the steps in Set Permissions for Workbooks and Views on page 350 to change this from Unspecified to Allowed.</p>
Download Workbook/ Save As	workbooks	Download a workbook from the server, and also save an edited workbook as a new workbook on the server. For more information, see Download Workbooks on page 308 and Grant Web Edit, Save, and Download Permissions on page 377.
Download Data Source	data sources	Download the data source from the server.
Delete	workbooks data sources views	Delete the resource.
Filter	workbooks	Modify filters in the view, keep only filters, and exclude data.

Permission	Applies to...	When allowed, users can...
	views	
Add Comments 	workbooks views	Add comments to views in a workbook. See Comment on Views on page 307 .
View Comments 	workbooks views	View the comments associated with the views in a workbook. See Comment on Views on page 307 .
Download Summary Data 	workbooks views	View the aggregated data in a view, or in the user's selection within the view, and download that data as a text file.
Download Full Data 	workbooks views	View the raw data behind each row in a view, as restricted by any marks the user has selected, and download the data as a text file.
Download Image/PDF 	workbooks views	Download each view as an image. For more information, see Download Views on page 309 .
Share Customized 	workbooks views	Make saved customizations to a view available for others to see. Users can create custom views using Custom Views in Tableau Server. For more information, see Custom Views on page 320 .
Move 	workbooks	Move workbooks between projects. Note: Only administrators can move data sources between projects.

Permission	Applies to...	When allowed, users can...
Set Permissions 	workbooks data sources views	Specify permissions for the resource. For workbooks, this permission extends to the views in a workbook.
Connect 	data sources	Connect to the data source. A user who accesses a view (in a project) that connects to a data source must have both View permissions for the view and Connect permission for the data source. Note: If a workbook is configured to show sheets as tabs, all views use the workbook permissions, even if different permissions are specified on an individual view.
Project Leader 	projects	Set permissions for all resources in a project and for the project itself. Can lock project permissions and edit default permissions.

Note: Tabbed views are views in a workbook that is published to the server with **Show Sheets as Tabs** enabled. Tabbed views use the workbook permissions instead of the view permissions. When you view the permissions for a tabbed view in a workbook, you see the workbook's permission rules in the Permissions window, not the view's permission rules. To edit tabbed view permissions, you must open the tabbed view's workbook permissions. The changes that you make to the workbook permissions affect all tabbed views in that workbook. When the workbook is saved again without tabs (or tabs are hidden), the default permissions are again applied to the workbook and views, but view permissions can then be edited.

Views in a workbook in a project with locked permissions will also use the workbook permissions. For more information, see [Lock Content Permissions to the Project](#) on page 369.

Grant Web Edit, Save, and Download Permissions

For a user to be able to edit, save, and download workbooks, they must have a site role that allows those actions, and specific capabilities in a user or group permission rule.

The following capabilities control whether a user can edit, save, and download views:

- **Web Edit**—determines whether the user can edit workbook views in Tableau Server.

To edit an existing workbook, a user must have a site role of **Interactor** or **Publisher**. The **Web Edit** capability must be set to **Allow** in the workbook permissions.

Note: Users with a site role of **Interactor** are not allowed to save or download workbooks.

- **Download/Save As**—determines whether the user sees the **Save** and **Save As** commands while they are editing a view, and whether they can save their changes to a new workbook. It also determines whether users can open a workbook on the server using Tableau Desktop.

To save changes to a workbook or save a workbook as a new workbook on Tableau Server, a user must have a site role of **Publisher**. The **Save** and **Download/Save As** capabilities must be set to **Allow** in the workbook permissions.

- **Save**—determines whether users can save changes to an existing workbook on the server (overwrite a workbook).

To save changes to a workbook, a user must have a site role of **Publisher**. The **Save** capability must be set to **Allow** in the workbook permissions.

Note: Setting the **Save** capability to **Denied** for a project disables saving to the entire project, as well as disabling overwriting the existing workbook.

To grant Web Edit permissions

1. Set the site role of the user to **Interactor** or **Publisher**. For more information, see [Change Site Roles on page 209](#).
2. In the permission rules for a group or user at the workbook level, set the **Web Edit** capability to **Allowed**.
3. Save the rule.

To grant Save and Download/Save As permissions

1. Set the site role of the user to **Publisher**. For more information, see [Change a Site Role](#).

Note: **Interactors** are not allowed to save or download workbooks.

2. Create a permission rule for a group or user at the project and workbook level. Set the following capabilities:

To allow users (Publisher site role) to edit and save changes to existing and new workbooks

Permission	For the project	For specified workbooks in the project
Web Edit	-	Allowed
Download/Save As	-	Allowed
Save	Allowed	Allowed

Note: To apply the default permissions to all workbooks within the project, lock content permissions to the project. For more information, see [Lock Content Permissions to the Project](#) on page 369.

To allow users (Publisher site role) to edit and save changes to new workbooks, but not overwrite existing workbooks

Permission	For the project	For specified workbooks in the project
Web Edit	-	Allowed
Download/Save As	-	Allowed
Save	Allowed	Denied

Important: In this scenario, permissions must be set manually on each workbook and the project permissions are not locked. If project permissions are locked, the permissions apply to all workbooks in the project.

3. Save the rule.

Note: When you deny **Save** permissions for a workbook, users can still click **Save** when editing the workbook in Tableau Server, but a message appears that tells users they do not have permission to overwrite the workbook and the changes will not be saved.

About permissions for views in workbooks

Permissions for views in workbooks are inherited from the workbook permissions.

If a user selects **Show sheets as tabs** when publishing a workbook from Tableau Desktop or saving it on Tableau Server, the workbook permissions override the permissions on individual views. When the workbook is saved again without tabs, the default permissions are applied to the workbook and views, but view permissions can then be edited.

See also

[Permissions Reference](#) on page 374

[Quick Start: Permissions](#)

[Quick Start: Lock Content Permissions to a Project](#) on page 341

[Permission Rules and User Permissions](#)

[Set Permissions for Workbooks and Views](#) on page 350

[Set Permissions for a Project](#) on page 359

[Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364

[Create Project-Based Permissions](#) below

[Site Roles for Users](#) on page 176

Create Project-Based Permissions

As an administrator, you can organize a collection of related workbooks and data sources in a project. You can then control access to that content by creating permission rules for groups of users who need similar access levels to publish or interact with that content.

Note: For this scenario, you set the permission rule for the All Users group for the project to **None**, which means that permissions are **Unspecified** for the All Users group.

Preparation

Before you begin creating projects and project-based permissions, document the projects and permission levels that you want users to have in each project.

This roadmap exercise helps you organize permissions to be most efficient to manage over time, and can help you identify any user or permission gaps in your solution.

Also read the following topics in the Tableau Server Help:

- [Manage Permissions](#) on page 336 and permissions-related topics
- [Projects](#) on page 136 and projects-related topics
- [Grant Web Edit, Save, and Download Permissions](#) on page 377

Step 1: Create projects and user groups

1. Sign in to Tableau Server with your administrator user name and password.
2. On the Projects page, click **New Project**.
3. Click **Groups**, and then click **New Group**.

Create groups that correspond to each project and access level. For example, for a project that allows users only to access the views, you might use a name similar to Project1_Viewer. For a project that allows interaction with the views, Project1_Interactor.
4. Click **Users**, and then click **Add Users**. Select one or more users in the list, select **Actions > Group Membership**, and then select a group for the users. Click **Save** to confirm the group membership.

Repeat this step to add users to other groups.

Step 2: Assign permissions at the project level

After you set up your projects and user groups, you can start assigning permissions. Repeat these steps for each project. Also see [Set Default Permissions for a Project, and its Workbooks and Data Sources](#) on page 364.

1. On the Projects page, select a project, and then select **Actions > Permissions**.
 2. For the **All Users** group permission rule, set the permission role template to **None**.

Click the actions menu (...) next to **All Users**, and then click **Edit**. Select **None** for **Project**, **Workbooks**, and **Data Sources**, and then click **Delete**. This means that all capabilities will be set to **Unspecified**.
 3. Click **Add a user or group rule**, select **Group**, and then select the group name in the list.

To edit an existing rule, click the actions menu (...) next to the permission rule name, and then click **Edit**.
 4. Select a permission role template for **Project**, **Workbooks**, and **Data Sources** to specify a predefined set of capabilities for the group or user.
 5. To further change capabilities included in the rule, click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**.

Click **Save** when you are done.
- Repeat steps 3-5 for each group or user requiring project permissions.

Note: You can optionally lock content permissions to the project to enforce the default permissions for all content in the project. This overwrites any previous permissions

assigned to workbooks and views in the project. For more information, see [Lock Content Permissions to the Project](#) on page 369.

Step 3: Check project permissions

- View the resulting user permissions.

Click a group name or user name in the permission rules list to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

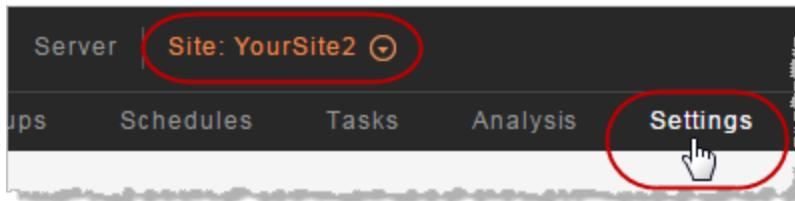
When you publish workbooks to the project, the permissions are updated accordingly.

For information on granting Save permissions to users, see [Grant Web Edit, Save, and Download Permissions](#) on page 377.

Enable Web Authoring

The ability for users to edit views in Tableau Server is a setting that administrators control. In addition to this setting being enabled, user must also have the **Web Edit** capability allowed in their permissions for a given content item.

- In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.



- In a site's Settings page, make sure **Allow users to use web authoring** is selected.



- In the permissions for a workbook or a view, make sure the permission rule for a user or group allows the **Web Edit** capability.

- If your site is already in production, and you want the change to take effect immediately, restart the server.

To confirm which sites allow web authoring, server-level administrators can display the **Sites** page.

Name	Users	Site Admins	Max Users	Storage Used	Max Storage	Status	Metrics	Web Authoring
Default	5	3	1,000	0 B	Server limit	Active		<input checked="" type="checkbox"/>
YourSite1	2,404	2	1,000	36.4 GB	Server limit	Active		<input checked="" type="checkbox"/>
YourSite2	68	2	Server limit	5.6 GB	Server limit	Active		<input checked="" type="checkbox"/>
YourSite3	3,354	3	Server limit	82.3 GB	Server limit	Active		<input checked="" type="checkbox"/>

For more information on web authoring and web editing in Tableau Server, also see these topics:

[Disable web authoring](#)

[The Web Authoring workspace](#)

[Edit a view in Tableau Server](#)

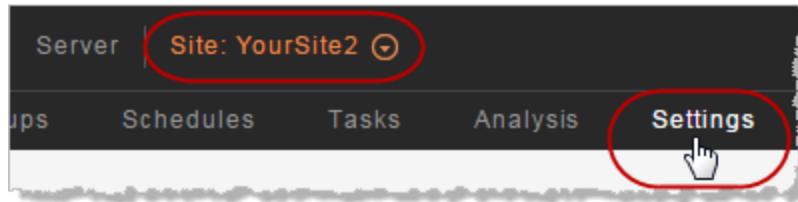
[Grant edit and save permissions](#)

Disable Web Authoring

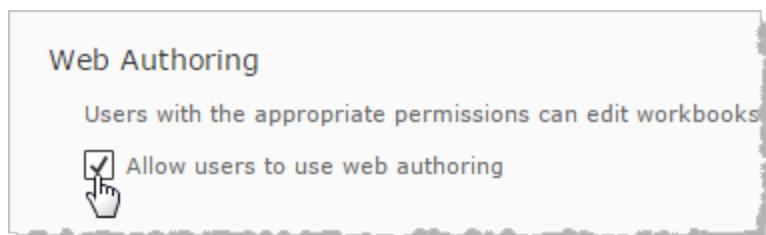
If you want users to be able to view published workbooks on Tableau Server but not access the web editing environment, you can use a site-level setting to disable authoring.

For example, you might have a select group of data analysts who use Tableau Desktop to create and publish workbooks, and a group of sales managers working in the field, who do not use Tableau Desktop but need to access the published dashboards from a web browser.

- In a web browser, sign in to the server as an administrator and go to the site for which you want to disable authoring.
- With Site selected, display the **Settings** page.



3. In the Site Settings page, clear the check box for **Allow users to use web authoring**.



If you disable web authoring while creating a new site, no cached sessions exist, and the setting takes effect immediately.

Otherwise, the change takes effect after server session caching expires or the next time a user signs in after signing out.

Until the change takes effect, users might have authoring access if they see an Edit link on a view, or if they enter the URL for the view's edit mode. For example, they bookmarked the URL while they had the view open for editing.

4. If your site is already in production, and you want the change to take effect immediately, restart the server.

To confirm which sites allow web authoring, server-level administrators can display the **Sites** page.

A screenshot of the 'Sites' page. At the top, there's a navigation bar with tabs: 'Status' (highlighted), 'Sites' (circled in red), 'Users', and 'Schedules'. Below is a table with columns: Name, Users, Site Admins, Max Users, Storage Used, Max Storage, Status, Metrics, and Web Authoring. A red box highlights the 'Web Authoring' column. The table data is as follows:

Embed Views into Webpages

You can embed interactive Tableau views into web pages, blogs, wiki pages, web applications, and intranet portals. Embedded views update as the underlying data changes, or as their workbooks are updated on Tableau Server. Embedded views follow the same licensing and permission restrictions used on Tableau Server. That is, to see a Tableau view that's embedded in a web page, the person accessing the view must also have an account on Tableau Server. As an alternative, if you have a core-based license, you can select **Enable Guest account**, which allows users to load the view without signing in.

You can embed views in the following ways:

- **Use the Share embed code:** The **Share** button at the top of each view provides embed code that you can copy and paste into your webpage.
- **Write your own embed code:** You can enhance the embed code that Tableau provides, or you can build your own code. Either way you can use parameters that control the toolbar, tabs, and more.
- **Use the Tableau JavaScript API:** You can use Tableau JavaScript objects in your own web application code. For more information, see [JavaScript API on page 670](#).

Note: For users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

Writing Embed Code

If you're writing your own embed code, you can take one of two approaches:

- **Use Tableau JavaScript:** This is the preferred approach. Use the embed code that Tableau generates as the starting point for your own code, adding or editing object parameters that control the toolbar, tabs, and more. The default embed code, which relies on a Tableau JavaScript file, is also the only way to control the load order of multiple embedded views.
- **Specify the View URL:** Embed a view using an Iframe or Image tag, where the source is the URL from the **Link** box of the **Share** dialog box. You may want to do this if you can't use JavaScript on your website. There may also be situations when all you can specify is an URL—such as if you're embedding a view using the [SharePoint Page Viewer Web Part](#).

When you embed a view, you should define a width and height that the view will be displayed in. If you do not do this, the client browser will arbitrarily pick a width and height.

Tableau JavaScript

The following code shows an example of embed code that is generated when you click **Share** on a published view. Special characters in the `host_url` parameter are URL encoded, and those in the `site_root` and `name` parameters are notated as HTML numeric character references.

```
<script type='text/javascript' src-
c='http://myserver/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width:800; height:600;'>
<object class='tableauViz' width='800' height='600' style-
e='display:none;'>
  <param name='host_url' value='http%3A%2F%2Fmyserver%2F' />
  <param name='site_root' value='t/Sales' />
  <param name='name' value='MyCoSales/SalesScoreCard' />
</object></div>
```

The source for the `<script>` tag is the URL for the Tableau Server JavaScript file, **viz_v1.js**. The JavaScript file handles assembling the full URL of the view that's displayed for your users. The `name` and `site_root` object parameters are the only required parameters; all other parameters are optional.

View URL as the Source

Here's an example of embedding the same view using an IFrame, where the source is the URL from the **Link** box of the **Share** dialog box:

```
<iframe src-
c="http://myserver/t/Sales/views/MyCoSales/SalesScoreCard
?:embed=yes&:tabs=yes&:toolbar=yes" width="800"
height="600"></iframe>
```

The `embed` URL parameter is required, and you can optionally include parameters that control the toolbar and revert options, among others. You can also add filters to the URL that control the specific data that shows when a view is loaded.

See also

For examples, see the [List of Embed Parameters](#) on the next page and the "Script Tag Examples" in the [Examples](#) on page 395 section.

List of Embed Parameters

You can embed a view using either an Iframe tag, which uses URL parameters, or a JavaScript tag, which uses object parameters. The following table lists both sets of parameters and how to use them.

Object Parameter	URL Parameter	Values	Description	Examples
customViews	:customViews	no	Hides the Remember my changes option.	<param name='customViews' value='no'> http://tabserver/views/DateTime/DateCalcs?:embed=yes&:customViews=no
-	:embed	yes	Required for URL parameter. Hides the top navigation area, making the view blend into your web page better.	http://tabserver/views/DateTime/DateCalcs?:embed=yes
filter	-	string	Customizes what is displayed when the view opens. Filtering by URL parameters is also possible.	<param name='filter' value='Team=Blue'>
-	:format	pdf;	Displays a view as a	http://tabserver/views/Sales/Q2?:format=pdf

Object Parameter	URL Parameter	Val- ues	Descrip- tion	Examples
		png	PDF or .png file.	
-	:high-dpi	false	Renders a view using standard DPI (dots per inch) for high resolution displays and devices.	<code>http://tableau-server/views/Sales/Q2?:highdpi=false</code>
-	:original_view	yes	If the name parameter refers to a workbook or sheet URL (and does not explicitly refer to a custom view) including this parameter displays the view as the original view when other custom views are available.	<code><param name='filter' value-e=':original_view=yes' /></code>
host_	-	strin-	The server	<code><param name='host_url' value-</code>

Object Parameter	URL Parameter	Val- ues	Descrip- tion	Examples
url		g	name as it appears in the URL.	= 'http://myserver.bigco.com/'> <param name="host_url" value="http://localhost/">
link-target	:link-target	string	The target window name for external hyperlinks.	<param name="linktarget" value="_blank"/> http://tabserver/views/DateTime/DateCalcs?:embed=yes&:linktarget=_blank
load-order	-	number	When multiple views are embedded, the default load order is the order in which the views are listed. Use this setting to override that order. Negative numbers are allowed.	<param name="load-order" value="2"/>
name	-	string	Required for object parameter. Workbook and sheet name and optionally,	<param name='name' value='MyCoSales/Sales' /> <param name='name' value="MyCoSales/Sales/jsmith@myco.com/EastCoastSales' />

Object Parameter	URL Parameter	Values	Description	Examples
			<p>a custom view (username@-domain/[custom view name]). If you refer to the Tableau Server URL to confirm the value of name, exclude the session ID (:iid=<n>) at the end of the URL.</p>	
path	-	string	<p>For trusted authentication only, cannot be used with the ticket parameter. Overrides value of the name parameter</p>	<pre data-bbox="736 1220 1396 1389"><param name='path' value='trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4' /></pre> <pre data-bbox="736 1410 1396 1558">http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&:tabs=yes</pre>

Object Parameter	URL Parameter	Values	Description	Examples
			<p>and is used as the URL. See the Trusted Authentication examples.</p>	
-	: record_per- formance	yes	<p>Starts a performance recording for a view. Add this at the end of the URL, immediately before the session ID (</p> <p>:iid= <n>). See Create a Performance Record-ing on page 443 for more information.</p>	<p><code>http://tabserver/#/views/Sales2013/MarginsByAreaCode?:record_performance=yes&:iid=1</code></p>
-	:re- fresh		<p>Re-renders the page. See Refresh</p>	<p><code>http://tabserver/views/DateTime/DateCalcs?:embed=yes&:refresh</code></p>

Object Parameter	URL Parameter	Values	Description	Examples
			Data on page 308 for details.	
	:render	true; false; number	<p>If client-side rendering is enabled (the default), setting to false forces server-side rendering for the session. If client-side rendering is disabled, setting to true enables it for the session. A number can be used to test a complexity threshold. See About Client-Side Rendering on page 437.</p>	<a :render='false"' href="http://tabserver/views/DateTime/DateCalcs?">http://tabserver/views/DateTime/DateCalcs?":render=false

Object Parameter	URL Parameter	Values	Description	Examples
-	:revert	all; filter-s; sorts; axes; shelves	Returns the item to its original state.	<code>http://tabserver/views/DateTime/DateCalc-s?:embed=yes&:revert=all</code>
site_root	-	string	Required. The site name. The Default site value is null (value=''). If your server is multi-site and you want to use trusted authentication, see the Trusted Authentication examples .	<pre><param name='site_root' value='#/Sales'/></pre> <pre><param name='site_root' value=''/></pre>
tabs	:tabs	yes; no	Displays or hides tabs.	<code><param name='tabs' value='yes' /></code>

Object Parameter	URL Parameter	Val- ues	Description	Examples
ticket	-	num-ber	For trusted authentication only, cannot be used with the path object parameter. Must be used with name object to construct the trusted ticket redemption URL. See the Trusted Authentication examples .	<param name='ticket' value='EtDpsm_Ew6rJY-9kRrALjauU'> http://tableauserver/trusted/EtDpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&:tabs=yes
tool-bar	:tool-bar	ye-s; no; top	The tool-bar is displayed by default on the bottom when this parameter is not set. When no the toolbar is excluded from the	<param name='toolbar' value=top'> http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:toolbar=no

Object Parameter	URL Parameter	Values	Description	Examples
			embedded view. When top, the toolbar is placed above the view.	
tool-tip	:tooltip	yes; no	Tooltips are displayed by default in a view when this parameter is not set. If set to no, tooltips are excluded from the embedded view.	<param name='tooltip' value='no'> http://tabserver/views/workbookQ4/SalesQ4?:embed=yes&:tooltip=no

Examples

Here are some examples of ways you can customize or work with your embed code.

Add Filters

You can pass filter values so the view opens showing just the data you want. For example, you may want to include a hyperlink from another part of your web application to an embedded sales performance view that only shows a specific region.

Script Tag Example

```
<script type='text/javascript' src-  
c='http://myserver/javascripts/api/viz_v1.js'>  
</script>  
<object class='tableauViz' width='800' height='600'
```

```

<param name='host_url' value='http://myserver/' />
<param name='site_root' value=' ' />
<param name='name' value='Superstore/Product' />
<param name='filter' value='Region=East' />
</object>
```

To pass through multiple filters, just separate each value with a comma. For example:

```
<param name='filter' value='Region=East,West' />
```

Iframe Tag Examples

```
<iframe src-
c="http://myserver/views/Superstore/Product?:embed=y&Region=East"
width="800" height="600"></iframe>

<iframe src-
="h-
tp://myserver/views/Superstore/Product?:embed=yes&Region=East,West"

width="800px" height="600px"></iframe>
```

For more information, see [Filter on Multiple Fields](#) below.

Filter on Multiple Fields

You can pass filters on as many fields as you want, including fields that are not in the original view.

Script Tag Example

```
<script type='text/javascript' src-
c='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='800' height='600' style-
e='display:none;'>
    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value=' ' />
    <param name='name' value='Superstore/Product' />
    <param name='filter' value='Region=Central, South&Customer Seg-
ment=Consumer, Home Office' />
</object>
```

Iframe Tag Example

```

<iframe src=
="h-
ttp://myserver/views/Superstore/Product?:embed=y&Region=Central,South&Segment=
Office"
width="800" height="600"></iframe>

```

The first box below shows an example of the URL you might get when you click **Share** on a view and copy the link in the **Link** field.

The second box shows how you might modify the URL and add it to an Iframe by deleting the `showShareOptions` and `display_count` parameters, adding filter parameters for Region and Segment, and adding width and height parameters, to create an embed link that displays only Consumer and Home Office products from the Central and South regions.

The screenshot shows two Tableau views side-by-side. The top view is a 'Product Drilldown' showing sales by product category over time. A red box highlights the 'Link' field in the sharing options, which contains the URL: `http://myserver/views/Superstore/Product?:embed=y&Region=Central,South&Segment=Office`. The bottom view shows the same data but with a modified URL: `<iframe src="http://myserver/views/Superstore/Product?:embed=y&Region=Central,South&Segment=Consumer,HomeOffice" width="800" height="600"></iframe>`. This modified URL includes the `width` and `height` attributes for the iframe.

Note: If a filter value contains a special character, such as a comma, replace the character with the URL encoding sequence for \ (backslash, %5c) followed by the URL encoding sequence for the special character. The backslash is needed to escape the special character. For example, the URL encoding sequence for \, (backslash, comma) is %5c%2c.

Filter Dates and Times

If you want to filter on a Date/Time field, include the value using the default Tableau format shown below:

`YYYY-mm-dd hh:mm:ss`

The time part uses a 24-hour clock. Many databases store all date values as Datetime fields, so you may need to pass a time value along with your date.

Script Tag Example

```
<script type='text/javascript' src-
c='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style-
e='display:none;'>
    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value=''/>
    <param name='name' value='Sales/Sales-Performance' />
    <param name='filter' value='Date=2012-12-01' />
</object>
```

This example filters on both a date field and a datetime field:

```
<param name='filter' value='2012-12-01%2022:18:00' />
```

Iframe Tag Example

```
<iframe src="http://myserver/Sales/SalesPer-
formance?:embed=yes&Date=2008-12-01%2022:18:00" width="800"
height="600"></iframe>
```

To filter multiple dates, separate each date with a comma.

Filter Measures

You can filter measures by including one or more values. There is no support for greater than, less than, or ranges. The example below filters to show only \$100 and \$200 sales.

Script Tag Example

```
<script type='text/javascript' src-
c='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='800' height='600' style-
e='display:none;'>
    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value=''/>
    <param name='name' value='Sales/Sales-Performance' />
    <param name='filter' value='Profit=100, 200' />
</object>
```

Iframe Tag Example

```
<iframe src="http://myserver/views/Sales/Sales-Performance?:embed=yes&Profit=100,200"
width="800" height="600"></iframe>
```

Control the Load Order of Multiple Views

You can control the order in which multiple views load for the people working with your views. This feature can only be accessed using embed code that relies on the Tableau JavaScript file.

In the following example, two views are embedded. The second view loads first, followed by the top view. If you embed multiple views and give them all the same load order value, or if you don't specify load order parameters, they are loaded in the order in which they appear on the page.

Script Tag Example

```
<script type='text/javascript' src-
c='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='600' height='400' style-
e='display:none;'>
    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value=' ' />
    <param name='name' value='MyCoSales/TopPerformers' />
    <param name='tabs' value='yes' />
    <param name='toolbar' value='yes' />
    <param name='filter' value='Salesperson=Top 5' />
    <param name='load-order' value='0' />
</object>
<script type='text/javascript' src-
c='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='600' height='400' style-
e='display:none;'>
    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value=' ' />
    <param name='name' value='MyCoSales/SalesScoreCard' />
    <param name='tabs' value='yes' />
    <param name='toolbar' value='yes' />
    <param name='load-order' value=' -1' />
</object>
```

Embed Code for Custom Views

When you embed a view of a workbook or sheet that has custom views available:

- If the embed code URL for the view explicitly refers to a custom view, that custom view will be displayed by default.
- If the embed code URL does not explicitly refer to a custom view, and a Default custom view has been defined, the Default custom view will be displayed in the embedded view by default.
- If no Default custom view has been defined, the original view will be displayed in the embedded view by default.

Note: To ensure the original view will be displayed by default in an embedded view, make sure the embed code URL for the name parameter does not explicitly refer to a custom view, and include the following filter parameter in the embed code: <param name='filter' value=':original_view=yes' />.

In the following example, the embed code will always display the original view of the Profit Analysis sheet in the Profit Analysis workbook, because the filter parameter is set to :original_yes, and the name parameter does not refer to a specific custom view in the URL for the sheet.

```
<script type='text/javascript' src-
c='http://mysite.myserver.com/javascripts/api/viz_
v1.js'></script>
<div class='tableauPlaceholder' style='width: 1496px; height:
749px;'>
<object class='tableauViz' width='1496' height='749' style-
e='display:none;'>
<param name='host_url' value='http://mysite.myserver.com' />
<param name='site_root' value=' ' />
<param name='name' value='ProfitAnalysis/ProfitAnalysis' />
<param name='tabs' value='yes' />
<param name='toolbar' value='yes' />
<param name='filter' value=':original_view=yes' /></ob-
ject></div>
```

In this example, the setting for the name parameter in this example specifically refers to the URL for a custom view named Furniture (in the Profit Analysis sheet in the Profit Analysis workbook).

```
<script type='text/javascript' src-
c='http://mysite.myserver.com/javascripts/api/viz_
v1.js'></script>
<div class='tableauPlaceholder' style='width: 1496px; height:
749px;'>
<object class='tableauViz' width='1496' height='749'
```

```

style='display:none;'>
<param name='host_url' value='http://mysite.myserver.com' />
<param name='site_root' value=' ' />
<param name='name' value-
e='ProfitAnalysis/ProfitAnalysis/Furniture' />
<param name='tabs' value='yes' />
<param name='toolbar' value='yes' /></object></div>

```

In this example, the `name` parameter does not refer to a specific custom view in the URL for the sheet, and the `original_view` parameter has not been specified. The embed code here will display the custom view that has been set to Default in the Profit Analysis sheet in the Profit Analysis workbook. However, if the original view is still the Default (no other custom view has been set to Default), then the original view will be displayed as the default view.

```

<script type='text/javascript' src-
c='http://mysite.myserver.com/javascripts/api/viz_
v1.js'></script>
<div class='tableauPlaceholder' style='width: 1496px; height:
749px;'>
<object class='tableauViz' width='1496' height='749' style-
e='display:none;'>
<param name='host_url' value='http://mysite.myserver.com' />
<param name='site_root' value=' ' />
<param name='name' value='ProfitAnalysis/ProfitAnalysis' />
<param name='tabs' value='yes' />
<param name='toolbar' value='yes' /></object></div>

```

Embed Views into SharePoint (Microsoft SSPI)

You can embed a Tableau Server view in a SharePoint page. To automatically authenticate Tableau Server users who access the embedded view you have two choices, both of which depend on which **user authentication method** was selected during Tableau Server Setup. You can use either **Active Directory** with **Enable automatic logon** to authenticate Tableau Server users (also known as using Microsoft SSPI), or you can use **Local Authentication**—and then also configure Tableau Server for **trusted authentication**.

This topic applies to the first option, where both Tableau Server and SharePoint are using Microsoft SSPI. If your Tableau Server is using **Local Authentication**, see [Embed Views into SharePoint \(Local Authentication\)](#) on page 406 for steps.

Requirements

Licensed users: Anyone who accesses an embedded view must be a licensed user on Tableau Server.

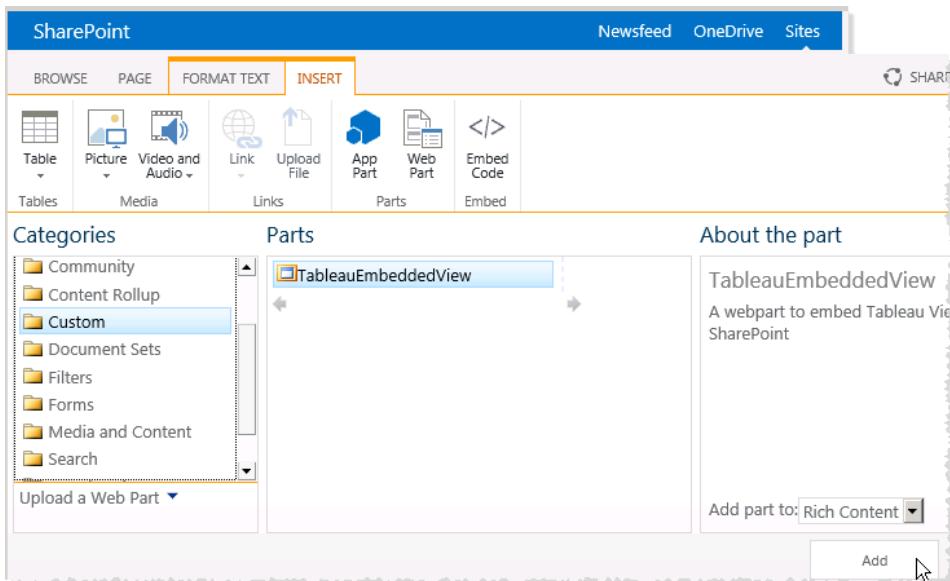
SharePoint version: Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server's security requirements.

TableauEmbeddedView web part: You must have a TableauEmbeddedView web part deployed to your SharePoint server before you can embed Tableau views in a SharePoint page. For sample SharePoint code and instructions for how to create a web part and deploy it to your SharePoint server, see C:\Program Files\Tableau\Tableau Server\<version>\extras\embedding\sharepoint. **Note:** The sample SharePoint code is provided as an example, and may require modification to work in your SharePoint deployment.

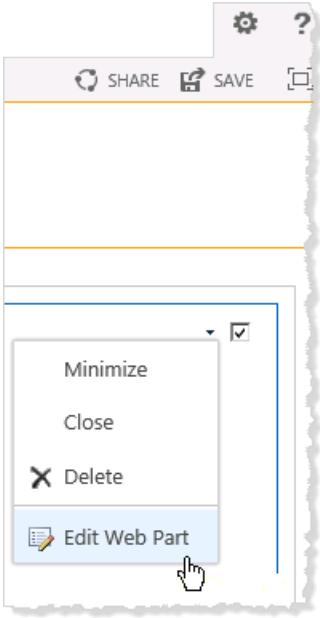
Embedding a View into SharePoint

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.
2. In the section of the page where you want to embed the view, on the **Insert** tab, click **Web Part**.
3. Under Categories, in the **Custom** (or **Miscellaneous**) folder, select **TableauEmbeddedView**, and then click **Add** in the lower-right corner.

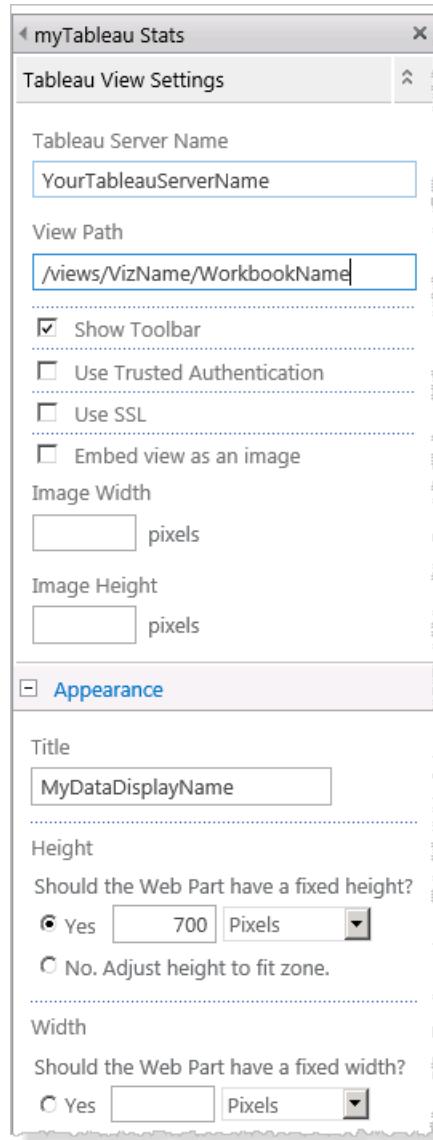


4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select **Edit Web Part**.



5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
 - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
 - In **View Path**, enter the path to the view you want to embed.
 - Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.
 - In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed

height (for example, 700 Pixels) and adjust the width to fit the zone.



6. Click **OK** to apply the changes and exit edit mode.

The view will be embedded into the web part that you just created. Your users will not need to log in to Tableau Server to see the embedded view, rather they will be automatically authenticated using Microsoft SSPI.

Embed Views into Wikis

You can easily embed a view into a wiki or other web page simply by putting the view inside an <iframe> tag.

1. Navigate to the wiki page you want to embed a view into.
2. Edit the page and add an `<iframe>` where the source is the URL from the **Email** box of the **Share View** dialog box. For example:

```
<iframe src="http://myserver/views/Date-Time/DateCalc-
s?:embed=yes&:toolbar=no"
width="800" height="600"></iframe>
```

3. Save your changes.

The view is embedded into the wiki page.

If both Tableau Server and the wiki are configured to use Microsoft SSPI, users accessing an embedded view on the wiki will be automatically signed in so they can see the view.

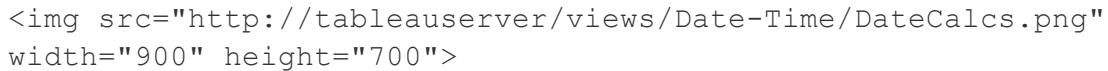
If the server and the wiki are not using the same method for authentication, users will first be asked to sign in to the server before they can see the view.



Embed Images

In addition to embedding a view into a `<script>` or `<iframe>` tag you can also embed the view as an image. When you embed an image the view is not interactive, however, it is updated every time the page fully reloads. That way the image shows the latest data even if the underlying data changes.

1. Navigate to the page where you want to embed the image.
2. Edit the page and add an `` tag where the source is the URL from the **Email** box of the **Share View** dialog box for the view, plus the .png file extension. For example:



Note:

Due to a temporary product limitation, the above approach will only work if the user accessing the embedded image also has an active web browser session with Tableau Server, and is signed in to Tableau Server using Microsoft SSPI.

Embed Views into SharePoint (Local Authentication)

You can embed a Tableau Server view in a SharePoint page. If Tableau Server is using Local Authentication for user authentication, there are some extra steps you need to take before you start embedding views.

This topic describes how to complete the following steps:

- Edit the security permissions for the TableauEmbeddedView.dll file.
- Install and deploy the TableauEmbeddedView.wsp file.
- Verify the web part's deployment.
- Embed a view in SharePoint using the Tableau web part.

Note: If your Tableau Server installation is using Active Directory for user authentication, you can start embedding views right away. For more information, see [Embed Views into SharePoint \(Microsoft SSPI\) on page 401](#).

Requirements

Users: To access an embedded view, users must be licensed Tableau Server users and their user name on SharePoint must be the same as their user name on Tableau Server.

SharePoint version: Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server's security requirements.

Edit Security Permissions for TableauEmbeddedView.dll

Edit the security permissions for TableauEmbeddedView.dll so that all users of the operating system can use it.

1. Locate the TableauEmbeddedView.dll and TableauEmbeddedView.wsp files that install with Tableau Server. If Tableau Server is installed on drive C, the files will be in the following directory:

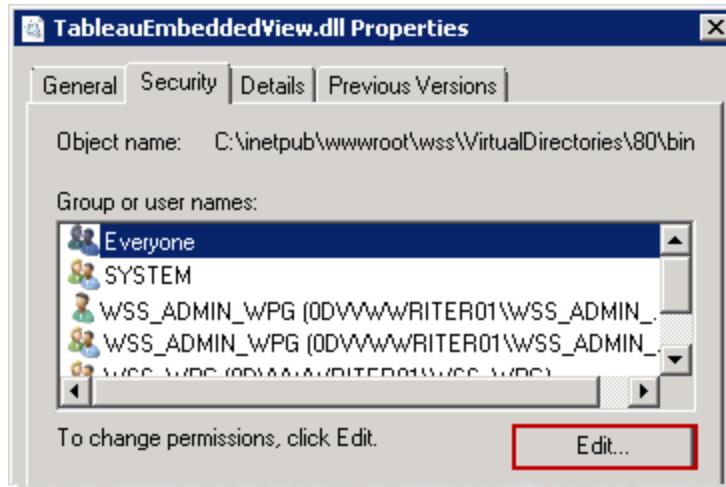
```
C:\Program Files\Tableau\Tableau  
Server\9.2\extras\embedding\sharepoint\
```

2. Copy the files to the root directory of your SharePoint server. The root directory is usually

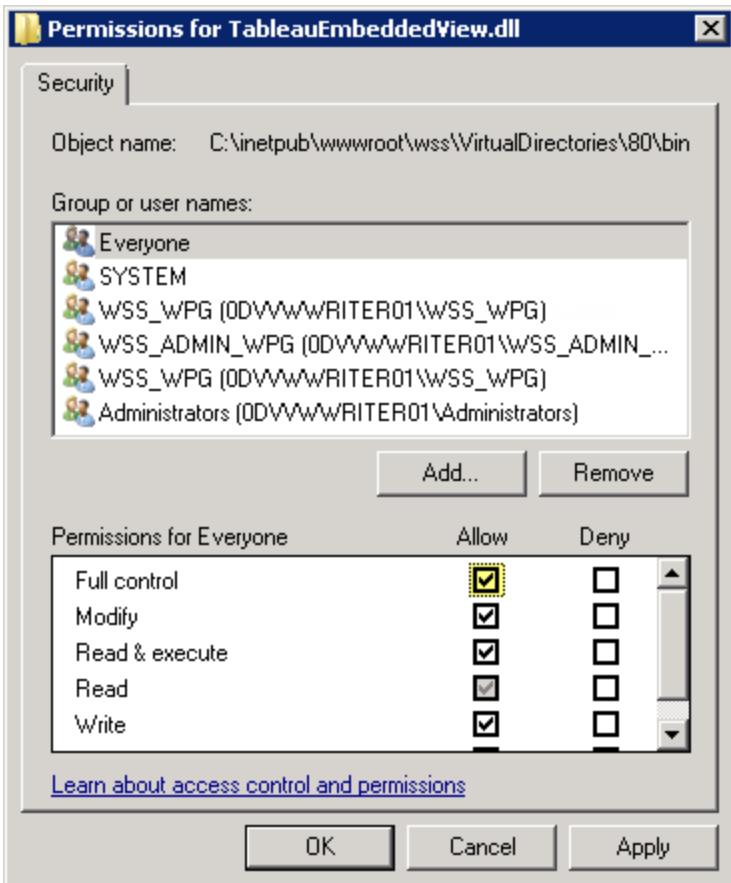
located at C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin,
for example:

C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin

3. To edit the security permissions on TableauEmbeddedView.dll, right-click **TableauEmbedded.dll** and then select **Properties > Security**.
4. Under **Group or user names**, select **Everyone**, and then click **Edit**.



5. Under **Permissions for Everyone**, for the **Full control** permission, select **Allow**.



6. Click **OK**.

Install and Deploy TableauEmbeddedView.wsp

The TableauEmbeddedView.wsp file gives SharePoint more information about what to do with the .dll file. You copied the TableauEmbeddedView.wsp file to the SharePoint root directory in the previous procedure. To install and deploy the .wsp file, follow these steps:

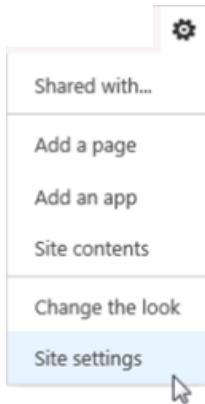
1. Open SharePoint 2013 Management Shell and enter the following command:

```
Add-SPSolution -LiteralPath  
"C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin\TableauEmbe  
ddedView.wsp"
```

2. On the SharePoint Central Administration home page, click **System Settings**.
3. In the **Farm Management** section, click **Manage farm solutions**.
4. On the Solution Management page, click the solution that you want to deploy.
5. On the Solution Properties page, click **Deploy Solution**.
6. On the Deploy Solution page, in the **Deploy When** section, select one of the following

options:

- **Now**
 - **At a specified time.** Specify a time by using the date and time boxes.
7. In the **Deploy To?** section, in the **A specific web application** list, click **All web applications** or select a specific Web application, and then click **OK**.
 8. Open your SharePoint site. Click the settings icon, and then select **Site settings**.

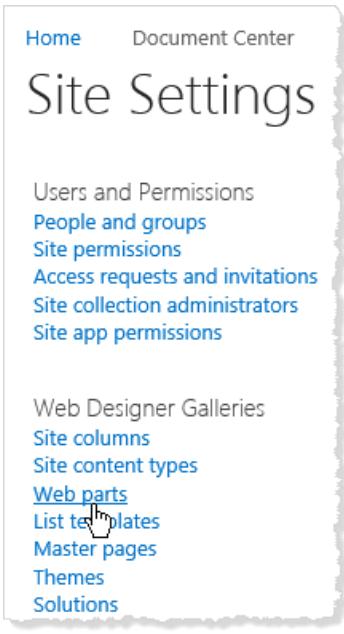


9. Under Site Collection Administration, click **Site collection features**.
10. Scroll to the TableauEmbeddedView feature and then click **Activate** to activate the feature.

Verify the Web Part's Deployment

In the following procedure, you will verify that the Tableau web part is installed.

1. Open your SharePoint site in a web browser.
It may take a few moments for the site to appear.
2. Click the settings icon, and then select **Site settings**.
3. Under **Web Designer Galleries**, click **Web parts**.

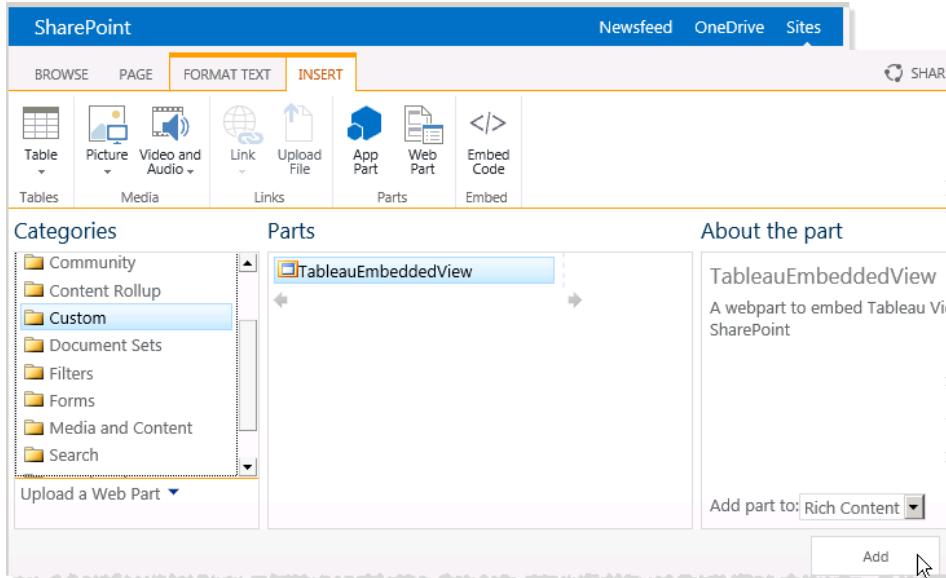


4. Confirm that **TableauEmbeddedView.webpart** is listed.

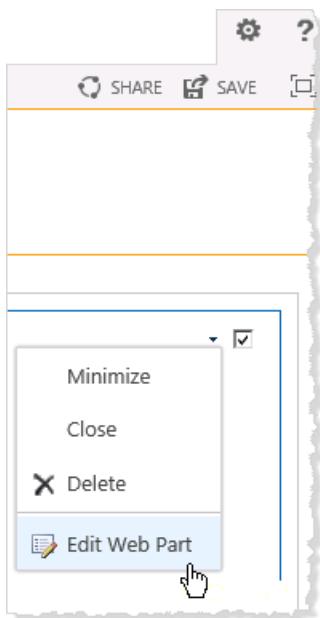
Embed a View Using the Tableau Web Part

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.
2. In the section of the page where you want to embed the view, on the **Insert** tab, click **Web Part**.
3. Under Categories, in the **Custom** (or **Miscellaneous**) folder, select **TableauEmbeddedView**, and then click **Add** in the lower-right corner.

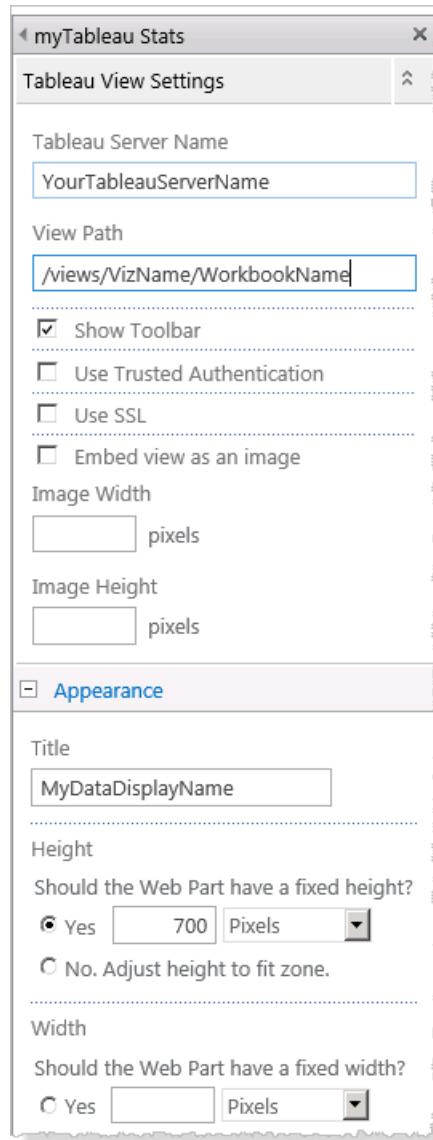


4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select **Edit Web Part**.



5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
 - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
 - In **View Path**, enter the path to the view you want to embed.

- Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.
- In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed height (for example, 700 Pixels) and adjust the width to fit the zone.



6. Click **OK** to apply the changes and exit edit mode.

Now the view is embedded in the page and users who access it will be automatically signed in based on their user name and password for SharePoint.

This is an example of embedding views into SharePoint using the provided .dll file. You can also embed views into other types of web applications. See [JavaScript API on page 670](#) for more information.

Security

There are four main components to security in Tableau Server:

Authorization

Authorization refers to what users are allowed to do with content and on Tableau Server after they have been authenticated. Authorization is determined by a combination of site role and permissions.

Note: Users can also separately be granted permission to work with data. For details, see [Data Security on page 417](#).

Site Roles

When a user is defined in Tableau Server, the user is assigned a role like Server Administrator, Site Administrator, Publisher, Interactor, or Viewer. The site role provides an initial authorization that indicates what the user is allowed to do. For example, server administrators (sometimes referred to as system administrators) can perform operations on all content anywhere on the server, regardless of what permissions are assigned to the content. In contrast, site administrators have unrestricted access to content on a specific site. Publishers can upload content to the server, whereas viewers can see content, but can't publish.

For more information about site roles, see [Site Roles for Users on page 176](#).

Users and Groups

In order to work with content on the server, users must have a user identity on the server. They can then sign in, as described in [Authentication on page 416](#).

Users can be organized into groups, which is useful for authorization. You can configure authorization for groups instead of for individual users. This can be more convenient than configuring authorization for individual users, especially if you want to make wholesale changes to the authorization of a collection of users. It's also convenient because as users change roles in your organization (for example, people leave or change jobs) it's often easier to add and remove users to groups than it would be to manage authorization for individual users.

Note: Any permissions that are granted to individual users take precedence over any permissions that users get from groups that they are in.

For more information, see [Users on page 173](#) and [Groups on page 159](#).

Permissions

Individual resources on the server have permissions that determine who is allowed to see and interact with the resources. Permissions are resource-based: they are attached to projects, workbooks, views, and data sources. The permissions specify which users or groups can work with the resource.

In order for a user to be able to work with a resource, the user or a group that the user belongs to must be explicitly allowed to use that resource. If no **Allowed** permissions are granted to the user or to a group that the user belongs to, the user cannot access the resource. To state it another way, permissions are implicitly denied, and users must explicitly be allowed to access resources.

You can also set a **Denied** permission for a user or group. A **Denied** permission always takes precedence over an **Allowed** permission.

A content owner has permission to work with his or her content. If you publish a workbook to the server, you are that workbook's owner, and you can view the workbook, interact with it, delete it, and so on. If required, an administrator can change ownership of content. For more information, see [Manage Ownership](#) on page 331.

Site Roles and Permissions (Resulting Permissions)

Site roles determine the maximum permissions that a user is allowed. For example, imagine that user Dave has been assigned a site role of Viewer. This role allows users to see content, but not to interact with it. On the server, you select a workbook and grant user Dave permissions to filter and web-edit the workbook, and to save, edit, move, and delete the workbook.

User Dave signs in to the server and views the workbook. However, he does not see any filters on workbook views, and he cannot delete the workbook, even though he has been granted these permissions for the workbook. When Tableau Server evaluates user Dave's permissions, it calculates *resulting permissions* that take into account both the workbook permissions and Dave's site role. In this case, the site role (Viewer) limits Dave's permissions to only viewing the workbook.

On the other hand, users can be granted *fewer* permissions than are set by their site role. For example, user Dave might have a site role of Interactor, which by default lets him filter values when he works with views. However, you can deny user Dave permissions for specific workbooks, such as denying him permission to filter data. In that case, the deny permission for user Dave takes precedence over the capabilities that are set by his Viewer site role, and Tableau won't let user Dave interact with views in that workbook.

Every site has an **All Users** group, and users who are added to the site are automatically added to the **All Users** group. By setting permissions for the **All Users** group, you can establish default permissions for users. For example, you can edit the permissions for a specific workbook and explicitly deny **Share Customized** permissions for the **All Users** group. In that

case, users on the site are not allowed to share customized views unless you add a rule that allows them to do so.

For more information, see [Manage Permissions](#) on page 336.

Authorization for Configuring Tableau Server

One or more users must have permissions to configure Tableau Server and to run `tabadmin` commands. These permissions are not managed through Tableau Server. Users who configure Tableau Server must have administrator permissions on the Windows computer where Tableau Server is running.

Authentication

Authentication establishes a user's identity. Tableau Server has its own user identity and authentication system that lets you determine who can sign in to Tableau Server and who can publish content to the server. This system also allows a personalized user experience for users who access your instance of Tableau Server.

Users sign in to Tableau Server by opening a browser and entering the name or IP address of the server. They are then prompted to enter their username and password:



User identity in Tableau Server

Any user who signs in and works with content in Tableau Server must have a user identity in the Tableau Server repository and must be assigned a site role. User identities can be added to Tableau Server in the server UI, using [tabcmd Commands](#) on page 556, or using the [REST API](#).

If the server is configured to use local authentication, when you add a user identity, you specify a username, a password and a site role. In that case, the Tableau Server repository is used exclusively to authenticate the user.

If the server is configured to use Active Directory authentication, the username and password is managed in Active Directory. In that case, when users sign in to the server, their username and password is verified using Active Directory.

For more information, see [Users](#) on page 173.

Single sign-on options for Tableau Server

Tableau Server supports several types of single sign-on (SSO). With SSO, users don't have to explicitly sign in to Tableau Server. Instead, the credentials they've used to authenticate already (for example, by signing in to your corporate network) are used to authenticate them to Tableau Server, and they can skip the step of entering a username and password to access Tableau Server. With SSO, the user's identity as established externally is mapped to a user identity defined in the Tableau Server repository.

Tableau Server supports these types of SSO:

- **SAML**. You can configure Tableau Server to use SAML (security assertion markup language) for SSO. With SAML, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Tableau Server that provides information about the user's identity. For more information, see [SAML](#) on page 472.
- **Kerberos**. If Kerberos is enabled in your environment and if the server is configured to use Active Directory authentication, you can provide users with access to Tableau Server based on their Windows identities. For more information, see [Kerberos](#) on page 503.
- **Trusted Authentication**. Trusted authentication lets you set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from a trusted web server, it assumes that the web server has already handled whatever authentication is necessary. Tableau Server receives the request with a redeemable token or ticket and presents the user with a personalized view which takes into consideration the user's role and permissions. For more information, see [Trusted Authentication](#) on page 451.

Authentication for the REST API

The REST API lets you manage and change Tableau Server resources programmatically, via HTTP. In order to make requests to the server, you must programmatically sign in to the server. The server sends an authentication token that you then add to subsequent requests. For more information, see [Signing In and Out \(Authentication\)](#) in the REST API documentation.

Data Security

Tableau provides several ways for you to control which users can see which data. For data sources that connect to live databases, you can also control whether users are prompted to provide database credentials when they click a published view. The following three options work together to achieve different results:

- **Database login account:** When you create a data source that connects to a live database, you choose between authenticating to the database through Windows NT or through the database's built-in security mechanism.
- **Authentication mode:** When you publish a data source or a workbook with a live database connection, you can choose an **Authentication mode**. Which modes are available depends on what you choose above.
- **User filters:** You can set filters in a workbook or data source that control which data a person sees in a published view, based on their Tableau Server login account.

The table below outlines some dependencies with the above options:

Database Connection Options		Data Security Questions		
Database login account uses...	Authentication mode	Is database security possible per Tableau Server user?	Are user filters the only way to restrict which data each user sees?	Are web caches shared among users?
Window NT Integrated Security (Windows Authentication)	<i>Server Run As account</i>	No	Yes	Yes
	<i>Impersonate via server Run As account</i>	Yes	No*	No
	<i>Viewer Credentials</i>	Yes	No*	No
Username and Password	<i>Prompt user:</i> Viewers are prompted for their database credentials when they click a view. Credentials can be saved.	Yes	No	No
	<i>Embedded credentials:</i> The workbook or data source publisher can embed their database credentials.	No	Yes	Yes

<i>Database Connection Options</i>		<i>Data Security Questions</i>		
Database login account uses...	Authentication mode	Is database security possible per Tableau Server user?	Are user filters the only way to restrict which data each user sees?	Are web caches shared among users?
	<i>Impersonate via embedded password:</i> Database credentials with impersonate permission are embedded.	Yes	No*	No

* Because it can create unexpected results, Tableau recommends that you not use this authentication mode with user filters.

User filters, the embedded credentials option and the impersonation modes have similar effects—when users click a view, they are not prompted for database credentials and they see only the data that pertains to them. However, user filters are applied in the workbook by authors, and the impersonation authentication modes rely on security policies defined by administrators in the database itself.

Some of the options described above require configuration steps that must happen during Tableau Server Setup or before you publish a workbook or data source. See the following topics for more information:

- [Server Settings \(General\) on page 257](#)
- [Enable Kerberos Delegation on page 510](#)
- [OAuth Connections on page 461](#)
- [Run As User on page 525](#)
- [SQL Server Impersonation on page 534](#)
- [User Filters and Data Source Filters](#) in the Tableau Desktop Help.

Related Topics

[Regenerate a Password for the Tableau Server PostgreSQL Database \(Repository\)](#)

When you install Tableau Server or upgrade from a previous version, the installation process generates a password for Tableau Server to use internally when it accesses the Repository PostgreSQL database. To help with security, the password generated during the installation process is unique to an installation. Because the password is used only by Tableau Server for access to the Repository, the password is not accessible to server administrators or other users.

Tableau Server can also generate an SSL certificate that can be used to protect internal communications to the Repository and other server components. Using SSL for internal communications between processes is optional. For more information, see [Configure Internal SSL](#) on page 494.

Note: If you need access to the Repository (for example, to monitor activity), you can use the administrative views that are built in to the server environment or create your own custom views. For more information, see [Administrative Views](#) on page 289 and [Create Custom Administrative Views](#) on page 300

[Regenerating the password and certificate](#)

If you need to generate a new password and certificate for internal use, you can use the `tabadmin regenerate_internal_tokens` command. For example, if you believe your installation of Tableau Server has been compromised, you should run the `regenerate_internal_tokens` command to generate a new password and SSL certificate.

Note: The SSL certificate is used for internal communication between server components and the PostgreSQL database and is independent of any SSL certificate that you might have on the server to use for HTTPS communication between the server and clients that connect to Tableau Server.

To manually regenerate a password and SSL certificate:

1. On the Tableau Server computer, open a command prompt as an administrator and navigate to <install directory>\Program Files\Tableau\Tableau Server\9.0\bin.
2. Enter the following:

```
tabadmin stop  
tabadmin regenerate_internal_tokens
```

```
tabadmin config  
tabadmin start
```

See [regenerate_internal_tokens](#) on page 607 for more information, including optional switches to specify regeneration of password or certificate.

Network Security

There are three main network interfaces in Tableau Server:

- **Client to Tableau Server:** The client can be a web browser, Tableau Desktop, or the [tabcmd](#) on page 552 utility.
- **Tableau Server to your database(s):** To refresh data extracts or handle live database connections, Tableau Server needs to communicate with your database(s).
- **Server component communication:** This applies to distributed deployments only.

Client to Tableau Server

A Tableau Server client can be a web browser, Tableau Desktop, or [tabcmd](#) commands. Communications between Tableau Server and its clients use standard HTTP requests and responses. Tableau Server can also be configured for HTTPS (see [Configure External SSL](#) on page 491). When Tableau Server is configured for SSL, all content and communications between clients are encrypted using SSL, and the HTTPS protocol is used for requests and responses.

Passwords are communicated from browsers and tabcmd to Tableau Server using 512-bit public/private key encryption. Tableau Server sends a public key to the browser, which uses the key to encrypt the password for transmission. Each encrypted transmission uses a key one time before it is discarded. This means that passwords are always secured regardless of the use of SSL. If SSL is enabled, SSL encryption is used in addition to the 512-bit public key encryption of passwords.

Note: The HTTP OPTIONS method for Tableau Server is a configurable setting. The OPTIONS method is disabled by default, which means that the server returns an HTTP 405 (Method Not Allowed) response for the HTTP OPTION. To enable the HTTP OPTIONS method, use the following [tabadmin](#) command:

```
tabadmin set wgserver.restrict_options_method false
```

Clickjack Protection

By default, Tableau Server has *clickjack protection* enabled. This helps prevent certain types of attacks in which the attacker overlays a transparent version of a page on top of an innocuous-looking page in order to lure a user into clicking links or entering information. With clickjack

protection enabled, Tableau Server imposes certain restrictions on embedding views. For more information, see [Clickjack Protection](#) below.

Tableau Server to your database

Tableau Server makes dynamic connections to databases to process result sets and refresh extracts. It uses native drivers to connect to databases whenever possible and relies on a generic ODBC adapter when native drivers are unavailable. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Tableau.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server's internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Communication with the repository

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. For more information, see [Configure Internal SSL](#) on page 494.

Server component communication in a cluster

There are two aspects to communication between Tableau Server components in a distributed server installation: trust and transmission. Each server in a Tableau cluster uses a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. Computers in the cluster running a gateway process accept requests from third parties (clients), unless they are fronted by a load balancer, in which case the load balancer receives the requests. Servers not running a gateway process only accept requests from other trusted members of the cluster. Trust is established by a whitelist of IP address, port, and protocol. If any of these are invalid, the request is ignored. All members of the cluster can communicate with each other.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server's internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Clickjack Protection

Tableau Server includes protection against clickjack attacks. *Clickjacking* is a type of attack against web pages in which the attacker tries to lure users into clicking or entering content by

displaying the page to attack in a transparent layer over an unrelated page. In the context of Tableau Server, an attacker might try to use a clickjack attack to capture user credentials or to get an authenticated user to change settings on your server. For more information about clickjack attacks, see [Clickjacking](#) on the Open Web Application Security Project website.

Note: Clickjack protection was available in previous versions of Tableau Server, but was disabled by default. New installations of Tableau Server 9.1 and later will always have clickjack protection on unless you explicitly disable it.

Effects of clickjack protection

When clickjack protection is enabled on Tableau Server, the behavior of pages loaded from Tableau Server changes in the following ways:

- Tableau Server adds the `X-Frame-Options: SAMEORIGIN` header to certain responses from the server. In the current versions of most browsers, this header prevents the content from being loaded into an `<iframe>` element, which helps prevent clickjacking attacks.
- The top-level page from Tableau Server cannot be loaded in `<iframe>` elements. This includes the sign-in page. One consequence is that you cannot host Tableau Server pages in an application that you create.
- Only views can be embedded.
- If an embedded view requires data source credentials, a message is displayed in the `<iframe>` element with a link to open the view in a secure window where the user can safely enter credentials. Users should always verify the address of the opened window before entering credentials.
- Views can be loaded only if they include the `:embed=y` parameter in the query string, as in this example:

```
http://<server>/views/Sales/CommissionModel?:embed=y
```

Note: View URLs that contain a hash mark (#) after the server name (for example, `http://<server>/#/views/Sales/CommissionModel?:embed=y`) are blocked when clickjack protection is enabled.

Disabling clickjack protection

You should leave clickjack protection enabled unless it is affecting how your users work with Tableau Server. If you want to disable clickjack protection, use the following `tabadmin` commands:

1. tabadmin stop
2. tabadmin set wgserver.clickjack_defense.enabled false
3. tabadmin config
4. tabadmin start

OAuth Connections

For Google BigQuery, Google Analytics, and Salesforce.com data sources, an alternative to storing sensitive database credentials with Tableau Server is to create connections using the **OAuth 2.0** standard.

When you create an OAuth connection, you give the data provider your approval for Tableau to access your data. The data provider then sends Tableau an **access token** that uniquely identifies requests from Tableau. For more information, see [Overview of the OAuth process below](#) below.

Using OAuth connections provides the following benefits:

- **Security:** Your database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau.
- **Convenience:** Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider.

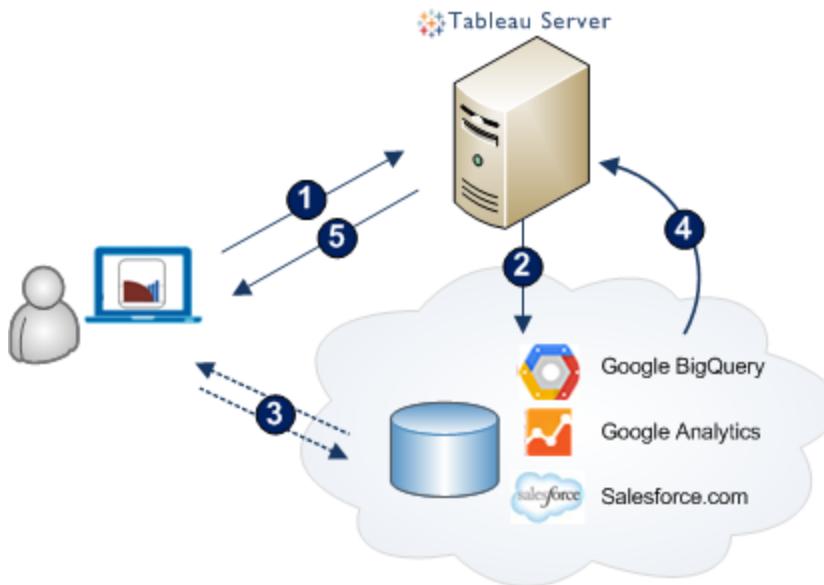
In addition, for live connections to Google BigQuery data, each workbook viewer can have a unique access token that identifies the user, rather than sharing a single user name and password credential.

Overview of the OAuth process

The following steps describe a workflow in the Tableau environment that calls the OAuth process.

1. You take an action that requires access to a cloud data source.
For example, you open a workbook that's published to Tableau Server.
2. Tableau directs you to the hosted data provider's sign-in page. The information that is sent to the hosted data provider identifies Tableau as the requesting site.
3. When you sign in to the hosted data source, it prompts you to confirm your authorization for Tableau Server access to the data.
4. Upon your confirmation, the data-source provider sends an access token back to Tableau Server.

5. Tableau Server presents your workbook and data to you.



The following other workflows can also use the OAuth process:

- Creating a workbook and connecting to the data source from Tableau Desktop or from Tableau Server.
- Publishing a data source from Tableau Desktop.
- Signing in to Tableau Server from an approved *client*, such as Tableau Mobile or Tableau Desktop.

Access tokens for data connections

You can save access tokens with data connections to enable direct access to the data after the initial authentication process. An access token is valid until a Tableau Server user deletes it, or the data provider revokes it.

It is possible to exceed the number of access tokens your data source provider allows. If that's the case, when a user creates a new token, the data provider uses length of time since last access to decide which token to invalidate to make room for the new one.

Access tokens for authentication from approved clients

By default, Tableau Server sites allow users to access their sites directly from approved Tableau clients, after users provide their credentials the first time they sign in. This type of authentication also uses OAuth access tokens to store the users' credentials securely.

For more information, see [Authentication for Connected Devices](#) on page 259

Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)

When you install Tableau Server or upgrade from a previous version, the installation process generates a password for Tableau Server to use internally when it accesses the Repository PostgreSQL database. To help with security, the password generated during the installation process is unique to an installation. Because the password is used only by Tableau Server for access to the Repository, the password is not accessible to server administrators or other users.

Tableau Server can also generate an SSL certificate that can be used to protect internal communications to the Repository and other server components. Using SSL for internal communications between processes is optional. For more information, see [Configure Internal SSL on page 494](#).

Note: If you need access to the Repository (for example, to monitor activity), you can use the administrative views that are built in to the server environment or create your own custom views. For more information, see [Administrative Views on page 289](#) and [Create Custom Administrative Views on page 300](#)

Regenerating the password and certificate

If you need to generate a new password and certificate for internal use, you can use the `tabadmin regenerate_internal_tokens` command. For example, if you believe your installation of Tableau Server has been compromised, you should run the `regenerate_internal_tokens` command to generate a new password and SSL certificate.

Note: The SSL certificate is used for internal communication between server components and the PostgreSQL database and is independent of any SSL certificate that you might have on the server to use for HTTPS communication between the server and clients that connect to Tableau Server.

To manually regenerate a password and SSL certificate:

1. On the Tableau Server computer, open a command prompt as an administrator and navigate to `<install directory>\Program Files\Tableau\Tableau Server\9.0\bin`.
2. Enter the following:

```
tabadmin stop  
tabadmin regenerate_internal_tokens  
tabadmin config  
tabadmin start
```

See [**regenerate_internal_tokens**](#) on page 607 for more information, including optional switches to specify regeneration of password or certificate.

Performance

Every server environment is unique, and there are many variables that can impact performance.

Variables affecting performance include:

- Hardware details, such as disk speed, memory, and processor cores
- The number of servers in your deployment
- Network traffic
- Usage factors such as workbook complexity, concurrent user activity, and data caching
- Tableau Server configuration settings, such as how many of each server process you're running
- Data considerations—such as data volume, database type, and database configuration.

Tableau Server is highly configurable to help you address these variables in your server environment and fine tune server performance. Because of this complexity, there is no single formula for improving server performance. However, there are some basic guidelines you can follow.

See the following topics for more information:

General Performance Guidelines

Hardware and Software

Use a 64-bit operating system and the 64-bit product: Although Tableau Server runs well on 32-bit Microsoft operating systems, for the best performance, choose a 64-bit operating system and install the 64-bit version of Tableau Server.

Add more cores and memory: Regardless of whether you're running Tableau Server on one computer or several, the general rule is that more CPU cores and more RAM will give you better performance. Make sure you meet Tableau Server's recommended [hardware and software requirements](#) and see the topic [When to Add Workers and Reconfigure](#) on the next page to assess whether you should add additional machines.

If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host.

Configuration

Schedule refreshes for off-peak hours: Backup tasks tend to stall other background tasks until the backup is completed. Use the [Background Tasks for Extracts](#) on page 295

administrative view to see your refresh and backup task schedules. Your refresh tasks should be scheduled for off-peak hours that don't overlap with your backup window.

Look at caching: Caching helps Tableau Server respond to client requests quickly, especially for views that connect to live databases. Confirm that **Refresh Less Often** on the [Data Connections tab](#) of the Configuration dialog box is selected.

Consider changing two session memory settings:

- **VizQL session timeout limit:** The default VizQL session timeout limit is 30 minutes. Even if a VizQL session is idle, it is still consuming memory and CPU cycles. If you can make do with a lower limit, use [tabadmin on page 583](#) to change the `vizqlserv-er.session.expiry.timeout` setting .
- **VizQL clear session:** By default, VizQL sessions are kept in memory even when a user navigates away from a view. This consumes a good deal of session memory. Instead, you can end sessions when users move away from a view by changing the value of the `vizqlserver.clear_session_on_unload` setting to `true` (default is `false`).

Assess your process configuration: Tableau Server is divided into six different components called server processes. While their default configuration is designed to work for a broad range of scenarios, you can also reconfigure them to achieve different performance goals. Specifically, you can control on which computers the processes run and how many are run. See [Improve Server Performance on the next page](#) for general guidelines for one-, two-, and three-node deployments.

When to Add Workers and Reconfigure

Tableau Server can scale up and out as your needs and requirements evolve. Here are some guidelines to help you figure out whether it's time to add more worker nodes to your system, reconfigure the server, or both:

- **More than 100 concurrent users:** If your deployment is user-intensive (>100 simultaneous viewers), it's important to have enough VizQL processes—but not so many that they exceed your hardware's capacity to handle them. Also, enabling the Tableau Server [Guest User account](#) can increase the number of potential simultaneous viewers beyond the user list you may think you have. The administrative view can help you gauge this. For more information, see [Actions by Specific User on page 293](#). For tips on how to configure or scale your deployment, see [Improve Server Performance on the next page](#).
- **Heavy use of extracts:** Extracts can consume a lot of memory and CPU resources. There's no one measurement that qualifies a site as extract-intensive. Having just a few, extremely large extracts could put your site in this category, as would having very many small extracts. Extract heavy sites benefit from isolating the data engine process on its own machine. For general guidelines, see [Improve Server Performance on the next page](#).

- **Frequent extract refreshes:** Refreshing an extract is a CPU-intensive task. Sites where extracts are frequently refreshed (for example, several times a day) are often helped by more emphasis on the background process, which handles refresh tasks. Use the [Background Tasks for Extracts on page 295](#) administrative view to see your current refresh rate. See [Improve Server Performance below](#) for details on how to scale.
- **Troubleshooting performance:** If views are slow to load or server performance is generally slow, there could be several causes. For more information, see [General Performance Guidelines on page 428](#).
- **Downtime potential:** If your server system is considered mission critical and requires a high level of availability, you can configure it so there's redundancy for the server processes that handle extracts, the repository, and the gateway. For more information, see [High Availability on page 82](#).

Note: To install Tableau Server on multiple nodes, you must have a Tableau Server—Multi-Machine Core license.

Improve Server Performance

Use the topics below for guidance on how to improve the performance of deployments that are extract-intensive, user-intensive, or both:

What's your goal?

One-
Machine
Example:
Extracts

How Many Processes to Run

Two-
Machine
Example:
Extracts

Where to Configure Processes

Two-
Machine
Example:
Viewing

Optimizing the Extracts and Workbooks

Three
Machine
Example:
Extracts
& View-

ing

[Assessing view responsiveness](#) on the next page

What's your goal?

Optimizing for extracts

The data engine process stores extracts and answers queries; the background process refreshes extracts. Because both are demanding of CPU resources, the best approach to improving performance for an extract-intensive deployment is to isolate these two processes from one another, and from the other server processes. This may take three machines. If you don't have three machines to work with, there are still strategies you can use.

Optimizing for users and viewing

The VizQL server process handles loading and rendering views for Tableau Server users. If you are trying to optimize your deployment for a high number of users and a lot of view interaction, this is the process you should focus on.

How Many Processes to Run

This topic assumes that you are running the 64-bit version of Tableau Server on a 64-bit operating system, on a computer with 8 cores and 16 GB of RAM.

In this configuration, two instances of each process should meet your needs. If your machine has just 4 cores or only meets the minimum RAM requirement for Tableau Server, which is 8 GB, your limit can be one instance of each process.

While the minimum installation requirement is 4 cores and 8 GB of RAM, we do not recommend load or scale testing a single node server using a 4-core machine. A single 4-core server is typically for small trials and prototyping. Large enterprise deployments should consider using 16-core servers for each node.

Background Process

A single background process can consume 100% of a single CPU core, and sometimes even more for certain tasks. As a result of this, the total number of instances you should run depends on the machine's available cores—as well as on what you're trying to improve. The deployment examples below use N to represent the machine's total number of cores, and each suggests a different strategy where the background process is concerned. When in doubt, start with the low end of the suggested range and assess performance before increasing the number.

Data Engine and Repository Processes

There are scenarios where the data engine process should be isolated on its own node—such as if you are trying to improve an extract-intensive deployment and you want to emphasize

querying more than extract refreshes. The deployment examples below provide specifics. Because the data engine stores real-time data, transferring it is a multi-phased procedure. For more information, see [Move the Data Engine and File Store Processes](#) on page 66

Another reason to isolate the data engine (and/or the repository) is to minimize your deployment's potential for downtime. Unless you're configuring for high availability, the repository can usually remain on the primary Tableau Server. For more information, see [High Availability](#) on page 82.

Where to Configure Processes

You configure the type and number of processes any machine is running using the Tableau Server Configuration dialog box. If you are adding new machines as part of your reconfiguration, they must already have Tableau Worker software installed on them. For more information, see [Install and Configure Worker Nodes](#) on page 76.

If you are reconfiguring the processes on your primary or standalone Tableau Server, see [Reconfigure Processes](#) on page 36.

Optimizing extracts and workbooks

Fast server performance with extracts is partly a function of the extracts and workbooks themselves. Workbook authors can help improve server performance by keeping the extract's data set short, through filtering or aggregating, and narrow, by hiding unused fields. Use the Tableau Desktop options **Hide All Unused Fields** and **Aggregate data for visible dimensions** to do this. For steps, see [Creating an Extract](#) (Tableau Desktop help). For general tips on building well-performing workbooks, search for "performance" in the Tableau Desktop help. To see how workbooks perform after they've been published to Tableau Server you can create a performance recording. For more information, see [Create a Performance Recording](#) on page 443.

Assessing view responsiveness

When a user opens a view, the components of the view are first retrieved and interpreted, then displayed in the user's web browser. For most views, the display rendering phase occurs in the user's web browser and in most cases, this yields the fastest results and highest level of interactive responsiveness. Handling most interactions in the client web browser reduces bandwidth and eliminates round-trip request latencies. If a view is very complex, Tableau Server handles the rendering phase on the server instead of in the client web browser—because that generally results in the best performance. If you find that views aren't as responsive as you'd like, you can test and change the threshold that causes views to be rendered by the server instead of in the client web browser. For more information, see [About Client-Side Rendering](#) on page 437.

One-Machine example: Optimized for heavy extract usage

This example shows a 64-bit, 8+ core, 16+ GB system configured for heavy extract usage.

This configuration would look like the following Process Status table on the Server Status page.

Server Status	
Process Status	
The real-time status of processes running in Tableau Server.	
Process	10.32.139.21
Gateway	✓
Application Server	✓
API Server	✓
VizQL Server	✓✓
Cache Server	✓✓
Search & Browse	✓
Backgrounder	✓✓
Data Server	✓✓
Data Engine	✓
File Store	✓
Repository	✓
Refresh Status	<input checked="" type="button"/> Active <input type="button"/> Busy <input checked="" type="button"/> Passive <input type="button"/> Unlicensed <input checked="" type="button"/> Down <input type="button"/> Status unavailable

Configuration Notes:

- The primary server runs 2 VizQL server processes, 2 cache server processes, and 2 data server processes. These are the recommended defaults from installation.
- As a general rule, run a cache server process for every VizQL server process on the node.
- Calculate the least number of background processes to run by taking the machine's total number of cores and divide it by 4. To determine the maximum number, divide by 2.
- Both the background and data engine processes are CPU-intensive and the configuration shown above balances them.
- Schedule extract refreshes for off-peak times to help the VizQL server, application server, data engine, and background processes to not compete with one another for system resources.

Two-Machine example: Optimized for heavy extract usage

This example shows the possible configuration for a two-machine Tableau Server deployment that handles heavy extract usage. Both machines are 64-bit, 8+ core, 16+ GB systems.

Note that the VizQL server, application server, data server, and data engine processes are isolated from the background processes.

With this configuration, the Server Status page would look like this:

Server Status		
Process Status		
The real-time status of processes running in Tableau Server.		
Process	Primary 10.32.139.21	Worker 1 10.32.139.22
Cluster Controller	✓	✓
Gateway	✓	✓
Application Server	✓	
API Server	✓	
VizQL Server	✓ ✓	
Cache Server	✓ ✓	✓ ✓
Search & Browse	✓	
Backgrounder		✓ ✓ ✓ ✓
Data Server	✓ ✓	
Data Engine	✓	
File Store	✓	
Repository	✓	
<input type="button" value="Refresh Status"/> <input checked="" type="checkbox"/> Active <input type="checkbox"/> Busy <input checked="" type="checkbox"/> Passive <input type="checkbox"/> Unlicensed <input checked="" type="checkbox"/> Down <input type="checkbox"/> Status unavailable		

Configuration Notes:

- The primary server runs 2 VizQL server processes, 2 cache server processes, and 2 data server processes. These are the recommended defaults from installation.
- As a general rule, run a cache server process for every VizQL server process on the node.
- Isolate the background processes on the worker. To figure out the minimum number of background processes to run, take the machine's total number of cores and divide it by 4. For the maximum number, divide by 2.
- Isolate the backgrounder processes from the VizQL server, application server, data server, and data engine processes.
- Adding cache servers on the worker node with backgrounders can make cache requests on behalf of users or jobs.

Two-Machine example: Optimized for viewing extracts

This example shows a two-machine deployment with light extract usage and heavier viewing. Both machines are 64-bit, 8+ core, 16+ GB systems.

The Process Status table for this configuration would look like this:

Server Status		
Process Status		
Process	Primary 10.32.139.21	Worker 1 10.32.139.22
Cluster Controller	✓	✓
Gateway	✓	✓
Application Server	✓	
API Server	✓	
VizQL Server	✓✓	
Cache Server	✓✓	✓✓
Search & Browse	✓	
Backgrounder		✓✓✓✓
Data Server	✓✓	
Data Engine	✓	✓
File Store	✓	✓
Repository	✓	

 Active
 Busy
 Passive
 Unlicensed
 Down
 Status unavailable

Configuration Notes:

- The primary server runs 2 VizQL server processes, 2 cache server processes, and 2 data server processes. These are the recommended defaults from installation.
- As a general rule, run a cache server process for every VizQL server process on the node.
- A minimum of 2 background processes should be run on the worker. The maximum number you should run is equal to the machine's total number of cores.
- Run the data engine process on both nodes to split view requests between the two nodes. In a deployment where extracts are refreshed infrequently, the data engine and background processes can be on the same node.
- If extract refresh jobs will be only run during off hours, you can add more background processes on each node to maximize their parallelism.
- Adding cache servers on the worker node with backgrounders can make cache requests on behalf of users or jobs.
- The number of nodes in the cluster is determined by the total number of cores and main memory available across all nodes.

Three-Machine example: Optimized for using and viewing extracts, and a high number of concurrent users

A three-machine configuration is the recommended minimum number of machines to achieve the best performance if you have both a high amount of extract refreshing and usage, and a high number of concurrent users. In this example, all machines are assumed to be 64-bit, 16 core, 16+ GB systems.

The Process Status table for this configuration would look like this:

Process Status			
Process	Primary 10.32.139.21	Worker 1 10.32.139.22	Worker 2 10.32.139.30
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server	✓	✓	
API Server	✓	✓	
VizQL Server	✓✓	✓✓	
Cache Server	✓✓	✓✓	✓✓
Search & Browse	✓	✓	
Backgrounder			✓✓✓✓
Data Server	✓✓	✓✓	
Data Engine	✓	✓	
File Store	✓	✓	
Repository	✓	✓	
Refresh Status		✓ Active ↻ Busy ✗ Passive ⚠ Unlicensed ✗ Down ◻ Status unavailable	

Configuration Notes:

- Run 2 VizQL server processes, 2 cache server processes, and 2 data server processes on the nodes that are not running the background processes. These are the recommended defaults from installation.
- As a general rule, run a cache server process for every VizQL server process.
- For this configuration, 16 cores are recommended for each node.
- The background processes are on their own machine so that their work does not compete with that of the other processes. Because the machine is dedicated to background processes and they can consume 100% of the CPU resources, the low end of the suggested range equals the total number of cores. Depending on the size of the data being refreshed, it's possible for some deployments to run up to twice as many background processes than cores and still obtain parallel speed-up.

- Run the data engine process on the primary and the worker that is not running background processes to split view requests between the two nodes.
- The user loads for the application server and data server processes can typically be handled by 1 process each but they can be set to 2 to provide redundancy.
- Under most conditions, the primary Tableau Server and the data engine will not be a bottleneck for the system's overall throughput as long as sufficient CPU cycles exist for them. To increase viewing capacity, add nodes dedicated to the VizQL server process. To increase capacity for refreshing extracts, add nodes dedicated to the background process.
- Adding cache servers on the worker node with backgrounders can make cache requests on behalf of users or jobs.

About Client-Side Rendering

Before a view's marks and data are displayed in a client web browser, they are retrieved, interpreted, and rendered. Tableau Server can perform this process in the client web browser or on the server. Client-side rendering is the default mode because handling the rendering and all interaction on the server can result in more network data transfer and round-trip delays. With client-side rendering, most view interactions are faster, because they are interpreted and rendered right there in the client.

Some views, however, are more efficiently rendered on the server where there's more computing power. Server-side rendering makes sense for a view that is complex to the extent that image files take up significantly less bandwidth than the data used to create the images. Also, because tablets usually have much slower performance than PCs, they can handle less view complexity. There are cases where a view opened from a PC's web browser might be client-rendered but the same view opened from a tablet's web browser is server-rendered.

Tableau Server is configured to automatically handle all of these situations using [The Threshold Calculation on the next page](#) as the trigger for rendering a view on the server instead of in the web browser. As the administrator, you can test or fine tune this setting for both PCs and tablets. See the topics below for more information.

Requirements

- **Supported browsers:** Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 <canvas> element, which is used by client-side rendering.
- **Polygons and the page history feature:** If a view uses polygons or the page history feature, server-side rendering is performed, even if client-side rendering is otherwise enabled.

The Threshold Calculation

When client-side rendering is enabled, Tableau Server uses a calculation to determine the view's complexity. If the complexity value exceeds 100 (for PC browsers) or 20 (for tablet browsers), the view is rendered on the server instead of in the web browser. Here's the calculation:

```
view complexity = (# of marks) + 3(# of headers) + 3(# of annotations) + 3(# of reference lines) + 6(# of unique custom shapes)
```

For example, if you have a view with 2,000 marks, 150 headers (you can sometimes determine this by adding the number of rows and columns in a view), 1 annotation, and 1 reference line, your equation would be:

$$2,000 + 3(150) + 3(1) + 3(1) = 2,456$$

Now take the current threshold value and divide it by 100, then multiply it by 5,000 (dividing the threshold by 100 is a normalization and multiplying by 5,000 is Tableau's scaling factor).

Assuming a current threshold value of 100, the equation would be as follows:

$$100/100 * 5,000 = 5,000$$

Compare the two sums. Knowing that 5,000 represents a complexity of 100, you can see that 2,456 represents about half the complexity (49). Therefore, to force server-side rendering for this particular view on a PC browser, you would need to set that threshold to 48. Keep in mind that interactions such as filtering may change the complexity of the view, and a session may switch rendering modes whenever the view's complexity changes.

Note: "Unique custom shapes" represents different images. You can have 2000 marks with one unique custom shape, and that part of the calculation would be $6 * 1$. With 500 marks, each with a different custom shape, that part of the calculation would be $6 * 500$.

See the topics below for details on how to test and configure client-side rendering.

Test with the URL Parameter

Tableau Server is configured to perform client-side rendering by default, as long as the requirements are met. To test server-side rendering on a session basis, type `? :render=false` at the end of the view's URL. For example:

`http://localhost/views/Supplies/MyView? :render=false`

If client-side rendering is disabled on Tableau Server, enter `? :render=true` to enable it for the session:

`http://localhost/views/Supplies/MyView? :render=true`

You can also test particular complexity thresholds on individual views to see if it's appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server's default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=80
```

Configure with the tabadmin set Options

You can use the tabadmin options `vizqlserver.browser.render` to disable or enable client-side rendering and `vizqlserver.browser.render_threshold` and `vizqlserver.browser.render_threshold_mobile` to change the thresholds for client-side rendering. See [tabadmin set options](#) on page 616 for details.

Tableau Server Processes

There are Tableau Server processes whose default configuration you can change to achieve different results. The topics [Improve Server Performance](#) on page 430 and [High Availability](#) on page 82 describe some of the approaches you can take. High-level status for each process is displayed on the server's Status page and more detailed information related to some of the processes—such as the background process—is in the [Administrative Views](#) on page 289 topic.

Note: Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node. File store is installed when you install data engine and cannot be installed separately. Every instance of a data engine process will always have one instance of the file store process present as well.

Architecturally, the 64-bit version of Tableau Server uses native, 64-bit processes; the 32-bit version of Tableau Server uses 32-bit processes. The exception is the data engine. If the 32-bit version of Tableau Server is installed on a 64-bit operating system, the 64-bit version of the data engine process is used.

For information on log files generated by these processes, see [Server Log File Locations](#) on page 645.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
API Server	wgserver.exe	Handles REST API calls	Yes	Unless you are using REST APIs for critical business processes, this service can be down without impacting the overall health of Tableau Server.
Application Server	vizportal.exe	Handles the web application, supports browsing and searching	Yes	Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.
Background Worker	backgrounder.exe	Executes server tasks, including extract refreshes, 'Run Now' tasks, and tasks initiated from tabcmd	No	A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.
Cache Server	redis-server.exe	Query cache	No	A query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and API server and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
Cluster Controller	clustercontroller.exe	Responsible for monitoring various components, detecting failures, and executing failover when needed	n/a	Included in the base install on every node.
Coordination Service	zookeeper.exe	In distributed installations, responsible for ensuring there is a quorum for making decisions during failover	n/a	Included in the base install on every node.
Data Engine	tdeserver64.exe tdeserver.exe (32-bit)	Stores data extracts and answers queries	Yes	The data engine's workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
				consumption is primarily based on the size of the data extracts being loaded. The 64-bit binary is used as the default on 64-bit operating systems, even if 32-bit Tableau Server is installed. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.
Data Server	dataserver.exe	Manages connections to Tableau Server data sources	Yes	Because it's a proxy, it's normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.
File Store	filestore.exe	Automatically replicates extracts across data engine nodes	n/a	Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.
Repository	postgres.exe	Tableau Server database, stores workbook and user metadata	n/a	Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository on page 48.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
Search & Browse	searchserver.exe	Handles fast search, filter, retrieval , and display of content metadata on the server	Yes	The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.
VizQL Server	vizqlserver.exe	Loads and renders views, computes and executes queries	Yes	Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.

Create a Performance Recording

With the Performance Recording feature in Tableau, you can record performance information about key events as you interact with workbooks. You then view performance metrics in a performance workbook that Tableau creates automatically. The steps you follow to create and view performance recording vary somewhat between Tableau Desktop and Tableau Server. However, the resulting performance workbooks have the same format in both Tableau Desktop and Tableau Server.

Use performance workbooks to analyze and troubleshoot performance issues pertaining to different events that are known to affect performance, including:

- Query execution
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation

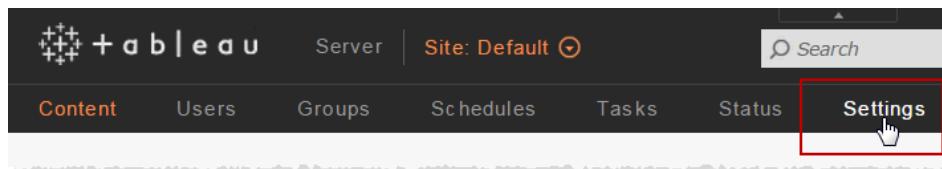
- Blending data
- Server blending (Tableau Server only)

Tableau support may request that you create performance workbooks as they assist you with diagnosing performance issues.

: Enable Performance Recording for a Site

By default, performance recording is not enabled for a site. A server administrator can enable performance recording site by site.

1. Navigate to the site for which you want to enable performance recording.
2. Click **Settings**:



3. Under Workbook Performance Metrics, select **Allow recording workbook performance metrics**.
4. Click **Save**.

: Start a Performance Recording for a View

1. Open the view for which you want to record performance.

When you open a view, Tableau Server appends ":iid=<n>" after the URL. This is a session ID. For example:

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1
```

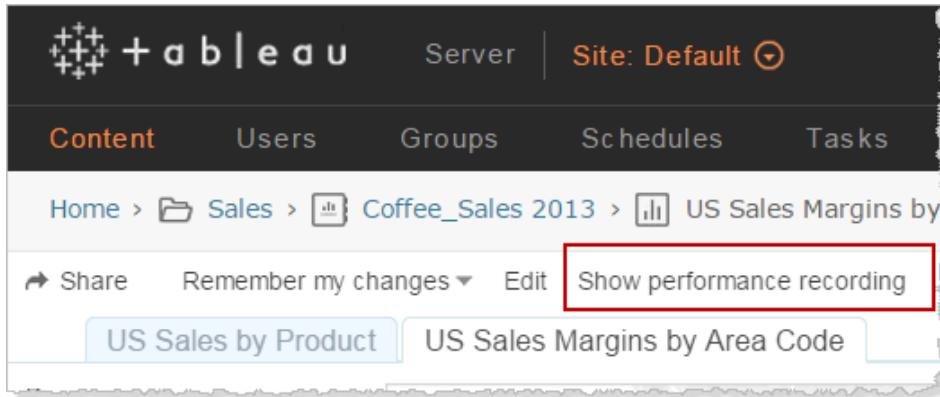
2. Type `:record_performance=yes&` at the end of the view URL, immediately before the session ID. For example:

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:record_performance=yes&:iid=1
```

3. Load the view.

A visual confirmation that performance recording has started is the **Show performance**

recording option in the view toolbar:



: View a Performance Recording

1. Click **Show performance recording** to open a performance workbook. This is an up-to-the-minute snapshot of performance data. You can continue taking additional snapshots as you continue working with the view; the performance data is cumulative.
2. Move to a different page or remove :record_performance=yes from the URL to stop recording.

Interpret a Performance Recording

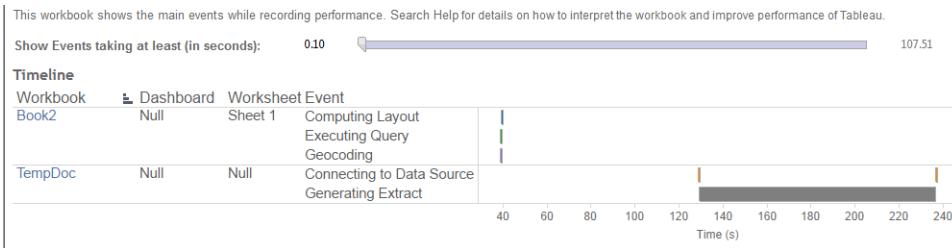
A performance recording workbook is a Tableau dashboard that contains three views: **Timeline**, **Events**, and **Query**.

For information on how to create a performance recording in Tableau Server, see [Create a Performance Recording on page 443](#).

Timeline

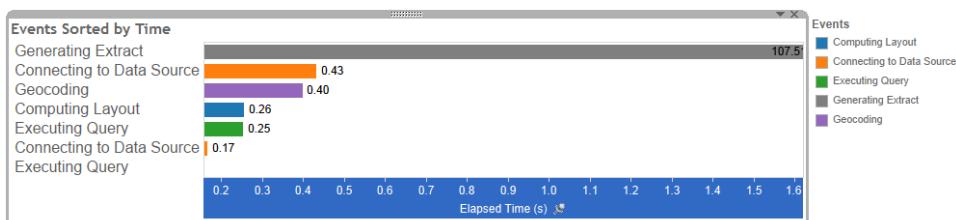
The uppermost view in a performance recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.

In the Timeline view, the **Workbook**, **Dashboard**, and **Worksheet** columns identify the context for events. The **Event** column identifies the nature of the event, and the final column shows each event's duration and how it compares chronologically to other recorded events:



Events

The middle view in a performance recording workbook shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.



Different colors indicate different types of events. The range of events that can be recorded is:

- Computing layouts.
If layouts are taking too long, consider simplifying your workbook.
- Connecting to data source.
Slow connections could be due to network issues or issues with the database server.
- Executing query.
If queries are taking too long, consult your database server's documentation.
- Generating extract.
To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.
- Geocoding.
To speed up geocoding performance, try using less data or filtering out data.
- Blending data.
To speed up data blending, try using less data or filtering out data.
- Server rendering.

You can speed up server rendering by running additional VizQL Server processes on additional machines.

Query

If you click on an **Executing Query** event in either the **Timeline** or **Events** section of a performance recording dashboard , the text for that query is displayed in the Query section. For example:

Query

```
SELECT "State"."ID" AS "ID",
"StateSynonyms"."Name" AS "State_Name",
"State"."ParentID" AS "State_ParentID"
FROM "StateSynonyms"
INNER JOIN "State" ON ("State"."ID" = "StateSynonyms"."ParentID") AND ("State"."MapCode" = "StateSynonyms"."MapCode")
```

Sometimes the query is truncated and you'll need to look in the Tableau log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau intelligently combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Proxy Servers

Tableau Server can be configured to work with a proxy server. In this type of environment, the proxy server acts as an intermediary between Tableau Server and the clients that are making requests for resources on Tableau Server. There are several ways to configure proxy servers—for example, as forward proxies or reverse proxies. These topics assume that you have already configured your proxy server, and now need to identify your proxy server to Tableau Server.

Notes for proxy support

Note the following about support for proxy environments when using Tableau Server:

- Tableau Server requires a proxy URL at the domain level. URLs that include a context path are not supported. For example, `tableau.mycompany.com` will work, but `www.mycompany.com/tableau` will not.
- If you will be using Kerberos authentication, you need to configure Tableau Server for your proxy before you configure Tableau Server for Kerberos. For more information, see [Configure Kerberos on page 508](#).
- Apache reverse proxy servers are not supported if Tableau Server is using SSPI (Active Directory with [Enable automatic logon](#)) for authenticating Tableau Server users. Apache reverse proxy servers are supported if Tableau Server is authenticating server users with just Active Directory (no [Enable automatic logon](#)), [Local authentication](#), or [SAML](#).

More information

Use the topics below for more information:

Prepare to Configure for a Proxy Environment

To configure Tableau Server to work with a proxy server, you will need the following information about your proxy server:

- **IP address:** The IP address of the proxy server. The IP address must be in IPv4 format, for example, `123.45.67.890`, and it must be a static IP.
- **FQDN:** The fully-qualified domain name of the proxy server. For example, `bigbox.example.com`. The FQDN should not include information beyond the domain name, such as `bigbox.example.com/tableau`. Tableau Server does not support context switching.
- **Non-FQDN:** Any non-fully-qualified domain names for the proxy server. Using the above example, the non-fully-qualified domain name of the proxy server would be `bigbox`.

- **Aliases:** Any aliases for the proxy server. Aliases are designated using CNAMEs (Canonical Name records). An example would be a proxy server with a CNAME of *bigbox.example.com* and aliases of *ftp.example.com* and *www.example.com*.

Configure Tableau to Work with a Proxy Server

After you collect the information described in [Prepare to Configure for a Proxy Environment](#) on the previous page, you can configure Tableau Server to work with a proxy by performing the following steps. For information on the settings below, see [tabadmin set options on page 616](#).

1. **Stop the server.**

2. Still in the Tableau Server bin directory, enter the following command, where *name* is the URL that will be used to reach Tableau Server through the proxy server:

```
tabadmin set gateway.public.host "name"
```

For example, if Tableau Server is reached by entering `tableau.example.com` in a browser address bar, enter this command:

```
tabadmin set gateway.public.host "tableau.example.com"
```

3. By default, Tableau assumes that the proxy server is listening on port 80 for external communications. To designate a different port, enter the following command, where *port_number* is the port:

```
tabadmin set gateway.public.port "port_number"
```

4. Now, enter the following command, where *server_ip_address* is the IPv4 address of the proxy server:

```
tabadmin set gateway.trusted "server_ip_address"
```

The value for *server* can be a comma-separated list, for example:

```
tabadmin set gateway.trusted "123.45.67.890, 123.45.67.880,  
123.45.67.870"
```

5. In the next command, you will provide any alternate names for the proxy server, such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases. These are the names a user might type in a browser. Separate each name with a comma:

```
tabadmin set gateway.trusted_hosts "name1, name2, name3"
```

For example:

```
tabadmin set gateway.trusted_hosts "proxy1.example.com,  
proxy1, ftp.example.com, www.example.com"
```

6. Type the following command to commit the configuration change:

```
tabadmin config
```

7. **Start the server** so the changes can take effect.

Trusted Authentication

When you embed Tableau Server views into webpages, everyone who visits the page must be a licensed user on Tableau Server. When users visit the page they are prompted to sign in to Tableau Server before they can see the view. If you already have a way of authenticating users on the webpage or within your web application, you can avoid this prompt and save your users from having to sign in twice by setting up trusted authentication.

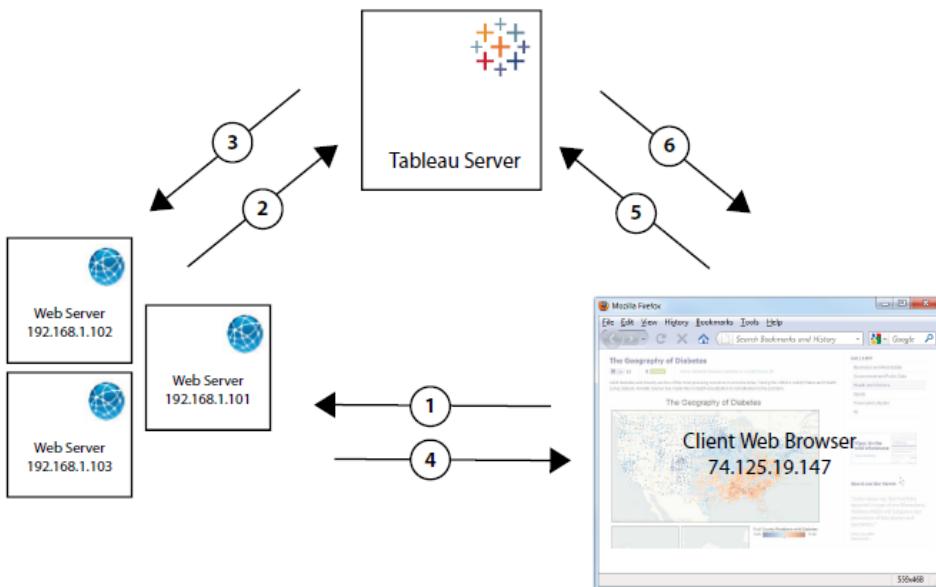
Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from these trusted web servers it assumes that your web server has handled whatever authentication is necessary.

If your web server uses SSPI (Security Support Provider Interface), you do not need to set up trusted authentication. You can embed views and your users will have secure access to them as long as they are licensed Tableau Server users and members of your Active Directory.

Note: Client browsers must be configured to [allow third-party cookies](#) if you want to use trusted authentication with embedded views.

How Trusted Authentication Works

The diagram below describes how trusted authentication works between the client's web browser, your web server(s) and Tableau Server.



- 1 User visits the webpage:**
When a user visits the webpage with the embedded Tableau Server view, it sends a GET request to your web server for the HTML for that page.
- 2 Web server POSTS to Tableau Server:** The web server sends a POST request to the trusted Tableau Server (for example, `http://tabaserver/trusted, not http://tabserver`). That POST request must have a `username` parameter. The `username` value must be the username for a licensed Tableau Server user. If the server is running multiple sites and the view is on a site other than the Default site, the POST request must also include a `target_site` parameter.
- 3 Tableau Server creates a ticket:** Tableau Server checks the IP address or host name of the web server (192.168.1.XXX in the above diagram) that sent the POST request. If it is set up as a trusted host then Tableau Server creates a ticket in the form of a unique 24-
- 4 Web server passes the URL to the browser:**
The web server constructs the URL for the view using either the view's URL or its object tag (if the view's embedded), and inserts it into the HTML for the page. The ticket is included (e.g., `http://tabserver/trusted/<ticket>/views/requestedviewname`). The web server passes all the HTML for the page back to the client's web browser.
- 5 Browser requests view from Tableau Server:** The client web browser sends a request to Tableau Server using a GET request that includes the URL with the ticket.
- 6 Tableau Server redeems the ticket:** Tableau Server sees that the web browser requested a URL with a ticket in it and redeems the ticket. Tickets must be redeemed within three minutes after they are issued. Once the ticket is redeemed, Tableau Server logs the user in, removes the ticket from the URL, and sends back the final URL for the embedded view.

character (URL-safe, Base64-encoded) string. Tableau Server responds to the POST request with that ticket. If there is an error and the ticket cannot be created Tableau Server responds with a value of -1.

Add Trusted IP Addresses or Host Names to Tableau Server

The first step in setting up trusted authentication is to configure Tableau Server to recognize and trust requests from one or more web servers:

1. Open a command prompt as an administrator and navigate to your Tableau Server bin directory (for example, C:\Program Files\Tableau\Tableau Server\9.2\bin).
2. Type the following command to stop Tableau Server:

```
tabadmin stop
```

3. Next, type the following command:

```
tabadmin set wgserver.trusted_hosts "<trusted IP addresses or host names>"
```

In the command above, <trusted IP addresses> should be a comma-separated list of the IPv4 addresses or host names of your web server(s). For example:

```
tabadmin set wgserver.trusted_hosts "192.168.1.101,  
192.168.1.102, 192.168.1.103"
```

or

```
tabadmin set wgserver.trusted_hosts "webserv1, webserv2, webserv3"
```

Notes:

The comma separated list should be in quotes, with one space after each comma. The web servers you specify must use static IP addresses, even if you use host names here ([learn more](#)).

4. If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in step 2, above) and Tableau Server, you also need to add them as trusted gateways. See [Configure Tableau to Work with a Proxy](#)

[Server](#) on page 449 for steps.

5. Type the following command to save the changes to all the server configuration files:

```
tabadmin config
```

6. Finally, type the following command to start the server again:

```
tabadmin start
```

Next, you need to [configure your web server to receive tickets from Tableau Server](#).

Get a Ticket from Tableau Server

After you've [added trusted IP addresses](#) to Tableau Server, you're ready to configure your web server to get tickets from Tableau Server via POST requests ([step 3 in the diagram](#)). The POST request must be sent to `http://<server name>/trusted`, not `http://tabserv`. For example `http://tabserv/trusted`.

Note: If SSL is enabled you must use https instead of http. For example: `https://tabserver/trusted`.

For code examples that you can use to create the POST request in Java, Ruby, and PHP, see the following:

```
C:\Program Files\Tableau\Tableau Server\9.2\extras\embedding
```

Here's the data you can use in a POST request to Tableau Server:

- **username=<username>** (required): The username for a licensed Tableau Server user. If you are using Local Authentication the username can be a simple string (for example, `username=jsmith`). If you are using Active Directory with multiple domains you must include the domain name with the user name (for example, `username=MyCo\jsmith`).
- **target_site=<site id>** (required if view not on Default site): Specifies the site containing the view if Tableau Server is running [multiple sites](#) and the view is on a site other than the Default site (for example, `target_site=Sales`). The value you use for `<site id>` should be the [Site ID](#) that was provided when the site was created. This value is case sensitive. If the **Site ID** is `SAles`, then the `target_site=SAles`.
- **client_ip=<IP address>** (optional): Used to specify the IP address of the computer whose web browser is accessing the view (for example, `client_ip=123.45.67.891`). It is not the IP address of the web server making the POST request of Tableau Server. If you decide to use this parameter, see [Optional: Configure Client IP Matching on page 457](#) for more information.

Tableau Server's response to the POST request will be a unique 24-character string (the ticket). If Tableau Server isn't able to process the request, the return will be -1. See [Ticket Value of -1 Returned from Tableau Server on page 457](#) for tips on how to correct this. Also, in order for users to successfully authenticate when they click an embedded view, their browsers must be configured to [allow third-party cookies](#).

Next, you need to add code that allows the web server to [construct an URL](#) for the view that includes the view's location and the ticket.

Display the View with the Ticket

After you [create the POST request](#), you need to write code that provides the web server with the view's location and the ticket from Tableau Server. It will use this information to display the view. How you specify it depends on whether the view is embedded, and if Tableau Server is running multiple sites.

Tableau Server View Examples

Here's an example of how to specify a view that users only access via Tableau Server (the view is not embedded):

```
http://tabserver/trusted/<ticket>/views/<workbook>/<view>
```

If Tableau Server is running [multiple sites](#) and the view is on a site other than the Default site, you need to add t/<site ID> to the path. For example:

```
http://tabserver/trusted/<ticket>/t/Sales/views/<workbook>/<view>
```

Use the same capitalization that you see in the Tableau Server URL.

Embedded View Examples

Here are some examples of how to specify embedded views. Because there are two approaches you can take with [embed code](#), both ways are provided below. Regardless of which you use, there is some information unique to trusted authentication that you must provide.

Script Tag Examples

This example uses the [ticket](#) object parameter:

```
<script type="text/javascript" src-  
c="http://myserver/javascripts/api/viz_v1.js"></script>  
<object class="tableauViz" width="800" height="600" style-  
e="display:none;">  
    <param name="name" value="MyCoSales/SalesScoreCard" />
```

```
<param name="ticket" value="EtDpsm_Ew6rJY-9kRrALjauU" />
</object>
```

Here's what the above example looks like for a multi-site Tableau Server, where the view is published on the Sales site:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style-
e="display:none;">
    <param name="site_root" value="/t/Sales" />
    <param name="name" value="MyCoSales/SalesScoreCard" />
    <param name="ticket" value="EtDpsm_Ew6rJY-9kRrALjauU" />
</object>
```

Instead of using ticket, you can use the path parameter to state the full path of the view explicitly. When path is used, you do not also need the name parameter, which is usually a required parameter in Tableau JavaScript embed code:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700" style-
e="display:none;">
    <param name="path" value="trusted/EtDpsm_Ew6rJY-9kRrAL-
jauU/views/MyCoSales/SalesScoreCard" />
</object>
```

Here's the same example, but for a multi-site server. Note that /t/<site ID> is used here:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700" style-
e="display:none;">
    <param name="path" value="trusted/EtDpsm_Ew6rJY-9kRrAL-
jauU/t/Sales/views/MyCoSales/SalesScoreCard" />
</object>
```

Iframe Tag Example

```
<iframe src="http://tabserver/trusted/EtDpsm_Ew6rJY-9kRrAL-
jauU/views/workbookQ4/SalesQ4?:embed=yes" width="800" height-
t="600"></iframe>
```

Optional: Configure Client IP Matching

By default, Tableau Server does not consider the client web browser IP address when it creates or redeems tickets. To change this, you need to do two things: specify an IP address using the `client_ip` parameter in the POST request that obtains the ticket, and follow the steps below to configure Tableau Server to enforce client IP address matching.

1. Open a command window and change directories to the location of Tableau Server's bin directory. The default location is `C:\Program Files\Tableau\Tableau Server\9.2\bin`

2. Open a command prompt as an administrator and type the following command:

```
tabadmin set wgserver.extended_trusted_ip_checking true
```

3. Then type the following command:

```
tabadmin configure
```

4. Finally, restart the server by typing the following:

```
tabadmin restart
```

Troubleshoot Trusted Authentication

Below are some common issues and errors you might encounter when you're configuring trusted authentication. Trusted authentication information is written to `ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\vizql-*.log`. To increase the logging level from `info` to `debug`, use the `tabadmin` setting `vizqlserver.trustedticket.log_level`.

For tips on testing trusted authentication, see the [Tableau Knowledge Base](#).

Ticket Value of -1 Returned from Tableau Server

Tableau Server returns -1 for the ticket value if it cannot issue the ticket as part of the trusted authentication process. The exact reason for this message is written to the file `production*.log` in the following folder:

`ProgramData\Tableau\Tableau Server\data\tabsvc\logs\wgserver`

and to the `vizql*.log` in the following folder:

`ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver`

Here are some things to confirm:

- **All web server host names or IP addresses are added to trusted hosts**

The IP address or host name for the computer sending the POST request must be in the list of trusted hosts on Tableau Server. See [Add Trusted IP Addresses or Host Names to Tableau Server](#) on page 453 to learn how to add IP addresses or host names to this list.

- **Value of `wgserver.trusted_hosts` is properly formatted**

The list of trusted hosts you provided using the `wgserver.trusted_hosts` setting must be a comma-separated list with a space after each comma. For example, the list should be similar to the following: 192.168.1.101, 192.168.1.102, 192.168.1.103, or bigbox1.example.lan, bixbox2.example.lan, bigbox3.example.lan.

- **IP addresses are IPv4**

If you are using IP addresses to specify trusted hosts, they must be in Internet Protocol version 4 (IPv4) format. An IPv4 address looks like this: 123.456.7.890. IPv6 addresses (for example, fe12::3c4a:5eab:6789:01c%34) are not supported as a way of inputting trusted hosts.

- **Username in POST request is a valid Tableau Server user**

The username you send in the POST request must be a licensed Tableau Server user with a Viewer or Interactor license level. You can see a list of users and their license levels by signing in to Tableau Server as an administrator and clicking the Licensing link on the left side of the page.

- **Username in POST request includes domain**

If Tableau Server is configured to use Local Authentication, the username that you send in the POST can be a simple string. However, if the server is configured for Active Directory you must include the domain name with the user name (domain\username). For example, the username parameter might be: `username=dev\jsmith`

- **Content-Type is specified**

If you are designing an ASP.NET or C# application, you need to declare the content type in your HTTP request. For example, `http.setRequestHeader ("Content-Type ", "application/x-www-form-urlencoded; charset=UTF-8 ")`. If you do not specify content type and Tableau Server returns a -1, the log files contain the error: "missing username and/or client_ip".

HTTP 401 - Not Authorized

If you receive a 401- Not Authorized error, you may have configured Tableau Server to use Active Directory with SSPI (see [Enable automatic login](#)). If your web server uses SSPI, you do not need to set up trusted authentication. You can embed views and your users will have access to them as long as they are licensed Tableau server users and members of your Active Directory.

HTTP 404 - File Not Found

You may receive this error if your program code references a Tableau Server URL that does not exist. For example, your web server may construct an invalid URL that cannot be found when the webpage tries to retrieve it.

Invalid User (SharePoint or C#)

You may encounter this error if you've configured Tableau Server for trusted authentication.

The example code for the SharePoint .dll references the following GET request:

```
SPContext.Current.Web.CurrentUser.Name
```

The above request will return the display name of the current Windows Active Directory user. If you want to use the login ID, then you will need to change the code to:

```
SPContext.Current.Web.CurrentUser.LoginName
```

After you make the change, recompile the SharePoint .dll.

Attempting to Retrieve the Ticket from the Wrong IP Address

You may encounter this error if you've configured Tableau Server for trusted authentication.

The client web browser IP address is not considered by default when redeeming the ticket. If Tableau Server is configured to enforce client IP address matching, make sure that the client's web browser IP address that is sent in the POST to Tableau Server is the same as when the browser tries to retrieve the embedded view. For example, in the Trusted Authentication diagram, if the [POST request in step 3](#) sends the parameter client_ip=74.125.19.147, then the [GET request in step 5](#) must come from that same IP address.

See [Optional: Configure Client IP Matching on page 457](#) to learn how to configure Tableau Server to enforce client IP address matching.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's

web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

An error occurred communicating with the server (403)

If Tableau Server is configured for trusted authentication, you may receive this error after opening a new view in a browser and attempting to navigate back to views you'd opened earlier. Tableau Server provides protection against unauthorized reuse of VizQL sessions through the tabadmin set option `vizqlserver.protect_sessions`, which is set to `true` by default. Because Tableau Server is configured for trusted authentication, you may not also need to enable `vizqlserver.protect_sessions`. To disable it, use [set on page 610](#) to change it to `false`.

OAuth Connections

For Google BigQuery, Google Analytics, and Salesforce.com data sources, an alternative to storing sensitive database credentials with Tableau Server is to create connections using the **OAuth 2.0** standard.

When you create an OAuth connection, you give the data provider your approval for Tableau to access your data. The data provider then sends Tableau an **access token** that uniquely identifies requests from Tableau. For more information, see [Overview of the OAuth process below](#) below.

Using OAuth connections provides the following benefits:

- **Security:** Your database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau.
- **Convenience:** Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider.

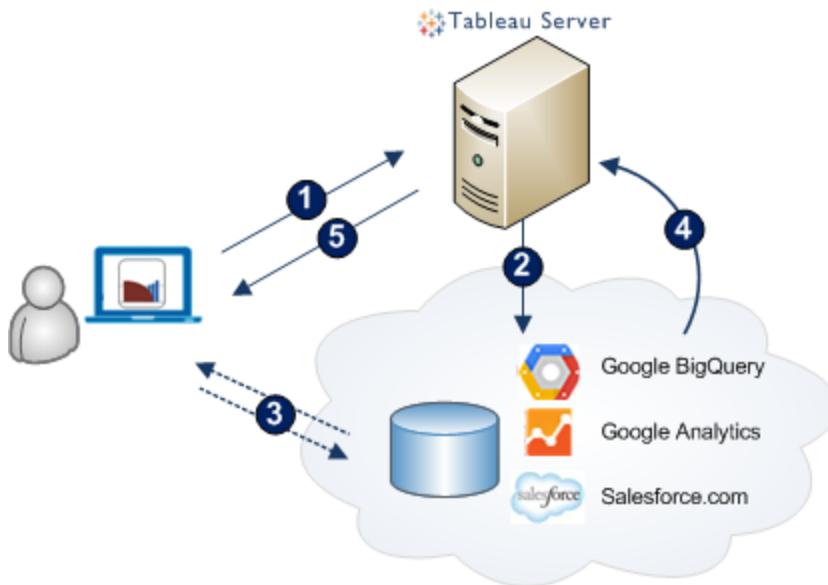
In addition, for live connections to Google BigQuery data, each workbook viewer can have a unique access token that identifies the user, rather than sharing a single user name and password credential.

Overview of the OAuth process

The following steps describe a workflow in the Tableau environment that calls the OAuth process.

1. You take an action that requires access to a cloud data source.
For example, you open a workbook that's published to Tableau Server.
2. Tableau directs you to the hosted data provider's sign-in page. The information that is sent to the hosted data provider identifies Tableau as the requesting site.
3. When you sign in to the hosted data source, it prompts you to confirm your authorization for Tableau Server access to the data.
4. Upon your confirmation, the data-source provider sends an access token back to Tableau Server.

5. Tableau Server presents your workbook and data to you.



The following other workflows can also use the OAuth process:

- Creating a workbook and connecting to the data source from Tableau Desktop or from Tableau Server.
- Publishing a data source from Tableau Desktop.
- Signing in to Tableau Server from an approved *client*, such as Tableau Mobile or Tableau Desktop.

Access tokens for data connections

You can save access tokens with data connections to enable direct access to the data after the initial authentication process. An access token is valid until a Tableau Server user deletes it, or the data provider revokes it.

It is possible to exceed the number of access tokens your data source provider allows. If that's the case, when a user creates a new token, the data provider uses length of time since last access to decide which token to invalidate to make room for the new one.

Access tokens for authentication from approved clients

By default, Tableau Server sites allow users to access their sites directly from approved Tableau clients, after users provide their credentials the first time they sign in. This type of authentication also uses OAuth access tokens to store the users' credentials securely.

For more information, see [Authentication for Connected Devices](#) on page 259

Configure the Server for OAuth Support

Instead of individual usernames and passwords, OAuth works through limited-purpose access tokens. Before you can obtain access tokens needed to create an OAuth connection in Tableau, you need to configure your server so that the data provider sending the token can recognize Tableau Server as a trusted destination. The following section describes how to prepare for setting up OAuth regardless of your data provider. The topics listed below it contain the steps for configuring your server for specific data providers.

Preparing for Configuring OAuth Support

Before you begin the configuration steps specific to your data provider, complete the following prerequisites:

- Obtain the fully qualified domain name of each Tableau Server node that will host views that connect to this data source. For example:

`https://sales.your_domain.com`

If you use Salesforce.com, you will need to provide an `https` address.

- Make sure at least one of your data-provider accounts is enabled for API access.

For **Google BigQuery** and **Google Analytics**, you need access to the developers console on the [Google Cloud Platform](#).

For **Salesforce.com**, you need access to the [Force.com platform](#).

- Make sure you have the latest drivers for the data source.

For Google BigQuery, use the 32-bit version.

You can download updated drivers from the [Drivers & Activation](#) page on the Tableau website.

Configure Settings for Your Data Provider

When you complete the OAuth-preparation steps, you can configure the appropriate settings with your data provider.

- [Set up OAuth for Google](#) below
- [Set up OAuth for Salesforce.com](#) on page 466

Set up OAuth for Google

This topic describes how to set up your Google BigQuery and Google Analytics data sources for OAuth. Complete these steps for each Tableau Server instance.

Note Before you complete these steps, make sure you have completed the prerequisites described in [Preparing for Configuring OAuth Support](#) on the previous page.

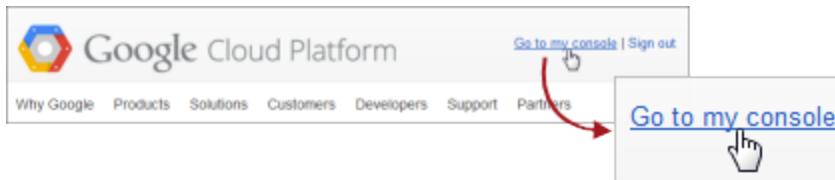
Set up OAuth by following these two procedures:

- Get required information from Google and enable API access.
- Use the information you obtained to configure your server.

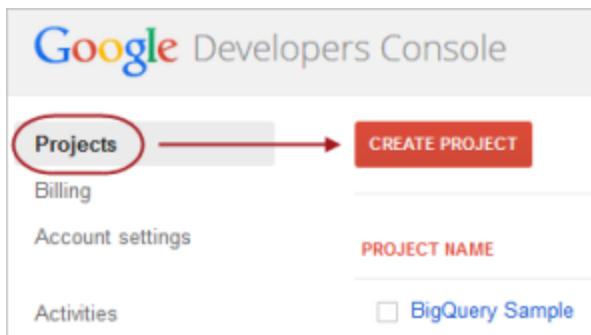
Obtain a Client ID and Enable Google APIs

Note These steps reflect the settings in the Google Cloud Platform console at the time of this writing. For more information, see [Using OAuth 2.0 for Web Server Applications](#) in the Google Developers Console Help.

1. Sign in to [Google Cloud Platform](#), and then click **Go to my console**.



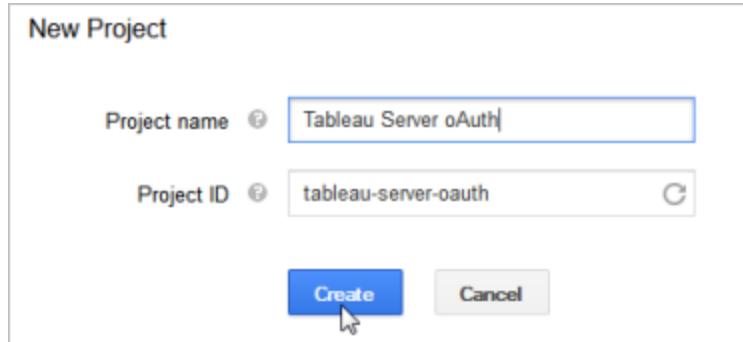
2. Select **Projects**, and on the Project page, click **Create Project**.



3. In the new project form that appears, complete the following:

- Give the project a meaningful name that reflects the Tableau Server instance for which you'll use this project.
- Determine whether you want to change the project ID.

Note After you create the project, you will not be able to change the project ID. For information, click the question mark icons.



4. Open the new project, and navigate to **APIs & auth > Credentials**.
5. Click **Create a New Client ID**, and in the Create Client ID page, complete the following:
 - Select **Web Application**.
 - For Authorized JavaScript Origins, type the local computer name of your Tableau Server.
 - For Authorized Redirect URI, replace the existing text with the Internet address for your server, and add the following text to the end of it: **auth/add_oauth_token**. For example:
`https://your_server_url.com/auth/add_oauth_token`
6. Click **Create Client ID**.
7. Copy the following values that Google returns, and paste them in a location that you can access from your Tableau Server computer:
 - Client ID
 - Client secret
 - Redirect URIs
8. In the Google Developer Console, with your new project open, select **APIs & auth > APIs**, and then set the status to **On** for **BigQuery API** or **Analytics API**.



Configure Tableau Server for Google OAuth

Using the information you obtained by completing the steps in [Obtain a Client ID and Enable Google APIs on page 464](#), configure your Tableau Server:

1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory.

```
cd C:\Program Files\Tableau\Tableau Server\<version>\bin
```

2. Type the following command to stop the server:

```
tabadmin stop
```

3. Type the following commands to configure the server with the client ID and client secret you obtained from Google, as well as your server URI. Press **Enter** after each command.

```
tabadmin set oauth.google.client_id <your_client_ID>
```

```
tabadmin set oauth.google.client_secret <your_client_secret>
```

```
tabadmin set oauth.google.redirect_uri <your_server_URI>
```

4. Type the following commands to complete the configuration and restart the server:

```
tabadmin config
```

```
tabadmin start
```

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see [Allow Saved Access Tokens on page 469](#).

Set up OAuth for Salesforce.com

This topic describes how to set up your Salesforce.com data sources for OAuth. Complete these steps for each Tableau Server instance.

Note: Before you complete these steps, make sure you have completed the prerequisites described in [Preparing for Configuring OAuth Support on page 463](#).

Set up OAuth by following these two procedures:

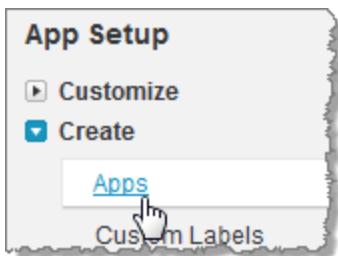
- Create a Connected App in Salesforce
- Use the information you obtained to configure your server.

Create a Connected Salesforce App

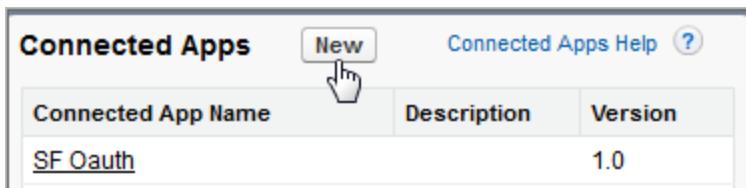
1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.



2. In the left navigation column, under App Setup, select **Create > Apps**.



3. In the Connected Apps section, click **New**.



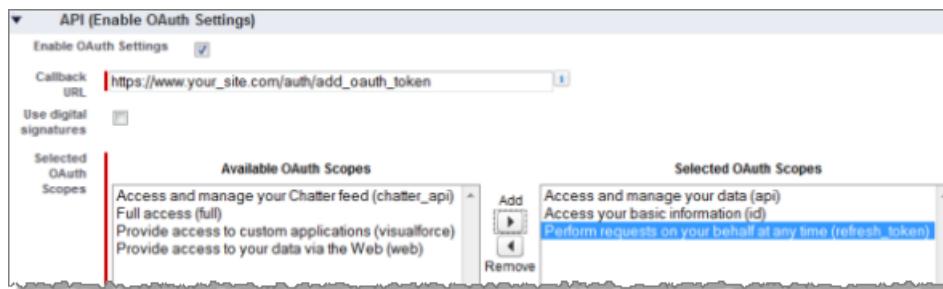
4. Complete the **Basic Information**, and in the API section, select **Enable OAuth Settings**.
5. In the new OAuth settings that appear, for **Callback URL**, type the fully qualified domain name of your server, using the https protocol, and append the following text to the URL: **auth/add_oauth_token**.

For example:

https://www.your_server.com/auth/add_oauth_token

6. Move the following items from Available OAuth Scopes to Selected OAuth Scopes:

- **Access and manage your data (api)**
- **Access your basic information (id)**
- **Perform requests on your behalf at any time (refresh_token)**



7. Click **Save**.

After you save the app, Salesforce populates the API section with the following IDs that you will use to configure Tableau Server:

- Consumer Key
- Consumer Secret
- Callback URL

Configure Tableau Server for Salesforce.com OAuth

1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory:

```
cd C:\Program Files\Tableau\Tableau Server\<version>\bin
```

2. Type the following command to stop the server:

```
tabadmin stop
```

3. Type the following commands to configure the server with the consumer ID and secret you obtained from Salesforce and the callback URL. Press **Enter** after each command:

```
tabadmin set oauth.salesforce.client_id <your_consumer_ID>
```

```
tabadmin set oauth.salesforce.client_secret <your_consumer_secret>
```

```
tabadmin set oauth.salesforce.redirect_uri <your_callback_URL_>
```

4. (Optional) To change the default login server, type the following command:

```
tabadmin set oauth.salesforce.server_base_url <URL>
```

By default, this is set to <https://login.salesforce.com>.

5. Type the following commands to complete the configuration and restart the server:

```
tabadmin config
```

```
tabadmin start
```

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see [Allow Saved Access Tokens below](#).

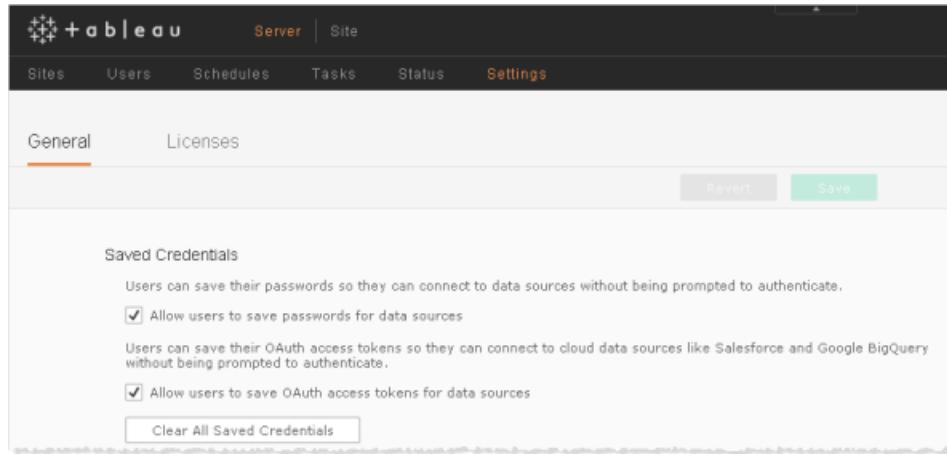
Allow Saved Access Tokens

After you configure Tableau Server for OAuth, you can decide to allow users to manage their own OAuth credentials, or you want to manage them centrally. If you want users to manage their own, you need to enable user profile settings from the server.

Note: If you have not yet configured your server to enable OAuth data connections, see the related topics listed below.

1. Sign in to Tableau Server as a server administrator.
2. Select **Server > Settings**.
3. Click **General**, and then select the following:
 - **Allow users to save passwords for data sources** (allows users to save their individual credentials with data sources).

- Allow users to save OAuth access tokens for data sources



4. Click **Save**.

After you select these check boxes, users will see a **Manage Credentials** section in their profile settings, where they can add access tokens for OAuth data connections.

Manage Credentials	
Salesforce	Add
Google BigQuery	Add
Google Analytics	Add
tableauonlineuser@gmail.com	
Delete Test	

Managing credentials centrally

Server administrators alternatively can manage OAuth credentials centrally. This can work well, for example, if multiple users work from the same data, and you have a dedicated user account for your data provider.

To manage credentials centrally, you do the following:

- Clear the check boxes described in the preceding procedure.
- Edit connection information as data sources are published.

When you edit the connection, you embed credentials that use an OAuth access token instead of an individual's user name and password.

When the settings for saving passwords and access tokens are not enabled, the Manage Credentials section is excluded from users' profile settings.

See also

[Set up OAuth for Google on page 463](#)

[Set up OAuth for Salesforce.com on page 466](#)

SAML

SAML (Security Assertion Markup Language) is an XML standard that allows secure web domains to exchange user authentication and authorization data. You can configure Tableau Server to use an external identity provider (IdP) to authenticate Tableau Server users over SAML 2.0. All user authentication is done outside of Tableau, regardless of whether you're using Active Directory or local authentication in Tableau Server to manage your user accounts on Tableau Server. This allows you to provide a single sign-on experience for your users across all the applications in your organization.

The SAML support in Tableau Server is for user authentication. It does not handle permissions and authorization having to do with Tableau Server content, such as workbooks.

Note: The IdP-provided authentication is a single-use, limited time token.

See the links below for more information about SAML:

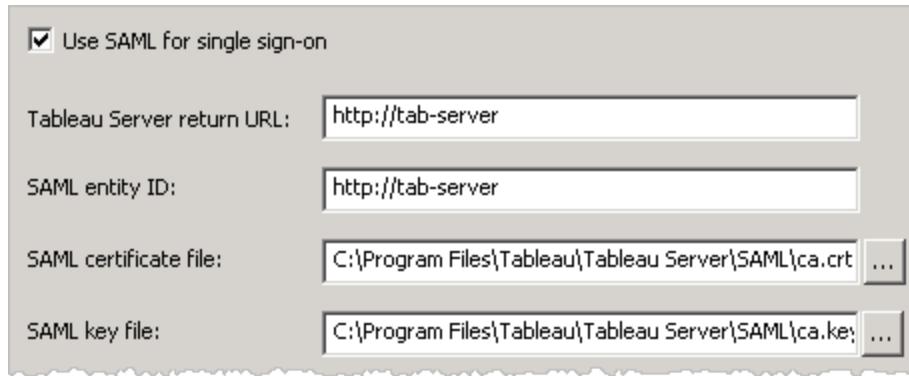
Quick Start: Single Sign-On with SAML

With Tableau's SAML support, you use an external identity provider (IdP) to authenticate Tableau Server users. All user authentication is done outside of Tableau, regardless of whether you're using Active Directory or local authentication in Tableau Server to manage your user accounts. This allows you to provide a single sign-on experience across all the applications in your organization. To configure Tableau Server for SAML, you need the following:

- **Certificate file:** A PEM-encoded x509 certificate with the extension **.crt**.
- **Certificate key file:** An RSA or DSA key file that is not password protected and that has a **.key** file extension.
- **IdP account:** Examples are PingFederate, SiteMinder, and OpenAM.
- **Matching usernames:** Tableau Server usernames and the usernames stored in the IdP must match. Ensure that the username you plan to use for your Tableau Server administrator account exists in your IdP before you run Setup.

1 Specify the Server and Certificates

Run Server Setup. After you configure your general settings in the Configuration utility, click the **SAML** tab and select **Use SAML for single sign-on**:

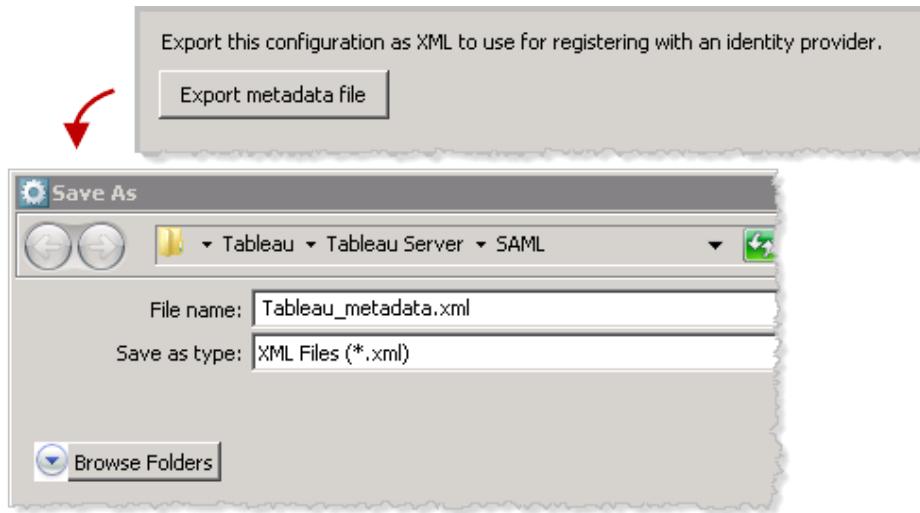


In the **Tableau Server return URL** text box, enter the customer-facing URL for your Tableau Server. Enter this same value for **SAML entity ID**.

Create a SAML folder under C:\Program Files\Tableau\Tableau Server and copy your **.crt** and **.key** files there. Enter that location in the next two fields.

2 Export Metadata from Tableau

Leaving the **SAML IdP metadata file** text box empty, click the **Export Metadata File** button.



Use the **.xml** file name of your choice.

In the next dialog box, save the XML file. You will need to provide this file to your IdP in the next step.

3 Export Metadata from the IdP

On the IdP's website, add your Tableau Server as a connection type for the IdP to authenticate. As part of this, you will import the Tableau metadata **.xml** file you created in step 2, and confirm that your IdP's settings use **username** as the attribute element to verify.

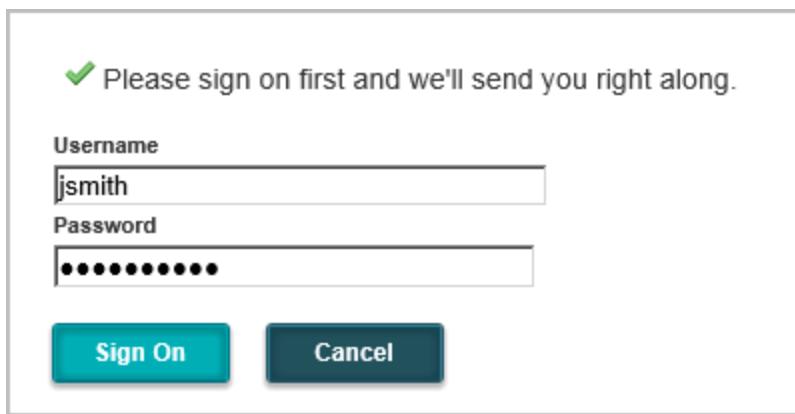
Next, export your IdP's metadata **.xml** file and copy it to the following folder on your Tableau Server:

C:\Program Files\Tableau\Tableau Server\SAML

4 Test the SAML Sign-On

On the **SAML** tab in the Tableau Configuration utility, enter the location to the IdP's file in the **SAML IdP metadata** file text box. Click OK. Finish Setup, creating an administrator account when prompted.

To test your changes, start a fresh web browser session to Tableau Server. You should note that the Sign On prompt is from your IdP and not Tableau:



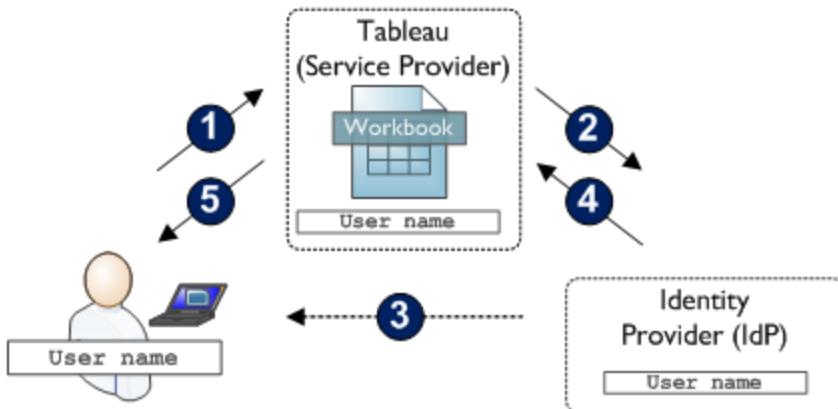
How SAML Authentication Works

SAML (Security Assertion Markup Language) is an open standard for exchanging authentication information between a service provider and an identity provider (IdP). A third-party IdP is used to authenticate users and to pass identity information to the service provider in the form of a digitally signed XML document. Tableau Server is a service provider. Examples of IdPs include PingOne and OneLogin.

When you use a trusted IdP for the SAML connection, you can provide a single sign-on (SSO) experience, in which your users can access their web applications, including Tableau Server, through one set of credentials. In this environment, only the IdP has access to users' credentials.

Tableau supports authentication initiated by the service provider. This means that your users must access Tableau Server from its sign-in page, rather than an IdP sign-in page.

The following image shows the single sign-on authentication sequence.



- 1 User navigates to the Tableau Server sign-in page or a published workbook, and enters the user name.
- 2 Tableau Server starts the authentication process and redirects the request to the registered IdP.
- 3 The IdP requests the user's password and, after confirming that the user name submitted is identical to the user name stored in the IdP assertions, authenticates the user.
- 4 The IdP returns a SAML success response to Tableau Server.
- 5 Tableau Server displays the page the user requested in step 1.

SAML Requirements

To configure Tableau Server for SAML, you need the following:

- **Certificate file:** A PEM-encoded x509 certificate with the file extension **.crt**. This file is used by Tableau Server, not the IdP. If you have an SSL certificate, you can use the same certificate with SAML. See [About the Certificate File](#) later in this topic for details.
- **Certificate key file:** An RSA or DSA private key file that is not password protected, and which has the file extension **.key**. This file is used by Tableau Server, not the IdP. The certificate key file must have the passphrase embedded in it. If you have an SSL certificate key file, you can use it for SAML as well. See [About the Certificate File](#) later in

this topic for details.

- **IdP account that supports SAML 2.0:** You need an account with an external identity provider. Some examples are PingFederate, SiteMinder, and Open AM. The IdP must support SAML 2.0.
- **IdP provider that supports import/export of XML metadata:** Your identity provider must support the import and export of XML metadata files. Manually generated files may appear to work, but Tableau Software Technical Support cannot assist with manual IdP metadata file generation or troubleshooting.

Additional notes about SAML compatibility with Tableau Server

Note the following about using SAML with Tableau Server:

- **SP-initiated:** Tableau Server only supports SAML authentication that begins at the service provider (SP).
- **No Active Directory automatic logon:** If you are using SAML and Tableau Server is also configured to use Active Directory for user management, do not also use **Enable automatic logon**.
- **No Kerberos:** Tableau Server does not support SAML and Kerberos together.
- **User identity in Tableau Server for tabcmd users:** To use **tabcmd** with the server, users must sign in to the server using the credentials of a user defined on the server; you cannot use **tabcmd** to sign in using SAML. An initial system administrator user is created when the server is first installed and configured, and you can add more users by creating them on the server or importing them from Active Directory. See [Add Users to the Server on page 188](#) for more information.
- **IdP provider that uses forms-based authentication:** Tableau Desktop requires forms-based authentication. If your IdP does not support forms-based authentication you can disable SAML for Tableau Desktop with the `wgserver.authentication.desktop_nosaml` command. See [tabadmin set options on page 616](#) for more information.
- **Distributed installations:** Clusters configured for SAML must have the same SAML certificate, SAML key, and SAML IdP metadata files on each Tableau Server that's running an application server process. See [Configure a Server Cluster for SAML](#) for details.
- **Login URL:** To sign in, your IdP must be configured with SAML Login that does a POST to `http(s)://<tableauserver>/wg/saml/SSO/index.html`.
- **Logout URL:** To sign out when authenticated with SAML, your IdP must be configured with a SAML Logout endpoint that does a POST to `http(s)://<tableauserver>/wg/saml/SingleLogout/index.html`.
- **Post-Logout Redirect URL:** By default, when you sign out of Tableau Server, the sign

in screen appears. You can use the `tabadmin set wgserver.saml.logout.redirect_url` command to specify an alternate page to display after you sign out.

- To specify an absolute URL, use a fully-qualified URL starting with `http://` or `https://` (for example, `tabadmin set wgserver.saml.redirect_url http://corpserver.bigco.com`).
- To specify a URL relative to the Tableau Server host, use a page starting with a / (slash) (for example, `tabadmin set wgserver.saml.redirect_url /ourlogoutpage.html`).
- To specify a URL relative to the Tableau Server SingleLogout API endpoint, use only a page name (for example, `tabadmin set wgserver.saml.redirect_url ourlogoutpage.html`).

Note: The post-logout redirect page cannot be hosted on the Tableau Server Apache Server.

Requirements for XML data

You configure SAML using metadata XML documents that are generated by Tableau Server and by your IdP. During the authentication process, the IdP and Tableau Server exchange authentication information using XML documents. To be sure that the XML that's used for SAML configuration and SAML-based authentication works correctly, review the following requirements. If the XML does not meet these requirements, you can experience errors when you configure SAML or during the authentication process.

- **HTTP POST:** Tableau Server only accepts HTTP POST requests for SAML communications. HTTP Redirect is not supported.

The SAML metadata XML document that is exported by Tableau Server should contain the following elements, with the `Binding` attribute set to `HTTP-POST`.

- The first element you should verify specifies the URL that the IdP redirects to after successful authentication:

```
<md:AssertionConsumerService  
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"  
Location="http(s)://YOUR-SERVER/wg/saml/SSO/index.html  
index="0" isDefault="true"/>
```

- The second element you should verify specifies the URL that the IdP will use for the logout endpoint:

```
<md:SingleLogoutService  
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
```

```
Location="http  
(s)://<tableauserver>/wg/saml/SingleLogout/index.html/">
```

In addition, the metadata XML document that is created by the IdP should contain the following `SingleSignOnService` element, with the `Binding` attribute set to HTTP-POST:

```
<md:SingleSignOnService  
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"  
Location="http(s)://<tableauserver>/wg/saml/SSO/index.html/">
```

- **Attribute named `username`:** You must configure your identity provider to return an assertion that includes the `username` value in the `<saml:AttributeStatement>` element in a format like the following example. Make sure that the attribute is typed as `xs:string`. (It should *not* be typed as `xs:any`.)

```
<saml:Assertion assertion-element-attributes>  
  <saml:Issuer>issuer-information</saml:Issuer>  
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">  
    ...  
  </Signature>  
  <saml:Subject>  
    ...  
  </saml:Subject>  
  <saml:Conditions condition-attributes >  
    ...  
  </saml:Conditions>  
  <saml:AuthnStatement authn-statement-attributes >  
    ...  
  </saml:AuthnStatement>  
  
  <saml:AttributeStatement>  
    <saml:Attribute Name="username"  
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-  
      format:basic">  
      <saml:AttributeValue  
        xmlns:xss="http://www.w3.org/2001/XMLSchema"  
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
        xsi:type="xs:string">  
        user-name  
      </saml:AttributeValue>  
    </saml:Attribute>
```

```
</saml:AttributeStatement>  
</saml:Assertion>
```

To change the SAML attribute that the **username** value is passed in, use the **tabadmin set** command to set the `wgserver.saml.idpattribute.username` value to a different attribute name. You must change the attribute if you use a global ID. The attribute name is case sensitive.

- **Matching usernames:** Tableau Server usernames and the usernames stored in the IdP must match. For example, if the username for Jane Smith is stored in PingFederate as `jsmith`, it must also be stored in Tableau Server as `jsmith`. If you are configuring SAML as part of Tableau Server Setup, part of Setup is creating the Tableau Server administrator account. Before you run Setup, make sure that the account you plan to use exists in your IdP. If you are using Active Directory authentication with Tableau Server and have multiple Active Directory domains (users belong to multiple domains or your Tableau Server installation includes multiple domains), the IdP must send both the domain and username for a user, and these must match the user exactly in Tableau Server (these can be sent either as `domain/username` or `username@domain`).

About the Certificate File

If you are using a PEM-encoded x509 certificate file for SSL, you can use the same file for SAML. When it's used for SSL, the certificate file is used to encrypt traffic. When it's used for SAML, the certificate is used for authentication.

Tableau Server does not support certificate and certificate key files for SAML if the certificate/key require a chain file. If your SSL certificate and certificate key file require a chain file, you need to generate a new certificate and key file to use for SAML.

Configure SAML

You can configure Tableau Server to use an external identity provider (IdP) to authenticate Tableau Server users over SAML. All user authentication is done outside of Tableau, regardless of whether you're using Active Directory or local authentication in Tableau Server to manage your user accounts on Tableau Server. This allows you to provide a single sign-on experience across all the applications in your organization.

Before you configure Tableau Server for SAML, make sure you meet the [SAML Requirements on page 475](#).

Configure SAML

To configure Tableau Server to use SAML:

1. Place the certificate files in a folder named SAML, parallel to the Tableau Server 9.2 folder. For example:

```
C:\Program Files\Tableau\Tableau Server\SAML
```

You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. If you are configuring SAML during Tableau Server setup, go to the SAML tab in the configuration utility.

If you are configuring SAML after you installing Tableau Server, open the Tableau Server Configuration Utility (**Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**) and then click the **SAML** tab.

3. On the SAML tab, select **Use SAML for single sign-on** and provide the location for each of the following:

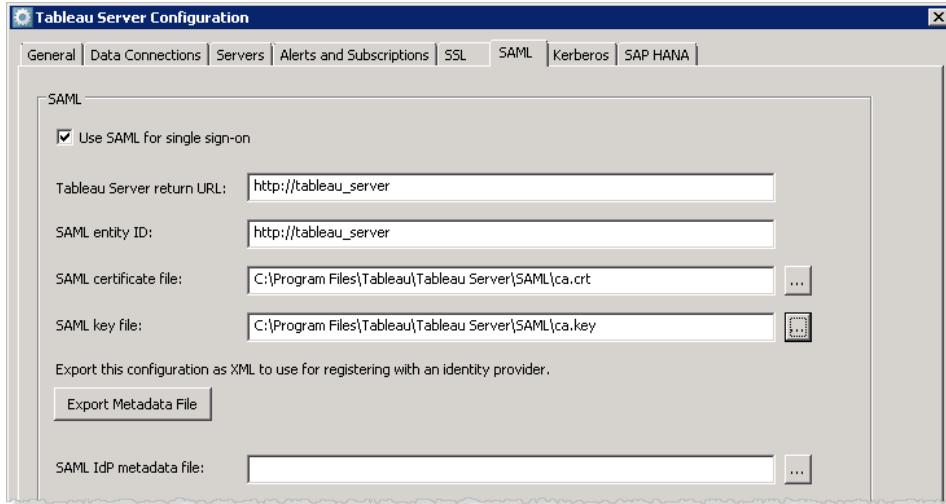
Tableau Server return URL—The URL that Tableau Server users will be accessing, such as `http://tableau_server`. Using `http://localhost` is not recommended. Using a URL with a trailing slash (for example, `http://tableau_server/`) is not supported.

SAML entity ID—The entity ID uniquely identifies your Tableau Server installation to the IdP. You can enter your Tableau Server URL again here, if you like, but it does not have to be your Tableau Server URL.

SAML certificate file—A PEM-encoded x509 certificate with the file extension `.crt`. This file is used by Tableau Server, not the IdP.

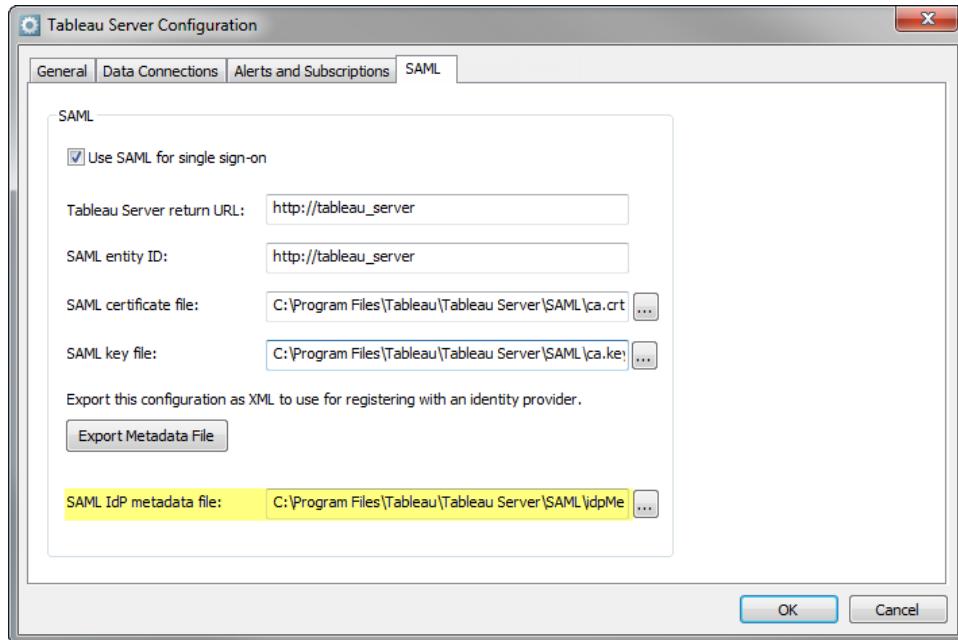
SAML certificate key file—An RSA or DSA private key file that is not password protected, and that has the file extension `.key`. This file is used by Tableau Server, not the IdP.

4. Leave the **SAML IdP metadata file** text box empty for now and click **Export Metadata File**.



5. A dialog box opens that allows you to save Tableau Server's SAML settings as an XML file. At this point, metadata from your IdP is not included.
Save the XML file with the name of your choice.
6. On your IdP's website or in its application:
 - Add Tableau Server as a Service Provider. Refer to your IdP's documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the file you saved in step 5.
 - Confirm that your IdP uses **username** as the attribute element to verify.
7. Still within your IdP, export your IdP's metadata XML file.
It's a good idea to verify that the metadata XML you get from the IdP includes a **SingleSignOnService** element in which the binding is set to HTTP-POST, as in the following example:

```
<md:SingleSignOnService Bind-
ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://SERVER-NAME:9031/idp/SSO.saml2"/>
```
8. Copy your IdP's metadata XML file to the following folder on the computer where Tableau Server is installed:
`C:\Program Files\Tableau\Tableau Server\SAML`
9. On the SAML tab in the Tableau Server Configuration dialog box, enter the location to the file in the **SAML IdP metadata file** text box:



10. Click OK. Tableau Server is now configured for SAML authentication.

Configure a Server Cluster for SAML

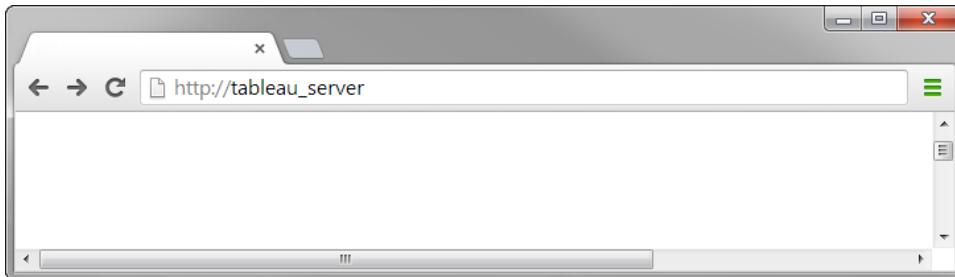
When you configure a Tableau Server cluster to use SAML, you place the same SAML certificate, SAML key, and SAML IdP metadata files on every computer that's running a Tableau application server process (also known as `vizportal.exe`). To configure a Tableau Server cluster to use SAML:

1. Configure the primary Tableau Server as described in the procedure above.
2. Place the same SAML certificate, SAML key, and SAML IdP metadata files that you used for the primary on each Tableau Worker that is running an application server process. Use the same folder location on the workers that you used on the primary. You do not need to do any additional configuration on the workers.

For example, consider a cluster that includes a primary Tableau Server and two workers. Application server processes are running on the primary and on Worker 2 and Worker 3. In this situation, you [configure the primary Tableau Server for SAML](#), and then copy the same SAML certificate, SAML key, and SAML IdP metadata files to the Worker 2 and Worker 3 computers. On the worker computers, put the SAML files in the `C:\Program Files\Tableau\Tableau Server\SAML` folder, just as they are on the primary computer.

Test Your Configuration

Test your SAML configuration by opening a new web browser instance and typing the Tableau Server name in the URL window:



You should note that the sign in prompt that appears is from your IdP and not Tableau Server:

A screenshot of a sign-in dialog box. At the top, there is a green checkmark icon followed by the text "Please sign on first and we'll send you right along.". Below this, there are two input fields: one labeled "Username" containing "jsmith" and another labeled "Password" containing a series of asterisks. At the bottom of the dialog are two buttons: a teal-colored "Sign On" button on the left and a dark blue "Cancel" button on the right.

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide Single Sign-on (SSO) for SAP HANA. HANA SSO is not dependent on SAML authentication to Tableau Server.

Note: You do not need to use SAML sign on with Tableau Server in order to use HANA SSO. You can sign in to Tableau Server using whatever method you choose.

With SSO for SAP HANA, Tableau Server functions as an Identity Provider (IdP) and this configuration allows you to provide a single sign-on experience for users making SAP HANA connections. As part of the configuration, you need to acquire a SAML certificate and key file for Tableau Server (these should be a public key certificate and private key). You need to also install the signed certificate in HANA. You can generate the certificate and key yourself, or get them from a Certificate Authority. For more information on generating a certificate/private key and configuring SAP HANA, see the [Tableau Knowledgebase](#).

Note: The SAP HANA driver version 1.00.9 or later must be installed on Tableau Server in order to use SSO for SAP HANA. The driver cannot encrypt the SAML assertion, so you may want to enable encryption for the SAML connections. For more information, see the [Tableau Knowledgebase](#).

Configure SSO for SAP HANA

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 9.2 folder.

For example:

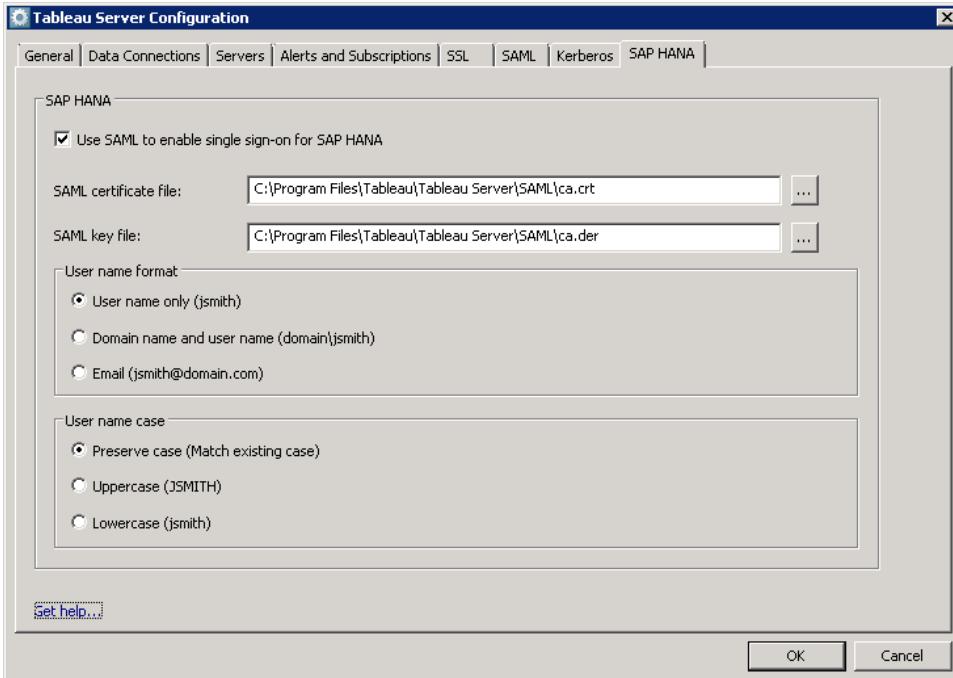
C:\Program Files\Tableau\Tableau Server\SAML

You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. After you install Tableau Server, run the Configuration utility (**Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**), and then click the **SAP HANA** tab.
3. Select **Use SAML to enable single sign-on for SAP HANA** and provide the location for each of the following:

SAML certificate file—A PEM-encoded x509 certificate with the file extension **.crt** or **.cert**. This file is used by Tableau Server, and must also be installed on HANA.

SAML private key file—A DER-encoded private key file that is not password protected, and that has the file extension **.der**. This file is only used by Tableau Server.



4. Select the format of the user name.
5. Select the case for the user name. This determines the case of the name when it is forwarded to the SAP HANA identity provider (IdP).

Troubleshoot SAML

Use the following topics to troubleshoot SAML issues.

SAML and Enable Automatic Logon

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select **Enable automatic logon**. **Enable automatic logon** and SAML cannot both be used on the same server installation.

HTTP Status 500 error when configuring SAML

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:

```
org.opensaml.saml2.metadata.provider.MetadataProviderException:  
User specified binding is not supported  
by the Identity Provider using profile
```

urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.
- The IdP is configured to use SP-initiated authentication. (IdP-initiated authentication is not supported.)>
- The IdP is configured to use HTTP-POST requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in [SAML Requirements on page 475](#).

Signing In from the Command Line

Even if Tableau Server is configured to use SAML, it is not used if you sign in to Tableau Server using the command line tools [tabcmd on page 552](#) or the [Tableau Data Extract command line utility](#) (provided with Tableau Desktop).

Login Failed

Login can fail with the following message:

```
Login failure: Identity Provider authentication successful for
user <username> from IdP. Failed to find the user in Tableau
Server.
```

This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `jsmith` it must be stored in Tableau Server as `jsmith`.

SAML Error Log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see [Archive Log Files on page 641](#).

Note: In Tableau Server 9.0 and later, to log SAML-related events, both `wgserver.log.level` and `vizportal.log.level` must be set to debug. For more information, see [Change Logging Levels](#) on page 652.

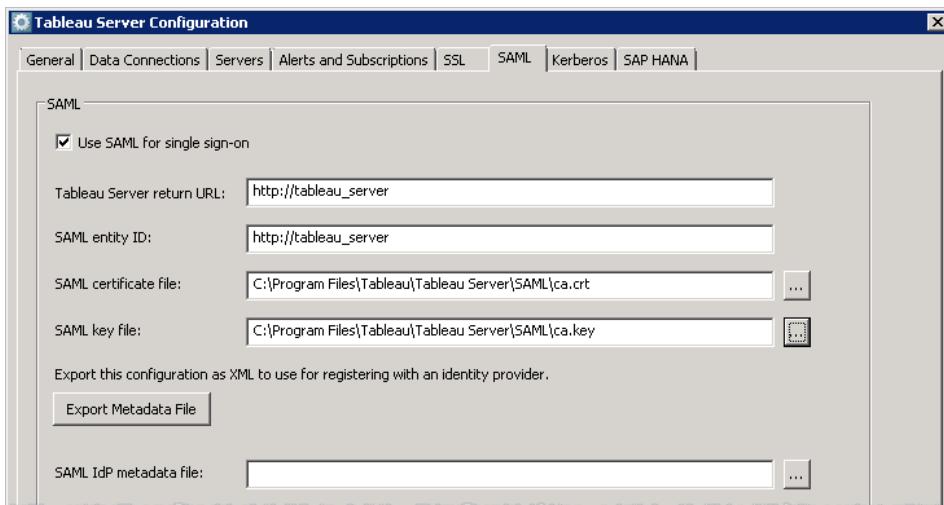
Check for SAML errors in the following files in the unzipped log file snapshot:

```
\wgserver\wgserver-<n>.log  
\vizportal\vizportal-<n>.log  
\wgserver\production.<nnnn>_<yyyy_mm_dd_hh_mm_ss>.log
```

In Tableau Server 9.0 and later, the application process (`vizportal.exe`) handles authentication, so SAML responses are logged by that process. The SAML setup process logs information in the logs for the api server process (`wgserver.exe`).

Trailing Slash

On the SAML tab, confirm that the **Tableau Server return URL** does not end with a trailing slash (correct: `http://tableau_server`; incorrect: `http://tableau_server/`):



Confirm Connectivity

Confirm that the Tableau Server you are configuring has either a routeable IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running telnet on Tableau Server and attempting to connect with the SAML IdP. For example: C :\telnet 12.360.325.10 80

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.

SSL

SSL (Secure Sockets Layer) is a standard security technology that establishes an encrypted link between a web server and clients. To use SSL, you need to install an SSL certificate on Tableau Server.

Tableau Server also supports mutual (two-way) SSL as an encryption and authentication method.

You can configure Tableau Server to use SSL in the following ways:

- Use SSL for external HTTP traffic.
- Use mutual (two-way) SSL between clients (Tableau Desktop, web browsers, and tabcmd.exe) and Tableau Server.
- Use SSL for all HTTP traffic between internal server components and the repository.

If you are using mutual SSL, each client also needs a certificate.

Note: Tableau Server uses SSL only for user authentication. Tableau Server does not use SSL to handle permissions and authorization for content hosted on Tableau Server, such as workbooks.

For more information, see the following topics:

Quick Start: Mutual (Two-Way) SSL Authentication

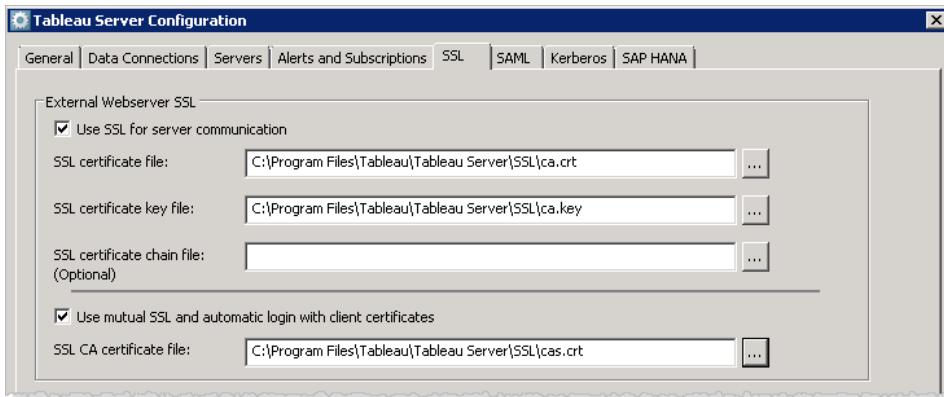
To provide a secure automatic sign-in experience with Tableau across all devices, use mutual SSL. With mutual SSL, when a client (Tableau Desktop, a web browser, or tabcmd.exe) with a valid certificate connects to Tableau Server, Tableau Server confirms the existence of a valid client certificate and automatically signs the user in, using the user name it finds in the certificate. If the client does not have a valid SSL certificate, Tableau Server refuses the connection. To configure Tableau Server for mutual SSL, you need the following:

- **Certificate file:** A PEM-encoded x509 certificate file with the extension **.crt**.
- **Certificate key file:** An RSA or DSA key file that is not password-protected and that has a **.key** file extension.
- **Certificate CA file:** A PEM-encoded x509 certificate file with the extension **.crt**.
- **Client certificate on client devices:** Tableau Server queries the client for an SSL certificate that it trusts, before it allows a connection to Tableau Server.

The certificate files should be in the `C:\Program Files\Tableau\Tableau Server\SSL` folder.

1 Use SSL for server communication

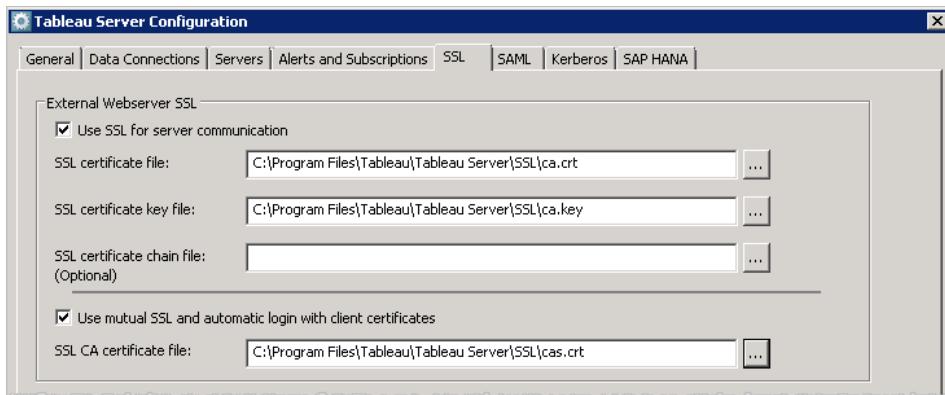
To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the Tableau Server Configuration utility after you have installed Tableau Server. Click the **SSL** tab, and then select **Use SSL for server communication**.



Specify values for **SSL certificate file** and **SSL certificate key file**.

2 Use mutual SSL

To add mutual authentication between the server and each client and allow for automatic sign-in experience, select **Use mutual SSL and automatic login with client certificates**.



Specify the **SSL CA certificate file**. The SSL CA certificate file identifies the certificate of the Certificate Authority (for example, Verisign). For information on how to configure multiple Certificate Authorities, see [Configure External SSL on the next page](#).

Click **OK** to close the Tableau Server Configuration utility, and then start Tableau Server.

Additional options for mutual SSL

Fallback authentication

When Tableau Server is configured for mutual SSL, authentication is automatic and a client must have a valid certificate. If you need a fallback option, use the `tabadmin set ssl.client_certificate_login.fallback_to_password true` command to configure Tableau Server to allow user name / password authentication. Setting this option to `true` allows Tableau Server to fall back to using user name and password for authentication if SSL certificate authentication fails.

Username mapping

When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate so the client can be automatically signed in. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

- **Local Authentication**—Tableau Server uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the CN (Common Name) by using the `tabadmin set ssl.client_certificate_login.mapping_strategy` command.

Certificate Revocation List (CRL)

You may need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly. To specify a CRL, use the `tabadmin set ssl.revocation.file` command. For more information, see [tabadmin set Commands](#).

Configure External SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the web browser and the server or Tableau Desktop and the server is protected. Steps on how to configure the server for SSL are described in the topic below; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server. If you are running a Tableau Server cluster and you want to use SSL, see [Configure SSL for a Cluster](#) on page 493, below, for recommendations.

1. Acquire an Apache SSL certificate from a trusted authority (for example, Verisign, Thawte, Comodo, GoDaddy). You can also use an internal certificate issued by your company. Wildcard certificates, which allow you to use SSL with many host names within the same domain, are also supported.

Some browsers will require additional configuration to accept certificates from certain providers. Refer to the documentation provided by your certificate authority.

2. Place the certificate files in a folder named SSL, parallel to the Tableau Server 9.2 folder. For example:

C:\Program Files\Tableau\Tableau Server\SSL

This location gives the account that's running Tableau Server the necessary permissions for the files.

Note: You may need to create this folder.

3. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 9.2 > Configure Tableau Server** on the Start menu.
4. In the Configuration Tableau Server dialog box, select the **SSL** tab.
5. Select **Use SSL for server communication** and provide the location for each of the following certificate files:
 - **SSL certificate file**—Must be a valid PEM-encoded x509 certificate with the extension .crt.
 - SSL certificate key file**—Must be a valid RSA or DSA key that has an embedded passphrase, and is not password protected with the file extension .key.
 - SSL certificate chain file (Optional for Tableau Server, required for Tableau Mobile and Tableau Desktop on the Mac)**—Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate. All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).
6. (optional) If you are using SSL for server communication and want to configure SSL communication between Tableau Server and clients using certificates on both the server and clients:
 - Select **Use mutual SSL and automatic login with client certificates**.
 - In **SSL CA certificate file**, browse to the location for the certificate file. The SSL CA certificate file must be a valid PEM-encoded x509 certificate with the extension .crt.

Note: If you have multiple trusted Certificate Authorities (CAs) you can copy and paste the entire contents of each CA certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines, into a new file, then save the file as CAs.crt. In **SSL CA certificate file**, browse to the location of this new file.

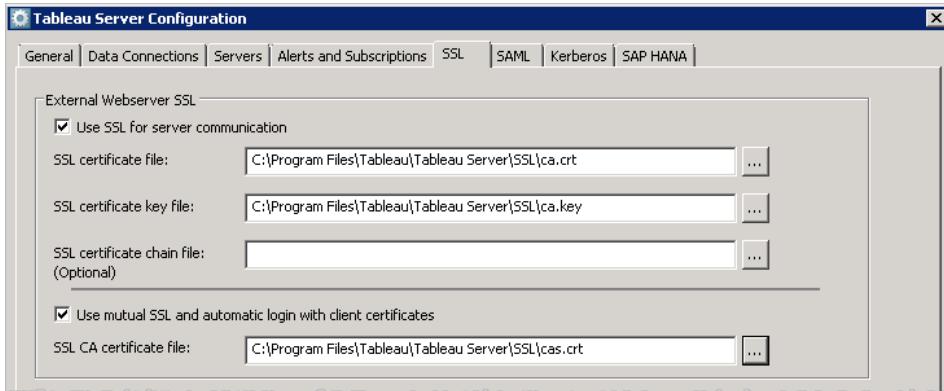
7. Click **OK**. The changes will take effect the next time the server is restarted.

When the server is configured for SSL, it accepts requests to the non-SSL port (default is port 80) and automatically redirects to the SSL port 443.

Note: Tableau Server only supports port 443 as the secure port. It cannot run on a computer where another application is using port 443.

SSL errors are logged in the install directory at the following location. Use this log to troubleshoot validation and encryption issues:

```
C:\ProgramData\Tableau\Tableau  
Server\data\tabsvc\logs\httpd\error.log
```



Configure SSL for a Cluster

You can configure a Tableau Server cluster to use SSL. If the primary Tableau Server computer is the only node that is running the gateway process (which it does by default), then that's the only place where you need to configure SSL. See the procedure above for steps.

SSL and Multiple Gateways

A highly available Tableau Server cluster can include multiple gateways, fronted by a load balancer ([learn more](#)). If you are configuring this type of cluster for SSL, you have two choices:

- **Configure your load balancer for SSL.** Traffic is encrypted from the client web browsers to the load balancer. Traffic from the load balancer to the Tableau Server gateway processes is not encrypted. No SSL configuration in Tableau Server is required, it's all handled by your load balancer.
- **Configure Tableau Server for SSL:** Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. See the procedure below for details.

Configure a Server Cluster for SSL

When you configure a Tableau Server cluster to use SSL, you place the SSL certificate and key files on every computer that's running a gateway process. To configure a Tableau Server cluster to use SSL:

1. Configure the load balancer for SSL passthrough. Refer to your load balancer's documentation for assistance.
2. Make sure that the SSL certificate you use was issued for the load balancer's host name.
3. Configure the primary Tableau Server node as described in the procedure above.
4. Place the same SSL certificate and key file that you used for the primary on each Tableau Server worker node that is running a gateway process. Use the same folder location on the workers that you used on the primary.

If you are using mutual ssl, place the SSL CA certificate file you used for the primary on each worker node that is running a gateway process. Use the same folder location that you used on the primary.

You do not need to do any additional configuration on the workers.

For example, say you have a cluster that includes a primary Tableau Server node and three worker nodes with gateway processes are running on the primary, Worker 2 and Worker 3. In this situation, you [configure the primary Tableau Server for SSL](#), then copy the same SSL certificate and key files to Worker 2 and Worker 3. Because these files are in `C:\Program Files\Tableau\Tableau Server\SSL` folder on the primary, they are in that same location on Worker 2 and Worker 3.

You can configure a Tableau Server cluster to use SSL. If the primary Tableau Server computer is the only node that is running the gateway process (which it does by default), then that's the only place where you need to configure SSL. See the procedure above for steps.

Configure Internal SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository.

1. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**.
2. In the Tableau Server Configuration dialog box, click the **SSL** tab.
3. Select one of the following options:
 - **Required for all connections**
When this option is selected, Tableau Server uses SSL for communications between the repository database and other server components. In addition, direct connections to Tableau Server (connections using the "tableau" or "readonly" users) must use SSL.
 - **Optional for direct user connections**
This option configures Tableau Server to use SSL between the repository and other server components and supports but does not require SSL for direct connections by "tableau" or "readonly" users.
 - **Off for all connections** (the default)
This option disables SSL for internal communications and direct connections.

4. Click **OK**.

For more information on downloading the public certificate for direct connections, see [Configure SSL for Direct Connections](#) below.

Configure SSL for Direct Connections

When Tableau Server is configured to use SSL internally, SSL connections are either optional or required for client machines making direct connections to the Tableau Server repository database. Direct connections include those using the "tableau" user or the "readonly" user.

To use SSL with direct connections, generate the SSL certificate file and copy it to the computer from which you will be making the direct connections.

1. Generate the SSL certificate file using the [regenerate_internal_tokens](#) on page 607 command.
2. Locate the SSL cert file by looking in the workgroup.yml file on the primary Tableau Server node.

The workgroup.yml file is located on the primary Tableau Server node in the `\ProgramData\Tableau\Tableau Server\data\tabsvc\config` folder.

The location of the SSL certificate and key files are listed in the file. For example:

```
pgsql.ssl.cert.file: C:/ProgramData/Tableau/Tableau Server-
/data/tabsvc/config/pgsql/server.crt
```

```
pgsql.ssl.key.file: C:/ProgramData/Tableau/Tableau Server-/data/tabsvc/config/pgsql/server.key
```

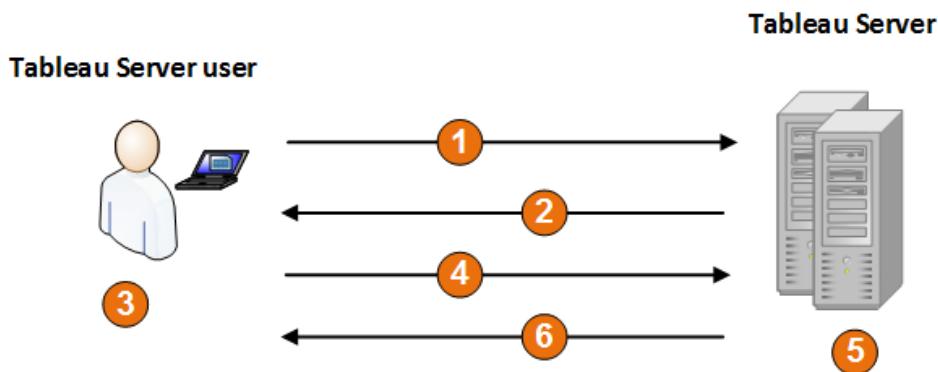
3. Copy the cert file to the computer that will be making the direct connection and import them into the computer's certificate store using the documentation from the operating system manufacturer.

Note: Do not copy the key file. This file should only be on the server.

How Mutual SSL Authentication Works

Mutual (or two-way) SSL authentication provides a combination of an encrypted data stream, mutual authentication of both server and client, and automatic sign-in convenience. To use mutual SSL with Tableau Server, you need an SSL certificate for Tableau Server and a certificate on each client that will connect to Tableau Server. You also need to configure Tableau Server to use mutual SSL. Tableau Server and client verify that each other has a valid certificate, and Tableau Server signs in the user automatically, based on the user name that Tableau Server finds in the client certificate.

The following image shows the sequence of events that occurs with mutual SSL.



- 1 The user navigates to Tableau Server.
- 2 Tableau Server sends its SSL certificate to the client computer.
- 3 The client computer verifies the Tableau Server certificate.

- 4 The client computer sends its certificate to Tableau Server.
- 5 Tableau Server verifies the client certificate.
- 6 Tableau Server signs the user in using the user name from the certificate.

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. Tableau Server can perform client mapping using different strategies, depending on the content of the client certificates in your organization.

This topic discusses the options for how a client certificate is mapped to a user identity and explains how to change how the server performs the mapping if necessary. Note that in order for you to understand how the mapping is performed and whether you need to change the default mapping for Tableau Server, you must understand how client certificates are structured in your organization.

- [Mapping options](#)
- [Changing the certificate mapping](#)
- [Ambiguous user names in multi-domain organizations](#)

Mapping options

Tableau Server can map a client certificate to a user identity by using one of the following approaches:

- **Use Active Directory.** If Tableau Server was configured during setup to use Active Directory for user authentication, when Tableau Server receives a client certificate, Tableau Server passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any explicit user name information in the certificate is ignored.

Note: This approach requires client certificates to be published for the user accounts in Active Directory.

- **Using the user principal name (UPN).** A client certificate can also be created so that the user name is in the user principal name (UPN) field of the certificate. Tableau Server can read the UPN value and use it to map to a user in Active Directory or to a local user.

- **Using the common name (CN).** A client certificates can also be created so that the user name is in the common name (CN) field of the certificate. Tableau Server can read the CN value and use it to map to a user in Active Directory or to a local user.

If the server is configured for Active Directory authentication, and if you're using UPN or CN mapping, the user name should be in one of these formats: `username`, `domain\username`, or `username@domain`. For example, the name must be `asmith`, `example.org\asmith`, or `asmith@example.org`.

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Changing the certificate mapping

The approach that Tableau Server uses to map a client certificate to a user identity is specified using the `ssl.client_certificate_login.mapping_strategy` setting. Possible values for this setting are `ldap` for Active Directory mapping, `upn` for UPN mapping, or `cn` for CN mapping.

When you first install and configure Tableau Server, the server makes default settings for the mappings. By default, if Tableau Server is configured to use Active Directory, the server also uses Active Directory for mapping the certificate to the user identity (`ssl.client_certificate_login.mapping_strategy` is set to `ldap`). If the server is configured to use local authentication, by default the server gets the user name value from the UPN field in the certificate (`ssl.client_certificate_login.mapping_strategy` is set to `upn`).

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the `tabadmin set` command to change the value of `ssl.client_certificate_login.mapping_strategy`. As an example, the following sequence of commands shows how to set the mapping to use the CN value:

```
tabadmin stop
tabadmin set ssl.client_certificate_login.mapping_strategy cn
tabadmin configure
tabadmin start
```

Ambiguous user names in multi-domain organizations

Under some circumstances, the user name in a UPN or CN field in the certificate can be ambiguous. This can have unexpected results when the user name is mapped to a user identity on the server. This can occur when all of the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.
- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains (for example,

asmith@example.org and asmith@example.com).

- The user name in the UPN or CN fields of the certificate does not include the domain as part of the user name—for example, the certificate simply includes asmith.

If Tableau Server gets a user name that has no domain, the server maps the user name to an identity using the default domain. This can result in incorrectly mapping the user name.

Important: Incorrect mapping of the user name can result in a user being granted an identity and permissions for a different user. To avoid this issue, you should make sure that the client certificates include full user names, with the domain.

To resolve this issue, the system administrator should make sure that the user name in the user's certificate is fully qualified with a domain name using the format asmith@example.org or example.org\asmith.

Troubleshooting Mutual SSL Authentication

This topic describes possible mutual (two-way) SSL authentication issues and their causes, the messages that users might see, and possible mitigation for the issues.

- The client is missing a certificate
- The client doesn't support mutual SSL authentication
- Client certificates are not published to Active Directory
- Users unexpectedly see a sign-in dialog box that displays an error message
- The user name in the UPN or CN fields is missing or invalid
- The user is signed in using unexpected user name (LDAP mapping)
- The user is signed in as incorrect user (UPN or CN mapping)

For more information about mutual SSL authentication and LDAP, UPN, and CN user mapping, see the following topics:

- [Quick Start: Mutual \(Two-Way\) SSL Authentication on page 489](#)
- [Mapping a Client Certificate to a User During Mutual Authentication on page 497](#)

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

The client is missing a certificate.

If the client has no client certificate, the user sees this message during authentication:

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

To resolve the issue, the user should contact the system administrator to generate a certificate for the client computer.

Invalid user name or password

The client doesn't support mutual SSL authentication.

Versions of Tableau Desktop older than version 9.1 do not support mutual SSL authentication. If an older version of Tableau Desktop is used to connect to Tableau Server that is configured for mutual SSL authentication, the following can occur:

- If Tableau Server is configured to use fallback authentication, the client displays a sign-in dialog box and the user can enter a user name and password.
- If the server is not configured to use fallback authentication, the user sees the following message and cannot connect to the server:

Invalid user name or password

For more information about fallback authentication, see [Quick Start: Mutual \(Two-Way\) SSL Authentication](#) on page 489.

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

Client certificates are not published to Active Directory.

If Tableau Server is configured to use Active Directory for authentication, and if user mapping is set to LDAP, Tableau Server sends the client certificate to Active Directory for authentication. However, if client certificates have not been published to Active Directory, authentication fails and the user sees the following message:

We couldn't find your user name in the client certificate.
Contact your Tableau Server administrator or sign in using your Tableau Server account.

To resolve this issue, the system administrator should make sure that client certificates are published to Active Directory. Alternatively, the server should be configured to use a different user mapping (UPN or CN), and the system administrator should be sure that client certificates contain user names in the UPN or CN fields.

Users unexpectedly see a sign-in dialog box that displays an error message

If Tableau Server is configured to use mutual SSL authentication and certificates are available for use with users' computers, a user should not see a sign-in dialog box, because Tableau Server uses the certificate to authenticate the user. However, if the server does not recognize the user name in the certificate, the user sees a sign-in dialog box with an error message that indicates why the certificate was not used. This can occur when all of the following conditions are true:

- Fallback authentication is enabled.
- If the server is using UPN or CN mapping, the user name in the certificate's UPN or CN field is not recognized. If the server is using LDAP mapping, the certificate is not mapped to the user in Active Directory.

To resolve this issue, the system administrator should do the following, depending on how user mapping is configured on Tableau Server:

- LDAP mapping: Make sure that the certificate is linked to the user, that the certificate is available for use with the user's computer, and that the user is configured as a Tableau Server user.
- UPN or CN mapping: Make sure that the certificate is available for the user's computer, that the user name is in the certificate's UPN or CN field, and that the user name matches the user name on Tableau Server (including domain).

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.

Certificate does not contain a valid Tableau Server user name.

The user name in the UPN or CN fields is missing or invalid

When Tableau Server is configured to use UPN or CN mapping, the server reads the user's name from the UPN or CN field of the certificate and then looks up the user name in Active Directory or in the local repository on Tableau Server. (The specific field that the server reads depends on which mapping—UPN or CN—the server is configured to use.) If the field that is supposed to contain the user name has nothing in it, the user sees the following message:

We couldn't find your user name in the client certificate.
Contact your Tableau Server administrator.

If a client certificate contains a user name but Active Directory and Tableau Server don't recognize the user name, the user sees the following message:

Certificate does not contain a valid Tableau Server user name.

This can occur when all of the following conditions are true:

- Tableau Server is configured to use UPN or CN mapping.
- Fallback authentication is not enabled.
- The client certificate has no user name in the UPN or CN field, or the user name in the UPN or CN field does not match a user name in Active Directory or on Tableau Server.

To resolve this issue, the system administrator should make sure that the user's certificate has the correct user name in the UPN or CN fields of the certificate.

The user is signed in using an unexpected user name (LDAP mapping)

When the server is configured to use Active Directory authentication and LDAP mapping, the certificate is linked to a user in Active Directory. If the certificate contains a user name in the UPN or CN field, that user name is ignored.

If the intention is that the user should be signed in with the user name in the UPN or CN fields, the server should be configured to use UPN or CN mapping.

The user is signed in as the incorrect user (UPN or CN mapping)

Under some circumstances, the user name in a UPN or CN field in the client certificate can be ambiguous. The result is that a user is signed in to the incorrect identity.

For more information about the conditions under which this issue can occur, see [Ambiguous user names in multi-domain organizations](#) in the topic [Mapping a Client Certificate to a User During Mutual Authentication](#) on page 497.

Kerberos

Kerberos is a three-way authentication protocol that relies on the use of a trusted third-party network service called the Key Distribution Center (KDC) to verify the identity of computers and provide for secure connections between the computers through the exchange of *tickets*. These tickets provide mutual authentication between computers or services, verifying that one has permission to access the other.

Tableau Server supports Kerberos authentication in an Active Directory Kerberos environment, with authentication to Tableau Server being handled by Kerberos.

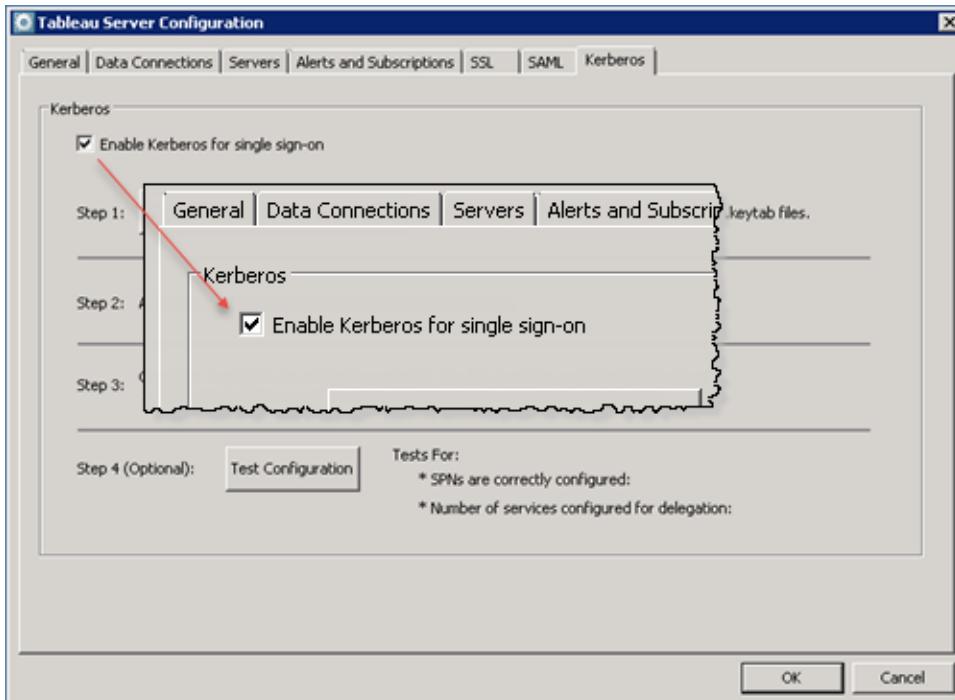
Note: The Kerberos support in Tableau Server is for user authentication. It does not handle internal permissions and authorization related to Tableau Server content, such as workbooks.

Quick Start: Single Sign-On with Kerberos

Tableau Server now supports Kerberos-based single sign-on (SSO). Users with Active Directory (AD) accounts in a Kerberos-enabled environment can now use SSO to connect to Tableau Server from Tableau Desktop and web browsers. In addition, Tableau Server can use Kerberos for authentication to Kerberos-enabled Microsoft SQL and MSAS data sources. When Tableau Server is configured for Kerberos, you can make SSO connections to Cloudera Impala databases using server managed credentials for Impala LDAP authentication.

1 Configure Tableau Server

After you install Tableau Server, run the Tableau Server Configuration utility. On the **Kerberos** tab select **Enable Kerberos for single sign-on**.



2 Generate the Configuration Script

Click **Export Kerberos Configuration Script** to generate a batch file that will configure Kerberos in AD for Tableau Server.



Save the file and then send it to your AD domain administrator to run.

3 Run the Configuration Script

The domain administrator needs to run the script from a command prompt on any computer in the domain by typing the name of the script.

When your domain administrator runs the configuration script, the script registers Service Principal Names (SPNs) for Tableau Server using the Run As User account, and generates a .keytab file for your environment. (The .keytab file is created in a \keytabs folder in the folder where the script was run.)

Have the domain administrator send you a copy of the .keytab file.

4 Copy the .keytab File

On the **Kerberos** tab of the Tableau Server Configuration utility, enter the path to the .keytab file in the text box in Step 3.



The utility will copy the file to each gateway node in the Tableau Server installation.

Click **Test Configuration** to verify that the configuration is correctly set up. If the SPNs are correctly set up, the test should display an OK. The number of services configured for delegation will be 0 (zero) unless you have completed the steps below in **Configure Kerberos Delegation in AD**.

Configure Kerberos Delegation in AD

To use Kerberos Authentication with SQL Server or MSAS data source, or to make SSO connections to Cloudera Impala, you need to configure Kerberos delegation in AD. You don't need to complete these steps if you will only be using Kerberos SSO to connect to Tableau Server. K-

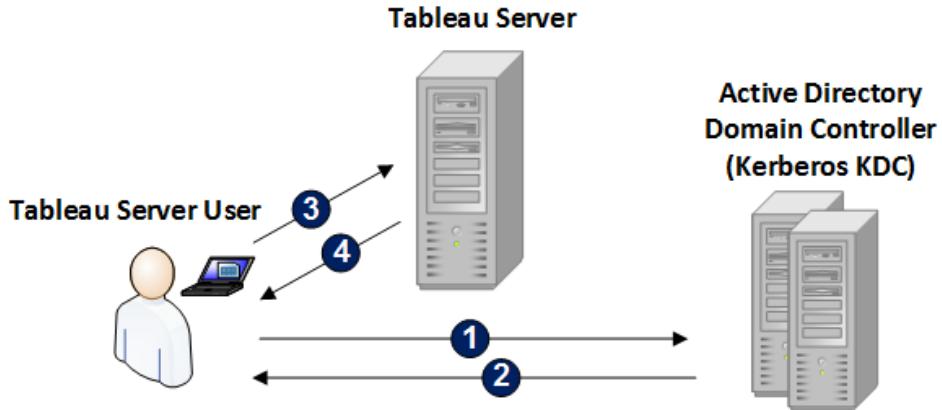
To configure Kerberos delegation in AD:

- Enable the Run As User to act as the operating system. For more information, see [Enable Run As User to Act as the Operating System](#) on page 511.
- Enable Kerberos delegation in AD. This step is specific to the supported connection type (s) that you will be using with Tableau:
 - **SQL Server** - See [Enabling Kerberos Delegation for SQL Server](#) in the Tableau Knowledge Base.
 - **MSAS** - See [Enabling Kerberos Delegation for MSAS](#) in the Tableau Knowledge Base.
 - **Impala** - See [Enabling Delegation for Cloudera Impala](#). in the Tableau Knowledge Base.

Kerberos Authentication in Tableau Server

When you configure Tableau Server for Kerberos in an Active Directory (AD) environment, the AD domain controller also serves as the Kerberos Key Distribution Center (KDC) and issues Ticket Granting Tickets to the other nodes in the domain. Users authenticated by the KDC do not have to authenticate further when connecting to Tableau Server.

The following is a diagram of the authentication workflow.



Kerberos Requirements

To use Kerberos authentication with Tableau Server, you need the following:

- **Windows Server:** Tableau Server must be installed on a server version of Windows. Non-server versions (including Windows 7 and Windows 8) do not support the `ktpass` command required for generating a keytab file.
- **Active Directory:**
 - Tableau Server must use Active Directory (AD) for authentication.
 - The domain must be an AD 2003 or later domain.
- **Run As User account:**
 - The Run As User account (the Tableau Server service account) must be an AD domain account. Local accounts, including NTAUTHORITY\NetworkService will not work.
 - The Run As User account must be in the same domain as the database services that will be delegated.

- Constrained delegation: The Run As User account must be granted access to the target database Service Principal Names (SPNs).
- Data Source authentication: If you plan to use Kerberos to authenticate to Microsoft SQL Server or MSAS databases, or with delegation for Single sign-on (SSO) to Cloudera Impala, enable the Run AS User account to act as part of the operating system. For more information, see [Enable Run As User to Act as the Operating System on page 511](#).
- **Single-Sign On (SSO)**: Users must be granted a Kerberos Ticket Granting Ticket (TGT) from Active Directory when they sign into their computers. This is standard behavior for domain-joined Windows computers and standard for Mac computers that use AD as their network account server. For more information on using Mac computers and Active Directory, see [Join your Mac to a network account server](#) in the Apple Knowledge Base.
- **External Load Balancer/Proxy Server**: If you are going to use Tableau Server with Kerberos in an environment that has external load balancers (ELBs) or proxy server, you need to set these up before you configure Kerberos in the Tableau Server Configuration utility. See [Add a Load Balancer on page 106](#) and [Configure Tableau to Work with a Proxy Server on page 449](#) for more information.
- **Smart Card Support**: Smart cards are supported when users sign into their workstations with a smartcard and this results in a Kerberos TGT being granted to the user from Active Directory.
- **iOS Browser Support**: An iOS user can use Kerberos authentication with mobile Safari if a Configuration Profile specifying the user's Kerberos identity is installed. See [Configuring an iOS Device for Kerberos Support](#) in the Tableau Knowledge Base.

For more information about browser support for Kerberos SSO, see [Browser Support for Kerberos SSO to Tableau Server](#) in the Tableau Knowledge Base.

External load balancers:

- If you are using an external load balancer or a reverse proxy, complete the configuration for the external load balancer or reverse proxy before configuring Tableau Server for Kerberos.

Note: If you configure these after configuring Tableau Server for Kerberos, the configuration script generated by the Tableau Server Configuration utility might use the wrong host names. See [Add a Load Balancer on page 106](#) and [Configure Tableau to Work with a Proxy Server on page 449](#) for more information.

To use Kerberos authentication for delegated access with data sources:

- **Data Sources:**
 - The supported data sources (SQL Server, MSAS, and Cloudera Impala) must be configured for Kerberos authentication.
 - The data sources must be on the same domain as Tableau Server (users can be on different domains).

Kerberos connections to Tableau Server are supported in the following configurations:

- Tableau Server requires constrained delegation, where the Run As User account is specifically granted rights to the target database SPNs. Unconstrained delegation is not supported.

Configure Kerberos

You can configure Tableau Server to use Kerberos. This allows you to provide a single sign-on experience across all the applications in your organization. Before you configure Tableau Server for Kerberos make sure you meet the [Kerberos Requirements on page 506](#).

1. Open a command prompt as an administrator and change directories to the location of Tableau Server's bin directory. The default location is C:\Program Files\Tableau\Tableau Server\9.0\bin.
2. Type the following command to stop Tableau Server:

```
tabadmin stop
```
3. Open the Tableau Server Configuration Utility (**Start > All Programs > Tableau Server 9.2 > Configure Tableau Server**), and then click the **Kerberos** tab.
4. Select **Enable Kerberos for single sign-on**.
5. Click **Export Kerberos Configuration Script**. The generated script configures your Active Directory domain to use Kerberos with Tableau Server. For more information, see [Kerberos Configuration Script on page 510](#).



Note: Verify the host names in the setspn lines of the script. If you are using an external load balancer or a reverse proxy, the host names should match the name

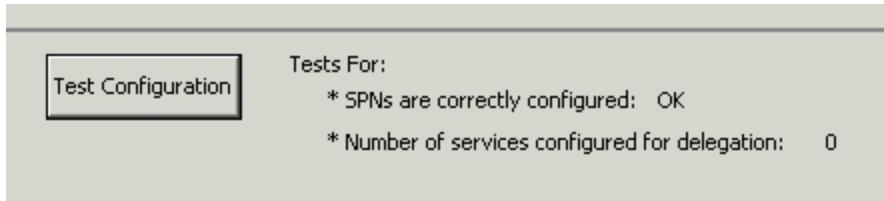
you used when you configured Tableau Server for the load balancer or proxy. If you have not configured Tableau Server for your proxy or external load balancer, do that and then re-export the Kerberos configuration script to ensure it has the correct host names. See [Add a Load Balancer](#) on page 106 and [Configure Tableau to Work with a Proxy Server](#) on page 449.

6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the .keytab file. The domain administrator should do the following:
 - Review the script to verify it contains correct values.
 - Run the script at a command prompt on any computer in the domain by typing the script name (not by double-clicking the script in Windows Explorer).

The script creates a file, `kerberos.keytab`, in a `\keytabs` folder in the location that the script was run.
7. Save a copy of the .keytab file created by the script to the Tableau Server computer. In Step 3, enter the path to the .keytab file, or click the browse button to navigate to the file. The keytab file will be copied to all the gateway nodes in your Tableau Server installation when you click **OK** in the Configuration utility.

Note: Do not rename the .keytab file. The script creates a file named `kerberos.keytab` and you need to save it with this name.

8. (optional) Click **Test Configuration** to confirm that your environment is configured correctly to use Kerberos with Tableau Server.



If you have not configured any data sources for Kerberos delegation, 0 is shown for the **Number of services configured for delegation**.

9. Click **OK** to save your Kerberos configuration.
10. Start Tableau Server.

Confirm Your SSO Configuration

Once Tableau Server has restarted, test your Kerberos configuration from a web browser on a different computer by typing the Tableau Server name in the URL window:



You should be automatically authenticated to Tableau Server.

Kerberos Configuration Script

When you click **Export Kerberos Configuration Script** in the Tableau Server Configuration utility, the `KerberosConfig.bat` script is generated. This script registers the Service Principal Names (SPNs) for Tableau Server in Active Directory (AD) and generates a Kerberos `.keytab` file.

SPNs - The script uses the `setspn` utility to register the SPNs for Tableau Server, using the Run As User account. These SPNs are used for generating the `.keytab` file, and for authenticating web browser connections to Tableau Server.

.keytab - The script uses the `ktpass` utility, to generate a `kerberos.keytab` file, located in the `\keytabs` folder in the folder where the script was run. The `.keytab` file contains the shared secret key for Tableau Server.

Note: The `setspn` and `ktpass` utilities may generate warning or errors. You can ignore these errors and warnings if the utilities run to completion.

Enable Kerberos Delegation

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer's name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Tableau Server requires constrained delegation, with the Run As User account specifically granted delegation rights to the target database Service Principal Names (SPNs). Delegation is not enabled in Active Directory by default.

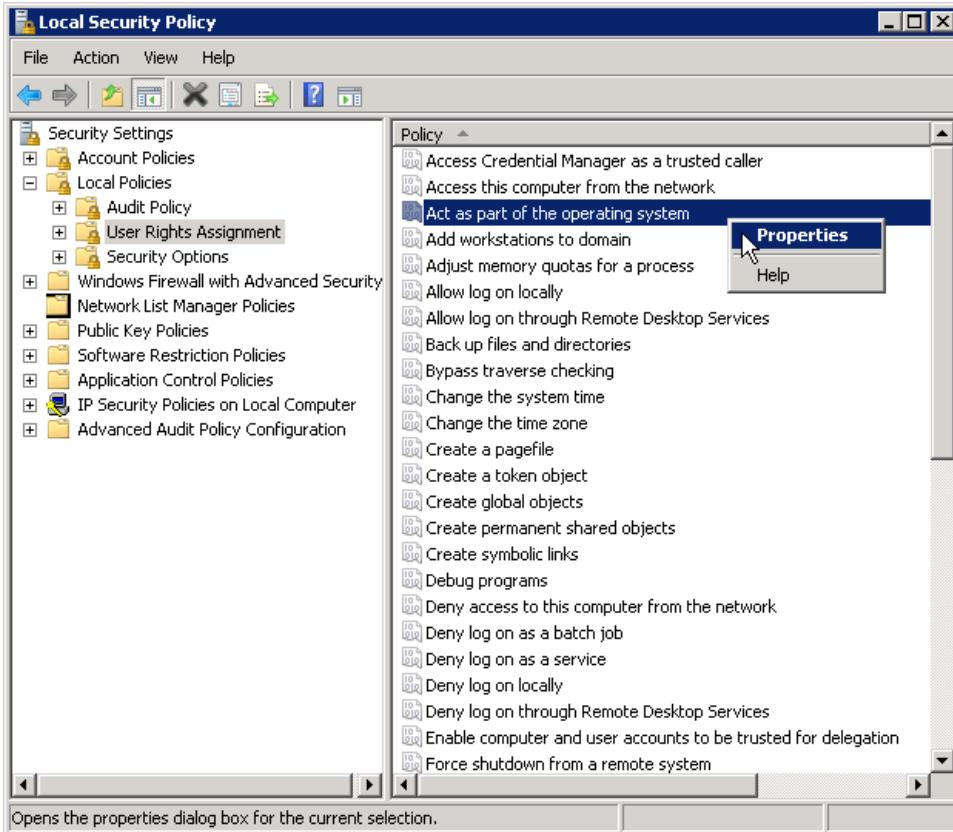
To configure Kerberos delegation:

1. On all nodes in Tableau Server, configure the Run As User to act as part of the operating system. For more information, see [Enable Run As User to Act as the Operating System below](#).
2. In Active Directory:
 - Configure SPNs for the data sources you will be using.
 - Enable Kerberos delegation for the data sources' SPNs
3. Enable delegation for data connections:
 - **SQL Server**—See [Enabling Kerberos Delegation for SQL Server](#) in the Tableau Knowledge Base.
 - **MSAS**—See [Enabling Kerberos Delegation for MSAS](#) in the Tableau Knowledge Base.
 - **Cloudera Impala**—In this case Kerberos must be enabled on Tableau Server but the connection does not use Kerberos. See [Enabling Delegation for Cloudera Impala](#) in the Tableau Knowledge Base.

Enable Run As User to Act as the Operating System

To use Kerberos delegation with Tableau Server, you must configure the Run As User account to act as the operating system on each Tableau Server node.

1. On the computer that is running Tableau Server, select **Start > Control Panel > Administrative Tools > Local Security Policy**.
2. In the Local Security Settings window, expand **Local Policies**, click **User Rights Assignments**, and then right-click **Act as part of the operating system** and select **Properties**.



3. In the Act as part of the operating system Properties window, click **Add User or Group**.
4. Type the <domain>\<username> for the Tableau Server Run As User account (for example: MYCOMPANY\tableau_server), and then click **Check Names**.
5. When the account resolves correctly, it is underlined. Click **OK**.
6. Click **OK** to close the Local Security Policy windows.

Troubleshoot Kerberos

The troubleshooting suggestions in this topic are divided into issues related to Single sign-on (SSO) on the server and issues with the delegated data sources.

Single Sign-on to Tableau Server

Kerberos Authentication Failed (unable to connect automatically to Tableau Server)

If you are using Kerberos for SSO and a user is prompted to sign in to Tableau Server when they connect with either a web browser or with Tableau Desktop, try these steps from the client computer:

Sign In



Enter your Windows username and password to sign in.

Username:

Password:

Tableau Server could not authenticate you automatically. Please sign in.

Kerberos Authentication Failed
Request ID: VEqjjgogkdlAABVsTO0AAADy

Sign In

Troubleshooting on the client computer

- **Account permissions**—Try to sign in to Tableau Server using the user's name and password. If they can't sign in to Tableau Server using their user name and password, they do not have permission to access Tableau Server and Kerberos authentication will fail.
- **Other accounts**—Try to connect with SSO to Tableau Server using other user accounts. If all users are affected, the problem may be in the Kerberos configuration.
- **Computer location**—Kerberos will not work when connecting from localhost. Clients must be connecting from a computer other than the Tableau Server computer.
- **URL address**—You cannot use Kerberos SSO when connecting using an IP address. In addition, the server name you use to access Tableau Server must match the name used in the Kerberos configuration (see [Key table entry](#), below).
- **TGT (Ticket Granting Ticket)**—Confirm that the client computer has a TGT from the Active Directory domain. Kerberos requires a TGT to sign in. To confirm the client computer has a TGT, type:
 - klist tgt at a command prompt on a Windows computer
or
klist at a terminal prompt on a Mac computer

The output should show a TGT for the user/domain trying to authenticate to Tableau Server.

The client computer may not have a TGT in the following circumstances:

- The client computer is using a VPN connection
- The client computer is not joined to the domain (for example, it is a non-

work computer being used at work)

- The user signed into the computer with a local (non-domain) account
- The computer is a Mac that is not using Active Directory as a network account server
- **Browser**—Check which browser the user is using to access the server
 - Internet Explorer (IE) and Chrome work "out of the box" on Windows
 - Safari works "out of the box" on Mac
 - Firefox requires additional configuration

For more information about browser support for Kerberos Single Sign-On (SSO), see [Browser Support for Kerberos SSO to Tableau Server](#) in the Tableau Knowledge Base.

Troubleshooting on the server

If you cannot solve the problem from the client computer, your next steps are to troubleshoot on the computer running Tableau Server. The administrator can use the request ID to locate the sign-in attempt in the Apache logs on Tableau Server.

- **Log files**—Check the Apache error.log for an error with the exact time/date of the failed sign-in attempt.
 - In a ziplog archive, these logs are in the \httpd folder.
 - On Tableau Server, these logs are in the \data\tabsvc\logs\httpd\ folder.
- **Key table entry**—If the error.log entry says "No key table entry matching HTTP/<servername>.<domain>.<org>@", for example:

```
[Fri Oct 24 10:58:46.087683 2014] [:error] [pid 2104:tid 4776] [client 10.10.1.62:56789] gss_acquire_cred() failed: Unspecified GSS failure. Minor code may provide more information (, No key table entry found matching HTTP/server-name.domain.com@)
```

This errors is a result of a mis-match between any of the following:

- **Tableau Server URL** - The URL used by the client computer to access the server.

This is the name that you type into Tableau Desktop or a browser address bar. It could be a shortname (<http://servername>) or a fully-qualified domain name (<http://servername.domain.com>)



- **DNS reverse lookup** for the server IP address

This looks up a DNS name using an IP address.

At a command prompt type:

```
pingservername
```

with the IP address returned by pinging the server, do a reverse DNS lookup type:

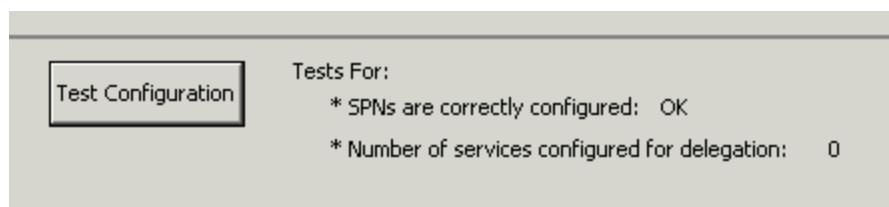
```
nslookup <ip address>
```

The Tableau Server computer name needs to match in:

- .keytab file
- Service Principal Name (SPN) for the server

Test Configuration and tabconfig.log

Use the Test Configuration button in the Tableau Server Configuration utility:



If your SPNs are correctly set up for Kerberos, **SPNs are correctly configured** shows OK.

If any services are configured for delegation, the number of configured services will appear. A value of 0 (zero) does not indicate a problem unless you are using delegation and Kerberos authentication to SQL Server or MSAS.

Look in `tabconfig.log` for any problems or errors. For example:

```
2014-10-17 10:58:16.545 -0700 ERROR root: No SPN entries found
```

If the test does not show successful results, run the configuration script again.

Data source SSO

Delegated data source access failures

Check the `vizqlserver` log files for "workgroup-auth-mode":

- In a ziplog archive, these logs are in the `\vizqlserver\Logs` folder
- On the Tableau Server, these logs are in the `\data\tabsvc\vizqlserver\Logs` folder

Look for "workgroup-auth-mode" in the log files. It should say "kerberos-impersonate" not "as-is".

OpenID Connect

You can configure Tableau Server to support OpenID Connect for single sign-in (SSO). OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server.

Configuring OpenID Connect involves several steps. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [Configure the Identity Provider \(IdP\) for OpenID Connect](#) on the next page
- [Configure Tableau Server for OpenID Connect](#) on page 519
- [Signing In to Tableau Server Using OpenID Connect](#) on page 522
- [Changing IdPs in Tableau Server for OpenID Connect](#) on page 524

Requirements for Using OpenID Connect

To use OpenID Connect with Tableau Server, you must have the following.

IdP account

You must have access to an IdP that supports the protocol, such as Google. You must also have an account with the IdP.

Local authentication

To use OpenID Connect on Tableau Server, the server must be configured to use local authentication. Active Directory authentication is not supported.

User names with email addresses

In Tableau Server, each user who can sign in must have an existing identity in Tableau Server—that is, you must previously have created a user for each person who will sign in. By default, the user's user name in Tableau Server must match the user name in the IdP. This is typically an email address—for example, if you use Google as the IdP, the user name in Tableau Server must be the user's Gmail address (`alice@gmail.com`). Using a complete email address in this way helps to guarantee the uniqueness of the user name in Tableau Server, even when two users have the same email but are on different email hosts.

Note: When you create a user identity in Tableau Server, you specify a user name, password, and optionally an email address. For using OpenID Connect, the user name is

the value that must match the user's name in the IdP. (The optional email address in the Tableau Server user identity is not used for OpenID authentication.)

Ignoring the domain name

You can configure Tableau to ignore the domain portion of an email address when matching the IdP user name in Tableau Server. In this scenario, the user name in the IdP might be `alice@example.com`, but this will match a user named `alice` in Tableau Server. Ignoring the domain name might be useful if you already have users defined in Tableau Server whose names match IdP user names except for the domain.

To configure Tableau Server to ignore domain names in user names from the IdP, use the following sequence of `tabadmin` commands:

```
tabadmin stop  
tabadmin set vizportal.openid.ignore_domain true  
tabadmin configure  
tabadmin start
```

Note: When you change the `vizportal.openid.ignore_domain` setting to ignore the domain in user names, all user names in Tableau Server must have a domain name.

Configure the Identity Provider (IdP) for OpenID Connect

This topic provides information about configuring an identity provider (IdP) to use OpenID Connect with Tableau Server. This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect on the previous page](#)
- Configure the Identity Provider (IdP) for OpenID Connect (you are here)
- [Configure Tableau Server for OpenID Connect on the next page](#)
- [Signing In to Tableau Server Using OpenID Connect on page 522](#)
- [Changing IdPs in Tableau Server for OpenID Connect on page 524](#)

Configure the IdP

Before you can use OpenID Connect with Tableau Server, you must have an account with an IdP and a project or application with the IdP. When you configure Tableau Server, you will need to be able to provide the following information:

- Provider client ID. This is the identifier that the IdP assigned to your application.
- Provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.
- Provider configuration URL. This is the URL at the provider's site that Tableau Server should send authentication requests to.

The following procedure provides an outline of the steps that you follow with the provider. As an example, the procedure discusses using Google as a provider. However, each provider has a somewhat different flow, so the specifics of the steps (and their order) might vary depending on your provider.

1. Register at the provider's developer site and sign in. For example, for Google, you can go to the Developers Console at this URL: <https://console.developers.google.com>
2. Create a new project, application, or relying party account.
3. In the developer dashboard, follow the steps for getting an OAuth 2.0 client ID and client secret. Record these values for later.

Note: Keep the client secret in a secure place.

4. On the developer site, find the URL of the endpoint that the IdP uses for OpenID Connect discovery. For example, Google uses the URL <https://accounts.google.com/.well-known/openid-configuration>. Record this URL for later.

The IdP configuration requires an additional step that you cannot finish until after you've configured Tableau Server, as described in [Configure Tableau Server for OpenID Connect below](#).

Configure Tableau Server for OpenID Connect

This topic describes how to configure Tableau Server to use OpenID Connect for single-sign on (SSO). This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect on page 517](#)
- [Configure the Identity Provider \(IdP\) for OpenID Connect on the previous page](#)
- Configure Tableau Server for OpenID Connect (you are here)
- [Signing In to Tableau Server Using OpenID Connect on page 522](#)
- [Changing IdPs in Tableau Server for OpenID Connect on page 524](#)

Note: Before you perform the steps described here, you must configure the OpenID identity provider (IdP) as described in [Configure the Identity Provider \(IdP\) for OpenID Connect on page 518](#).

Important notes

Before you configure Tableau Server for OpenID Connect, make sure you read these notes.

- You can use OpenID Connect with Tableau Server only if the server is configured to use local authentication. OpenID Connect is not available if the server is configured to use Active Directory authentication. For more information, see [Configure General Server Options on page 12](#).
- We recommend that you configure Tableau Server to use SSL for external communications. This helps to maintain secure communications between Tableau Server and the IdP during the exchange of authentication information. For details, see [Configure External SSL on page 491](#).

If you are configuring OpenID Connect during the initial configuration of Tableau Server (the first time the configuration utility runs), there is no option to set up SSL. In that case, we recommend that you finish the installation, then return to the configuration to set up SSL and then configure OpenID.

Note If you want to use external SSL for Tableau Server, it's generally more convenient to do that before you configure OpenID Connect. If you configure SSL after you've already configured OpenID, you need to return to the IdP and update the configuration that you made previously. For example, you need to change the protocol for the Tableau Server external URL from `http://` to `https://`.

Configure the server

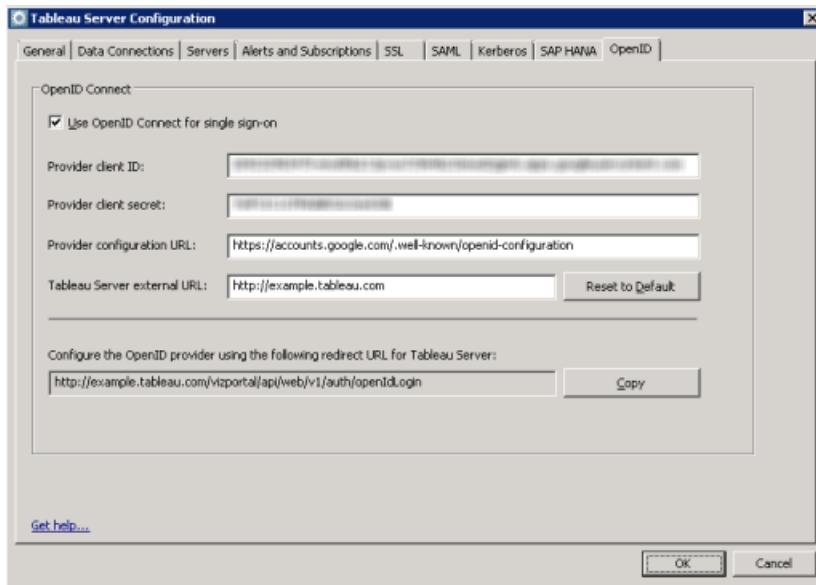
To configure Tableau Server for OpenID Connect, follow these steps.

1. Log in as an administrator to the computer where Tableau Server is running.
 2. If the server is running, stop it (Windows Start > **All Applications** > **Tableau Server** > **Stop Tableau Server**).
- Tip:** You can also stop the server by using the `tabadmin stop` command.
3. Run the Tableau Server Configuration tool (Windows Start > **All Applications** > **Tableau Server** > **Configure Tableau Server**).
 4. Click the **OpenID** tab.

5. Select the **Use OpenID Connect for single sign-on** option.
6. Fill in the **Provider client ID** and **Provider client secret** boxes with the values you recorded earlier.
7. In the **Provider configuration URL** box, enter the URL that the IdP uses for OpenID Connect discovery.
8. In the **Tableau Server external URL** box, enter the URL of your server. This is typically the public name of your server, such as `http://example.tableau.com`.

When you initially configure OpenID, the **Provider configuration URL** box contains a default value that's constructed based on the name of the server (`gateway.public.host`) and the gateway port, if any (`gateway.public.port`). In addition, by default the protocol is set to `https://` if SSL is enabled for the server.

Note: Make sure that you update the external URL if the default value is not the URL for how your server can be reached from an external source.



9. Copy the URL in the box labeled **Configure the OpenID provider using the following redirect URL for Tableau Server**. You'll use this value in the next procedure to finish configuring the IdP.
10. Start the server (Windows Start > All Applications > Tableau Server > Start Tableau Server).

Tip: You can also start the server by using the `tabadmin start` command.

Add the redirect URL to the IdP configuration

After you configure Tableau Server, you finish the IdP configuration using the server's redirect URL.

1. Return to the IdP portal where you set up the project or application.
2. Edit the project configuration and find the redirect URL.
3. Enter the redirect URL that you copied in the previous procedure.

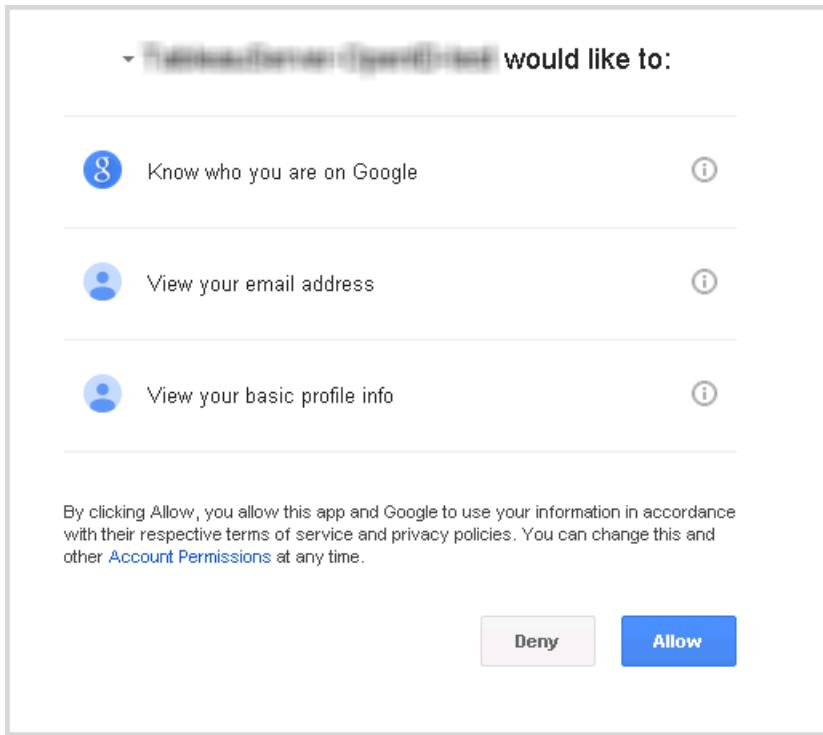
Signing In to Tableau Server Using OpenID Connect

This topic provides information about signing in to Tableau Server using OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect on page 517](#)
- [Configure the Identity Provider \(IdP\) for OpenID Connect on page 518](#)
- [Configure Tableau Server for OpenID Connect on page 519](#)
- Signing In to Tableau Server Using OpenID Connect (you are here)
- [Changing IdPs in Tableau Server for OpenID Connect on page 524](#)

Signing in using OpenID Connect

Once Tableau Server has been configured to use OpenID Connect, users who access the server and aren't already signed in are redirected to the IdP site, where they are prompted to sign in. Users enter the credentials that they have with the IdP. In many cases, the user is also asked to authorize the IdP to share information with Tableau Server, as in the following example:



When a user signs in using OpenID Connect, the IdP sends a unique user identifier (known in OpenID as the sub value) as part of the information that's redirected to Tableau Server. This sub value is associated with the user's Tableau user identity.

Restricting sign-in to server administrators for command-line tools

Command-line tools for working with Tableau Server (`tabcmd`, `tabadmin`, and `tableau.com`) do not support sign-in using OpenID Connect. When OpenID Connect is enabled for the server, these tools still require sign-in using a Tableau Server username and password.

Even if users normally authenticate using OpenID Connect, each user has a Tableau Server username and password. This means that users could use command-line tools like `tabcmd`. As a security measure, you can make sure that *only* server administrators can use command-line tools. To do this, use `tabadmin set to set wgserver.authentication.restricted to true`. When this setting is `true`, only server administrators can sign in to Tableau Server using a username and password; all other users *must* sign in to the server using a single sign-on (SSO) option like OpenID Connect. The effect is that users who are not administrators also cannot then use command-line tools. To make this change, do the following:

1. Stop the server.
2. Run the following sequence of `tabadmin` commands:

```
tabadmin set wgserver.authentication.restricted true  
tabadmin configure
```

3. Start the server.

Changing IdPs in Tableau Server for OpenID Connect

This topic provides information about changing an identity provider (IdP) if you have configured Tableau Server to use OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect](#) on page 517
- [Configure the Identity Provider \(IdP\) for OpenID Connect](#) on page 518
- [Configure Tableau Server for OpenID Connect](#) on page 519
- [Signing In to Tableau Server Using OpenID Connect](#) on page 522
- Changing IdPs in Tableau Server for OpenID Connect (you are here)

Changing providers

You might decide to change the IdP that Tableau Server is configured to use. To do so, you follow the procedure that you used to configure the first IdP: establish an account, get a customer ID and secret, configure Tableau Server with that information, and provide the IdP with the redirect URL for Tableau Server. For more information, see [Configure Tableau Server for OpenID Connect](#) on page 519.

However, you also need to perform an additional step: you must clear any user identifiers (`sub` values) that have already been associated with Tableau Server users. The new IdP will have different `sub` values for each user, and you must clear the existing ones so that Tableau Server can store a new `sub` value when the user signs in using the new IdP.

To clear `sub` values for users, use the `tabadmin reset_openid_sub` command. You can reset (that is, clear) `sub` values for an individual user, as in the following example:

```
tabadmin reset_openid_sub Alice
```

You can also clear the `sub` value for all users using this command:

```
tabadmin reset_openid_sub all
```

Run As User

You can use a dedicated Active Directory (AD) user account for the Tableau Server service to run under, called a Run As User account. Some administrators choose to do this when published workbooks on Tableau Server connect to live data sources. The server's default Network Service account (NT AUTHORITY\NetworkService) doesn't have the correct permissions for connecting to data sources on other computers. A correctly configured AD account does.

For data sources that require Windows (NT) authentication, the AD account can also automatically handle the authentication process, attempting first to authenticate using Kerberos and then using NTLM if that fails, thus shielding users from prompts for credentials when the workbook connects to the live data source. Finally, a Run As User AD account that is dedicated to a specific resource is often less problematic to manage than an AD account associated with a person.

To configure Tableau Server to use a Run As User account, follow the procedures below. If you are running a [distributed installation](#) of Tableau Server, these steps should be performed on workers as well as on the primary. Also note that the steps under [Required Run As User Account Settings](#) on page 528 may vary from site to site.

To use Kerberos delegation with Tableau Server, you must configure the Run As User account to act as the operating system on each Tableau Server node. For more information, see [Enable Run As User to Act as the Operating System](#) on page 511.

Note: If you are installing Tableau Server with your Run As User account in hand, before you run Setup, confirm that the Windows Secondary Login service has the correct values for Log On and Startup. See [Verify Tableau Service Settings](#) on the next page for more information.

Identify the Account

Your first step is to identify or create an Active Directory account for the Tableau Server service to run under. This will be the Tableau Server's Run As User account, and it should have the following:

- Permissions for connecting to the data source with at least read access.
- Credentials to allow Tableau Server to satisfy the NT authentication process with the data source. Microsoft data sources that perform NT authentication include Microsoft SQL Server and Microsoft Analytical Services (MSAS), but not Access or Excel.
- Permissions to query your Active Directory domain controller for users and groups. A

user account created on the local machine that Tableau Server is running on probably won't have these permissions.

Verify Tableau Service Settings

Confirm that Tableau services are assigned the correct Log On and Startup values. If you are running a **distributed installation** of Tableau Server, perform these steps on the workers as well as on the primary.

1. Log on as administrator to the computer running Tableau Server.
2. On the Tableau Server computer, select **Start > Control Panel > Administrative Tools > Computer Management > Services and Applications > Services**.
3. Open Services and Applications, then click **Services**. Confirm that the following services have the correct settings:

Service Name	Logon Value	Startup Value
FLEXnet Licensing Service	Local System	Manual
Secondary Logon	Local System	Automatic
Tableau Server Application Manager (tabsvc)	<domain>\<username> This is the Run As user account. See below.	Automatic
Tableau Server License Manager (tablicsrv)	Local System	Automatic

Note: Do not change the default settings on the **Recovery** tab of the **Tableau Server Application Manager Properties** dialog box; leave the settings for failure recovery as **Take No Action**. If you change these settings, Tableau Server will restart after being stopped via the **tabadmin** command or **Stop Tableau Server** command.

Changing the Log On Value

To change the **Log On** value for Tableau Server (tabsvc) to the Run As User account:

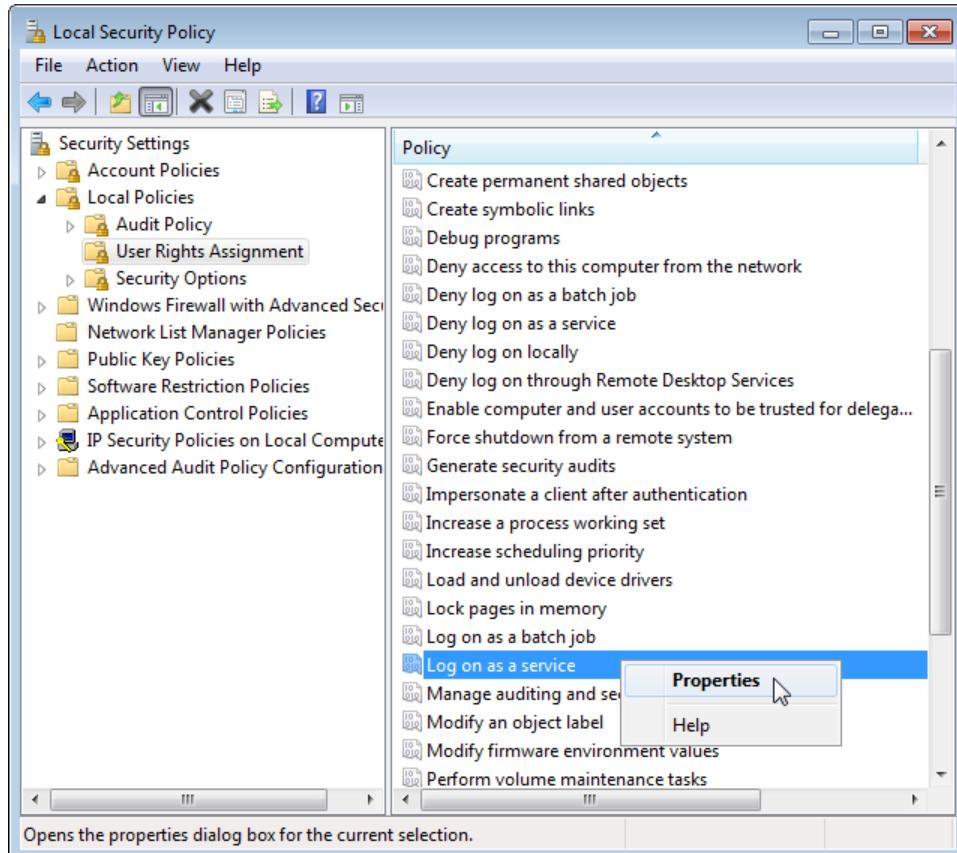
1. Select **Start > All Programs > Tableau Server > Stop Tableau Server**.
2. Select **Start > All Programs > Tableau Server > Configure Tableau Server**.
3. On the General tab, enter the domain, user name, and password for Tableau Server's Run As User account.

4. Click **OK**, and then select **Start > All Programs > Tableau Server > Start Tableau Server**.

Prepare the Local Security Policy

If your Run As User account isn't an administrator on the Tableau Server machine (primary and workers, if you're running a distributed installation), you must prepare the machine's local security policy so that the Tableau Server Run As User account can log onto the machine as a service and make configuration changes. To do this:

1. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.
2. In the Local Security Settings window, open Local Policies, highlight User Rights Assignments, then right-click **Log on as a service** and select **Properties**.



3. In the Log on as a service Properties window, click **Add User or Group**.
4. Type the `<domain>\<username>` for the Tableau Server Run As User account (for example: MYCO\tableau_server), and click **Check Names**.
5. When the account resolves correctly, it is underlined. Click **OK**.

6. Repeat these steps to add the Run As account to the **Allow log on locally** policy.
7. Repeat these steps to remove the Run As account from the **Deny log on locally** policy.
8. Click **OK** to close the Local Security Settings windows.

Required Run As User Account Settings

The Run As User account needs permissions that allow it to read, execute, and sometimes modify files.

Note: Do not hide the files created by the Tableau Server installer.

Depending on the account you used as a starting point, it may already have the correct permissions. Any time that you change the server Run As account, you should confirm that it meets the following requirements. If you're running a distributed installation, this applies to both the primary server and worker nodes.

Grant Read and Execute Permissions

The account the Tableau Server service runs under needs permission to read and execute files in the path where Tableau Server is installed.

For example, if Tableau is installed on the D drive, the account needs permissions for D:\Program Files\Tableau and D:\ProgramData\Tableau, including all folders and files in all subfolders.

Any time the server's Run As User account is changed, confirm or configure the following:

1. On the computer hosting Tableau Server (and on Tableau Worker computers, if distributed), use Windows Explorer to right-click the drive on which Tableau is installed, for example **Local Disk (C:)**, and select **Properties**.
2. In the Local Disk Properties Window, select the **Security** tab.
3. Click **Edit**, then **Add**.
4. In the Select Users, Computers, Service Accounts, or Groups dialog box, type the <domain>\<username> for the Tableau Server Run As User account. Do not use a group account.
5. Click **Check Names** to resolve the account, then **OK** to confirm.
6. With the Tableau Server Run As User account highlighted, confirm that it has **Read & execute** permissions. Selecting **Read & execute** automatically selects **List folder contents** and **Read**.
7. Click **OK** to exit.

Grant Modify Permissions

The account also needs the ability to do things like create log files. Confirm the account has permission to read and execute files in the path where Tableau Server is installed and in the associated location where Tableau stores its data.

Confirm or configure permissions by doing the following:

1. Navigate to the following folders:

```
<installation drive>:\Program Files\Tableau  
<installation drive>:\ProgramData\Tableau\
```

ProgramData is hidden by default, so you may not see it without making it visible.

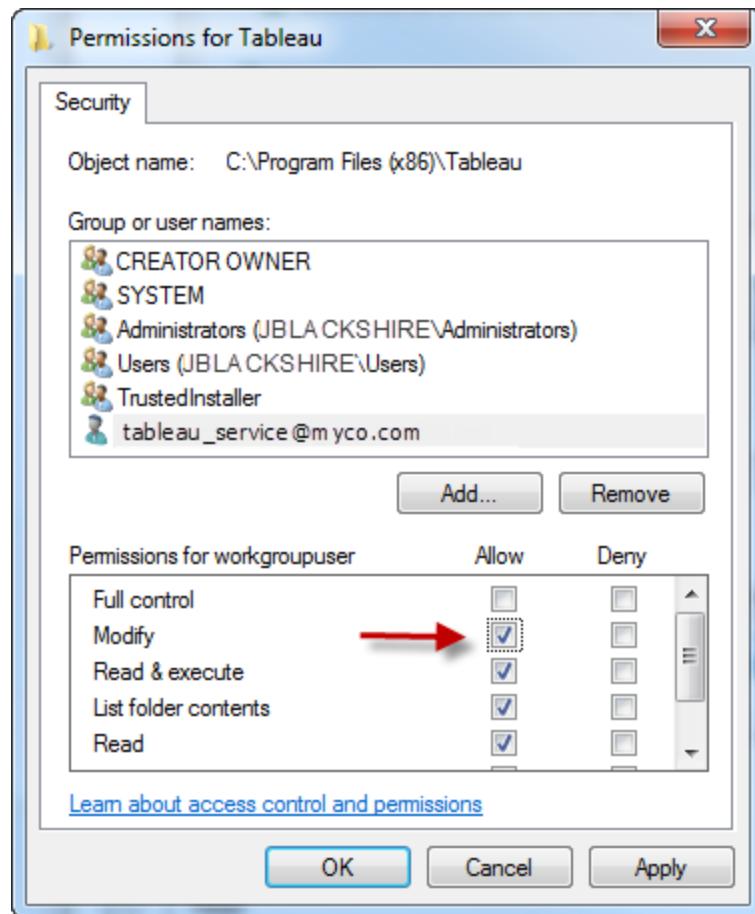
The drive may vary depending on where Tableau Server is installed. If you are running the 32-bit Tableau Server on a 64-bit operating system, you will need to go to

```
<installation drive>:\Program Files (x86)\Tableau instead of  
<installation drive>:\Program Files\Tableau.
```

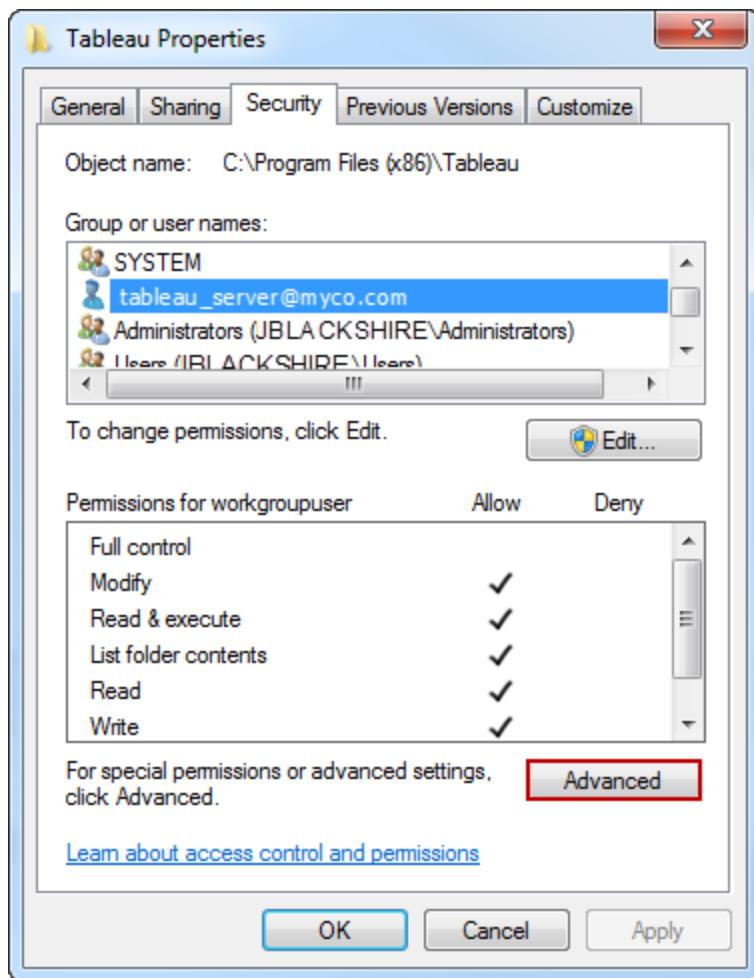
2. Right-click the folder, select **Properties**, and click the **Security** tab:

- Click **Edit**, then **Add**.
- Type the <domain>\<username> for the Tableau Server Run As User account.
- Click **Check Names** to resolve the account, then **OK** to confirm.
- With the Tableau Server Run As User account highlighted, confirm that it has **Modify** permissions. Selecting **Modify** automatically grants all permissions

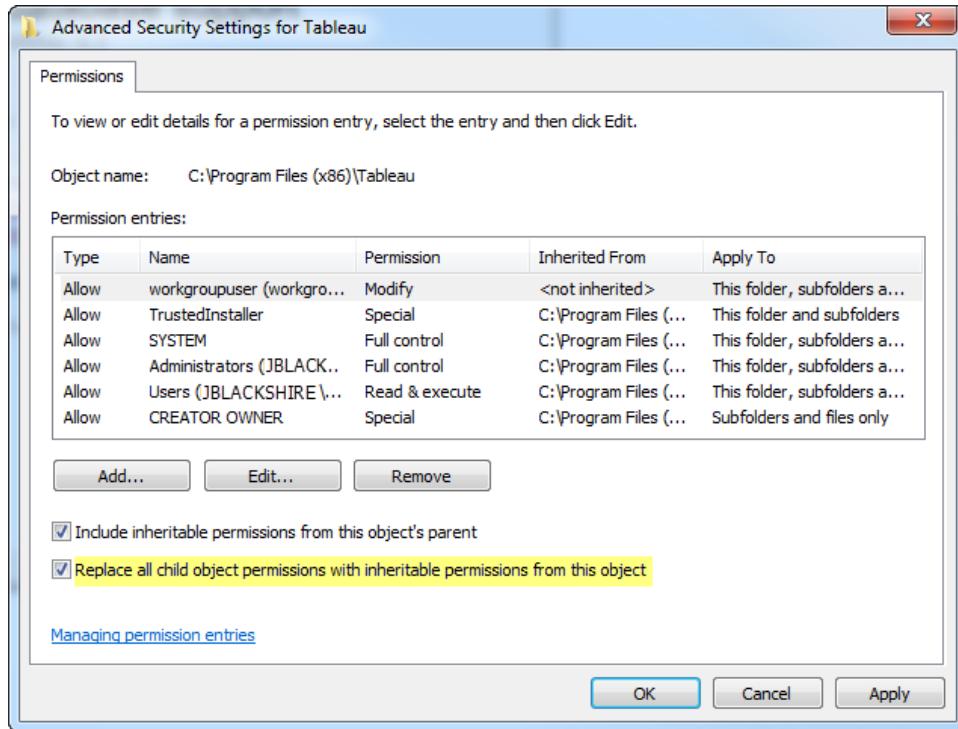
except for **Full Control** and **Special Permissions**:



3. For each folder from step 1 above, on the Tableau Properties Security tab, click **Advanced**:



4. In the Advanced Security Settings for Tableau window, click **Change Permissions**.
5. In the Advanced Security Settings for Tableau dialog box, highlight the Run As User account and select the **Replace all child object permissions with inheritable permissions from this object** check box:



6. Click **OK** to apply changes to all subfolders and files - this may take a few minutes. It's typical to receive several error messages from Windows when you apply these changes. There's no need to cancel the process; instead, click **Continue**.
7. Click **OK** to confirm changes, then click **OK** in the Tableau Properties dialog box.

Modify Registry Settings

The following step is optional and not seen in most environments. If the registry security is highly restrictive, grant the Tableau Server Run As User account read and write permissions to the registry branches listed below. The registry keys vary according to whether you installed the 32- or 64-bit version of Tableau Server and, in the case of the 32-bit Tableau Server, whether you installed on a 32- or 64-bit operating system. The 64-bit Tableau Server can only be installed on a 64-bit operating system.

64-bit Tableau Server Installations

- HKEY_CURRENT_USER\Software\Tableau
- HKEY_LOCAL_MACHINE\Software\Tableau

32-bit Tableau Server Installations

- HKEY_CURRENT_USER\Software\Tableau

and

- 32-bit operating systems: HKEY_LOCAL_MACHINE\Software\Tableau
- 64-bit operating systems: HKEY_LOCAL_MACHINE\Software\Wow6432Node\Tableau

Confirm Domain Two-Way Trust

Confirm that there is a two-way trust between domains if any of the following are true:

- The machines hosting the Tableau Server and the data source are on separate domains.
- Tableau Server users are on a separate domain from Tableau Server or the data source.

Configure Data Source Connection Settings

To automatically authenticate your users when the workbook they're accessing connects to a live, NT-authenticated data source, configure your Tableau data connection with the **Use Windows NT Integrated security** option selected:

Windows NT Integrated Security

Authenticates with the server's Run As User account

Username and Password

Each Tableau Server user is prompted for database credentials



SQL Server Impersonation

Impersonation in the context of Tableau Server means allowing one user account to act on behalf of another user account. You can configure Tableau and Microsoft SQL Server to perform database user impersonation, so that the SQL Server database account used by Tableau Server queries on behalf of SQL Server database users, who are also Tableau users.

The main benefit of this feature is it allows administrators to implement and control their data security policy in one place: their databases. When Tableau users access a view with a live connection to a SQL Server database, the view only displays what the users' database permissions authorize them to see. An additional benefit is that the users don't have to respond to a database sign-in prompt when they open the view. Also, workbook publishers don't have to rely on user-specific filters to restrict what's seen in views.

Use the topics below for more information on what you need to use this feature.

Impersonation Requirements

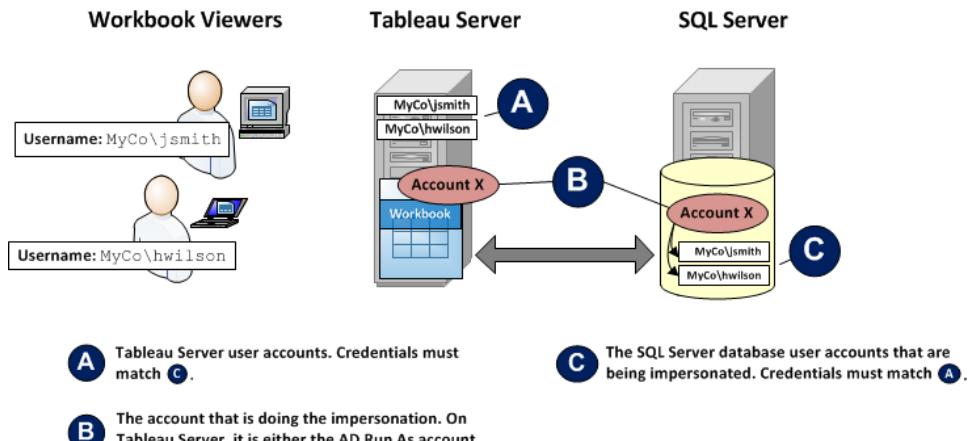
Here's what you need to use feature:

- **Live connections to SQL Server only:** Impersonation can only be used for views that have a live connection to a SQL Server database, version 2005 or newer.
- **Individual database accounts:** Each person who'll be accessing the view must have an explicit, individual account in the SQL Server database to which the view connects. Members of an Active Directory (AD) group cannot be impersonated. For example, if Jane Smith is a member of the AD group Sales, and her database administrator adds the Sales AD group to the SQL Server database, Jane cannot be impersonated.
- **Matching credentials and authentication type:** The credentials of each Tableau user's account and their Tableau user authentication type must match their credentials and authentication type in the SQL Server database. In other words, if Jane Smith's Tableau Server user account has a username of MyCo\jsmith and Tableau Server is using Active Directory for user authentication, her username on the SQL Server database must also be MyCo\jsmith and SQL Server must be using Windows Integrated Authentication.
- **SQL Server prerequisites:** In SQL Server you should have a data security table, a view that enforces data security, and you should require that your database users use the view.
- **SQL IMPERSONATE account:** You need a SQL Server database account that has IMPERSONATE permission for the above database users. This is either an account with the sysadmin role or one that has been granted IMPERSONATE permission for each individual user account (see the [MSDN article on EXECUTE AS](#)). This SQL Server account must also be one of two accounts on the Tableau side of things:

- The Tableau Server Run As User account (see [Impersonate with a Run As User Account](#) below).
- The workbook publisher's account (see [Impersonate with Embedded SQL Credentials](#) on page 537).

How Impersonation Works

Here's an illustration of how database user impersonation works:



In the above illustration, Jane Smith (MyCo\jsmith) is a West Coast sales representative and Henry Wilson (MyCo\hwilson) covers the East. In the SQL Server database, the account permissions for Jane's account, MyCo\jsmith, only give her access to West Coast data. Henry's account, MyCo\hwilson, can only access data for the East Coast.

A view has been created that displays data for the entire country. It has a live connection to a SQL Server database. Both users sign in to Tableau Server and click the view. Tableau Server connects to SQL Server using a database account with IMPERSONATE permission for each user's database account. This account acts on behalf of each user's database account.

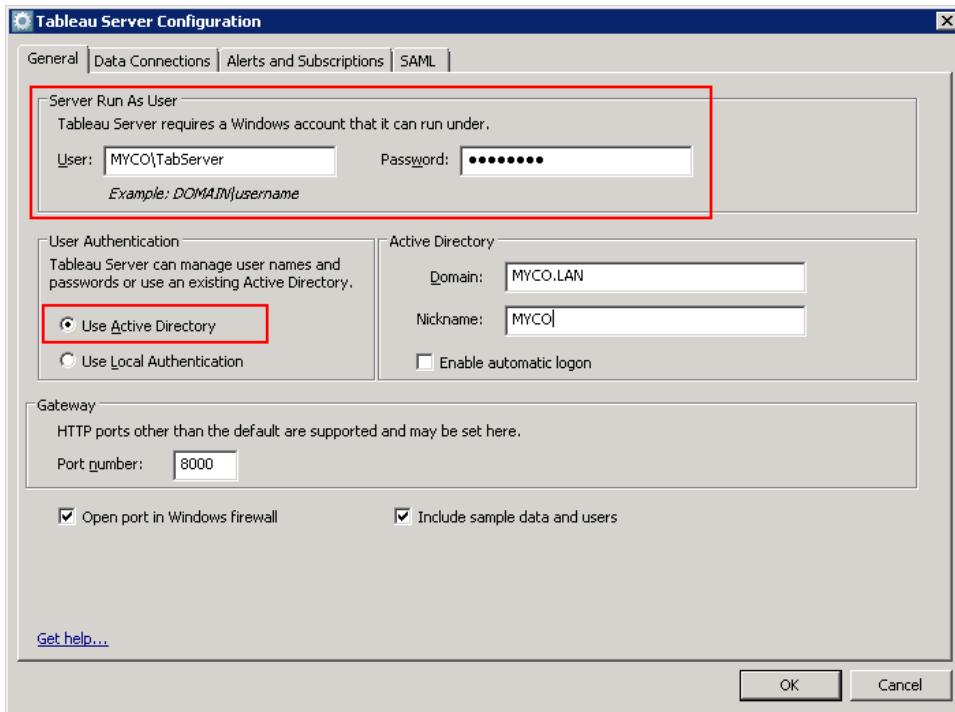
When the view displays, it is restricted by each user's individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.

Impersonate with a Run As User Account

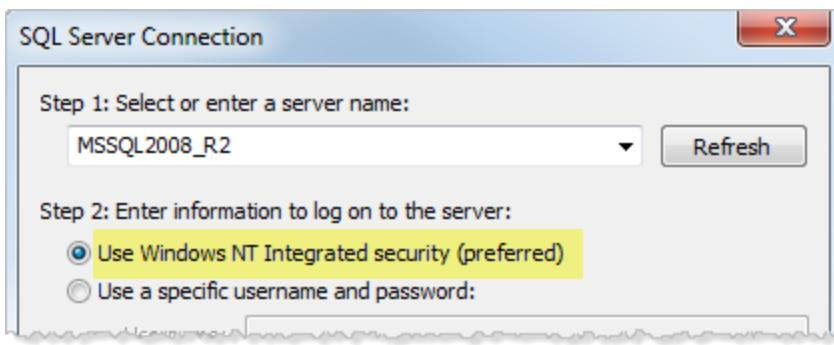
Impersonating via a Run As User account is the recommended way to perform impersonation. The Run As User account is an Active Directory (AD) account the Tableau Server service can run under on the machine hosting Tableau Server (see [Run As User](#) on page 525). This same account must have IMPERSONATE permission for the database user accounts in SQL Server. From a data security standpoint, using the Tableau Server Run As account for impersonation gives the administrator the most control.

To set up impersonation with a Run As User account:

1. When you configure Tableau Server as part of Setup, under **Server Run As User**, enter the Run As User AD account that has IMPERSONATE permission for the user accounts. Under **User Authentication**, select **Use Active Directory**:

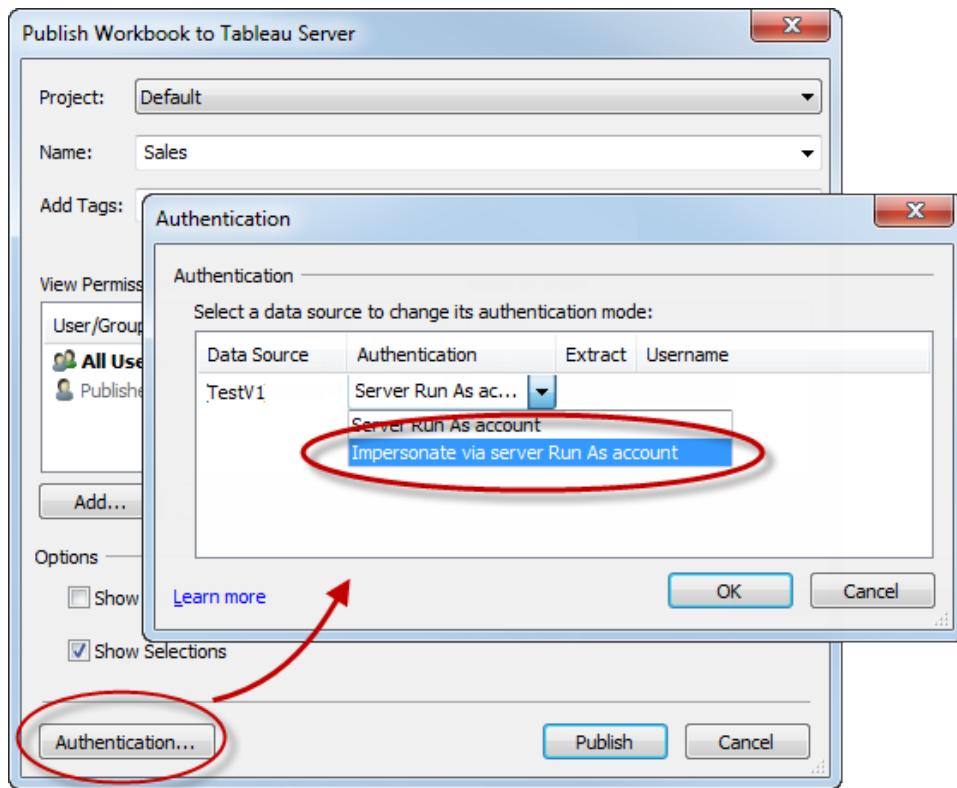


2. Click **OK** to finish configuration.
3. Create a workbook in Tableau Desktop. When you create the data connection, select **Use Windows NT Integrated security** for the workbook's live connection to a SQL Server database:



4. In Tableau Desktop, publish the workbook to Tableau Server (**Server > Publish Workbook**).

5. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via server Run As account** from the drop-down list:



6. Click **OK**.
7. Test the connection by signing into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Impersonate with Embedded SQL Credentials

You can also perform impersonation by having the person who publishes a view embed their SQL Server account credentials in the view. Tableau Server can be running under any type of account, but it will use these credentials, supplied by the publisher, to connect to the database.

This may be the right choice for your site if the account that handles the impersonation cannot be an Active Directory (AD) account and if you're comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

Note:

To use this approach, **Embedded Credentials** must be enabled on the server Settings page in Tableau Server:

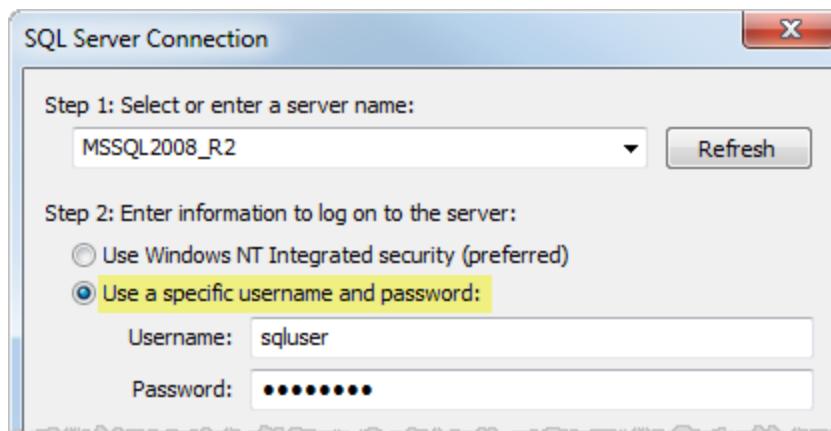
Embedded Credentials

Users can attach data source credentials to a workbook; people then connect to the workbook data.

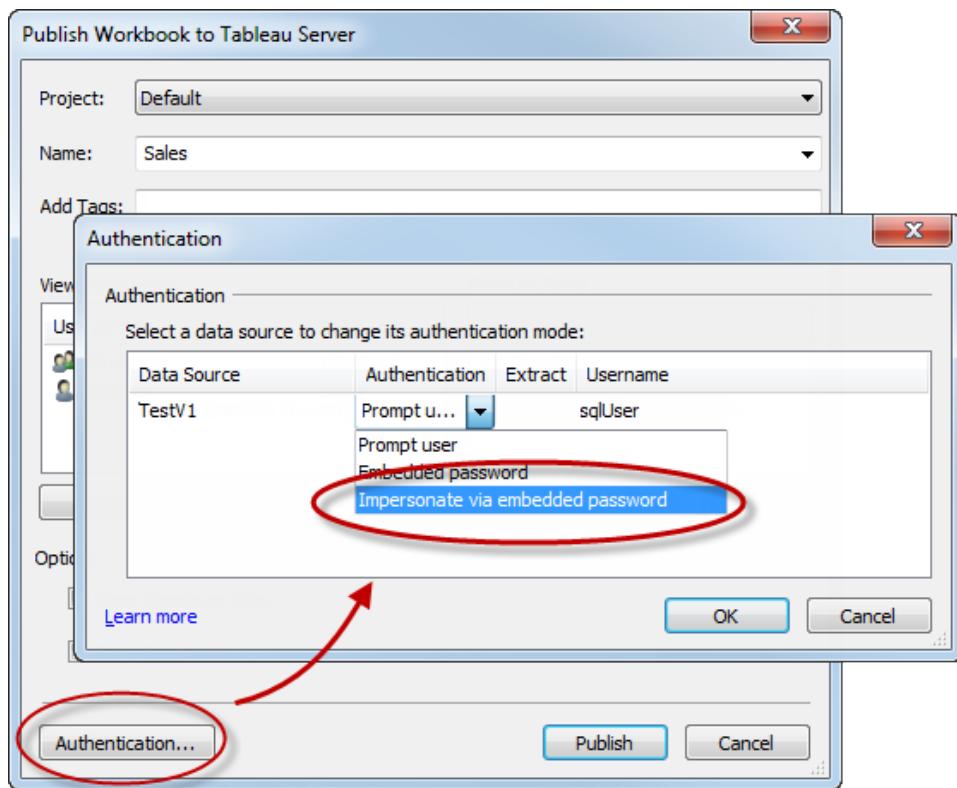
Allow embedded credentials

To impersonate with the workbook publisher's SQL account:

1. In Tableau Desktop, create a workbook. When you create the data connection, select Use a specific username and password for the workbook's live connection to a SQL Server database:



2. Publish the workbook to Tableau Server (**Server > Publish Workbook**).
3. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via embedded password** from the drop-down list:



4. Click **OK**.
5. Test the connection by signing in to Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Tableau Server Ports

The following table lists the ports that Tableau Server uses by default, and which must be available for binding. If you install multiple instances of a process (Cache Server for example) on a node, consecutive ports are used, starting at the base port. If Windows Firewall is enabled, Tableau Server will open the ports it needs for internal communication between processes. (There are circumstances when you may need to take action in addition. If you are [making an external connection to the Tableau Server database](#) you may need to open ports manually. If you have a distributed installation with a worker running Windows 7, see the [Tableau Knowledge Base](#).)

Dynamic port remapping

When dynamic port remapping is enabled (the default), Tableau Server first attempts to bind to the default ports, or to user-configured ports if they are defined. If the ports are not available, Tableau Server attempts to remap processes to other ports, starting at port 8000. The gateway port and ssl port are not remapped. When next restarted, Tableau Server will revert to using the default or configured ports.

When dynamic port remapping is disabled, Tableau Server does not attempt to remap processes and if a conflict is detected, Tableau Server will not start.

Note: Port conflicts can affect how JMX ports are determined. For more information, see [Enable the JMX Ports on page 549](#).

You can disable dynamic port remapping using the `tabadmin set service.port_remapping.enabled` command. For more information, see [tabadmin set options on page 616](#).

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
80	TCP	Gateway	X			gateway.public.port, workerX.gateway.port
443	TCP	SSL. When Tableau Server is configured for SSL, the application server	X			--

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		redirects requests to this port.				
2233	UDP	Server Resource Manager UDP port used for communication between Tableau Server processes. The Server Resource Manager monitors memory and CPU usage of Tableau Server processes (backgrounder.exe, dataserver.exe, tab-protosrv.exe, tdeserver.exe, vizportal.exe, vizqlserver.exe, wgserver.exe).	X			resource_manager_port
3729	TCP	Tableau Server setup	X			--
373-0-3731	TCP	Tableau worker servers in distributed and highly available environments (the primary		X	X	--

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		Tableau Server does not listen on these ports).				
5000	UDP	Server Worker Manager process (tabad-mwrk.exe) that is used for auto-discovery of worker servers in a distributed environment.	X			
6379	TCP	Cache Server process (redis-server.exe). Base port 6379. Consecutive ports after 6379 are used, up to the number of processes.	X			workerX.cacheserver.port
800-0-8059	TCP	API Server process (wgserver.exe). Base port 8000. Consecutive ports after 8000 are used, up to the number of processes. Tableau Server installs one API Server process on each node that has	X			wgserver.port Note: For historical reasons, the names of port settings for the API Server process sometimes include wgserver.

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		one or more Application Server processes.				
8060	TCP	PostgreSQL database	X			pgsql.port
8061	TCP	Firebird. Used for geocoding and custom geocoding.	X			firebird.port
8062	TCP	PostgreSQL database	X			pgsqlX.port
8080	TCP	Solr, Tomcat HTTP, and Repository processes	X			solr.port, tomcat.http.port, repository.port These parameters must be set to the same value.
8085	TCP	Tomcat HTTP	X			tomcat.server.port
8250	TCP	Background tasks	X			workerX.backgrounder.port
8350	TCP	Background tasks	X			
8600	TCP	Application Server process (vizportal.exe). Base port 8600. Consecutive ports after 8600 are used, up to the number of processes.	X			workerX.vizportal.port
8700	TCP	Application Server process	X			

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		(vizportal.exe)				
8755	TCP	Tableau Administrative process	X			tabadminservice.port
910-0-9199	TCP	VizQL Server process (base port 9100). Consecutive ports after 9100, up to the number of processes, are also used. By default, Tableau Server installs with two VizQL Server processes (ports 9100 and 9101).	X			vizqlserver.port
9200, 9400	TCP	VizQL Server process	X			
9345	TCP	File Store service		X	X	filestore.port
9346	TCP	File Store status service		X	X	filestore.status.port
970-0-9899	TCP	Data Server process (base port 9700). Consecutive ports after 9700, up to the number of processes, are also used. By default, Tableau	X			dataserver.port

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		Server installs with two Data Server processes (ports 9700 and 9701).				
9800, 1000-0	TCP	Data Server process	X			
1100-0	TCP	Search server		X	X	workerX.searchserver.port
1110-0	TCP	Search server		X	X	workerX.search-server.startup.port
1200-0	TCP	Coordination controller (ZooKeeper) client port	X			workerX.zookeeper.port
1201-2	TCP	Cluster Controller process		X	X	cluster.status.port
1300-0	TCP	Coordination controller (ZooKeeper) leader port	X			zoo-keeper.config.leaderPort
1400-0	TCP	Coordination controller (ZooKeeper) leader election port	X			zoo-keep-er.config.leaderElectPort
2700-0–2700-9	TCP	Workers and primary server to communicate licensing inform-		X	X	--

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		ation in distributed and highly available environments.				
	TCP	One additional port is dynamically chosen for workers and the primary server to communicate licensing information in distributed and highly available environments. Instead, you can specify a fixed port (27010 is recommended). See the Tableau Knowledge Base for details. Installations where the primary server is in a DMZ <i>must</i> use these directions.	X	X	--	
2704-2	TCP	Data Engine process. Tableau Server installs with one Data Engine process. There can be up	X			dataengine.port

Port	TCP/U-DP	Used by ...	TYPE OF INSTALLATION			Parameter
			All	Distributed	High Availability	
		to two Data Engine processes per node.				

Edit the Default Ports

Tableau Server processes are configured to use certain ports on the computer where the server is installed. For more information, see [Tableau Server Ports on page 540](#).

In general, you do not need to make changes to the port assignments for the server processes. However, if the computer that's running Tableau Server is also running other software that uses ports (this is not recommended), it's possible that the port assignments for Tableau Server processes conflict with ports used by the other software. In that case, you can assign different ports to Tableau Server processes.

To modify the ports used by Tableau Server processes, you use command line administrative tool ([tabadmin on page 583](#)). For example, the default port for the application server process (vizportal.exe) is 8000. You can use the **tabadmin** parameter `workerX.vizportal.port` to change it to a different port.

Follow the steps below to change the Tableau Server port configuration. If you are enabling the server's JMX ports, see [Enable the JMX Ports on page 549](#)

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Modify a port value by typing one of the following commands:

```
tabadmin set <workerX>.<parameter> <new port value>
```

```
tabadmin set <parameter> <new port value>
```

where:

- <workerX> indicates which machine in a cluster you want to change the process port for. The placeholder *X* refers to the worker number—*worker0* is the primary server (or the only server if you are not running a distributed server), *worker1* is the first worker server, *worker2* is the second worker server, and so on. If you are running a distributed server and you want to edit the default port for a process

on all machines in the cluster, you need to run the command (from a command prompt on the primary) once for each machine in the cluster.

- <parameter> is the server process that you are setting the port for, such as `wgserver.port`.
- <new port value> is the new port number you want the server process to use.

Here's an example that sets the port on the primary or standalone server to 8020 for the application server process (`vizportal`):

```
tabadmin set worker0.vizportal.port 8020
```

The following example sets the port for a 3-machine cluster (one primary and two workers) to 9200 for the VizQL server process.

```
tabadmin set worker0.vizqlserver.port 9200
```

```
tabadmin set worker1.vizqlserver.port 9200
```

```
tabadmin set worker2.vizqlserver.port 9200
```

You can use the following parameters to modify the corresponding ports—see [Tableau Server Ports on page 540](#) for a complete list of `tabadmin` parameters that can be set.

Port to Change	Parameter	Multiple workers?
80	<code>gateway.public.port</code>	No
80	<code>gateway.ports</code>	Yes
6379	<code>cacheserver.port</code>	Yes
8000	<code>wgserver.port</code>	Yes
8060	<code>pgsql.port</code>	Yes
8600	<code>vizportal.port</code>	Yes
9100	<code>vizqlserver.port</code>	Yes
9345	<code>filestore.port</code>	Yes
9700	<code>dataserver.port</code>	Yes
11000	<code>searchserver.port</code>	Yes

Note: You should not change port assignments for processes that are not listed in this table. Changing other ports can cause Tableau Server to stop working.

3. After you make the necessary port configuration changes, restart Tableau Server by typing the following:

```
tabadmin restart
```

While the server is restarting it will be unavailable to all users. Be sure to warn your users of the outage prior to this operation or schedule this maintenance during non-business hours.

Enable the JMX Ports

To help you work through a problem with Tableau Server, Tableau Support may ask you to enable the server's JMX ports. These ports can be useful for monitoring and troubleshooting, usually with a tool like JConsole.

To enable the JMX ports on Tableau Server:

1. **Stop the server.**
2. Enter the following command:

```
tabadmin set service.jmx_enabled true
```

3. Enter the configure command:

```
tabadmin configure
```

4. **Start the server.**

JMX Port List

Here's the list of JMX ports, all of which are disabled by default. When these ports are enabled, they are used for all types of installations: single-server, distributed, and highly available:

Port	Used by this server process ...	Parameter
8300 - 8359	Application server JMX. Determined by the application server port(s) + 300.	--
8550	Background monitor JMX. Determined by the background port of 8250 + 300.	--
9095	Service monitor JMX.	svcmonitor.jmx.port

Port	Used by this server process ...	Parameter
9400 - 9499	VizQL server JMX. Determined by the VizQL server port(s) + 300.	--
10000 - 10299	Data server JMX. Determined by the data server port(s) + 300.	--

How the JMX Ports Are Determined

By default, the JMX ports for the application server (8300 - 8359), backgrounder (8550), VizQL server (9400 - 9599), and the data server (10000 - 10299) are assigned using the formula "base port + 300". (See [Tableau Server Ports on page 540](#) for a list of the default base ports.) In addition, if there are multiple instances of a process, each will have a JMX port. For example, if you configure Tableau Server to run four instances of the application server process, ports 8000 (default base port), 8001, 8002, and 8003 are used. Application server JMX ports 8300 (base port + 300), 8301, 8302, and 8303 are then bound to their respective process instances.

If dynamic port remapping is enabled (which is the default) and if a port conflict is detected, JMX ports are not determined using the "base port + 300" formula. Instead, both base ports and JMX ports are assigned to available ports starting at port 8000. No offset is used for JMX ports; they are assigned the next available port, just like base ports are. If it's important that you have a fixed JMX port, you can disable port remapping or change the base ports so that there are no port conflicts.

Even though they're not directly used by Tableau Server, if a JMX port is being used by another application, Tableau Server processes won't run. In addition, JMX ports cannot be edited directly using tabadmin. You change a JMX port by changing the base port for its process. In other words, if port 10000 isn't available for the data server JMX process, you use tabadmin (as described in [Edit the Default Ports on page 547](#)) to change the data server base port from 9700 to 9800. This will move the data server JMX port to 11000.

To reduce security risks, it's a good practice to configure your firewall to block outside traffic to the JMX ports.

Restore the Default Value for a Port

You can restore the default value for a port by following the procedure below:

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Restore the default port value by typing the following:

```
tabadmin set <workerX>.<parameter> --default
```

If Tableau Server is running on one machine, <workerX> is worker0. If you're running a cluster, worker0 is the primary, worker1 is your first worker server, worker2 is your second, and so on.

Here's an example:

```
tabadmin set worker0.wgserver.port --default
```

3. Restart Tableau Server by typing the following:

```
tabadmin restart
```

tabcmd

The tabcmd utility is one of the two command line tools that installs with Tableau Server (the other is [tabadmin on page 583](#)). The commands provided through tabcmd can help you automate common tasks, such as publishing workbooks in batches and administering users and groups. The tabcmd utility installs in the Tableau Server bin folder (C:\Program Files\Tableau Server\9.2\bin), but you can install and run tabcmd on another machine as well. For more information, see the following topics.

Install tabcmd

By default, the tabcmd command line utility installs with Tableau Server to the server's bin folder (for example, C:\Program Files\Tableau\Tableau Server\9.2\bin). You can run it from there. For administrative flexibility, you can also install it on another machine.

If you installed the tabcmd command line utility on computers that are not running Tableau Server and you are upgrading Tableau Server to a new major version (version 9.1 to version 9.2 for example), Tableau recommends you also upgrade standalone installations of tabcmd to avoid any potential incompatibilities between versions.

To install tabcmd on another machine:

1. Navigate to the extras folder on Tableau Server:

```
C:\Program Files\Tableau\Tableau Server--  
\9.2\extras\TabcmdInstaller.exe
```

2. Copy TabcmdInstaller.exe to the computer where you want to install it.
3. Double-click TabcmdInstaller.exe to run it.
4. Follow the prompts to install tabcmd.

Because tabcmd is a command line tool, and due to some limitations with the Windows operating system, Tableau recommends that you install tabcmd in a folder named tabcmd at the root of the C:\ drive (C:\tabcmd).

Running the tabcmd Setup program does not automatically add tabcmd to the Windows PATH variable, you will need to either explicitly call tabcmd using its full path or add its directory to the PATH variable.

How to Use tabcmd

The basic steps for using tabcmd are as follows:

1. Open the Command Prompt as an administrator.
2. Change to the Tableau Server bin folder.

For example: cd C:\Program Files\Tableau\Tableau Server\9.2\bin

Or you can include the location in the command.

3. Run the tabcmd command.

When you use tabcmd, you must establish an authenticated server session. The session identifies the Tableau Server and the Tableau Server user running the session. You can start a session first, and then specify your command next, or you can start a session and execute a command all at once. If you are using tabcmd to perform more than one task, you must run each task one after the other (serially), rather than in parallel.

Commands (such as `login`) and the options (such as `-s`, `-u`, etc.) are *not* case sensitive, but the values you provide (such as `p@ssw0rd` or `User@Example.com`) *are* case sensitive.

Examples

The following command demonstrates starting a session with the Tableau Server named `tabserver.myco.com`:

```
tabcmd login -s http://tabserver.myco.com -u admin -p p@ssw0rd!
```

The next example shows a command that deletes a workbook named `Sales_Workbook`:

```
tabcmd delete "Sales_Workbook"
```

Here's how to accomplish all of the above with one command—note that you do not need `login` here:

```
tabcmd delete "Sales_Workbook" -s http://tabserver.myco.com -u admin -p p@ssw0rd!
```

A Tableau Server can run multiple sites. When a workbook is on the Default site of a multi-site server you don't need to specify Default, the above command is sufficient. However, if the command applies to something on a site other than Default, you need to specify the site ID for that site (see [Login on page 573](#)). Here's the same command for a workbook that's on the West Coast Sales site (site ID `wsales`):

```
tabcmd delete "Sales_Workbook" -s http://tabserver.myco.com -t wsales -u admin -p p@ssw0rd!
```

The options `-s`, `-t`, `-u`, and `-p` are among the tabcmd global variables, which can be used with any command.

For more information, see [tabcmd Commands on page 556](#).

Status messages and logs

When a command is successful, tabcmd returns a status code of zero. A full error message for non-zero status codes is printed to **stderr**. In addition, informative or progress messages may be printed to **stdout**.

A full log named **tabcmd.log** that includes debugging, progress, and error messages is written to **C:\Users\<username>\AppData\Local\Tableau**.

tabcmd Global Options

The table below shows the options that are used by all commands. The **--server**, **--user**, and **--password** options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

Option (short)	Option (long)	Argument	Description
-h	--help		Displays the help for the command.
-c	--use-certificate		Use client certificate to sign in. Required when mutual SSL is enabled. For more information, see Configure External SSL on page 491 .
-s	--server	Tableau Server URL	Required at least once to begin session.
-u	--user	Tableau Server user-name	Required at least once to begin session.
-p	--password	Tableau Server password	Required at least once to begin session. You can alternatively use the -P option.
	--password-file	filename.txt	Allows the password to be stored in the given file rather than the command line for increased security.
-t	--site	Tableau Server site ID	Indicates that the command applies to the site specified by the site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

Option (short)	Option (long)	Argument	Description
-x	--proxy	Host:Port	Uses the specified HTTP proxy.
	--no-prompt		When specified, the command will not prompt for a password. If no valid password is provided the command will fail.
	--no-proxy		When specified, an HTTP proxy will not be used.
	--no-cert-check		When specified, tabcmd (the client) does not validate the server's SSL certificate.
	--[no-] cookie		When specified, the session id is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session id. By default the session is saved.
	--timeout	seconds	Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.
	--		Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command. tabcmd export --csv -f "D:\export10.csv" -- -

Option (short)	Option (long)	Argument	Description
			430105/Sheet1

tabcmd Commands

Here are the commands that can be used with the tabcmd command line tool:

- addusers (to group)
- creategroup
- createproject
- createsite
- createsiteusers
- createusers
- delete *workbook-name or datasource-name*
- deletegroup
- deleteproject
- deletesite
- deletesiteusers
- deleteusers
- editdomain
- editsite
- export
- get *url*
- listdomains
- listsites
- login
- logout
- publish
- refreshextracts
- removeusers
- runschedule

`set`
`syncgroup`
`version`

addusers *group-name*

Adds users to the specified group.

Example

```
tabcmd addusers "Development" --users "users.csv"
```

Option (short)	Option (long)	Argument	Description
	--users	filename.csv	Add the users in the given file to the specified group. The file should be a simple list with one user name per line. The users should already be created on Tableau Server. See also CSV Import File Guidelines on page 200 .
	-- [no-]complete		When <code>set</code> to <code>complete</code> this option requires that all rows be valid for any change to succeed. If not specified, <code>-complete</code> is used.

creategroup *group-name*

Creates a group. Use `addusers` (for local groups) and `syncgroup` (for Active Directory groups) commands to add users after the group has been created.

Example

```
tabcmd creategroup "Development"
```

createproject *project-name*

Creates a project.

Example

```
tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing quarterly sales reports."
```

Option (short)	Option (long)	Argument	Description
-n	--name	name	Specify the name of the project that you want to create.
-d	--description	description	Specify a description for the project.

createsite *site-name*

Creates a site.

Examples

Create a site named West Coast Sales. A site ID of WestCoastSales will be automatically created, the site will have no storage quota limit, and site administrators will be able to add and remove users:

```
tabcmd createsite "West Coast Sales"
```

Create a site named West Coast Sales with a site ID of wsales:

```
tabcmd createsite "West Coast Sales" -r "wcoast"
```

Prevent site administrators from adding users to the site:

```
tabcmd createsite "West Coast Sales" --no-site-mode
```

Set a storage quota, in MB:

```
tabcmd createsite "West Coast Sales" --storage-quota 100
```

Option (short)	Option (long)	Argument	Description
-r	--url	site ID	Used in URLs to specify the site. Different from the site name.
	--user-quota	number of users	Maximum number of users that can be added to the site.
	--[no-]site-mode		Allow or deny site administrators the ability to add users to or remove users from the site.
	--storage-quota	number of MB	In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

createsiteusers *filename.csv*

Adds users to a site, based on information supplied in a comma-separated values (CSV) file. If the user is not already created on the server, the command creates the user before adding that user to the site.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, role, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see [CSV Import File Guidelines on page 200](#). As an alternative to including role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the `--role` option.

If the server is configured to use local authentication, the information in the CSV file is used to create users. If the server is configured to use Active Directory authentication, user information is imported from Active Directory to the server. In that case, any password and friendly name information in the CSV file is ignored. In that case, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not unique across domains, you must include the domain as part of the user name (for example, `example\Adam` or `adam@example`).

By default, users are added to the site that you are logged in to. To add users to a different site, include the global `--site` option and specify that site. (You must have permissions to create users on the site you specify.)

If the server contains multiple sites, you cannot assign the `ServerAdministrator` role to a user by using the `createsiteusers` command. (Use `createusers` instead.) If you specify the `ServerAdministrator` role for the `role` option, the command returns an error. If the CSV file includes `System` as value for administrator, the value is ignored and the user is assigned the `Unlicensed` role. However, if the server contains only one site (the default site), you can assign the `ServerAdministrator` role or specify `system` for the administrator value; in that case, the `createsiteusers` command works like the `createusers` command.

By default, this command creates users using a synchronous operation (it waits for all operations to complete before proceeding). You can use the `--no-wait` option to specify an asynchronous operation.

Example

```
tabcmd createsiteusers "users.csv" --role "Interactor"
```

Option (short)	Option (long)	Argument	Description
	<code>--admin-type</code>	Site or None	(Deprecated. Use the <code>--role</code> option instead.) Assigns or removes the site administrator

Option (short)	Option (long)	Argument	Description
			right for any user who does not already have an administrator setting in the CSV file. The default is <code>None</code> for new users and unchanged for existing users. If the server contains multiple sites; system administrators cannot be created or demoted using <code>createsiteusers</code> . (Use <code>createusers</code> instead.)
	--com- plete		Requires that all rows be valid for any change to succeed. This is the default setting.
	--license	Interactor, Viewer, or Unlicensed	(Deprecated. Use the --role option instead.) Specifies the license level for any user who does not already have a license level setting in the CSV file. The default is <code>Unlicensed</code> for new users and unchanged for existing users.
			<p>Note: License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting in Tableau Server 9.0.</p>
	--no-com- plete		Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.
	--no-pub- lisher		(Deprecated. Use the --role option instead.) Disallows publishing rights for any users who

Option (short)	Option (long)	Argument	Description
			do not already have a publisher setting in the CSV file. This is a default value for new users.
	--nowait		Do not wait for asynchronous jobs to complete.
	--pub-lisher		(Deprecated. Use the --role option instead.) Assigns publishing rights for any user who does not already have a publisher setting in the CSV file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.
-r	--role	ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed	<p>Specifies a site role for any user who does not already have a role specified in the CSV file. The default is Unlicensed for new users and unchanged for existing users.</p> <p>If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.</p> <div style="background-color: #f0f8ff; padding: 10px; border-radius: 10px;"> <p>Note: You cannot assign the ServerAdministrator role if the server has more than one site. In that case, use the createuser command.</p> </div>

Option (short)	Option (long)	Argument	Description
			<p>Note: If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.</p>
	--silent-progress		Do not display progress messages for the command.

createusers *filename.csv*

Create users in Tableau Server, based on information supplied in a comma-separated values (CSV) file.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, role, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see [CSV Import File Guidelines on page 200](#). As an alternative to including role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the --role option.

If the server has only one site (the default site), the user is created and added to the site. If the server has multiple sites, the user is created but is not added to any site. To add users to a site, use [createsiteusers](#).

If the server is configured to use local authentication, the information in the CSV file is used to create users. If the server is configured to use Active Directory authentication, user information is imported from Active Directory to the server. In that case, any password and friendly name information in the CSV file is ignored. In that case, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not unique across domains, you must include the domain as part of the user name (for example, example\Adam or adam@example).

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

Example

```
tabcmd createusers "users.csv" --role "ServerAdministrator"
tabcmd createusers "users.csv"
```

Option (short)	Option (long)	Argument	Description
	--admin-type	Site or None	(Deprecated. Use the --role option instead.) Assigns or removes the site administrator right for any user who does not already have an administrator setting in the CSV file. The default is None for new users and unchanged for existing users.
	--complete		Requires that all rows be valid for any change to succeed. This is the default setting.
	--license	Interactor, Viewer, or Unlicensed	(Deprecated. Use the --role option instead.) Specifies the license level for any user who does not already have a license level setting in the CSV file. The default is Unlicensed for new users and unchanged for existing users.
			<p>Note: License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting with Tableau Server 9.0.</p>
	--no-complete		Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.
	--no-publisher		(Deprecated. Use the --role option instead.) Disallows publishing rights for any users who

Option (short)	Option (long)	Argument	Description
			do not already have a publisher setting in the CSV file. This is a default value for new users.
	--nowait		Do not wait for asynchronous jobs to complete.
	--pub- lisher		(Deprecated. Use the --role option instead.) Assigns publishing rights for any user who does not already have a publisher setting in the CSV file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.
-r	--role	ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, Unli- censedWithPublish, or Unlicensed	<p>Specifies a role for any user who does not already have a role specified in the CSV file. The default is Unlicensed for new users and unchanged for existing users.</p> <p>On a multi-site server, the command does not assign the user to a site. Therefore, the only roles that the command will assign are ServerAdministrator and Unlicensed. In that case, if you specify a different role (like Publisher or Viewer), the command assigns the Unlicensed role.</p> <p>On a single-site server, the user is created and added to the default site using the role that you specify.</p> <p>If you have a user-based server installation, and if the command</p>

Option (short)	Option (long)	Argument	Description
			<p>creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.</p> <p>Note: If you specify a <code>role</code> option, you cannot also include <code>license</code>, <code>publisher</code>, <code>no-publisher</code>, or <code>administrator</code> options.</p>
	--silent-progress		Do not display progress messages for the command.

delete *workbook-name* or *datasource-name*

Deletes the specified workbook or data source from the server.

This command takes the name of the workbook or data source as it is on the server, not the file name when it was published.

Example

```
tabcmd delete "Sales_Analysis"
```

Option (short)	Option (long)	Argument	Description
-r	--project	Project name	The name of the project containing the workbook or data source you want to delete. If not specified, the “Default” project is assumed.
	--workbook	Workbook name	The name of the workbook you want to delete.
	--data-source	Data source name	The name of the data source you want to delete.

deletegroup *group-name*

Deletes the specified group from the server.

Example

```
tabcmd deletegroup "Development"
```

deleteproject *project-name*

Deletes the specified project from the server.

Example

```
tabcmd deleteproject "Designs"
```

deletesite *site-name*

Deletes the specified site from the server.

Example

```
tabcmd deletesite "Development"
```

deletesiteusers *filename.csv*

Removes users from from the site that you are logged in to. The users to be removed are specified in a file that contains a simple list of one user name per line. (No additional information is required beyond the user name.)

By default, if the server has only one site, or if the user belongs to only one site, the user is also removed from the server. On a Tableau Server Enterprise installation, if the server contains multiple sites, users who are assigned the role of **Server Administrator** are removed from the site but are not removed from the server.

If the user owns content, the user's role is change to **Unlicensed**, but the user is not removed from the server or the site. The content is still owned by that user. To remove the user completely, you must change the owner of the content and then try removing the user again.

If the user was imported from Active Directory, the user is removed from the site and possibly from the server. However, the user is not deleted from Active Directory.

Example

```
tabcmd deletesiteusers "users.csv"
```

deleteusers *filename.csv*

Deletes the users listed in the specified comma-separated values (CSV) file.

The CSV file should contain a simple list of one user name per line.

Example

```
tabcmd deleteusers "users.csv"
```

Option (short)	Option (long)	Argument	Description
	-- [no-] complete		When set to --complete this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.

editdomain

Changes the nickname or full domain name of an Active Directory domain on the server.

You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to sign in. However, if the user name that you are currently signed in with exists in both the current domain and the new domain, you can modify the full name for the current domain.

To see a list of domains, use [listdomains](#).

Examples

```
tabcmd editdomain --id 2 --nickname "new-nickname"
```

```
tabcmd editdomain --id 3 --name "new-name"
```

Option (long)	Argument	Description
--id	Domain ID	The ID of domain to change. To get a list of domain IDs, use listdomains .
--name	Domain name	The new name for the domain.
--nickname	Domain nickname	The new nickname for the domain.

editsite *site-name*

Changes the name of a site or its web folder name. You can also use this command to allow or deny site administrators the ability to add and remove users. If site administrators have user management rights, you can specify how many users they can add to a site.

Examples

```
tabcmd editsite wc_sales --site-name "West Coast Sales"
```

```

tabcmd editsite wc_sales --site-id "wsales"
tabcmd editsite wsales --status ACTIVE
tabcmd editsite wsales --user-quota 50

```

Option (long)	Argument	Description
--site-name	Name to change the site to	The name of the site that's displayed.
--site-id	The site ID to change the site to	Used in the URL to uniquely identify the site.
--user-quota	Number of users	Maximum number of users who can be members of the site.
--[no-] site-mode		Allow or prevent site administrators from adding users to the site.
--status	ACTIVE or SUSPENDED	Activate or suspend a site.
--storage-quota	Number of MB	In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

export

Exports a view or workbook from Tableau Server and saves it to a file. This command can also export just the data used for a view.

Note the following when you use this command:

- **Permissions:** To export, you must have the **Export Image** permission. By default, this permission is Allowed or Inherited for all roles, although permissions can be set per workbook or view.
- **Exporting data:** To export just the data for a view, use the `--csv` option. This exports the summary data used in a view to a .csv file.
- **Specifying the view, workbook, or data to export:** You specify this using the "workbook/view" string as it appears in the URL for the workbook or view, not using its "friendly name," and excluding the `:iid=<n>` session ID at the end of the URL. For example, to export the Tableau sample view *Investment Growth* from the *Finance* workbook, you would use the string `Finance/InvestmentGrowth`, not `Finance/Investment Growth`, or `Finance/InvestmentGrowth?:iid=1`. Use `-t <site_id>` if the server is running multiple sites and the view or workbook is on a site other than Default.

To export a workbook, you still include a valid view in the string you use. Using the above example, to export the *Finance* workbook, you would use the string `Finance/InvestmentGrowth`. Finally, to export a workbook, it must have been published with **Show Sheets as Tabs** selected in the Tableau Desktop Publish dialog box.

- **The saved file's format:** Your format options depend on what's being exported. A workbook can only be exported as a PDF using the `--fullpdf` argument. A view can be exported as a PDF (`--pdf`) or a PNG (`--png`).
- **The saved file's name and location (optional):** If you don't provide a name, it will be derived from the view or workbook name. If you don't provide a location, the file will be saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

Note: You must include a file name extension such as `.csv` or `.pdf`. The command does not automatically add an extension to the file name that you provide.

- **Dashboard web page objects not included in PDF exports:** A dashboard can optionally include a web page object. If you are performing an export to PDF of a dashboard that includes a web page object, the web page object won't be included in the PDF.

Clearing the Cache to Use Real-Time Data

You can optionally add the URL parameter `?refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using `tabcmd` with your own scripting and the `refresh` URL parameter is being used a great deal, this can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

Examples

Views

```
tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report.csv"  
tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f  
"C:\Tableau_Workbooks\Weekly-Reports.pdf"  
tabcmd export "Finance/InvestmentGrowth" --png  
tabcmd export "Finance/InvestmentGrowth?refresh=yes" --png
```

Workbooks

```
tabcmd export "Q1Sales/Sales_Report" --fullpdf
```

```
tabcmd export #/Sales "Sales/Sales_Analysis" --fullpdf --pagesize
tabloid -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"
```

Option (short)	Option (long)	Argument	Description
-f	--filename	The name and extension to use for the saved file	Saves the file with the given filename.
	--csv		View only. Export the view's data (summary data) in CSV format.
	--pdf		View only. Export as a PDF.
	--png		View only. Export as an image in PNG format.
	--fullpdf		Workbook only. Export as a PDF. The workbook must have been published with Show Sheets as Tabs enabled.
	--pagelayout	landscape, portrait	Sets the page orientation of the exported PDF. If not specified, its Tableau Desktop setting will be used.
	--pagesize	unspecified, letter, legal, note folio, tabloid, ledger, statement, executive, a3, a4, a5, b4, b5, quarto	Sets the page size of the exported PDF. Default is letter.
	--width	Number of pixels	Sets the width. Default is 800 px.
	--height	Number of pixels	Sets the height. Default is 600 px.

get url

Gets the resource from Tableau Server that's represented by the specified URL. The result is returned as a file.

Note the following when you use this command:

- **Permissions:** To get a file, you must have the **Download/Web Save As** permission. By default, this permission is allowed or inherited for all roles, although permissions can be set per workbook or view.
- **File extension:** The URL must include a file extension, for example, `"/views/Finance/InvestmentGrowth.csv"`. The extension (.csv) determines what's returned. A view can be returned in PDF, PNG, CSV (summary data only), or XML (information only) format. A Tableau workbook is returned as a TWB if it connects to a published data source or uses a live connection, or a TWBX if it connects to a data extract.

To figure out the correct extension, you can use a web browser to navigate to the item on Tableau Server and add the file extension to the end of the URL.

When you type the URL for the GET request, exclude the session ID (`:iid=<n>`) that appears at the end of the file name. For example, use

`"/views/Finance/InvestmentGrowth.pdf"` instead of
`"/views/Finance/InvestmentGrowth?:iid=3.pdf".`

Note: If you are downloading a view to a PDF or PNG file, and if you include a `--filename` parameter that includes the .pdf or .png extension, you do not have to include a .pdf or .png extension in the URL.

- **The saved file's name and location (optional):** The name you use for `--filename` should include the file extension. If you don't provide a name and file extension, both will be derived from the URL string. If you don't provide a location, the file is saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.
- **PNG size (optional):** If the saved file is a PNG, you can specify the size, in pixels, in the URL.

Clearing the Cache to Use Real-Time Data

You can optionally add the URL parameter `?refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using tabcmd with your own scripting, using the `refresh` parameter a great deal can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

Examples

Views

```
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename  
"Weekly-Report.png"  
  
tabcmd get "/views/Finance/InvestmentGrowth.pdf" -f  
"Q1Growth.pdf"  
  
tabcmd get "/views/Finance/InvestmentGrowth" -f "Q1Growth.pdf"  
  
tabcmd get "/views/Finance/InvestmentGrowth.csv"  
  
tabcmd get "/views/Finance/InvestmentGrowth.png?:size=640,480" -f  
growth.png  
  
tabcmd get "/views/Finance/InvestmentGrowth.png?:refresh=yes" -f  
growth.png
```

Workbooks

```
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Work-  
books\Weekly-Reports.twb"  
  
tabcmd get "/workbooks/Sales.xml"
```

Other

```
tabcmd get "/users.xml" --filename "UserList.xml"
```

Option (short)	Option (long)	Argument	Description
-f	--filename	Name to save the file as	Saves the file with the given file- name.

listdomains

Displays a list of the Active Domain domains that are in use on the server, along with their nicknames and IDs. If the server is configured to use local authentication, the command returns only the domain name `local`.

Example

```
tabcmd listdomains
```

listsites

Returns a list of sites to which the logged in user belongs.

Example

```
tabcmd listsites -u adam -pw P@ssword!
```

login

Logs in a Tableau Server user.

Use the `--server`, `--site`, `--username`, `--password` global options to create a session.

Note: When you use the **tabcmd login** command, you cannot use SAML single sign-on (SSO), even if the server is configured to use SAML. To log in, you must pass the user name and password of a user who has been created on the server. You will have the permissions of the Tableau Server user that you're signed in as. For more information, see [Site Roles for Users on page 176](#) and [Manage Permissions on page 336](#).

If you want to log in using the same information you've already used to create a session, just specify the `--password` option. The server and user name stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify the port.

You need the `--site` (`-t`) option only if the server is running multiple sites and you are logging in to a site other than the Default site. If you do not provide a password you will be prompted for one. If the `--no-prompt` option is specified and no password is provided the command will fail.

Once you log in, the session will continue until it expires on the server or the `logout` command is run.

Example

Logs you in to the Tableau Server running on your local machine:

```
tabcmd login -s http://localhost -u jsmith -p p@ssw0rd!
```

Logs you in to the Sales site on sales-server:

```
tabcmd login -s http://sales-server -t Sales -u administrator -p p@ssw0rd!
```

```
tabcmd login -s http://sales-server:8000 -t Sales -u administrator -p p@ssw0rd!
```

Logs you in to the Sales site on sales-server using SSL but does not validate the server's SSL certificate:

```
tabcmd login --no-certcheck -s https://sales-server -t Sales -u administrator -p p@ssw0rd!
```

Establishes a forward proxy and port for localhost:

```
tabcmd login --proxy myfwdproxyserver:8888 -s http://localhost -u jsmith -p p@ssW0rd!
```

Logs you in to the reverse proxy using SSL:

```
tabcmd login -s https://myreverseproxy -u jsmith -p p@ssW0rd!
```

Option (short)	Option (long)	Argument	Description
-s	--server	server URL	If you are running the command from an on-premises Tableau Server computer, you can use http://localhost. Otherwise, specify the computer's URL, such as http://bigbox.myco.com or http://bigbox. For Tableau Online specify the URLhttps://online.tableau.com.
-t	--site	site ID	Include this option if the server has multiple sites, and you are logging in to a site other than the Default site. The site ID is used in the URL to uniquely identify the site. For example, a site named West Coast Sales might have a site ID of west-coast-sales.
-u	--username	user name	The user name of the user logging in. For Tableau Online, the user name is the user's email address.
-p	--password	password	Password for the user specified for --username. If you do not provide a password you will be prompted for one.
	--pass-word-file	filename.txt	Allows the password to be stored in the given file rather than the command line, for increased security.
-x	--proxy	Host:Port	Use to specify the HTTP proxy server and port for the tabcmd request.

Option (short)	Option (long)	Argument	Description
	--no-prompt		Do not prompt for a password. If no password is specified, the <code>login</code> command will fail.
	--no-proxy		Do not use an HTTP proxy server.
	--cookie		Saves the session ID on login. Subsequent commands will not require a login. This value is the default for the command.
	--no-cookie		Do not save the session ID information after a successful login. Subsequent commands will require a login.
	--timeout SECONDS	Number of seconds	The number of seconds the server should wait before processing the <code>login</code> command. Default: 30 seconds.

logout

Logs out of the server.

Example

```
tabcmd logout
```

publish *filename.twb(x), filename.tds(x), or filename.tde*

Publishes the specified workbook (.twb(x)), data source (.tds(x)), or data extract (.tde) to Tableau Server.

If you are publishing a workbook, by default, all sheets in the workbook are published without database user names or passwords.

The permissions initially assigned to the workbook or data source are copied from the project that the file is published to. Permissions for the published resource can be changed after the file has been published.

If the workbook contains user filters, one of the thumbnail options must be specified.

Example

```
tabcmd publish "analysis.twbx" -n "Sales_Analysis"  
--db-username "jsmith" --db-password "p@ssw0rd"
```

```
tabcmd publish "analysis_sfdc.tde" -n "Sales Analysis"  
--oauth-username "username" --save-oauth
```

If the file is not in the same directory as tabcmd, include the full path to the file.

Example

```
tabcmd publish "C:\Tableau Workbooks\analysis.twbx" -n "Sales Analysis" --db-username "jsmith" --db-password "p@ssw0rd"
```

```
tabcmd publish "C:\Tableau Workbooks\analysis_sfdc.tde" -n "Sales Analysis" --oauth-username "username" --save-oauth
```

Option (short)	Option (long)	Argument	Description
-n	--name	Name of the workbook or data source on the server	If omitted, the workbook, data source, or data extract will be named after filename.
-o	--overwrite		Overwrites the workbook, data source, or data extract if it already exists on the server.
-r	--project	Name of a project	Publishes the workbook, data source, or data extract into the specified project. Publishes to the “Default” project if not specified.
	--db-user-name		Use this option to publish a database user name with the workbook, data source, or data extract.
	--db-pass-word		Use this option to publish a database password with the workbook, data source, or data extract.
	--save-db-password		Stores the provided database password on the server.
	--oauth-username	Email address of the user account	Connects the user through a preconfigured OAuth connection, if the user already has a saved access token for the cloud data source specified in --name. Access tokens are managed in user preferences. For existing OAuth connections to the data source, use this option instead of --db-username and --db-password.
	--save-oauth		Saves the credential specified by --oauth-username as an embedded cre-

Option (short)	Option (long)	Argument	Description
			<p>dential with the published workbook or data source.</p> <p>Subsequently, when the publisher or server administrator signs in to the server and edits the connection for that workbook or data source, the connection settings will show this OAuth credential as embedded in the content.</p> <p>If you want to schedule extract refreshes after publishing, you must include this option with <code>--oauth-username</code>. This is analogous to using <code>--save-db-password</code> with a traditional database connection.</p>
	<code>--thumb-nail-user-name</code>		If the workbook contains user filters, the thumbnails will be generated based on what the specified user can see. Cannot be specified when <code>--thumbnail-group</code> option is set.
	<code>--thumb-nail-group</code>		If the workbook contains user filters the thumbnails will be generated based on what the specified group can see. Cannot be specified when <code>--thumbnail-username</code> option is set.
	<code>--tabbed</code>		When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.
	<code>--append</code>		Append the extract file to the existing data source.
	<code>--replace</code>		Use the extract file to replace the existing data source.
	<code>--disable-uploader</code>		Disable the incremental file uploader.
	<code>--restart</code>		Restart the file upload.

refreshextracts *workbook-name* or *datasource-name*

Performs a full or incremental refresh of extracts belonging to the specified workbook or data source.

This command takes the name of the workbook or data source as it appears on the server, not the file name when it was published. Only an administrator or the owner of the workbook or data source is allowed to perform this operation.

Examples

```
tabcmd refreshextracts --datasource sales_ds  
tabcmd refreshextracts --workbook "My Workbook"  
tabcmd refreshextracts --url SalesAnalysis
```

Option (short)	Option (long)	Argument	Description
	--incremental		Runs the incremental refresh operation.
	--synchronous		Adds the full refresh operation to the queue used by the Backgrounder process, to be run as soon as a Backgrounder process is available. If a Backgrounder process is available, the operation is run immediately. The refresh operation appears on the Background Tasks report. During a synchronous refresh, tabcmd maintains a live connection to the server while the refresh operation is underway, polling every second until the background job is done.
	--workbook	Name of a workbook	The name of the workbook containing extracts to refresh. If the workbook has spaces in its name, enclose it in quotes.
	--datasource	Name of a data source	The name of the data source containing extracts to refresh.
	--project	Name of a project	Use with --workbook or --datasource to identify a workbook or data source in a project other than <i>Default</i> . If not specified, the Default project is assumed.

Option (short)	Option (long)	Argument	Description
	--url	URL name of a workbook	The name of the workbook as it appears in the URL. A workbook published as "Sales Analysis" has a URL name of "SalesAnalysis".

removeusers *group-name*

Removes users from the specified group.

Example

```
tabcmd removeusers "Development" --users "users.csv"
```

Option (short)	Option (long)	Argument	Description
	--users	filename.csv	Remove the users in the given file from the specified group. The file should be a simple list with one user name per line.
	-- [no-] complete		Requires that all rows be valid for any change to succeed. If not specified --complete is used.

runschedule *schedule-name*

Runs the specified schedule.

This command takes the name of the schedule as it is on the server.

For Tableau Online, the command can be run within the scope of a single site, using site administrator permissions.

Example

```
tabcmd runschedule "5AM Sales Refresh"
```

set *setting*

Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server.

Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- allow_scheduling
- embedded_credentials
- remember_passwords_forever

Example

```
tabcmd set embedded_credentials
```

syncgroup *group-name*

Synchronizes a Tableau Server group with an Active Directory group. If the Tableau Server group does not already exist, it is created and synchronized with the specified Active Directory group.

Example

```
tabcmd syncgroup "Development"
```

Note: If you synchronize a group that you are a member of, changes that you make using this command do not apply to your user. For example, if you use this command to remove the administrator right from users in a group that you are a member of, you are still an administrator when the command finishes.

Option (short)	Option (long)	Argument	Description
	--administrator	System, Site, or None	(Deprecated. Use the --role option instead.) Assigns or removes the administrator right for users in the group. The None option removes the administrator right from all users in the group (except you, if you are a member of the group that you are synchronizing). If you do not include this option, users who are added to the group after you run the command are not assigned the administrator right.

Option (short)	Option (long)	Argument	Description
	--complete		Requires that all rows be valid for any change to succeed. This is the default setting.
	--license	Interactor, Viewer, or Unlicensed	(Deprecated. Use the --role option instead.) Specifies the license level for users in the group.
			<p>Note: License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting in Tableau Server 9.0.</p>
	--no-complete		Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.
	--no-publisher		(Deprecated. Use the --role option instead.) Disallows publishing rights for users in the group.
	--publisher		(Deprecated. Use the --role option instead.) Assigns publishing rights to users in the group.
-r	--role	ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, Unli-	Specifies a role for users in the group. The default is Unlicensed.

Option (short)	Option (long)	Argument	Description
		censedWithPublish, or Unlicensed	<p>Note: If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.</p>
	--silent-progress		Do not display progress messages for the command.

version

Displays the version information for the current installation of the tabcmd utility.

Example

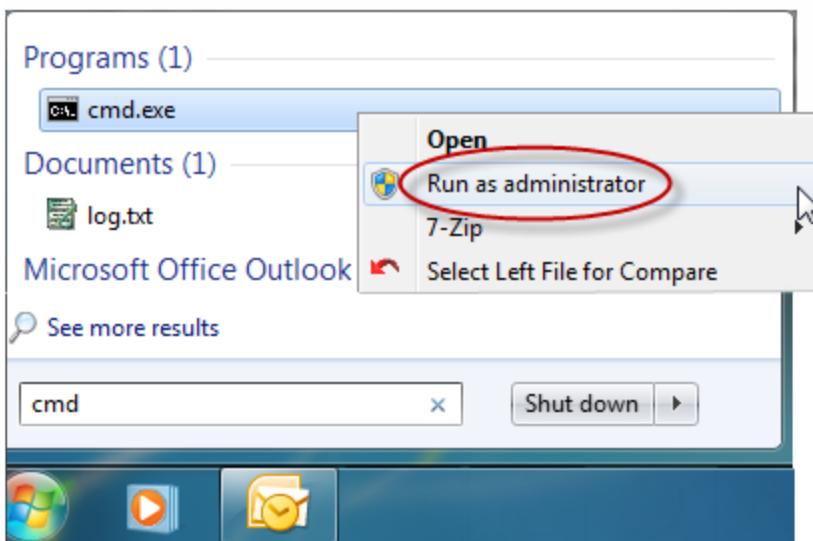
```
tabcmd version
```

tabadmin

You can perform certain administrative tasks and change Tableau Server configuration settings using the tabadmin command line tool. It installs with Tableau Server by default and cannot be installed on other computers. For more information, see the following topics.

How to Use tabadmin

tabadmin allows you to perform administrative tasks from the command line on Tableau Server. It installs with Tableau Server by default and cannot be installed on other machines. The first step to using tabadmin is to open a command prompt as an administrator:



Next, navigate to Tableau Server's bin directory by entering the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

You're now ready to enter [tabadmin commands](#).

Change Tableau Server's Configuration from the Command Line

When you enter a command that changes the server's configuration (a `tabadmin set` command for example), you need to follow a sequence of commands:

1. [Stop the server](#) before issuing the command.
2. Enter the appropriate command to make the configuration change.
3. Run `tabadmin config` to push the change out to all of the server's configuration files.
4. [Start Tableau Server again](#).

Example

Change the server's configuration using the `tabadmin set` command:

```
tabadmin stop  
tabadmin set [option-name value]  
tabadmin config  
tabadmin start
```

Display Command Line Help

Use the `tabadmin` built-in help to get a quick description of a command.

To display help for all `tabadmin` commands enter:

```
tabadmin help commands
```

To see help for a specific command, enter `tabadmin help <command>`. For example:

```
tabadmin help set
```

tabadmin Commands

Here are the commands that can be used with the `tabadmin` command line tool:

[activate](#) on page 586

[administrator](#) on page 587

[assetkeys](#) on page 587

[autostart](#) on page 589

[backup](#) on page 589

[cleanup](#) on page 590

[clearcache](#) on page 592

[configure](#) on page 593

[customize](#) on page 594

[dbpass](#) on page 594

[decommission](#) on page 596

[delete_webdataconnector](#) on

page 596

[exportsite](#) on page 597

failoverprimary on page 599
failoverrepository on page 599
get_openid_redirect_url on
page 600
importsite on page 600
importsite_verified on page 602
import_webdataconnector on
page 603
licenses on page 604
list_webdataconnectors on
page 605
manage_global_credentials on
page 605
passwd on page 606
reccommission on page 607
regenerate_internal_tokens on
page 607
reindex on page 608
reset_openid_sub on page 608
restart on page 609
restore on page 609
set on page 610
sitestate on page 611
start on page 612
status on page 613
stop on page 613
validate on page 614
warmup on page 615
ziplogs on page 615

activate

Activates or returns a Tableau Server license online or offline.

Examples

Activate a license offline:

```
tabadmin activate --tlf <file.tlf>
```

Return a license offline:

```
tabadmin activate --tlr <file.tlr>
```

Activate a license online:

```
tabadmin activate --activate <license>
```

Return a license online:

```
tabadmin activate --return <license>
```

Option (short)	Option (long)	Argument	Description
	--tlf	FILE	For offline activation. If you are offline during Setup, you are prompted to save a .tlq file, which you submit to Tableau. Tableau sends you a .tlf file. You use this .tlf file to activate Tableau Server.
	--tlr	FILE	For offline deactivation. The file you use as the argument is the .tlr file that you receive from Tableau.
	--activ- ate		Activate the specified license.
	--return		Return the specified license.

See Also

[Activate Tableau Offline on page 10](#)

administrator

Grants or removes the system administrator capability to the named user. This command does not apply to site administrators.

Examples

Remove the system administrator capability from user hwilson:

```
tabadmin administrator hwilson false
```

Give the system administrator capability to user jsmith:

```
tabadmin administrator jsmith true
```

assetkeys

Creates a new key to encrypt sensitive information, such as credentials for external databases, stored within the Tableau repository, which is a PostgreSQL database that Tableau Server uses internally. The key you create with this command can contain either a passphrase that you specify or one that's randomly generated.

If you specify your key's passphrase, it's a best practice for it to be at least eight characters long. You should also take character sets into consideration. A strong passphrase should contain characters from at least three of the following character sets:

- Lowercase a-z
- Uppercase A-Z
- Digits 0-9
- Non-alphabetic characters

The new key is encrypted and stored in the following key file: **asset_keys.yml** (ProgramData\Tableau\Tableau Server\data\tabsvc\config). If the key file is lost or corrupted, you can use the `assetkeys --validate` command to recreate it.

If you use the `assetkeys` command then later create and restore a backup file (.tsbak), you will need to run the `tabadmin assetkeys --validate` command after restoring the backup file. By design, backup files do not contain custom encryption keys—even though some data may be encrypted with them. This protects the encrypted values in case the backup file falls into the wrong hands. When you run `tabadmin assetkeys --validate` after a backup restore, you are prompted to enter the key's passphrase.

Examples

Have Tableau Server generate a key and passphrase for you:

```
tabadmin assetkeys --auto_create
```

Generate a key using a passphrase that you specify. You are prompted to enter a passphrase, which will not be displayed as you type:

```
tabadmin assetkeys --create
```

Use the contents of a file as the passphrase:

```
tabadmin assetkeys --create_from_file C:\test\key\password.txt
```

Confirm that the key file **asset_keys.yml** in ProgramData\Tableau\Tableau Server\data\tabsvc\config is valid and consistent with the metadata in the Tableau Repository:

```
tabadmin assetkeys --validate
```

Recreate the file **asset_keys.yml** which is now corrupted or missing from ProgramData\Tableau\Tableau Server\data\tabsvc\config:

```
tabadmin assetkeys --validate
```

You will be prompted for the passphrase.

Option (short)	Option (long)	Argument	Description
	--auto_create	[length]	Generates a random passphrase to generate the key. Takes an optional argument for the length of the passphrase. You should record the passphrase and keep it in a safe place, as it will be required by --validate if assetkeys.yml is lost or corrupted.
	--create	PASSPHRASE	Creates the passphrase of your choice to be used as the key. Your passphrase should be at least 10 characters long and not based on words found in the dictionary.
	--create_from_file	FILE	Generates a key using the contents of a file that you provide as the passphrase.
	--validate		Confirms that all asset keys being used internally by Tableau Server are up-to-date. If you lose the asset_keys.yml file (for example, due to file corruption), you can use the --validate option to recreate it. To successfully recreate the key file,

Option (short)	Option (long)	Argument	Description
			you must supply the passphrase used to generate any keys currently in use.

See Also

[Security on page 414](#)

autostart

Specifies whether Tableau Server starts at system start-up time. By default, Tableau Server starts when the computer on which it's installed starts. If `autostart` is set to `off`, you will need to start Tableau Server either using `tabadmin start` or the Start menu.

Example

Display Tableau Server's auto-start status:

```
tabadmin autostart
```

Start Tableau Server when the operating system starts:

```
tabadmin autostart on
```

Do not start Tableau Server when the operating system starts:

```
tabadmin autostart off
```

backup

Creates a backup of the data managed by Tableau Server. This data includes Tableau's own PostgreSQL database, which contains workbook and user metadata, data extract (.tde) files, and configuration data. If you have imported `web data connectors` using the [`import_webdataconnector` on page 603](#) command, the backup process saves copies of the connectors as well. You do not need to stop Tableau Server before you create a backup file.

By default, the backup file is put into the directory where you are running the `tabadmin backup` command. To put the backup file into a specific location, you can include full path with the backup file name. You can also use the `--userdir` option to put the backup file into a known location.

Note: The command adds the .tsbak extension to the file name that you specify unless the name already contains that extension.

Examples

Create a backup file in the current directory named **tabserv.tsbak**:

```
tabadmin backup tabserv.tsbak
```

Create a backup file in the C:\backups\tableau folder named **tabserv.tsbak**:

```
tabadmin backup C:\backups\tableau\tabserv.tsbak
```

Append the current date to the backup file name and put temporary files created during the backup process in C:\mytemp\tableau. The backup file **tabserv.tsbak** is created in the directory where you are running the command from:

```
tabadmin backup tabserv.tsbak -d -t C:\mytemp\tableau
```

Option (short)	Option (long)	Argument	Description
-d	--date		Appends the current date to the backup file name.
-u	--user-dir		Places the backup file in the ProgramData\Tableau\Tableau Server folder.
-t	--tempdir	PATH	Specifies the location for temporary files created during the backup process.

See Also

[Back Up the Tableau Data](#) on page 631

cleanup

Reduces the disk space consumed by Tableau Server. Running `tabadmin cleanup` removes log files, temporary files, and select rows in Tableau Server's PostgreSQL database. If Tableau Server is installed on multiple computers in a cluster, the command can also reset the information maintained by the coordination server that is used to synchronize between nodes and to manage failover.

The effect of the **cleanup** command depends on whether the server is running or stopped. For more information, see [Remove Unneeded Files](#) on page 634.

Examples

Remove log files, temporary files, and HTTP request entries in the PostgreSQL database:

```
tabadmin cleanup
```

Remove log files and temporary files (leave HTTP request entries in the database untouched):

```
tabadmin cleanup --restart
```

Option (short)	Option (long)	Argument	Description
-r	--restart		Stops Tableau Server, runs the cleanup command, and starts the server again.
	--reset-coordination		In addition to performing a normal cleanup, removes log files, transaction logs, and snapshots that are maintained by the Tableau Server coordination service (zookeeper) when Tableau Server is running on multiple computers in a cluster. Note that using this option completely resets the coordin-

Option (short)	Option (long)	Argument	Description
			<p>ation service, meaning all state maintained by the coordination service is removed. This option also does the equivalent of a tabadmin configure command. For guidelines about when to reset the coordination service, see Troubleshoot Server Processes on page 242.</p>

See Also

[Remove Unneeded Files](#) on page 634

clearcache

Clears the information being cached by the Cache Server process (redis-server.exe). The cache stores information used to render views in order to help speed rendering. Clearing the cache is useful if metadata about views or data sources that might be cached has changed, and those changes should take effect before the resource is removed from the cache in the normal course of server processing. For example, clearing the cache can be useful if you change permissions on a workbook or view and it's important that the changed permissions take effect immediately.

You must stop the server before you run this command. If you are running Tableau Server in a cluster, you must run the command on each computer where the Cache Server process is running.

Examples

```
tabadmin clearcache
```

See Also

[Tableau Server Processes](#) on page 637

configure

Updates Tableau Server's configuration by forcing an update to all the files in ProgramData\Tableau\Tableau Server\data\tabsvc\<area>. This update includes refreshing the master service configuration file, workgroup.yml (ProgramData\Tableau\Tableau Server\data\tabsvc\config). When you make a configuration change, it's a best practice to run tabadmin configure (or tabadmin config) to ensure that all files affecting the server's configuration are completely updated.

If you are running Tableau Server in a distributed environment and if you have imported [web data connectors](#) using the [import_webdataconnector](#) on page 603 command or deleted them using the [delete_webdataconnector](#) on page 596 command, the configure command makes sure that any web data connectors are correctly distributed (imported or deleted) in all nodes where the gateway process is running.

Examples

```
tabadmin configure
```

```
tabadmin config
```

See Also

[Reconfigure the Server](#) on page 35

[set](#) on page 610

[tabadmin set options](#) on page 616

customize

Customizes the name and logo that are used by Tableau Server. Note that even if you use this command, the copyright information at the bottom of every server page will list Tableau's copyright information.

Example

Change the product name used in tooltips from "Tableau Server" to "My Company Server":

```
tabadmin customize name "My Company Server"
```

Change the default logo to your own logo (up to 160 x 160 px but not smaller than 32 x 32 px):

```
tabadmin customize logo "C:\My Pictures\example.png"
```

Reset the product name to the default:

```
tabadmin customize name -d
```

Option (short)	Option (long)	Argument	Description
-d	-- default	name logo	Resets the name or logo to its default value.
	name	NAME	Sets the name to the value in the argument.
	logo	FILE	Sets the logo to the image referenced in the path.

See Also

[Change the Name or Logo on page 285](#)

dbpass

Enables external access to Tableau's PostgreSQL database (the repository). After you use the dbpass command to allow access to the database, you can connect to and query it using Tableau Desktop to create your own administrative views.

```
tabadmin dbpass [--disable] [--username <username>] [password]
```

Note: The --username option is valid starting with Tableau Server 8.2.5. In earlier versions dbpass only enabled the "tableau" user and you could not specify the user.

8.2.5 added a second user called "readonly" and introduced the ability to specify the user you are enabling access for.

Examples

Enable access for the `tableau` user and set the password to `p@ssword`:

```
tabadmin dbpass p@ssword
```

Enable access for the `readonly` user and set the password to `p@ssword`:

```
tabadmin dbpass --username readonly p@ssword
```

Disable external access for the default (`tableau`) user:

```
tabadmin dbpass --disable
```

or

```
tabadmin dbpass --disable --username tableau
```

Disable external access for the `readonly` user:

```
tabadmin dbpass --disable --username readonly
```

Option (long)	Argument	Description
<code>--dis-</code> <code>able</code>		Disable external access to Tableau's PostgreSQL database for the default remote user (<code>tableau</code>) or, starting in 8.2.5, if a user name is specified, disable remote access for that user.
<code>--user-</code> <code>name</code>	<code>tableau</code> or <code>readonly</code>	Change the password for the specified user, or, if used with the <code>--disable</code> option, disable access for the specified user. Options for users are <code>tableau</code> and <code>readonly</code> . This option is valid in Tableau Server 8.2.5 or higher.
	<code>password</code> provided by user	Enable remote access to Tableau's PostgreSQL database for the default remote user (<code>tableau</code>) or, starting in 8.2.5, if a user name is specified, enable access for that user with the given password.

See Also

[Create Custom Administrative Views on page 300](#)

[Enabling External Access to the Tableau Server Database on page 301](#)

decommission

Prepares Tableau Server File Store nodes for removal from the distributed installation. This command puts the specified nodes into read-only mode so new content cannot be added to the File Store, and makes sure that all content on the node also exists on another File Store node. This command can be run while Tableau Server is running.

Note: Remove a decommissioned File Store node before restarting Tableau Server. Restarting automatically re-activates any decommissioned File Store nodes.

```
tabadmin decommission <node1 node2 ...>
```

Examples

Decommission worker2:

```
tabadmin decommission worker2
```

Decommission two nodes by IP address:

```
tabadmin decommission 10.32.139.30 10.32.139.22
```

Option (long)	Argument	Description
	<node1 node 2 node 3...>	List of File Store nodes (servers) to decommission. Separate multiple nodes with a space.

See Also

[Distributed Environments](#) on page 73

[Maintain a Distributed Environment](#) on page 82

delete_webdataconnector

Removes the specified web data connector from the server, or removes all web data connectors. If the web data connector is installed on a cluster, this command removes the specified connector or all connectors from all computers in the cluster.

Note: If the server is running in a distributed environment and the delete process is partially successful, users can still access the connector. For more information, see [Web Data Connectors in Tableau Server](#) on page 276.

Examples

```
tabadmin delete_webdataconnector connector1.html
```

```
tabadmin delete_webdataconnector --all
```

Option (short)	Option (long)	Argument	Description
	--all		<p>Removes all web data connectors from Tableau Server. When you use this option, you do not specify a connector name.</p> <p>If the server is configured as a cluster, the command removes all connectors from all the nodes where they are installed.</p>

See Also

[import_webdataconnector](#) on page 603

[list_webdataconnectors](#) on page 605

[Web Data Connectors in Tableau Server](#) on page 276

exportsite

Exports a Tableau Server site, including its users, workbooks, projects, extracts, and data connections, and places it in a file with a .zip file extension. You can then use the exported site file to provision a new site by using the [importsite](#) on page 600 and [importsite_verified](#) on page 602 commands.

You don't need to stop Tableau Server before you use the exportsite command. Tableau Server will lock the site being exported during the export process.

Note: When you import a site that you exported earlier, each user and schedule that is being imported must match an existing user and schedule. For suggestions about how to manage the export and import process to match users and schedules, see [Tips for importing to a target with fewer users or schedules than the source site](#).

Examples

```
tabadmin exportsite <site ID> --file <PATH>
```

or

```
tabadmin exportsite <site ID> --file <FILE>
```

Export the site whose site ID is **finance** to a file named **finance_export.zip** and place it in Program Files\Tableau\Tableau Server\9.2\bin:

```
tabadmin exportsite finance --file finance_export
```

Export the Default site. The site ID for the Default site is "" (double quotes, no space).

```
tabadmin exportsite "" --file finance_export
```

If you are using Windows PowerShell to run the command, enclose the double quotes for the Default site within single quotes ('"'). For example: `tabadmin exportsite '''' --file finance_export`

Export the Default site to a file named **finance_export.zip** and place it in C:\temp\exported sites instead of in the Tableau Server bin directory. Because the path contains a space, it's contained by quotes:

```
tabadmin exportsite "" --file "C:\temp\exported sites\finance_export"
```

Export the site whose site ID is **finance**, name the export site file **financesite.zip**, place the file in C:\sites\exported, and write temporary run-time files to C:\temp_files:

```
tabadmin exportsite finance --file C:\sites\exported\financesite  
--tempdir C:\temp_files
```

Option (short)	Option (long)	Argument	Description
	--file	FILE or PATH	The name or name and location (path) of the exported site file to be created. If you don't specify a path, Tableau Server's bin directory is the assumed location (Program Files\Tableau\Tableau Server\9.2\bin).
	--tempdir		The location of temporary files created during export. Use this option if you don't have write access to the Tableau Server installation directory. This option does not determine where the export site file is created.

See Also

[Import or Export a Site on page 127](#)

failoverprimary

Identifies a second installation of the primary Tableau Server as the backup primary, or if the primary has failed, identify the backup primary as the new primary and the former primary as the new backup.

Example

```
tabadmin failoverprimary --primary <computer name(s) or IPv4 address(es)>
```

Option (short)	Option (long)	Argument	Description
	--primary	Computer name(s) or IPv4 address(es)	The Tableau Server machine that's acting as the cluster's primary.

See Also

[Understanding High Availability on page 88](#)

[Configure for Failover and Multiple Gateways on page 94](#)

[Use a Backup Primary on page 108](#)

failoverrepository

Identifies a second installation of the PostGRES repository as the active repository.

Tableau Server must be running when you run the failoverrepository command.

```
tabadmin failoverrepository --target <computer name or IPv4 address> | --preferred
```

Example

```
tabadmin failoverrepository --target worker_server2
```

Note: This command is persistent. The failover repository remains the active repository until you issue the command again. If you have a preferred active repository configured, use the --preferred option to switch back to that repository.

Option (short)	Option (long)	Argument	Description
	--target	Computer name or IPv4 address	The Tableau Server repository node to failover to.
	--preferred		Failover to the repository node that is specified as the preferred active repository.

See Also

[Understanding High Availability on page 88](#)

[Configure for Failover and Multiple Gateways on page 94](#)

[Use a Backup Primary on page 108](#)

get_openid_redirect_url

If Tableau Server is configured to use OpenID Connect for authentication, gets the URL that is used to redirect users from the identity provider (IdP) to Tableau Server after a successful sign-in.

Example

```
tabadmin get_openid_redirect_url
```

See Also

[OpenID Connect on page 517](#)

[Configure Tableau Server for OpenID Connect on page 519](#)

importsite

Imports a site into Tableau Server. The importsite command is the first of two commands you use to import a site into Tableau Server. To run this command, you need the following:

- **An exported site file.** Tableau Server administrators create this file using the [exportsite](#) on page 597 command. If you have a site on Tableau Online and you want to import it into your own on-premises installation of Tableau Server, request an exported site file from Tableau Customer Support.
- **The site ID for the target site.** The target site is the Tableau Server site into which you want to import. The target site must already exist when you run the importsite command; you can't create it as part of the command. The site ID for Tableau Server's default site is "" (double quotes, no space).

The contents of the site that you import will replace (not amend) the contents of the target site. For example, if your target site has a workbook named **MyDashboard.twbx** and the site you are importing does not have this workbook, the import process will remove **MyDashboard.twbx** from the target site.

When you run the importsite command, the command creates a temporary directory containing mapping files in comma-separated-value (CSV) format that define how the exported site's assets (users, workbooks, projects, extracts, and data sources) will be mapped when the site has been imported. It is important that you verify these details. Use a text editor or Microsoft Excel to open the mapping files and make any changes. Any entries with ??? (question marks) represent mappings that couldn't be handled and must be edited. After you verify the mappings, finish the import process using the [importsite_verified on the next page](#) command.

Note: When you import a site that you exported earlier, each user and schedule that is being imported must match an existing user and schedule. For suggestions about how to manage the export and import process to match users and schedules, see [Tips for importing to a target with fewer users or schedules than the source site](#).

Examples

```
tabadmin importsite <site ID> --file <PATH>
```

or

```
tabadmin importsite <site ID> --file <FILE>
```

Import the file **sales_site.zip** located in C:\tableau\exported to a site whose site ID is **wsales**:

```
tabadmin importsite wsales --file C:\tableau\exported\sales_site.zip
```

Import the file **sales_site.zip**, which is located in located in C:\Program Files\Tableau\Tableau Server\9.2\bin, to the Default site. The site ID for the Default site is "" (double quotes, no space).

```
tabadmin importsite "" --file sales_site.zip
```

The mapping files for you to verify are placed in ProgramData\Tableau\Tableau Server\data\tabsvc\temp\import_<site ID>_<datetime>\mappings. To specify a different directory, use the `--tempdir` option.

Place the files to be verified in C:\temp\site_to_import:

Skip the verification step (not recommended):

```
tabadmin importsite wsales --file "C:\tableau\exported\sales_site.zip" -no-verify
```

Option (short)	Option (long)	Argument	Description
	--file	PATH	The name and location of the exported site file you are importing. If you don't specify a path, Tableau Server's bin directory is the assumed location (Program Files\Tableau\Tableau Server\9.2\bin).
	--no-verify		<p>Skips the verification step and imports the exported site file directly to its new location in your Tableau Server installation. If you choose this option, you not need to use the <code>importsite_verified</code> command.</p> <p>Note: Importing a site without verifying the mappings is not recommended.</p>
	--tempdir	PATH	The directory where you will verify that the site files have the correct mappings. If you don't specify this option, files are placed in a directory under ProgramData\Tableau\Tableau Server\data\tabsvc\temp.

See Also

[Import or Export a Site on page 127](#)

[importsite_verified](#)

Performs the second part of an import process for a site on Tableau Server. Before you can use `importsite_verified`, you must first use [importsite](#) on page 600.

The `importsite_verified` command reads from a directory containing CSV files that you have verified, and imports a new site into Tableau Server based on how the site's assets are

mapped in the CSV files. The site that receives the import (the target site) must already exist on Tableau Server.

During the import process, Tableau Server locks the site receiving the import.

Examples

```
tabadmin importsite_verified <target site ID> --importjobdir  
<PATH>
```

Import files from the directory C:\temp\site_to_import to the site whose site ID is **esale**:

Option (short)	Option (long)	Argument	Description
	--import-jobdir	PATH	The directory containing CSV files whose mappings you have verified.

See Also

[Import or Export a Site on page 127](#)

[import_webdataconnector](#)

Installs a web data connector on the server. Users who create workbooks can then reference the web data connector as a data source.

Important: Before you import a web data connector, make sure that the JavaScript code in the connector does not implement any functionality that should not be on your server.

When the `import_webdataconnector` command finishes importing the connector, the command displays the server URL of the connector. When users want to reference the web data connector as a data source, they need to know this URL. (You can also view the URLs of connectors on your server by using the [list_webdataconnectors on page 605](#) command.)

If the web data connector includes references to an external file, such as to a .css file or .js file, you must make sure that the external file is available from the server, either over the web or as a local file. If the connector references a local file, the local file must be in the same folder as the connector's .html file relative paths to subdirectories are not supported for imported web connectors. (Make sure that the `<link>` or `<script>` element in the connector correctly references the file as a peer of the connector file.) If the external file is local, you must use the `import_webdataconnector` command to import the external file separately.

If the server includes multiple computers in a cluster, the web data connector is imported to each computer where a gateway process is running.

Examples

```
tabadmin import_webdataconnector connector1.html  
tabadmin import_webdataconnector  
c:\webdataconnectors\connector1.html --overwrite  
tabadmin import_webdataconnector  
\myshare\webdataconnectors\connector2.html --overwrite  
tabadmin import_webdataconnector connector1.css
```

Note: The connector name can contain only these characters: a-zA-Z0-9 ()~.-_.

Option (short)	Option (long)	Argument	Description
	--over-write		Overwrites any existing file on the server that has the same name as the file that you are importing.

See Also

[delete_webdataconnector](#) on page 596

[list_webdataconnectors](#) on the next page

[Web Data Connectors in Tableau Server](#) on page 276

licenses

Displays license information for Tableau Server.

Examples

```
tabadmin licenses  
tabadmin licenses -p
```

Option (short)	Option (long)	Argument	Description
-p	--processor_cores		Display the physical core count for the current machine.

list_webdataconnectors

Displays the names or URLs of web data connectors that are installed on the server.

Examples

List the names of the web data connectors.

```
tabadmin list_webdataconnectors
```

List the URLs of the web data connectors.

```
tabadmin list_webdataconnectors --urls
```

Option (short)	Option (long)	Argument	Description
	--urls		Specifies that the command should list URLs instead of names.

See Also

[import_webdataconnector](#) on page 603

[delete_webdataconnector](#) on page 596

[Web Data Connectors in Tableau Server](#) on page 276

manage_global_credentials

Manages credentials for delegated data access on Tableau Server. Use this command to specify the credentials for a proxy user that is used to access a data source that does not support single-sign on via Kerberos.

Examples

```
tabadmin manage_global_credentials --add --server <server> --user <username> --password <password>
```

Add credentials for a server named my-server.

```
tabadmin manage_global_credentials --add --server my-server --
```

```
user jsmith --password p@ssword
```

Option (short)	Option (long)	Argument	Description
	--add		Add credentials for the specified server.
	--remove		Remove credentials
	--show		Show current credentials
-s	--server	server	Server for which credentials are being managed
-u	--user-name	user	User name for connecting to a server
-p	--pass-word	password	Password for connecting to a server
-o	--over-ride		Override existing credentials

See Also

[Enabling Delegation for Cloudera Impala](#) in the Tableau Knowledge Base.

passwd

Resets the password for a Tableau Server account. After typing the command, you are prompted to enter a new password for the user.

You can only use this command if Tableau Server's user authentication is set to Local Authentication. When authentication is set to Active Directory, passwords are handled by Active Directory, not Tableau Server.

Examples

```
tabadmin passwd <username>
```

Reset the password for server user **jsmith**:

```
tabadmin passwd jsmith
```

See Also

[Configure General Server Options](#) on page 12

recommission

Reverts a decommissioned file store node in read-only mode to an active read/write state. Use spaces to separate multiple nodes.

Examples

```
tabadmin recommission <computer name(s) or IPv4 address(es)>
```

Recommission file store node by IP address:

```
tabadmin recommission 10.32.139.29
```

See Also

[Distributed Environments on page 73](#)

[Maintain a Distributed Environment on page 82](#)

regenerate_internal_tokens

Creates new security tokens that Tableau Server uses internally. These tokens include the passwords used by Tableau Server to access the repository, and the certificates used to validate internal SSL connections between Tableau Server components and the repository.

Running this command stops Tableau Server, so you will need to restart Tableau Server after you run the command.

Example

```
tabadmin regenerate_internal_tokens --passwords
```

```
tabadmin regenerate_internal_tokens --certs
```

Option (short)	Option (long)	Argument	Description
	--certs	None	Regenerates key pair for internal SSL connections.
	--pass-words	None	Regenerates passwords for the Postgres database.
	None	None	Regenerates key pair for internal SSL connections

Option (short)	Option (long)	Argument	Description
			and passwords for Postgres database. Note: The key pair is regenerated only if internal SSL is configured .
	--restart	None	Restart Tableau Server after regenerating tokens.

See Also

[Regenerate a Password for the Tableau Server PostgreSQL Database \(Repository\) on page 426](#)

[Security on page 414](#)

reindex

Rebuilds the search index for Tableau Server. In rare instances, you may need to rebuild the index if searches on the server return incomplete or incorrect results, or if the Search & Browse process is down for an extended period. You can use this command if users cannot sign in to the server because no sites are listed after they enter their credentials.

Note: The recommended way to reindex Search is to run this command while Tableau Server is stopped. Reindexing while the server is running can result in content, including sites and projects, temporarily disappearing.

Examples

```
tabadmin reindex
```

Reindex the server

See Also

[Rebuild the Search Index on page 254](#)

reset_openid_sub

Clears the user identifier (`sub` value) that binds a user identity in Tableau Server to a specific OpenID Connect identity provider (IdP).

If Tableau Server is configured to use OpenID Connect for authentication, the first time a user signs in to Tableau Server using the IdP, Tableau stores the `sub` value sent by the IdP with the user information in Tableau Server. The `sub` provides a unique identity for that user with the IdP. If you change IdPs for OpenID Connect, you must remove the `sub` value for the user. That way, when the user signs in using the new IdP, Tableau can store a new `sub` value.

Example

```
tabadmin reset_openid_sub Alice
```

This command clears the `sub` value for the user named Alice.

```
tabadmin reset_openid_sub all
```

This command clears the `sub` value for all users on the server.

See Also

[OpenID Connect on page 517](#)

[Changing IdPs in Tableau Server for OpenID Connect on page 524](#)

restart

Stops and starts all Tableau Server processes.

Example

```
tabadmin restart
```

restore

Restores a Tableau Server backup file (.tsbak) to a Tableau Server installation. When you restore a .tsbak file, the contents of the Tableau PostgreSQL database, data extracts, and configuration files are overwritten with the content in the backup file. If the backup was made after [web data connectors](#) were imported to the server using the [`import_webdataconnector` on page 603](#) command, the restore process restores the connectors as well. Using the `--no-config` option restores everything but the server's configuration.

Examples

Restore a file named **tabserv.tsbak** located in C:\mybackups and then restart the server:

```
tabadmin restore C:\mybackups\tbserv.tsbak --restart
```

Restore a file named **tabserv.tsbak** located in the Tableau Server bin directory and then restart the server:

```
tabadmin restore tabserv.tsbak --restart
```

Restore a file named **tabserv.tsbak** located in C:\mybackups, retaining everything but the server's configuration, but don't restart the server:

```
tabadmin restore --no-config C:\mybackups\tbserv.tsbak
```

Option (short)	Option (long)	Argument	Description
	--no-con-fig		Restore the Tableau Server backup file including the data but excluding the server's configuration.
	--par-allel-pg-restore		Run the restore process for the PostgreSQL repository as a parallel job.
	--restart		Restart the service when the restore process has completed.

See Also

[Restore from a Backup on page 632](#)

[Recover Extracts from a Backup on page 633](#)

set

Allows you to change the value of [Tableau Server configuration options](#). If the parameter you're setting begins with a hyphen, enclose the parameter's value in both double- and single-quotes.

Examples

```
tabadmin set [option-name value]
```

Set the backgrounder query limit to 2.5 hours (9000 seconds):

```
tabadmin set backgrounder.querylimit 9000
```

Set the wgserver virtual memory parameter to -Xmx512m:

```
tabadmin set wgserver.vmopts "'-Xmx512m'"
```

Set the wgserver virtual memory parameter to a range of -Xmx512m -Xss2048k:

```
tabadmin set wgserver.vmopts "'-Xmx512m -Xss2048k'"
```

Option (short)	Option (long)	Argument	Description
-d	--default		Reset the parameter to its default value.

See Also

[tabadmin set options on page 616](#)

sitestate

Activates (unlocks) or suspends a site. You can use this command to activate a site that was locked because of a site import failure. When a site is suspended, the only Tableau Server user who can access it is the system administrator.

Note: To specify the default site, use "" for the site ID.

Examples

```
tabadmin sitestate <site ID> --status <active|suspended>
```

Activate a site whose site ID is **wsales**:

```
tabadmin sitestate wsales --status active
```

Activate the Default site. The site ID for the Default site is "" (double quotes, no space).

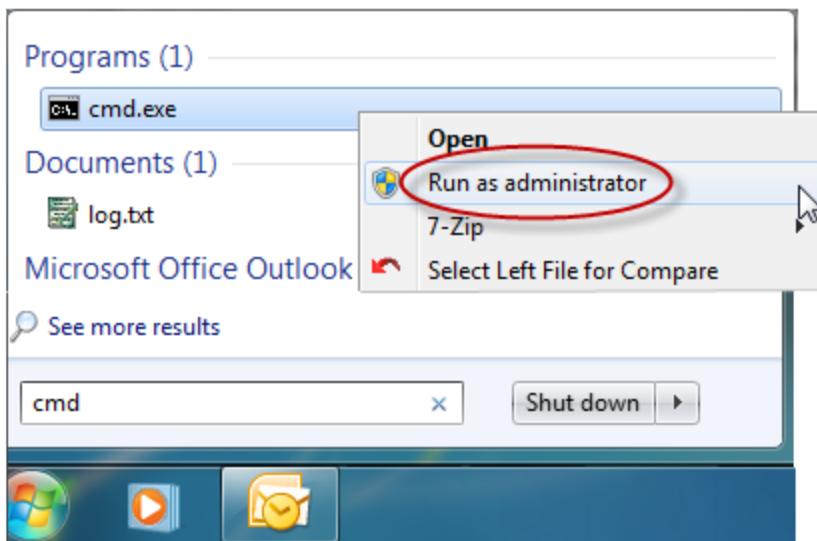
```
tabadmin sitestate "" --status active
```

Option (short)	Option (long)	Argument	Description
	--status	active or suspended	Specifies whether to activate or suspend the specified site.

start

Starts all Tableau Server processes. To use tabadmin start:

1. Open a command prompt as an administrator:



2. Type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

3. Type the following to start the server:

```
tabadmin start
```

Examples

```
tabadmin start
```

```
tabadmin start --wait 1200
```

Option (short)	Option (long)	Argument	Description
	--wait	number of seconds	Number of seconds after starting after which Tableau Server is ready to accept client requests. The default is 600 seconds.

status

Tells you whether or not Tableau Server is running and, if you use the `--verbose` option, gives you details on individual server process status, including whether a process is running and its process ID. The `tabadmin status` command obtains its information by connecting to the Windows Service tabsvc.exe, which in turn queries the tabspawn executables for each process. Because of this, it can sometimes display different information for the server processes than the status table on the [Maintenance page](#), which queries the processes directly.

Examples

```
tabadmin status
```

```
tabadmin status --verbose
```

Option (short)	Option (long)	Argument	Description
<code>-v</code>	<code>--verb-</code> <code>ose</code>		Returns a list of all the Tableau Server processes, their process IDs, and their status.

See Also

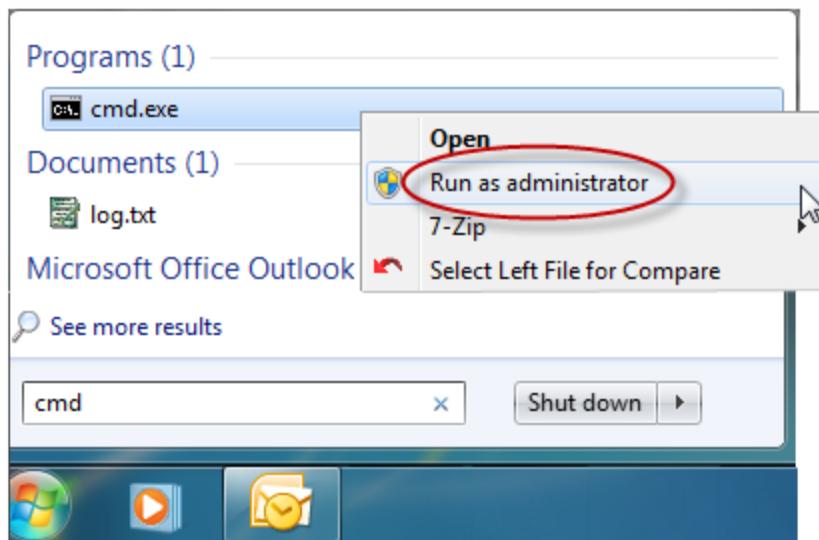
[Server Settings \(General\) on page 257](#)

[Tableau Server Processes on page 637](#)

stop

Stops all Tableau Server processes. To use `tabadmin stop`:

1. Open a command prompt as an administrator:



2. Type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

3. Type the following to stop the server:

```
tabadmin stop
```

validate

Confirms whether your Tableau Server environment meets the minimum requirements for running the 32-bit version of Tableau Server. Running with the -x option validates 64-bit requirements. If you are currently running the 32-bit version of Tableau Server, running this command before you upgrade can help you confirm whether your current hardware (cores), disk space, and RAM are sufficient for the 64-bit version.

Example

```
tabadmin validate
```

Option (short)	Option (long)	Argument	Description
-x	--x64		Validate current machine cores, memory and available disk space against 64-bit Tableau

Option (short)	Option (long)	Argument	Description
			Server requirements.
	--skiptempIPv6		Skip validating that temporary IPv6 addresses are disabled.

warmup

Causes every VizQL server process to load the vizql DLL file, resulting in faster load times when server users first load views. Administrators can run this command, or script it to be run, after a Tableau Server restart.

Example

```
tabadmin warmup
```

ziplogs

Creates an archive (.zip) containing Tableau Server log files, without removing the log files themselves. If you are running a Tableau Server cluster, log files from worker servers are included in the archive that's created.

Examples

Create an archive in the Tableau Server bin directory named **logs.zip**:

```
tabadmin ziplogs
```

Create an archive in the Tableau Server bin directory named **mylogs.zip**:

```
tabadmin ziplogs mylogs.zip
```

Create an archive in the Tableau Server bin directory named **mylogs.zip** that includes logs dated January 31, 2014 up to the present, excluding earlier logs:

```
tabadmin ziplogs -d 01/31/2014 mylogs.zip
```

Option (short)	Option (long)	Argument	Description
-n	--with-net-stat-info		Include information about the server environment in the .zip file.
-p	--with-postgresql-data		Include data from Tableau Server's PostgreSQL database. If Tableau Server is stopped, make a copy of the pgsql\data folder. If Tableau Server is running, get the data as binary dump files.
-l	--with-latest-dump		Limit the included log files to only the most recent ones to help reduce file size. By default, the 10 most recent log files are included.
-f	--force		Overwrites the existing log file of the same name.
-d	--min-imumdate	[mm/dd/yyyy]	Log files with this date, up to the present, are included in the .zip file. Logs dated earlier are excluded from the file. If not specified, up to seven days worth of data is included.
-a	--all		Include all log files in the .zip file. Data from Tableau Server's PostgreSQL database is still excluded.

See Also

[Work with Log Files on page 636](#)

[Archive Logs on Command Line \(tabadmin\) on page 644](#)

tabadmin set options

Use the table below to learn more about Tableau Server options you can configure using the [set on page 610](#) command. See [Tableau Server Ports on page 540](#) for a complete list of ports.

Option	Default Value	Description
api.server.enabled	true	Allows access to the REST API on page 671 . By default, this functionality is enabled.
auditing.enabled	true	Allows access to the PostgreSQL (Tableau Server's own database) historical auditing

Option	Default Value	Description
		tables. See Create Custom Administrative Views on page 300 for details.
backgrounder.extra_timeout_in_seconds	1800	The number of seconds beyond the setting in <code>backgrounder.querylimit</code> before a background task is canceled. This setting makes sure that tasks do not hold up subsequent jobs if they are stalled. The setting applies to processes listed in <code>backgrounder.timeout_tasks</code> . To disable backgrounder timeouts, set the value of <code>backgrounder.extra_timeout_in_seconds</code> to "" (an empty string).
backgrounder.querylimit	7200	Longest allowable time for completing a single extract refresh task, in seconds (7200 seconds = 2 hours).
backgrounder.reset_schedules_on_startup	true	Controls when to run background tasks that were scheduled to run at a time when the server was stopped. When set to <code>true</code> (the default), tasks are run at their next scheduled time. When set to <code>false</code> , all tasks that were scheduled to run when the server was stopped are run, simultaneously, at server startup, including times when the Tableau Server backup file (.tsbak) is restored.
backgrounder.timeout_tasks	refresh_extracts, increment_extracts, subscription_notify, single_subscription_notify	The list of tasks that can be canceled if they run longer than the combined values in <code>backgrounder.querylimit</code> and <code>backgrounder.extra_timeout_in_seconds</code> . The list of tasks is delimited with commas. The default list represents all the possible values for this setting.

Option	Default Value	Description
clustercontroller.zk_session_timeout_ms	300000	The length of time, in milliseconds, that Cluster Controller will wait for the Coordination Service (ZooKeeper), before determining that failover is required.
dataengine.port	27042	Port that the data engine runs on.
dataserver.port	9700	Port that the data server runs on.
gateway.public.host	Name of the machine	The name (URL) of the server, used for external access to Tableau Server. If Tableau Server is configured to work with a proxy server or external load balancer, it is the name entered in a browser address bar to reach Tableau Server. For example, if Tableau Server is reached by entering <code>tableau-example.com</code> , the name for <code>gateway.public.host</code> is <code>tableau.example.com</code> .
gateway.public.port	80 (443 if SSL)	Applies to proxy server environments only. The external port the proxy server listens on.
gateway.timeout	1800	Longest amount of time, in seconds, that the gateway will wait for certain events before failing a request (1800 seconds = 30 minutes).
gateway.trusted	IP address of proxy server machine	Applies to proxy server environments only. The IP address(es) or host name(s) of the proxy server.
gateway.trusted_hosts	Alternate name(s) of proxy server	Applies to proxy server environments only. Any alternate host name(s) for the proxy server.
install.firewall.allowedprograms.manage	true	Controls whether Tableau Server can modify firewall rules. When set to <code>true</code> (the default), Tableau Server can change firewall rules. Change this to <code>false</code> if you have modified firewall rules and do not want them changed.
java.heap.size	128m	Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.
mon-	30000	The length of time, in milliseconds, that Cluster

Option	Default Value	Description
itor-ing.dataengine.connection_timeout		Controller will wait for the data engine, before determining that a connection timeout occurred. The default is 30,000 milliseconds (30 seconds).
native_api.-connection.limit.<connection class>		Set parallel query limit for the specified data source (connection class). This overrides the global limit for the data source. For information about specific connection class strings, see the Tableau Knowledge Base .
native_api.-connection.limit.globallimit	16	Global limit for parallel queries. Default is 16 except for Amazon Redshift which has a default of 2. For information about configuring parallel queries in Tableau Server, see the Tableau Knowledge Base .
pgsql.port	8060	Port that PostgreSQL listens on.
rsync.timeout	600	Longest allowable time, in seconds, for completing file synchronization (600 seconds = 10 minutes). File synchronization occurs as part of configuring high availability , or moving the data engine and repository processes.
server.log.level	info	<p>The logging level for logs written to ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\Logs*.txt</p> <p>Set to <code>debug</code> for more information. When set to <code>debug</code>, logging is set to pre-8.2 verbosity. Using the <code>debug</code> setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 652 for more information.</p>
service.jmx_enabled	false	Setting to <code>true</code> enables JMX ports for optional monitoring and troubleshooting. See Enable the JMX Ports on page 549 for details.
service.max_procs	# of processes	Maximum number of server processes .

Option	Default Value	Description
service.port_remapping.enabled	true	<p>Determines whether or not Tableau Server will attempt to dynamically remap ports when the default or configured ports are unavailable. Setting to <code>false</code> disables dynamic port remapping. See Tableau Server Ports on page 540 for more information.</p>
session.ipsticky	false	<p>Makes client sessions valid only for the IP address that was used to sign in. If a request is made from an IP address different from that associated with the session token, the session token is considered invalid.</p> <p>In certain circumstances—for example, when Tableau Server is being accessed by computers with known and static IP addresses—this setting can yield improved security.</p>
		<p>Note: Consider carefully whether this setting will help your server security. This setting requires that the client have a unique IP address and an IP address that stays the same for the duration of the session. For example, different users who are behind a proxy might look like they have the same IP address (namely, the IP address of the proxy); in that case, one user might have access to another user's session. In other circumstances, users might have a dynamic IP address, and their address might change during the course of the session. If so, the user has to sign in again.</p>
solr.rebuild_index_timeout	3600	<p>When Tableau Server is upgraded or when a <code>.tsbak</code> file is restored, the background task rebuilds the search index. This setting controls the timeout setting for that task (3600 seconds = 60 minutes).</p>

Option	Default Value	Description
ssl.client_certificate_login.-fallback_to_password	false	<p>Specifies if Tableau Server should use user name and password for authentication if SSL authentication fails.</p> <p>Valid options are <code>false</code> (the default) and <code>true</code>.</p> <p>By default, when configured for mutual SSL, Tableau Server does not allow a connection if SSL authentication fails. Set this to <code>true</code> to allow user name and password authentication if SSL authentication fails.</p>
ssl.client_certificate_login.mapping_strategy	UPN or LDAP	<p>Specifies the method to be used for retrieving the user name from the certificate. Options are LDAP, UPN, or CN.</p> <p>The default depends on how Tableau Server is configured for user authentication:</p> <ul style="list-style-type: none"> When Tableau Server authentication is configured for Local Authentication, the default is UPN (User Principal Name). When Tableau Server authentication is configured for Active Directory (AD), the default is LDAP (Lightweight Directory Access Protocol). <p>CN (Common Name) is an option the administrator can set for either authentication type.</p>
ssl.revocation.file		<p>Specifies the file path for an SSL CA Certificate Revocation List (CRL) file.</p> <p>Example: <code>tabadmin set ssl.revocation.file "c:\Program Files\Tableau\Tableau Server\SSL\ca-bundle-client.crl</code></p>
subscriptions.enabled	false	<p>Controls whether subscriptions are configurable system-wide. See Manage Subscriptions on page 231.</p>
subscriptions.timeout	1800	<p>Number of seconds after which the background</p>

Option	Default Value	Description
		process handling a subscription times out.
tomcat.https.port	8443	SSL port for Tomcat (unused).
tomcat.server.port	8085	Port that tomcat listens on for shutdown messages.
vizportal.adsync.update_system_user	false	Specifies whether email addresses and display names of users are changed (even when changed in Active Directory) when an Active Directory group is synchronized in Tableau Server. To ensure that user email addresses and display names are updated during synchronization, set <code>vizportal.adsync.update_system_user</code> to <code>true</code> , and then restart the server.
vizportal.log.level	info	<p>The logging level for vizportal Java components. Logs are written to <code>ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizportal*.log</code>. Set to <code>debug</code> for more information. Using the <code>debug</code> setting can significantly impact performance, so you should only use this setting when directed to do so by Tableau Support. See Change Logging Levels on page 652 for more information.</p>
vizqlserver.allow_insecure_scripts	false	Allows a workbook to be published to the server from Tableau Desktop, and to be opened from the server, even if the workbook contains SQL or R expressions that are potentially unsafe (for example, a SQL expression that could potentially allow SQL injection). When this setting is <code>false</code> (the default), publishing a workbook or opening it from the server results in an error message, and the workbook is blocked. You should set this value to <code>true</code> only if you want to use workbooks that contain SQL or R expressions that have been detected as potentially

Option	Default Value	Description
		unsafe, and only if the workbooks come from a safe source and you have verified that they do not contain an unsafe expression.
vizqlserver.browser.render	true	Views under the threshold set by <code>vizqlserver.browser.render_threshold</code> or <code>vizqlserver.browser.render_threshold_mobile</code> are rendered by the client web browser instead of by the server. See About Client-Side Rendering on page 437 for details.
vizqlserver.browser.render_threshold	100	The default value (100) represents a high level of complexity for a view displayed on a PC. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the PC's web browser.
vizqlserver.browser.render_threshold_mobile	20	The default value (20) represents a high level of complexity for a view displayed on a tablet. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the tablet's web browser.
vizqlserver.clear_session_on_unload	false	Determines whether or not VizQL sessions are kept in memory when a user navigates away from a view or closes their browser. The default value (false) keeps sessions in memory. To close VizQL sessions on leaving a view or closing a browser, set this to <code>true</code> . See General Performance Guidelines on page 428 for more information.
vizqlserver.geosearch_cache_size	5	Sets the maximum number of different geographic search locale/language data sets that can be loaded into server memory at the same time. When the server receives a geographic search request for locale/language data set

Option	Default Value	Description
		that is not in memory, it will load the set into memory. If loading the data set will exceed the specified limit, the least recently used locale/language data set is cleared from memory so the requested one can be loaded. The minimum value is 1. Each cache takes approximately 60 MB in memory (so if you set this to 10, the memory usage would be 600 MB (60 * 10)).
vizqlserver.log.level	info	The logging level for vizqlserver Java components. Logs are written to ProgramData\Tableau\Tableau Server\data\t-absvc\logs\vizqlserver*.log. Set to debug for more information. Using the debug setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 652 for more information.
vizqlserver.port	9100	Base port for the VizQL servers.
vizqlserver.protect_sessions	true	When set to true (the default), prevents VizQL sessions from being reused after the original user signs out.
vizqlserver.querylimit	1800	Longest allowable time for updating a view, in seconds.
vizqlserver.rserve.host		Specifies an Rserve host. This setting, and the three settings immediately below, supports R functionality in workbooks.R is an open source software programming language and a software environment for statistical computing and graphics. In Tableau Desktop, you can use a set of four functions to pass R expressions to an Rserve server and obtain a result. If you upload a workbook that uses any of these functions, you should configure Tableau Server for an

Option	Default Value	Description
		Rserve connection, by configuring this option and the three following. Otherwise, any worksheets that use R functionality will be unavailable. See R Connection in the Tableau Desktop help for further details.
vizqlserver.rserve.port	6311	Specifies an Rserve port. This setting supports R functionality in workbooks.
vizqlserver.rserve.username		Specifies an Rserve username. This setting supports R functionality in workbooks. Not all Rserve hosts require a username and password.
vizqlserver.rserve.password		Specifies an Rserve password. This setting supports R functionality in workbooks. Not all Rserve hosts require a username and password.
vizqlserver.session.expiry.minimum	5	Number of minutes of idle time after which a VizQL session is eligible to be discarded if the VizQL process starts to run out of memory.
vizqlserver.session.expiry.timeout	30	Number of minutes of idle time after which a VizQL session is discarded.
vizqlserver.showdownload	true	Controls the display of the Download button in views.
vizqlserver.showshare	true	Controls the display of the Share button in views.
vizqlserver.trustedticket.log_level	info	<p>The logging level for trusted authentication. The logs are written to <code>ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\vizql*.log</code>.</p> <p>Set to <code>debug</code> for more information. Using the debug level can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 652 for more information.</p>

Option	Default Value	Description
vizqlserver.trustedticket.token_length	24	Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.
vizqlserver.trustedticket.use_DEPRECATED_9digit_token	false	When set to <code>true</code> , tickets are 9 digits long (as in version 8.0 and earlier) and the setting <code>vizqlserver.trustedticket.token_length</code> is ignored.
vizqlserver.url_scheme_whitelist		Adds to the protocols to whitelist when using URL actions on views and dashboards. <code>http</code> , <code>https</code> , <code>gopher</code> , <code>news</code> , <code>ftp</code> , and <code>mailto</code> are whitelisted by default.
webdataconnector.enabled	true	When this setting is <code>true</code> , you can use <code>tabadmin</code> commands to manage web data connectors on the server, and web data connectors are included when you back up and restore the server. If the setting is <code>false</code> , web data connectors that are on the server are not included during backup and restore. For more information, see Web Data Connectors in Tableau Server on page 276.
web-data-connector.refresh.enabled	true	When this setting is <code>true</code> , the server supports doing refreshes for web data connector-based data sources. For more information, see Web Data Connectors in Tableau Server on page 276.
wgserver.audit_history_expiration_days	183	Number of days after which historical events records are removed from the PostgreSQL database (the Tableau Server database). See Create Custom Administrative Views on page 300 for details.
wgserv-server.authentication.desktop_nosaml	false	Controls whether or not Tableau Desktop uses SAML for authentication. Use this option when your IdP does not use forms-based authentication. Valid options are <code>true</code> and <code>false</code> . By default this is not set, so the behavior is equi-

Option	Default Value	Description
		valent to setting it to <code>false</code> . Set this to <code>true</code> to disable SAML authentication for Tableau Desktop.
<code>wgserv-er.authentication.app_nosaml</code>	<code>false</code>	Serves as the above setting for the Tableau Mobile app.
<code>wgserv-er.authentication.restricted</code>	<code>false</code>	Controls whether users can sign in to Tableau Server using a Tableau Server username and password. This setting is useful in scenarios where users normally sign in to the server using single sign-on (SSO), such as by using SAML, OpenID Connect, or Kerberos. In these cases, the user also has a Tableau Server username and password. If <code>wgserv-er.authentication.restricted</code> is set to <code>true</code> , only system administrators can sign in to Tableau Server using a username and password; all other users <i>must</i> sign in to the server using SSO. Setting <code>wgserv-er.authentication.restricted</code> to <code>true</code> also has the effect of restricting user access to command-line tools like <code>tabcmd</code> and <code>tabconfig</code> . These tools do not support SSO, and therefore require a user to sign in using a Tableau Server. If the setting is <code>true</code> , users who are not system administrator cannot use these command-line tools.
<code>wgserver.change_owner.enabled</code>	<code>true</code>	Controls whether the ownership of a workbook, data source or project can be changed. Other options include <code>false</code> and <code>adminonly</code> . See Manage Ownership on page 331 for details.
<code>wgserver.clickjack_defense.enabled</code>	<code>true</code>	When set to <code>true</code> , helps prevents a malicious person from "clickjacking" a Tableau Server user. In a clickjack attack, the target page is displayed transparently over a second page, and the attacker gets the user to click or enter information in the target page while the user thinks he or she is interacting with the second

Option	Default Value	Description
		<p>page.</p> <p>For more information, see Clickjack Protection on page 422.</p>
wgserver.domain.fqdn	value of %USERDOMAIN%	The fully qualified domain name of the Active Directory server to use.
wgserver.log.level	info	<p>The logging level for wgserver Java components. Logs are written to ProgramData\Tableau\Tableau Server\data\tabsvc\logs\wgserver*.log.</p> <p>Set to <code>debug</code> for more information. Using the <code>debug</code> setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 652 for more information.</p>
wgserver.password_auto-complete.enabled	false	Controls whether web browsers are allowed to automatically complete password fields.
wgserver.restrict_options_method	true	Controls whether Tableau Server accepts HTTP OPTIONS requests. If this option is set to <code>true</code> , the server returns HTTP 405 (Method Not Allowed) for HTTP OPTIONS requests.
wgserver.saml.idpattribute.username	username	Specifies the attribute used by the IdP for SAML authentication. The default is <code>username</code> . For more information, see SAML on page 472 .
wgserver.saml.logout.enabled	true	Specifies whether SAML logout is enabled for Tableau Server. The default is <code>true</code> . This setting only applies if SAML authentication is enabled for Tableau Server.
wgserver.saml.logout.redirect_url		Specifies the post-logout landing page for SAML authentication. The default is the standard server sign-in page. You can specify an absolute or a relative URL. For more information, see SAML Requirements .

Option	Default Value	Description
wgserver.saml.maxassertiontime	3000	Specifies the maximum number of seconds, from creation, that an assertion is usable.
wgserver.saml.maxauthenticationage	7200	Specifies the maximum number of seconds allowed between user's authentication and processing of the AuthNResponse message.
wgserver.saml.responseskew	180	Sets the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed.
wgserver.session.apply_lifetime_limit	false	Controls whether there is a session lifetime for server sessions. Set this to <code>true</code> to configure a server session lifetime.
wgserver.session.lifetime_limit	1440	The number of minutes a server session lasts if a session lifetime is set. The default is 1440 minutes (24 hours). If <code>wgserver.session.apply_lifetime_limit</code> is <code>false</code> (the default) this is ignored.
wgserver.session.idle_limit	240	The number of minutes of idle time before a sign-in to the web application times out.
workerX.gateway.port	80 (443 if SSL)	External port that Apache listens on for workerX. worker0.gateway.port is Tableau Server's external port. In a distributed environment, worker0 is the primary Tableau Server.
workerX.vizqlserver.procs	# of processes	Number of VizQL servers.
workerX.vizqlserver.port	9100	Base port for the vizQL server on workerX.
workerX.wgserver.port	8000	Base port for the web application server on workerX.
workerX.wgserver.procs	# of processors	Number of web application server processes.
zoo-keeper.config.dataLogDir		Specifies the directory and file path for ZooKeeper transaction logs. By default ZooKeeper transaction logs are written to the Tableau data directory (for example <code>c:\T-</code>)

Option	Default Value	Description
		<p>ableau\Tableau Server-\data\tabsvc\zookeeper\0\data). Use this option to specify a different location.</p> <p>The drive and path apply to all nodes in a cluster. The location will be created if it does not exist. The drive must exist and be writable on all nodes. This should not be a UNC path to a share.</p> <p>ZooKeeper recommends that transaction logs be written to a dedicated drive to optimize performance.</p> <p>Example: tabadmin set zookeeper.config.dataLogDir "d:\Tableau\Tableau Server\zookeeper"</p>

Restore a Setting to its Default Value

You can restore the default value for a Tableau Server configuration setting by doing the following:

1. **Stop the server.**
2. Still in the bin directory, restore the default value for a particular setting by typing the following:

```
tabadmin set option-name --default
```

For example, to set the tabadmin **vizqlserver.session.expiry.timeout** option back to its default value of 30 minutes, you would type the following:

```
tabadmin set vizqlserver.session.expiry.timeout --default
```

Alternatively, you can use the shorter **-d** command. For example:

```
tabadmin set vizqlserver.querylimit -d
```

3. Next, run the configure command:

```
tabadmin configure
```

4. **Start the server.**

Database Maintenance

A Tableau Server administrator should perform regular database maintenance, monitor disk usage on the server, and clean up unnecessary files to free up space on the server. Taking these steps can help ensure that Tableau Server runs with maximum efficiency.

You can use the tabadmin command line tool to back up and restore your Tableau data, and to clean up (remove) unnecessary log and temporary files. Tableau data includes Tableau Server's own PostgreSQL database, which stores workbook and user metadata, data extract (.tde) files, and server configuration data. Tableau Server log files capture activity and can help you diagnose problems. Logs are written to folders on the server and you can archive and remove them to save disk space. Use the commands described in the topics below, along with the built-in Windows task scheduler to automate backing up data and cleaning up unnecessary files.

Note: You can only use backups made with the `tabadmin backup` command when restoring Tableau Server data. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

Back Up the Tableau Data

It is important to back up your Tableau data so you can restore published views and other information in the case of a system failure. The data managed by Tableau Server consists of Tableau's own PostgreSQL database, which contains workbook and user metadata, data extract (.tde) files, and configuration data. When you use tabadmin to create a backup, all these things are placed in a single file with a .tsbak extension. If you are running a [distributed installation](#) of Tableau Server this step is performed on the primary, even if the data engine, which handles the .tde files, is on a worker. Only backups created using tabadmin can be used when restoring Tableau data.

Store the .tsbak on a computer that is not a part of your Tableau Server installation.

Tableau Uninstall, which is the first step to upgrading to a new version, also automatically creates a .tsbak file. This same .tsbak file is used to automatically migrate your data to your newer version.

Create a backup of your Tableau data using the procedure below.

Note: Running the `backup` command also removes Tableau Server log files older than seven days as well as some of the information displayed in certain Tableau Server [Administrative Views](#) on page 289.

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```
2. Create a backup file by typing `tabadmin backup <filename>`, where `<filename>` is the name or location and name of your backup file. Starting with version 8.1, there is no need to stop the server before you create the backup. For example:

```
tabadmin backup tabserver
```

or

```
tabadmin backup C:\backups\tableau\tabserver
```

You can also optionally use `-d` to append the current date to the file name.

Add `-t` followed by a path, to specify a location for temporary files that are created during the backup process. The path for the temporary files is not the location where the backup file will be written. For example:

```
tabadmin backup tabserver -t C:\mytemp\tableau
```

In the above example, the backup file `tabserver.tsbak` will be created in the Tableau Server bin directory (`C:\Program Files\Tableau\Tableau Server\9.2\bin`) not in `C:\mytemp\tableau`.

Restore from a Backup

When you use `tabadmin` to restore your Tableau data, the contents of the Tableau PostgreSQL database, data extracts, and configuration files are overwritten with the content in the backup file (.tsbak). If you are running a [distributed installation](#) of Tableau Server, this step is performed on the primary.

Note: Only backups created using `tabadmin` can be used when restoring Tableau data.

To restore from a database backup file:

1. Stop the server by typing:

```
tabadmin stop
```

2. Restore the database from a backup file by typing:

```
tabadmin restore <filename>
```

In the above line, replace `<filename>` with the name of the backup file you want to restore from.

To restore only the data and no configuration settings, type the following instead:

```
tabadmin restore --no-config <filename>
```

3. Restart the server by typing:

```
tabadmin start
```

4. If you ran the `tabadmin assetkeys` command at any time before you created the backup file that you're now restoring, run the following command:

```
tabadmin assetkeys --validate
```

You'll be prompted to enter the passphrase needed to re-create the custom encryption keys in use in the backup file.

When you restore a .tsbak file, Tableau Server automatically creates a copy of its current data folder, names it `tabsvc.bak-*`, and places it in `ProgramData\Tableau\Tableau Server\data`. This folder is an emergency backup of Tableau Server data which Tableau Support may be able to use in case something goes wrong during backup restoration.

Once a restoration is complete, it's safe to remove any `tabsvc.bak-*` folders from `ProgramData\Tableau\Tableau Server\data` to free additional disk space. In Tableau Server clusters, `tabsvc.bak-*` folders are created on each machine running Tableau Server.

Note: Do not remove the `tabsvc` folder, which is also located under `ProgramData\Tableau\Tableau Server\data`. It contains Tableau Server data. Remove only the `tabsvc.bak-*` folders.

Recover Extracts from a Backup

The file `uninstall-<version>.tsbak` (for example, `uninstall-9.1.tsbak`) is created as part of the uninstall process. After you upgrade to version 9.2, you can use this file to restore data extracts—for example, if you mistakenly deleted the `dataengine` folder during the upgrade. To use `uninstall-<version>.tsbak` to restore data extracts:

1. **Stop the server.**
2. From within your version 9.2 Tableau Server bin directory, type the following:

Windows Server 2012, Windows Server 2008, Windows 7, Windows 8: `tabadmin restore \ProgramData\Tableau\Tableau Server\uninstall-9.1.tsbak`

32-bit Tableau Server installed on 64-bit Windows Server: `tabadmin restore \Program Files (x86)\Tableau\Tableau Server\uninstall-9.1.tsbak`

```
32-bit Tableau Server installed on 32-bit Windows Server: tabadmin restore  
\Program Files\Tableau\Tableau Server\uninstall-9.1.tsbak
```

Remove Unneeded Files

As a best practice, you should monitor space usage on your server. If you need to make more space available, you can use the [cleanup](#) on page 590 command to remove log files, temporary files, and unneeded entries in the PostgreSQL database. If you might need older logs for troubleshooting, you should create a log file archive before doing the cleanup. For more information, see [Archive Logs on Command Line \(tabadmin\) on page 644](#).

To perform a cleanup, use this command:

```
tabadmin cleanup
```

You can add the `restart`, option, which is the equivalent of running `tabadmin stop`, `tabadmin cleanup`, and then `tabadmin start`:

```
tabadmin cleanup --restart
```

The files and database entries that are removed by `tabadmin cleanup` command depend on whether Tableau Server is running or stopped. Therefore, to clean up all possible files and database entries, you should run `tabadmin cleanup` twice: once when Tableau Server is running, and once when it is stopped. Here's a summary of what's removed when you run `tabadmin cleanup` with the server running and stopped.

When you run `tabadmin cleanup` with Tableau Server stopped:

- All log files are removed from `ProgramData\Tableau\Tableau Server\data\tabsvc\logs`. ([Log files from ProgramData\Tableau\Tableau Server\logs are not removed](#).)
- Temporary files are removed from `ProgramData\Tableau\Tableau Server\temp` and `ProgramData\Tableau\Tableau Server\data\tabsvc\temp`.
- No rows for HTTP requests are removed from the `http_requests` table of the Tableau Server PostgreSQL database, because the database is not accessible when the server is stopped.

When you run `tabadmin cleanup` with Tableau Server running:

- Log files older than the log file rotation interval are removed from `ProgramData\Tableau\Tableau Server\data\tabsvc\logs`. (By default, the rotation interval is one day.) Active logs and log files from `ProgramData\Tableau\Tableau Server\logs` are not removed.
- Temporary files are not removed.
- Files that are in use (that is, locked by the operating system) are not removed.

- Rows for HTTP requests that are older than seven days are removed from the `http_requests` table of the Tableau Server PostgreSQL database.

Note: Rows for HTTP requests older than seven days are also removed when you back up Tableau data. For more information, see [Back Up the Tableau Data on page 631](#).

More Information

For more information about the `http_requests` table, see [Create Custom Administrative Views on page 300](#).

For tips on how to automate running the cleanup and backup commands, refer to the following Knowledge Base article: [Server Backup and Maintenance Automation](#)

If you have created a log file archive but you no longer need it, you can remove it from the server by using the **Delete Snapshot** option on the Status page. For more information, see [Archive Logs on Status Page \(Snapshot\) on page 642](#).

Troubleshooting

Use the following topics to troubleshoot issues you may be having with Tableau Server. For tips on troubleshooting trusted authentication, see [Troubleshoot Trusted Authentication on page 457](#):

Work with Log Files

Tableau Server creates log files as a normal part of its activities. You may need to use the server log files when you are troubleshooting issues with Tableau Server or if Tableau Support requests logs to help you resolve an issue.

You can create a zipped log file archive (snapshot) from the command line on the server, or using the Generate Snapshot option on the Maintenance page. The zipped archive contains copies of the logs you can copy or download using a web browser, and send to Tableau Support. Once you have a copy of the archive, you can delete the archive from your server. For more information on creating, downloading and deleting log file archives, see [Archive Logs on Status Page \(Snapshot\) on page 642](#).

This collection of topics provides information about how to create log file archives, the contents of specific log files, and details about when and how you might want to look at a log.

Investigating Tableau Server Issues

The range and complexity of possible issues with Tableau Server means that there is no simple process you can use to investigate all problems, but a general approach would include these steps:

1. **Clean up** existing log files to reduce their size. For more information, see [Remove Unneeded Files on page 634](#).
2. **Set the appropriate logging level**. This is something that Tableau Support will instruct you on. For more information, see [Change Logging Levels on page 652](#).
3. **Reproduce the issue** you are troubleshooting so the logs capture the events related to the problem.
4. **Create an archive** of the logs. For more information see [Archive Log Files on page 641](#).

Important: Use this archive when looking at the log files. You should not edit, move or delete any files directly on the server.

5. **Review the server configuration file** (`\config\tabsvc.yml`) to get a basic understanding of the server environment.
6. **Review the admin log** (`\logs\tabadmin.log`) to understand any maintenance

that has been done on the server.

Search for `run as: <script>` to find entries specific to tabadmin activity.

7. Review the Apache logs (`\httpd\access.####_##_##_##_##_#.log` and `\httpd\error.log`) for requests that may be related to the issue you are investigating.

The Apache logs will contain a fair amount of "noise" that does not apply to issues you are experiencing.

- If you find a request that seems to be related to your issue, search `\wgserver` and `\vizqlserver` for entries that include the unique request ID from the Apache logs.
- Look for the response code and message associated with the request ID.
- Search for the name of the workbook, view, dashboard, or data source that is related to your issue. Make sure to look for a relevant timestamp.
- If you find a request that seems to be related to your issue, look at the response code associated with the request. (200s are good, 500s indicate problems.)
- Locate the unique request ID associated with the request you've identified (the unique request ID is a 24 character alphanumeric string at the very end of the request).

8. Review the log archive further to search for other messages and possible errors.

- Use the request ID from the Apache logs to search the `\wgserver` and `\vizqlserver` folders of the log archive for files containing related log entries. Look for indications of a problem (for example, error messages or long-running queries).

9. Contact support

If you are not able to solve the issue yourself, or if requested by Tableau Support, send the zipped archive to Tableau.

See the following topics for more information:

Tableau Server Processes

There are Tableau Server processes whose default configuration you can change to achieve different results. The topics [Improve Server Performance on page 430](#) and [High Availability on page 82](#) describe some of the approaches you can take. High-level status for each process is displayed on the server's Status page and more detailed information related to some of the processes—such as the background process—is in the [Administrative Views on page 289](#) topic.

Note: Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node. File store is installed when you install data engine and

cannot be installed separately. Every instance of a data engine process will always have one instance of the file store process present as well.

Architecturally, the 64-bit version of Tableau Server uses native, 64-bit processes; the 32-bit version of Tableau Server uses 32-bit processes. The exception is the data engine. If the 32-bit version of Tableau Server is installed on a 64-bit operating system, the 64-bit version of the data engine process is used.

For information on log files generated by these processes, see [Server Log File Locations on page 645](#).

Process	File Name	Purpose	Multi-threaded?	Performance Characteristics
API Server	wgserver.exe	Handles REST API calls	Yes	Unless you are using REST APIs for critical business processes, this service can be down without impacting the overall health of Tableau Server.
Application Server	vizportal.exe	Handles the web application, supports browsing and searching	Yes	Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.
Background	backgrounder.exe	Executes server tasks, including extract refreshes, 'Run Now' tasks, and tasks initiated from tabcmd	No	A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.

Process	File Name	Purpos e	Multi- Threaded?	Performance Characteristics
Cache Server	redis-server.exe	Query cache	No	A query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and API server and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.
Cluster Controller	clustercontroller.exe	Responsible for monitoring various components, detecting failures, and executing failover when needed	n/a	Included in the base install on every node.
Coordination Service	zookeeper.exe	In distributed installations, responsible for ensuring there is a quorum for making decisions during	n/a	Included in the base install on every node.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
		failover		
Data Engine	tdeserver64.exe	Stores data extracts and answers queries	Yes	<p>The data engine's workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory consumption is primarily based on the size of the data extracts being loaded.</p> <p>The 64-bit binary is used as the default on 64-bit operating systems, even if 32-bit Tableau Server is installed. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.</p>
	tdeserver.exe (32-bit)			
Data Server	dataserver.exe	Manages connections to Tableau Server data sources	Yes	Because it's a proxy, it's normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.
File Store	filestore.exe	Automatically replicates extracts across data	n/a	Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.

Process	File Name	Purpose	Multi-Threaded?	Performance Characteristics
		engine nodes		
Repository	postgres.exe	Tableau Server database, stores workbook and user metadata	n/a	Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository on page 48.
Search & Browse	searchserver.exe	Handles fast search, filter, retrieval, and display of content metadata on the server	Yes	The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.
VizQL Server	vizqlserver.exe	Loads and renders views, computes and executes queries	Yes	Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.

Archive Log Files

You can create archives (snapshots) of log files in two different ways: from the Status page using a browser, or from a command prompt using `tabadmin` on Tableau Server. Creating a log file archive gives you a zipped snapshot of logs that you can use for troubleshooting or to send to Tableau Support for help with an issue.

Archive Logs on Status Page (Snapshot)

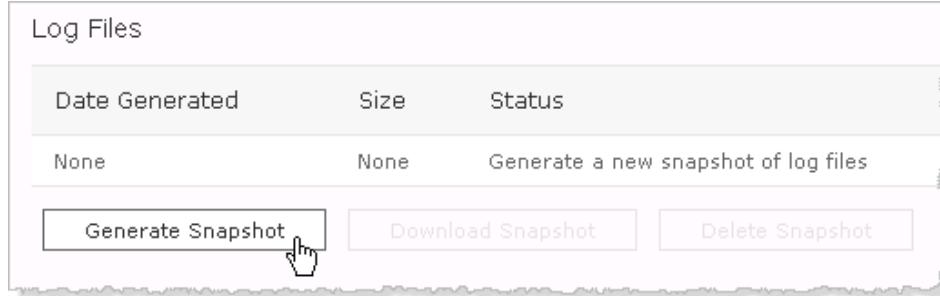
You can generate and download a snapshot (archive) of the Tableau Server log files from a web browser, without opening a command prompt. This zipped snapshot contains a copy of up to seven days of log file data from Tableau Server and any worker servers (if you have a distributed environment). The snapshot process does not change or remove either the Tableau Server log files or the log archives created with tabadmin.

Note To specify the amount of data you want to collect or the name of the zip file you are creating, use tabadmin to create an archive of server logs. For more information, see [Archive Logs on Command Line \(tabadmin\) on page 644](#).

To generate a snapshot of server log files:

1. Open the Status page:
 - Multi-site: Select **Server > Status** .
 - Single-site: Select **Status**.
2. Click **Generate Snapshot** to create a snapshot of the Tableau Server logs. The Generate Snapshot button is available only if there is no existing snapshot.

Note: This option is available whether or not you have created log archives with tabadmin.



3. Select the number of days of logs you want to include. The default is **Last 7 days**, but you might want to select fewer if you want to reduce the size of the zip file. For example, if you just reproduced an issue and are collecting logs related to the issue, you may want to select **Today** to create the smallest zip file necessary.
4. Click **Download Snapshot** to download the log snapshot to your web browser's default download location. This option is available after you create a snapshot.

Google Chrome shows you the download in the bottom of the window:

The screenshot shows a user interface for managing log files. At the top, a section titled "Log Files" displays a table with columns for Date Generated, Size, and Status. A single row is shown: "Dec 22, 2014, 3:07 PM" (Date Generated), "49.8 MB" (Size), and "Snapshot ready to download. Contains logs from previous seven days." (Status). Below the table are three buttons: "Generate Snapshot", "Download Snapshot", and "Delete Snapshot".

Rebuild Search Index

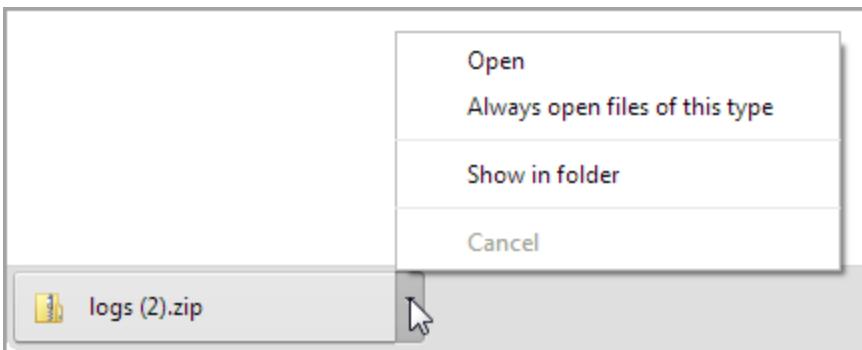
You may need to rebuild the search index if the Search & Browse process is slow.

Rebuild Search Index

File list:

- logs (1).zip

5. Click the arrow and then click **Open** to unzip the snapshot or **Show in folder** to see where it was downloaded:



6. (Optional) Click **Delete Snapshot** to delete a log snapshot. This option is available after you create a snapshot. You need to delete the existing snapshot before you can create a new one.

The screenshot shows the "Log Files" interface again. The table and file list are identical to the first screenshot. The "Delete Snapshot" button at the bottom of the table is highlighted with a red box and a cursor arrow pointing to it.

For example, you might want to delete the snapshot that you created before an event that you want to investigate.

Archive Logs on Command Line (tabadmin)

You can archive Tableau Server log files using the `tabadmin ziplogs` command. This command creates a zip file containing all of the log files and is useful when you're working with Tableau Support. The `ziplogs` command does not remove the log files, rather it copies them into a zip file. If you are running a [distributed installation](#) of Tableau Server, perform this step from the primary server. Any worker logs will be included in the zip file.

Note: The `tabadmin ziplogs` command may generate messages like "zip error: Nothing to do!" These are generally related to specific steps in the zip process and do not mean the log file archive is empty or the entire archive process failed.

To create log file archives:

1. Open a command prompt as administrator and navigate to the Tableau Server bin directory. For example:

```
cd "C:\Program Files\Tableau\Tableau Server\9.2\bin"
```

2. Stop Tableau Server by typing:

```
tabadmin stop
```

3. Create the zip file by typing `tabadmin ziplogs -l -n <filename>` where `<filename>` is the name of the zipped file you want to create. Choose a unique name with no spaces. Tableau will not overwrite an existing file. For example:

```
tabadmin ziplogs -l -n my_logs
```

If you don't specify a file name, the file is named `logs.zip`. You can also use `-d mm/dd/yyyy` to only include logs generated since a certain date. For example:

```
tabadmin ziplogs -l -n -d 12/14/2014
```

The above command creates a zipped file named `logs.zip` that includes logs dated December 14, 2014 up to the present; earlier logs are excluded. The `-n` option captures information about the server environment, including which ports are in use. To see a list of all the `ziplogs` options, type `tabadmin ziplogs -h`.

4. Restart Tableau Server by typing:

```
tabadmin restart
```

You can find the zipped log file in the Tableau Server bin directory.

Server Log File Locations

By default, Tableau Server log file archives are gathered in a zip file called `logs.zip` (you can specify a different name if you create the archive using `tabadmin`). You can copy the archive from the server to a local computer and open it there, or send it to Tableau Support. When you unzip the archive, a series of folders are created with related log files. This table explains the possible contents of each folder, along with the original location the files came from on the Tableau Server, the process that created the log files, and details about the files.

The Tableau Server log directory is `C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs` if you installed Tableau Server on drive C, unless otherwise noted in the table below.

Log Archive File Locations

Files/folders in <code>logs.zip</code>	Details	Files	Generated by	Location on Tableau Server
build-version.txt	The build version of Tableau Server.			
tabsvc.yml				\config
wgserver.checksum				
assetkey-encryption	Logs related to repository encryption.	assetkeyencryption.log	tabadmin assetkeys	\logs\assetkeyencryption
backgrounder	Logs related to subscriptions and scheduled activities like extract refreshes, "Run Now" tasks, and tab-cmd tasks.	backgrounder-.log spawn.####.log tomcat-#.#####-##.##.log	backgrounder.exe	\logs\backgrounder
	Logs related to the Cache		redis-server.exe	\cacheserver

cacheserver	Server process.			
cluster-controller	Logs related to the Cluster Controller process.	clustercontroller.log clustercontroller.log-#####-##-##	cluster-controller.exe	\clustercontroller
config	Configuration files. This is a good place to start gathering information when troubleshooting. Confirm that the configuration settings are what you expect.	connections.yml workgroup.yml	Tableau Server Configuration	\config
data-collector				\logs\datacollector
dataengine	There will be a tdeserver log file for each day with information about data extracts and queries, and responses to VizQL server requests.	tdeserver_####_##_##_##_.log	tdeserver.exe tdeserver64.exe	\logs\dataengine
dataserver	Information about connections to Tableau	dataserver-#.log	dataserver.exe	\logs\dataserver

	Server data sources.			
httpd 	Apache logs. Look here for authentication entries. Each request in the Apache log will have a request ID associated with it. This request ID is used throughout the server logs and you can use it to associate log entries with a request.	access.#####-##-##.##-##-##log error.log startup.log	Apache daemon	\logs\httpd
licensing 				\logs\licensing
logs 	This is the location of the logs of most interest and usefulness. Look here after reviewing the configuration files. tabadmin.log is never overwritten or truncated so it contains all the details.	tabadmin.log tabconfig.log tablicsrv.log tabsrvlic.log wgserver.war.deploy.log		\logs

	notify-tabadmin.log contains errors from tabadmin.log (the errors are also included in tabadmin.log). tablicsrv.log and tabsrvlic.log are related to licensing.			
 pgsql	PostgreSQL database logs, including files related to launching server processes. Tableau data extracts are stored in the PostgreSQL database.		tabspawn	\logs\pgsql
 repository			postgres.exe	\logs\repository
 service		notify-tabsvc.log tabsvc.log		\logs\service
 solr	Related to search indexing.			\logs\solr
 svcmonitor				\logs\svcmonitor
 tabad-	Related to log archives cre-			\log-s\tabadminservice

minservice	ated using the Generate a Snapshot of Server Log Files option.			
tabadmwrk	Server Worker Manager process that is used for auto-discovery of worker servers in a distributed environment.		tabadmwrk.exe	\logs\tabadmwrk
vizportal				\logs\vizportal
vizqlserver	Related to showing and interacting with views. When running multiple instances of VizQL Server, the instances are distinguished by port number. notify-production logs contain exceptional events.	vizql-0.log.#####-##-## spawn.#####.log	vizqlserver.exe	\logs\vizqlserver
vizqlserver-\logs	Most files are in JSON format. tabprotosrv.txt	backgrounder_#####_#####_##_##_##_##.txt dataserver_#####_		\vizqlserver\logs

	<p>is created when you open data or connect to data.</p>	<pre>#####_##_##_##_ ##_##.txt tabadmin_##### ##_##_##_## ##.txt tabprotosrv.txt vizqlserver_##### #####_##_##_## ##_##.txt wgserver_##### #####_##_##_## ##_##.txt tdserver_ vizqlserver_##### #####_##_##_## ##_##.txt</pre>		
	<p>wgserver</p> <p>Information related to administrative tasks, workbook and permissions management, authentication, sign-ins, initial view requests, and publishing requests.</p> <p>Browsing, searching.</p> <p>Instances of wgserver are distinguished by port number, immediately following</p>	<pre>db-migrate_##### ##_##_##_## ##.log migrate.log notify-production.##### #####_##_##_## ##_##.log production.#####.log production.##### #####_##_##_## ##_##.log spawn.#####.log tomcat-#.##### ##_##.log wgserver-#.log</pre>	wgserver.exe	\logs\wgserver

	<p>"production" or "notify-production".</p> <p>notify-production logs contain exceptional events.</p> <p>There will be a separate production.n_### file for each backgrounder process for each day.</p> <p>notify-production.n_### correlates to production.n_### but contains only errors.</p>			
	<p>zookeeper</p> <p>Information related to the Tableau Server Coordination Service.</p>	<p>spawn #####.log</p> <p>zookeeper-#.log</p> <p>zookeeper-#.log ##### _##_##</p>	<p>zookeeper.exe</p>	\logs\zookeeper

Tableau Server log files can be found in the following folders on the server:

[Tableau Service Logs](#)

The following log files track activities related to the web application, database, and index:

C:\ProgramData\Tableau\Tableau Server\data\tabsvc

VizQL Logs

These log files track activities related to displaying views, such as querying the database and generating images:

C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Logs

Temporary Files

Any file that starts with exe_ in the folder below is a Tableau Server file and can be deleted.

C:\ProgramData\Tableau\Tableau Server\temp

Change Logging Levels

By default, Tableau Server logs events at the **Info** level. You can change this if you need to gather more information (if you are working with Tableau Support, for example). As a best practice you should not increase logging levels except when troubleshooting an issue.

Logging Levels

The following logging levels are listed in order of increasing amount of information logged:

- off
- fatal
- error
- warn
- info (the default)
- debug
- trace

Note: Increasing the log level to debug or trace increases the amount of information being logged and can have a significant impact to performance. You should only set a logging level to debug when investigating a specific issue. Reproduce the issue and then reset the logging level back to info.

Change Logging Levels

Set logging levels for Tableau Server using one of several **tabadmin set** commands. The command you use depends on which component of Tableau Server you want to change the logging level for.

Command	Location of affected logs (path begins with \ProgramData\Tableau\Tableau Server\data\tabsvc)
---------	---

server.log.level	\vizqlserver\Logs*.txt
vizportal.log.level	\vizportal*.log
vizqlserver.log.level	\vizqlserver*.log
wgserver.log.level	\wgserver*.log

For more information, see [tabadmin set options](#) on page 616.

You need to stop Tableau Server before changing the logging levels, and restart it afterward. If you are running a [distributed installation](#) of Tableau Server, set logging levels from the primary server.

To change the logging level:

1. Open a command prompt as administrator and navigate to the Tableau Server bin directory.

If Tableau Server is installed on the C drive:

```
C:\Program Files\Tableau\Tableau Server\9.2\bin
```

or

```
C:\Program Files (x86)\Tableau\Tableau Server\9.2\bin
```

2. Stop Tableau Server by typing:

```
tabadmin stop
```

3. Set the logging level to by typing `tabadmin set [command] [option]`

where `[command]` is `server.log.level`, `vizqlserver.log.level` or `wgserver.log.level`

and `[option]` is a valid logging level.

Examples:

- `tabadmin set server.log.level debug`
- `tabadmin set vizqlserver.log.level warn`
- `tabadmin set vizportal.log.level debug`
- `tabadmin set wgserver.log.level off`

4. Restart Tableau Server by typing:

```
tabadmin restart
```

Reset Logging Levels

After you gather the information related to the issue you are investigating, reset the logging levels so there is no lingering performance impact.

Reset the logging level back to its default (info) using the appropriate command with a -d option.

Examples:

- `tabadmin set server.log.level -d`
`tabadmin set vizportal.log.level -d`
- `tabadmin set vizqlserver.log.level -d`
- `tabadmin set wgserver.log.level -d`

Handle an Unlicensed Server

Tableau offers two licensing models: user-based and core-based. User-based licensing requires each active user account to be covered by a license. User-based licenses have a defined capacity, or number of users that it allows. Each user is assigned a unique user name on the server and is required to identify himself when connecting to the server.

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. If you have a Tableau Server—Multi-Machine Core license, you can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

If you have a Tableau Server—Single-Machine Core license, you can install Tableau Server on only one computer. The number of processor cores on the computer must not exceed the number of cores that you have licensed.

Unlicensed User-Based Server

The most common reason for a server that has user-based licensing to be unlicensed is an expired product key or an expired maintenance contract. You can see your products keys and add new ones by selecting **Start > All Programs > Tableau Server > Manage Product Keys**.

Unlicensed Core-Based Server

A core-based server can become unlicensed for a variety of reasons. A common problem is that the primary or a worker machine has more cores than the license allows. When the server is unlicensed you may not be able to start or administer the server. You can, however, manage

your licenses using the [tabadmin command line tool](#). Follow the steps below to see a list of your licenses and number of cores by machine.

1. Open a command prompt and type the following: cd C:\Program Files\Tableau\Tableau Server\9.2\bin
2. Type the following: tabadmin licenses.

Handle an Unlicensed VizQL Server Process

There are several status indicators on the Tableau Server Status page that help you understand the state of Tableau Server processes. An orange-color status box, "Unlicensed", indicates that one of the VizQL server processes is unable to retrieve the Tableau Server license information.

Process Status		
The real-time status of processes running in Tableau Server.		
Process	Primary 10.32.139.21	Worker 10.32.139.22
Gateway	✓	✓
Application Server	✓	✓
API Server	✓	✓
VizQL Server	✓✓	⚠
Cache Server	✓✓	✓✓
Search & Browse	✓	✓
Backgrounder	✓	✓
Data Server	✓✓	✓✓
Data Engine	✓	✓
File Store	✓	✓
Repository	✓	✓
<button>Refresh Status</button>	✓ Active ⌚ Busy ✗ Passive ⚠ Unlicensed ✗ Down ◻ Status unavailable	

There may be several reasons why the process is unable to access this information. For example, there may be network issues preventing a VizQL process, which is running on a worker machine, from communicating with the primary machine. Or, the process may be getting sent more requests than it can accept at that time and can't handle the licensing request. As a result, some of your users may be able to access views while others cannot.

To resolve the problem, [stop](#), then [start](#) Tableau Server.

VizQL 'Out of Memory' Error

In 32-bit versions of Tableau Server, if a VizQL process reaches its limit of concurrent viewing sessions you may see an 'Out of Memory' error, which will also be written to the *vizqlserver*.txt* logs located here:

C:\ProgramData\Tableau\Tableau
Server\data\tabsvc\vizqlserver\Logs

The VizQL process doesn't terminate when this error occurs, but it will not accept additional connections. You can handle this problem by doing the following:

- **Upgrading to the 64-bit version of Tableau Server:** See [Upgrade to 9.2 on page 51](#) for details.
- **Increasing the number of VizQL processes:** This may mean that you need to add one or more workers. See [Install and Configure Worker Nodes on page 76](#) for how to do this.
- **Edit `vizqlserver.session.expiry.timeout`:** Use tabadmin to change the `vizqlserver.session.expiry.timeout` setting from its default (30 minutes) to a shorter time period such as 10 or 5 minutes. This will allow idle sessions to expire sooner, thus freeing memory for new sessions.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

Troubleshoot Data Sources

For users to work with Tableau Server data sources, up to three things need to be in place:

- **Permissions for the data source:** Anyone connecting to a data source must have the **Connect** and **View** permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be licensed to Publish and also have the **Write/Save As** and **Download/Web Save As** permissions. See [Manage Permissions on page 336](#) and [Set Permissions for a Data Source on page 354](#) for more information.

Multidimensional (cube) data sources have to be downloaded and used in Tableau Desktop, so they require **Download/Web Save As** permission. For more information about cubes in Tableau, see [Cube Data Sources on page 275](#).

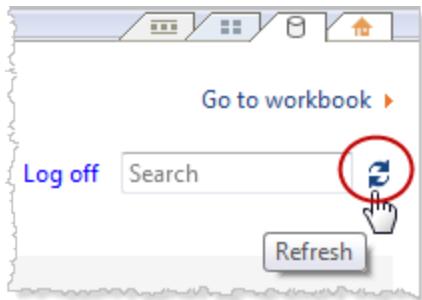
- **Ability to authenticate to the database:** There are several ways you can connect to data in Tableau and control who has access to what. Basically, whichever entity is connecting to the database must be able to authenticate. The entity could be Tableau Server performing an extract refresh. It could be a Tableau Desktop user connecting to a data source that then connects to a live database. It could also be a Tableau Server user who's accessing a view that connects to a live database. Refer to [Data Security on page 417](#) to learn more about your options.
- **Database drivers:** If the person who created and published the data source in Tableau Desktop needed to install additional database drivers, you may need to install them on Tableau Server as well. If you are running a distributed installation of Tableau Server where, for example, the data server process is running on a worker server, any required database drivers must be installed there as well as on the primary server. Other processes require drivers as well. See [Database Drivers on page 78](#) for more information.

Data Source Error Messages

Here are some errors that workbook authors and other users may encounter as they work with data sources and views:

Permission to access this Tableau Server data source denied: Connecting to a data source requires the Connect permission. See [Manage Permissions on page 336](#) and [Set Permissions for a Data Source on page 354](#) for more information.

Data source not found: Someone working with a view may see this error if a data source is removed from Tableau Server or if their Connect to Data page needs to be updated. To update the Connect to Data page in Tableau Desktop, click the Refresh icon:



Unable to connect to this Tableau Server data source: This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct settings.

Unable to list Tableau Server data sources: This error may occur if a user is trying to access Tableau Server data sources and there are connectivity issues between Tableau Server and Tableau Desktop.

Can't connect with a cube data source: To use a published multidimensional (cube) data source, you must download the data source and use it in Tableau Desktop. Verify that you have the **Download/Web Save As** permission for the data source. For more information about cubes in Tableau, see [Cube Data Sources on page 275](#).

Troubleshoot Subscriptions

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials:** Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.
- **Database temporarily down:** If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.
- **Background process timeout:** By default, the background process that handles subscriptions times out after 30 minutes. In the majority of cases, this is plenty of time. However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the [Background Tasks for Non Extracts on page 297](#) admin view to see if that's the case. To increase the timeout threshold, use the tabadmin option `subscriptions.timeout`.

Can't subscribe

If you can see a view on Tableau Server and it has a subscription icon () in the upper right corner, you can subscribe to it.

Two things need to be in place for you to subscribe to a view: Tableau Server needs to be correctly configured (described in [Manage Subscriptions on page 231](#)) and the view you're subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn't being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the [Tableau Desktop help](#) for details).

No subscription icon

It's possible to see a view on Tableau Server but be unable to subscribe to it. This happens for views with live database connections, where you're prompted for your database credentials when you first click the view. A subscription includes a view (or workbook), data, and a schedule. To deliver the data piece, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don't require credentials or have them embedded.

You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is configured for trusted authentication. See [Subscription Requirements](#) for more information.

Receiving invalid or "broken" subscriptions

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don't work, or receiving subscriptions even though they've unsubscribed from the view or workbook.

Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")

You may see the above error in Windows Event Viewer if subscriptions appear to be sent (according to the [Background Tasks for Extracts](#) on page 295 admin view), yet subscriptions aren't arriving, and your SMTP server is using encrypted (SSL) sessions. Subscriptions are only supported for unencrypted SMTP connections. The solution is to use an unencrypted SMTP server.

Custom scripts not working after upgrade to 8.1

To support better session management, starting with version 8.1, a hash tag (#) was added to the end of view URLs. If you had custom subscriptions scripting that generated views as PDFs or PNGs you may need to update your scripts to allow for the hash tag.

For example, prior to version 8.1, view URLs use this syntax:

`http://tableauserver/views/SuperStore/sheet1`. To generate a view as a PNG, `.png` could be added to the end of the URL. For example,
`http://tableauserver/views/SuperStore/sheet1.png`.

In versions 8.1, 8.2, or 8.3, view URLs use this syntax:

`http://tableauserver/views/SuperStore/sheet1#1`. To generate a PNG, add

.png before the hash tag. For example:

`http://tableauserver/views/SuperStore/sheet1.png#1`

Custom scripts not working after upgrade to 9.0

In version 9.0, the session ID at the end of server URLs is indicated by an "iid" parameter, `:iid=<n>`. For example,
`http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1`. This parameter replaces the hash tag "`#<n>`" used for the session ID in 8.x versions of Tableau Server.

If you use custom subscriptions scripts that generate views as PDFs or PNGs, you may need to update your scripts by removing the hash tag and number (`#<n>`), and inserting the `?iid=` session ID parameter before the number.

Starting in version 9.0, view URLs use this syntax:

`http://tableauserver/views/SuperStore/sheet1?:iid=2`.

To generate a PNG in version 9.0 and later, add .png before the session ID:

`http://tableauserver/views/SuperStore/sheet1.png?:iid=2`

Troubleshoot SAML

Use the following topics to troubleshoot SAML issues.

SAML and Enable Automatic Logon

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select **Enable automatic logon**. **Enable automatic logon** and SAML cannot both be used on the same server installation.

HTTP Status 500 error when configuring SAML

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:

```
org.opensaml.saml2.metadata.provider.MetadataProviderException:  
User specified binding is not supported  
by the Identity Provider using profile  
urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser
```

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.

- The IdP is configured to use SP-initiated authentication. (IdP-initiated authentication is not supported.)>
- The IdP is configured to use HTTP-POST requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in [SAML Requirements on page 475](#).

Signing In from the Command Line

Even if Tableau Server is configured to use SAML, it is not used if you sign in to Tableau Server using the command line tools [tabcmd](#) on page 552 or the [Tableau Data Extract command line utility](#) (provided with Tableau Desktop).

Login Failed

Login can fail with the following message:

```
Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.
```

This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `jsmith` it must be stored in Tableau Server as `jsmith`.

SAML Error Log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see [Archive Log Files on page 641](#).

Note: In Tableau Server 9.0 and later, to log SAML-related events, both `wgserver.log.level` and `vizportal.log.level` must be set to `debug`. For more information, see [Change Logging Levels on page 652](#).

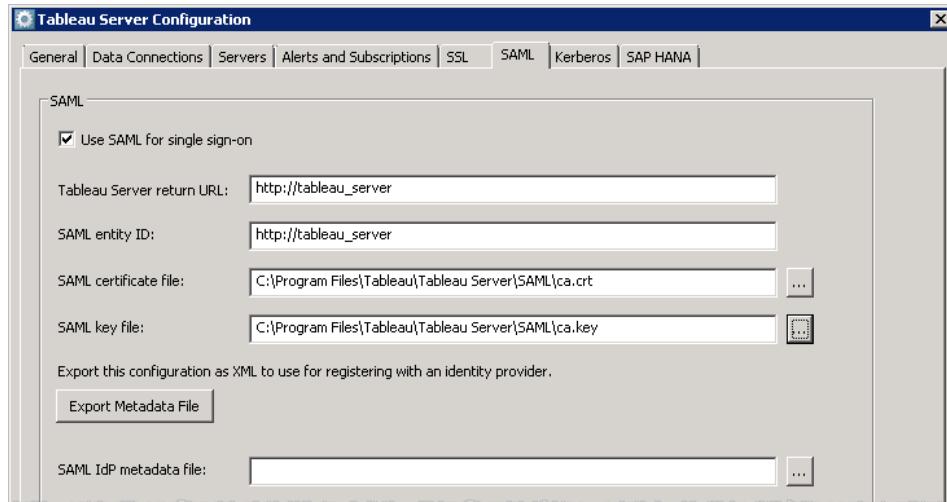
Check for SAML errors in the following files in the unzipped log file snapshot:

```
\wgserver\wgserver-<n>.log  
\vizportal\vizportal-<n>.log  
\wgserver\production.<nnnn>_<yyyy_mm_dd_hh_mm_ss>.log
```

In Tableau Server 9.0 and later, the application process (vizportal.exe) handles authentication, so SAML responses are logged by that process. The SAML setup process logs information in the logs for the api server process (wgserver.exe).

Trailing Slash

On the SAML tab, confirm that the **Tableau Server return URL** does not end with a trailing slash (correct: `http://tableau_server`; incorrect: `http://tableau_server/`):



Confirm Connectivity

Confirm that the Tableau Server you are configuring has either a routeable IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running telnet on Tableau Server and attempting to connect with the SAML IdP. For example: `C:\telnet 12.360.325.10 80`

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.

Troubleshooting Mutual SSL Authentication

This topic describes possible mutual (two-way) SSL authentication issues and their causes, the messages that users might see, and possible mitigation for the issues.

- The client is missing a certificate
- The client doesn't support mutual SSL authentication
- Client certificates are not published to Active Directory
- Users unexpectedly see a sign-in dialog box that displays an error message
- The user name in the UPN or CN fields is missing or invalid
- The user is signed in using unexpected user name (LDAP mapping)
- The user is signed in as incorrect user (UPN or CN mapping)

For more information about mutual SSL authentication and LDAP, UPN, and CN user mapping, see the following topics:

- [Quick Start: Mutual \(Two-Way\) SSL Authentication](#) on page 489
- [Mapping a Client Certificate to a User During Mutual Authentication](#) on page 497

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

The client is missing a certificate.

If the client has no client certificate, the user sees this message during authentication:

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

To resolve the issue, the user should contact the system administrator to generate a certificate for the client computer.

Invalid user name or password

The client doesn't support mutual SSL authentication.

Versions of Tableau Desktop older than version 9.1 do not support mutual SSL authentication. If an older version of Tableau Desktop is used to connect to Tableau Server that is configured for mutual SSL authentication, the following can occur:

- If Tableau Server is configured to use fallback authentication, the client displays a sign-in dialog box and the user can enter a user name and password.
- If the server is not configured to use fallback authentication, the user sees the following message and cannot connect to the server:

Invalid user name or password

For more information about fallback authentication, see [Quick Start: Mutual \(Two-Way\) SSL Authentication](#) on page 489.

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

Client certificates are not published to Active Directory.

If Tableau Server is configured to use Active Directory for authentication, and if user mapping is set to LDAP, Tableau Server sends the client certificate to Active Directory for authentication. However, if client certificates have not been published to Active Directory, authentication fails and the user sees the following message:

We couldn't find your user name in the client certificate.

Contact your Tableau Server administrator or sign in using your Tableau Server account.

To resolve this issue, the system administrator should make sure that client certificates are published to Active Directory. Alternatively, the server should be configured to use a different user mapping (UPN or CN), and the system administrator should be sure that client certificates contain user names in the UPN or CN fields.

Users unexpectedly see a sign-in dialog box that displays an error message

If Tableau Server is configured to use mutual SSL authentication and certificates are available for use with users' computers, a user should not see a sign-in dialog box, because Tableau Server uses the certificate to authenticate the user. However, if the server does not recognize the user name in the certificate, the user sees a sign-in dialog box with an error message that indicates why the certificate was not used. This can occur when all of the following conditions are true:

- Fallback authentication is enabled.
- If the server is using UPN or CN mapping, the user name in the certificate's UPN or CN field is not recognized. If the server is using LDAP mapping, the certificate is not mapped to the user in Active Directory.

To resolve this issue, the system administrator should do the following, depending on how user mapping is configured on Tableau Server:

- LDAP mapping: Make sure that the certificate is linked to the user, that the certificate is available for use with the user's computer, and that the user is configured as a Tableau Server user.
- UPN or CN mapping: Make sure that the certificate is available for the user's computer, that the user name is in the certificate's UPN or CN field, and that the user name matches the user name on Tableau Server (including domain).

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.

Certificate does not contain a valid Tableau Server user name.

The user name in the UPN or CN fields is missing or invalid

When Tableau Server is configured to use UPN or CN mapping, the server reads the user's name from the UPN or CN field of the certificate and then looks up the user name in Active Directory or in the local repository on Tableau Server. (The specific field that the server reads depends on which mapping—UPN or CN—the server is configured to use.) If the field that is supposed to contain the user name has nothing in it, the user sees the following message:

We couldn't find your user name in the client certificate.
Contact your Tableau Server administrator.

If a client certificate contains a user name but Active Directory and Tableau Server don't recognize the user name, the user sees the following message:

Certificate does not contain a valid Tableau Server user name.

This can occur when all of the following conditions are true:

- Tableau Server is configured to use UPN or CN mapping.
- Fallback authentication is not enabled.
- The client certificate has no user name in the UPN or CN field, or the user name in the UPN or CN field does not match a user name in Active Directory or on Tableau Server.

To resolve this issue, the system administrator should make sure that the user's certificate has the correct user name in the UPN or CN fields of the certificate.

The user is signed in using an unexpected user name (LDAP mapping)

When the server is configured to use Active Directory authentication and LDAP mapping, the certificate is linked to a user in Active Directory. If the certificate contains a user name in the UPN or CN field, that user name is ignored.

If the intention is that the user should be signed in with the user name in the UPN or CN fields, the server should be configured to use UPN or CN mapping.

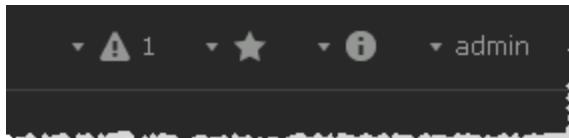
The user is signed in as the incorrect user (UPN or CN mapping)

Under some circumstances, the user name in a UPN or CN field in the client certificate can be ambiguous. The result is that a user is signed in to the incorrect identity.

For more information about the conditions under which this issue can occur, see [Ambiguous user names in multi-domain organizations](#) in the topic **Mapping a Client Certificate to a User During Mutual Authentication** on page 497.

Handle Extract Refresh Alerts

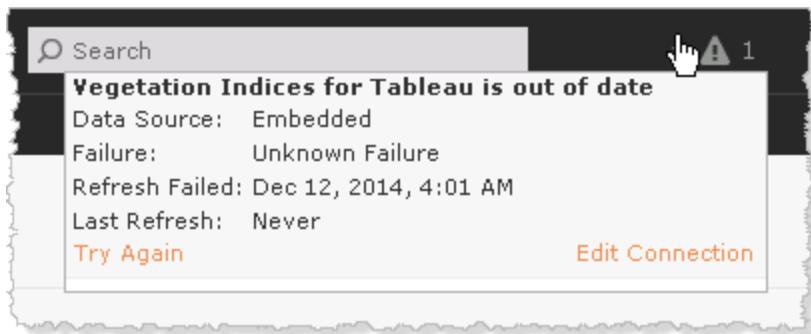
If scheduled extract refreshes did not succeed, Tableau displays an Alerts menu in the upper right corner:



You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn't be refreshed
- The author of a workbook that connects to a data source that couldn't be refreshed

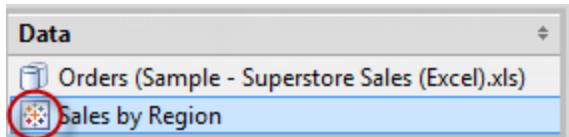
When you open the Alerts menu you can see more information about the refresh failure(s):



When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the **Data source** (for example, **Data source: sales_data**), it means that the data source is a **Tableau Server data source**. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon :



Resolving Extract Refresh Problems

You can resolve some extract refresh problems by clicking the **Edit connection info** link in the alert, and then entering the missing information, and clicking **Save**:

The screenshot shows the 'Edit Data Connection' dialog box. It contains the following fields:

Server:	SQL2008r2	undo x
Server Port:	6060	undo x
Database Username:	jsmith	undo x
Password:	*****	
Confirm Password:	*****	

Below the fields is a blue 'Test Connection' button. At the bottom right are two orange buttons: 'Save' and 'Cancel'.

If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

Tip: Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see [Troubleshoot Server Processes on page 242](#).

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.
2. Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server,

use the `tabadmin restart` command. This will stop all the processes associated with Tableau Server and then restart them.

3. Clean up files associated with the Coordination Service (ZooKeeper). To clean up Coordination Service files, use the `tabadmin cleanup --reset-coordination` command.

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tabadmin status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tabadmin restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Installing Tableau Server

Install fails due to hardware requirements

Starting with version 9.0, Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The minimum requirements are designed to minimize issues that result from running Tableau Server on under-powered computers. The requirements apply to both primary server computers and worker computers. The minimum requirements are lower for the 32-bit version of Tableau Server. If you are unable to install the 64-bit version due to hardware limitations, you may be able to use the 32-bit version instead.

For details on minimum hardware requirements, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#) on page 59.

Upgrading Tableau Server

Migrating Extracts to the File Store

Tableau Server 9.2 introduced a more reliable storage mechanism for data extracts called the File Store. Upgrading from a previous version requires migration of the extracts. This can take

a long time (up to several hours) if you have a large number of extracts or extracts that have a lot of data. During migration a message displays:

```
Migrating extracts to File Store  
This process may take up to several hours.
```

If the migration progress appears to be stalled or stuck, you can verify that migration is continuing by watching the `tabadmin.log`. An entry is written to this log for each extract that is migrated.

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts. As a part of upgrading to version 9.0, the Setup program migrates extracts to the new File Store and this takes space.

To free up disk space:

1. Zip and save logs using the `tabadmin ziplogs` command.

After you create the `ziplogs` file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tabadmin cleanup` command. For more information, see [Remove Unneeded Files](#) on page 634

Reindexing Tableau Server Search & Browse

Other problems that can be solved by reindexing Search & Browse

Other symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, rebuild the Search & Browse index using the `tabadmin reindex` command.

JavaScript API

With Tableau's JavaScript API you can integrate Tableau visualizations into your own web applications. The API lets you tightly control your users' interactions and combine functionality that otherwise couldn't be combined. For example, you can code a single control that filters a group of marks, selects some of those marks, and presents their data for download.

To learn more, see [JavaScript API](#).

REST API

With the REST API you can manage and change Tableau Server resources programmatically, via HTTP. The API gives you simple access to the functionality behind the data sources, projects, workbooks, site users, and sites on a Tableau server. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

To learn more, see [REST API](#).

Contact Us

Directory of worldwide offices

Sales

Contact

Support

1. Search our support resources.
2. Review the search results to see if your question is answered.
3. If you can't find what you need, scroll to the bottom of the search results, and click **Continue and Create Case**.



Copyright

© 2015 Tableau Software, Incorporated and its licensors. All rights reserved.

Patent www.tableau.com/ip.

Portions of the code copyright © 2002 The Board of Trustees of the Leland Stanford Junior University. All rights reserved.

Tableau's installation includes an unmodified executable version of the Firebird database. The source code for that database can be found at <http://www.firebirdsql.org>

For a listing of third party copyright notices please refer to the following file that is installed with Tableau Server:

C:\Program Files\Tableau\Tableau Server\9.2\COPYRIGHTS.rtf

Note: If you installed 32-bit Tableau Server on a 64-bit operating system, it will be in

C:\Program Files (x86)\Tableau\Tableau Server\9.2\COPYRIGHTS.rtf

This product includes software developed by Andy Clark.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product is Client Software as defined in Tableau Software's End User Software License Agreement.

Edit and Create Workbooks

Users with the appropriate permissions for the web authoring environment can edit existing workbooks or create new ones.

When you sign in to Tableau Server, the Content page appears by default. Content (workbooks and data sources) that you have access to appear here as a result of either of the following processes:

- A Tableau Desktop user publishes a workbook or data source to Tableau Server.
- A Tableau Server user creates and saves a workbook in the web editing environment.

Build a View

You can build a new view by creating a new sheet in an existing workbook or by creating a new workbook. This topic shows how to connect to a published data source and build a view in a new workbook.

The following procedure shows how to build a view that incorporates information about sales by category and region. It uses the **Superstore** sample data source that comes with Tableau Desktop, and is published to Tableau Server. If you do not have access to the data source shown, you can connect to your own published data and create a similar view using measures and dimensions from it.

1. On the **Content** page of a site, select **Data Sources**.
2. In the data sources list, select the check box next to the data source you want to use, and then select **Actions > New Workbook**.

A new, blank workbook opens in the Tableau Server web editing environment.

Note: The **New Workbook** option is not available if the data source is a for a cube-based database. For more information see [Cube Data Sources](#) on page 275.

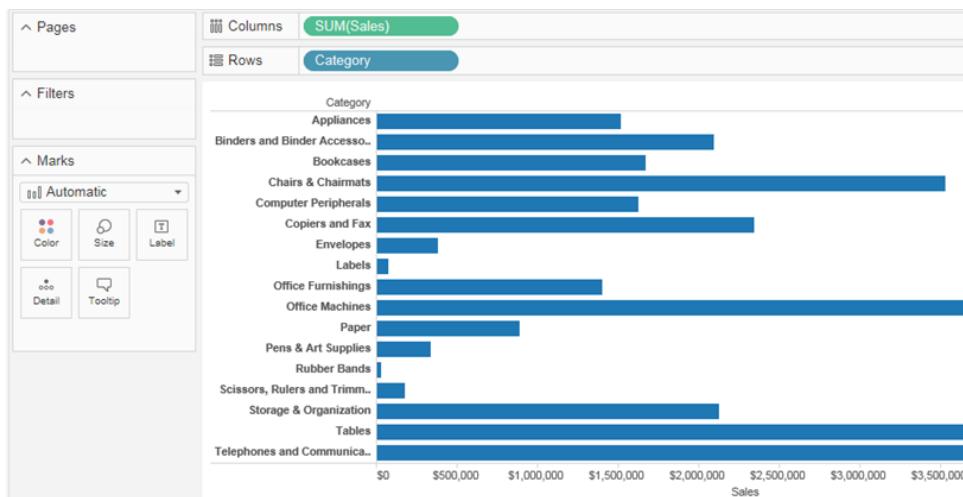
3. From the Measures pane, drag **Sales** to the Columns shelf.

The screenshot shows the Tableau Data pane on the left side of the interface. The 'Dimensions' section is expanded, displaying categories like Customer, Order, Location, and Product. The 'Measures' section is also expanded, listing items such as Sales, Profit, and Quantity. The item 'Sales' is highlighted with a red oval. A large black arrow points from the circled 'Sales' measure up towards the 'SUM(Sales)' button in the top right corner of the pane.

4. From the Dimensions pane, expand **Product** to display its sub-categories, and then drag **Category** to the Rows shelf.

The screenshot shows the Tableau Data pane. On the left, under Dimensions, the Product category is expanded, and its sub-item 'Category' is highlighted with a red oval. The right side of the pane shows the Analytics shelf with 'SUM(Sales)' assigned to Columns and 'Category' assigned to Rows. The Marks shelf is set to 'Automatic'.

Tableau now has enough to convert the data into a visualization (view), in this case a horizontal bar chart.



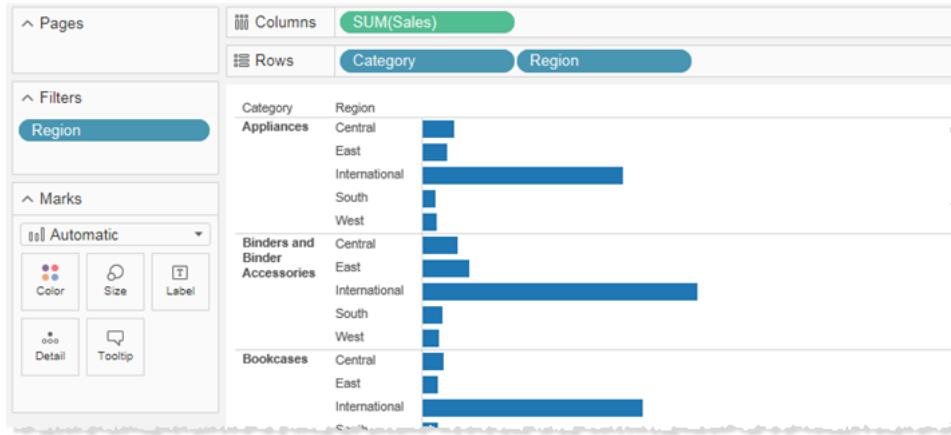
- From the Dimensions pane, drag **Region** to the Rows shelf.

The view now contains another layer of data—the categories are broken out by region.

Now suppose you want to view and compare sales by category in a single region. You can accomplish this using a filter.

6. From the Dimensions pane, drag **Region** to the Filters shelf.

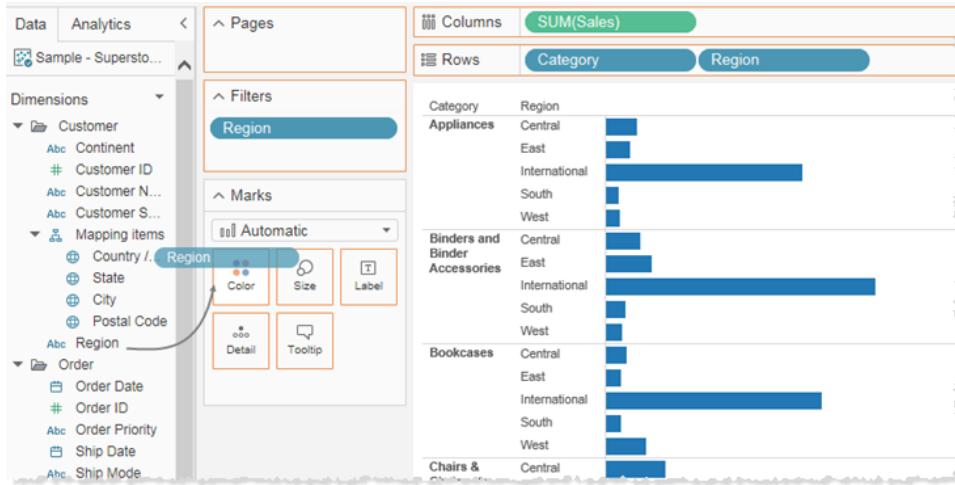
As you hover over the Filters shelf, a small triangle at the left of the field indicates that you can drop **Region** onto the shelf.



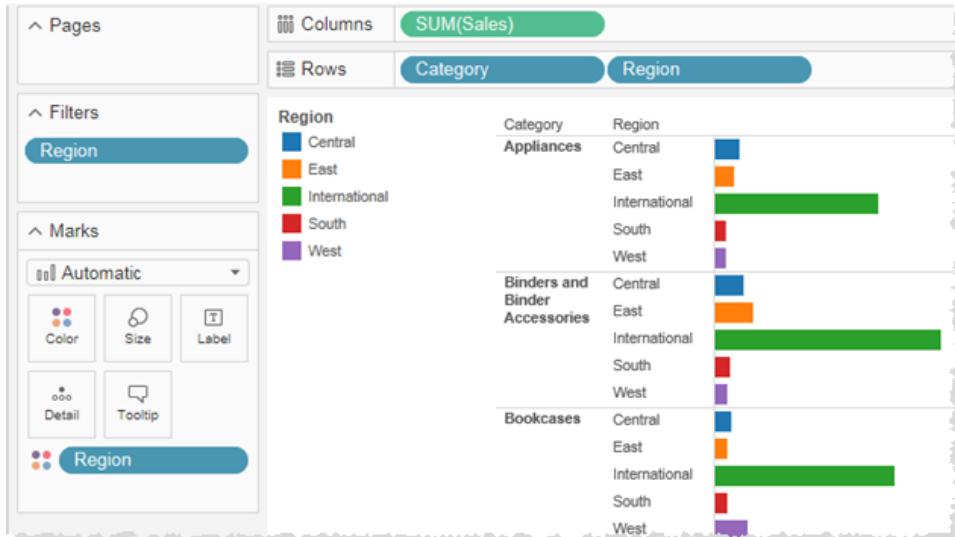
A Filter control appears at the right edge of the page.



7. Clear the check boxes for all but one region that you want to analyze, and then select them all again.
8. You can enhance the visualization using color. Drag **Region** to **Color** on the Marks card.

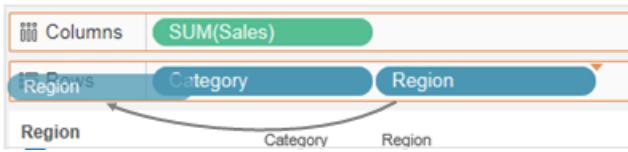


You now have a useful view that allows you to compare sales of different product categories across regions:

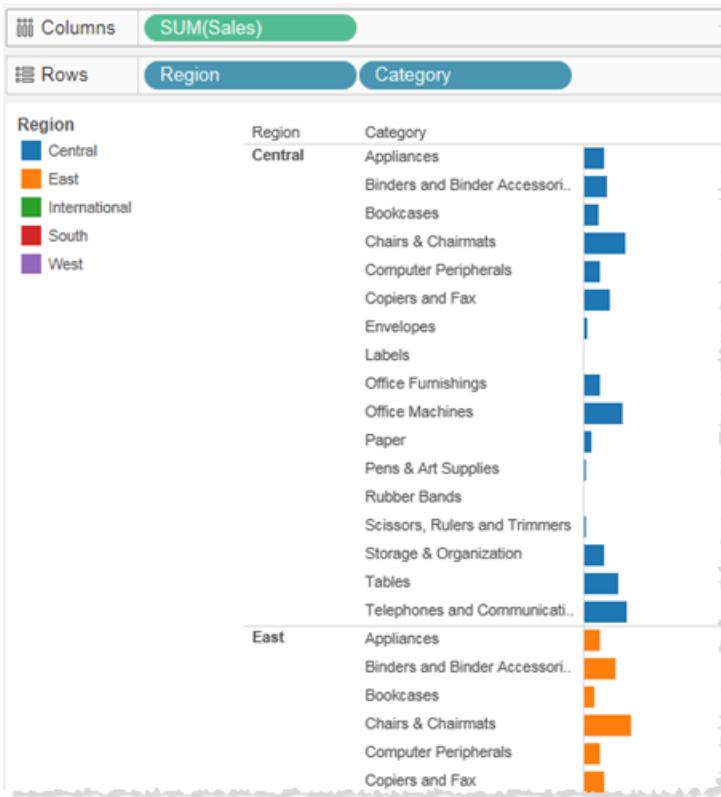


Tip: To learn about selecting a different color palette for the bars or resizing them, see [Marks](#) on page 695.

- Instead of focusing on regional sales of each product, maybe you prefer a view that lets you more easily analyze a region's overall product sales. On the Rows shelf, drag **Region** to the left of **Category**.



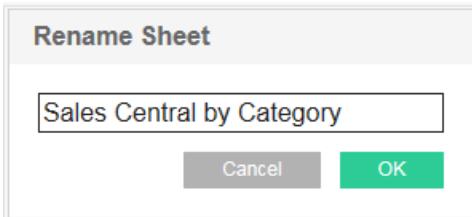
The view refreshes to show sales of all products by region.



10. If you decide that you prefer the previous version of the view, you can click **Undo** in the Toolbar.
11. If you want to create a second worksheet, select the **New Worksheet** tab at the bottom of the view.

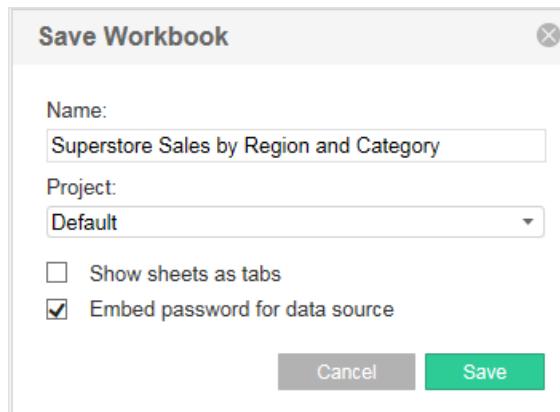


Select the worksheet tab and select **Rename Sheet** to give it a more descriptive name.



12. Click **Save** to save the workbook. In the **Save Workbook** dialog box, complete the following steps:

- Specify the workbook name, and leave **Project** set to **Default**.
- Select **Show sheets as tabs** if you created multiple sheets and want their tabs to appear at the bottom of the view.
- Select **Embed password for data source** if you want users who do not have an account on the database to be able to see the view.
- When you are finished, click **Save**.



Edit a View

In the Views section, you can open a view for editing in the following ways:

- In thumbnail view, select a thumbnail, then select **Actions > Edit View**.
- In thumbnail view, select a thumbnail, then click the Actions menu and select **Edit View**.
- Click a view to open it, and then click **Edit** at the top of the view.

If the workbook publisher did not embed database credentials, you are prompted to provide them.

Save or Discard Changes

While you are editing a view, you can save or discard changes by clicking the options in the header above the view area.



When you save your work, even though you entered the authoring environment from a single view, the complete workbook is saved, including other views you may or may not have edited.

- **Save** overwrites the original workbook.

Note: This option is only shown if you have permission to overwrite the workbook.

- **Save As** creates a new workbook in the same project.

If you want to keep both the original version of a view and your edited version, use **Save As** to create a new workbook.

If you select **Show sheets as tabs** in the **Save Workbook** dialog box, the workbook permissions override the permissions on individual views within the workbook until the workbook is saved again without tabs.

- **Revert** discards edits and returns to the last saved version of the workbook.
- **Done** exits the authoring environment.

If you have unsaved changes, you are prompted to save them. If you do not save changes, the unsaved changes are still present when you return to the authoring mode for that view for as long as you remain signed in to the current server session.

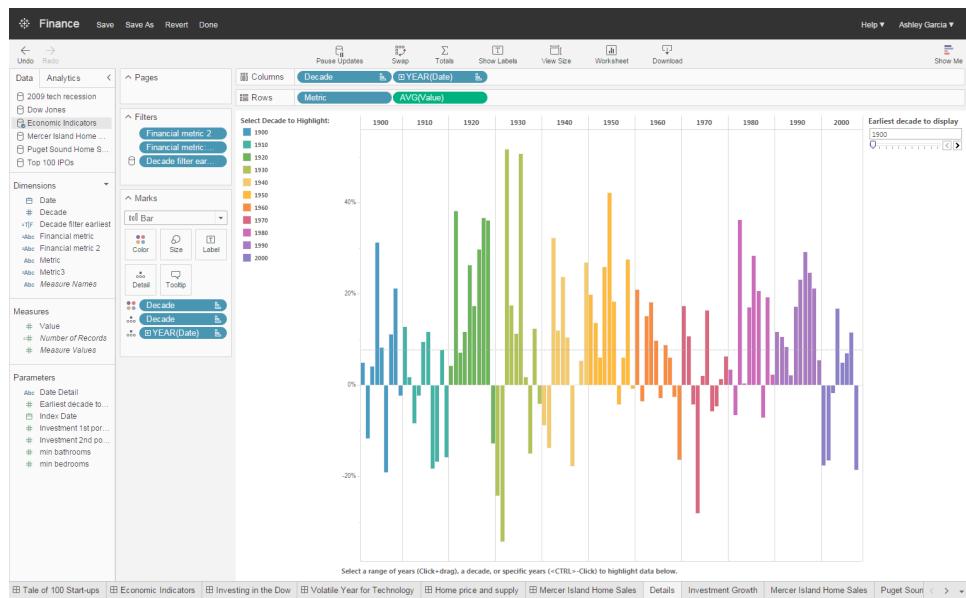
The options available to you to save workbooks depend on the permissions granted by your administrator. For more information, see [Grant Web Edit, Save, and Download Permissions](#) on page 377.

The Web Authoring Workspace

The web authoring environment in Tableau Server is similar to Tableau Desktop. You can edit views and interact with dashboards in your workbook, but not stories.

Worksheets

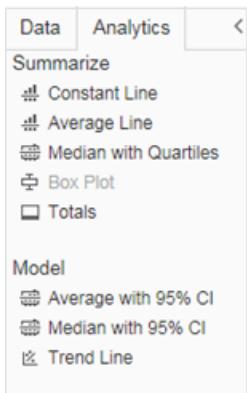
The web authoring workspace for views will look familiar. Like in Tableau Desktop, the Data and Analytics panes appear on the left side of the workspace.



The data pane includes the names of the data sources included in the workbook, and the fields, parameters, and sets included in the active data source.



The Analytics pane includes common analytic features in Tableau. You can drag reference lines, forecasts, trend lines, and other objects into your view from the Analytics pane.



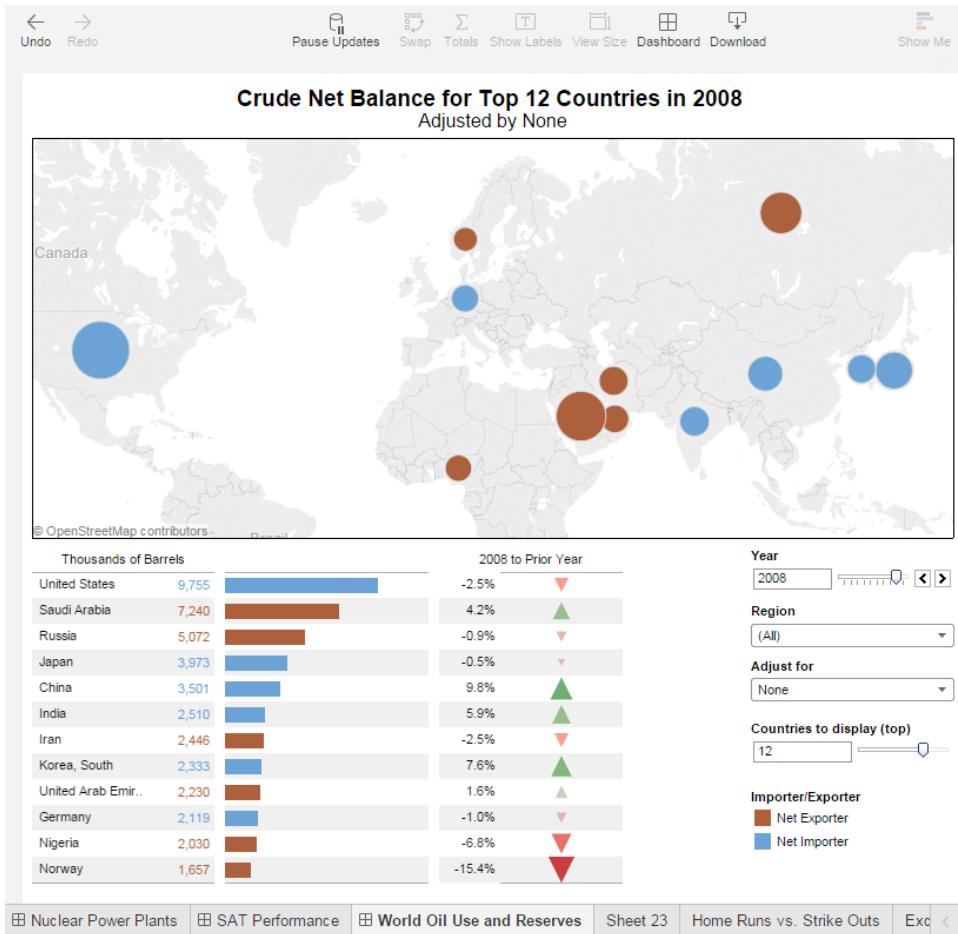
A **toolbar** with options for editing the view is displayed at the top of the workspace. The **Marks** card, and **Pages** and **Filters** shelves are displayed the left of the view. **Columns** and **Rows** for measures and dimensions shelves are displayed above the view. **Tabs** for worksheets and dashboards included in the workbook appear at the bottom of the workspace.



Select the **New Sheet** tab to begin creating a new view.

Dashboards

Select a dashboard tab in your workbook, and then choose **Edit** in the toolbar to begin web authoring. You can interact with dashboards as you would in Tableau Desktop, though full editing functionality is not enabled.



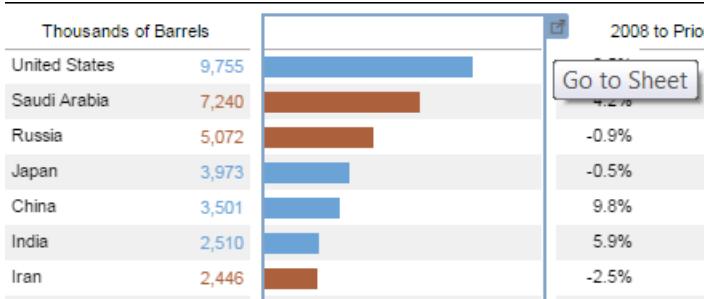
For example, sort data by selecting the sort button next to a view name on your dashboard. Or, select field names to **Keep Only**, **Exclude**, **View Data**, or **Group Members**. These actions will affect associated underlying worksheets.

Thousands of Barrels	2008 to Pr
United States	9,755
Saudi Arabia	7,240
Russia	5,072
Japan	3,973
China	3,501
India	2,510
Iran	2,446
Korea, South	2,333
United Arab Emir..	2,230
Germany	2,119
Nigeria	2,030
Norway	1,657

✓ Keep Only X Exclude

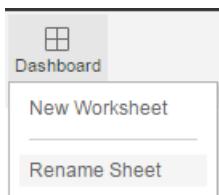
Country / Region: **United States**
Importer/Exporter: **Net Importer**
Imports, Exports, and Balance: **9,755**

To easily update or review views in your workbook, select a view on your dashboard to reveal **Go to Sheet**.



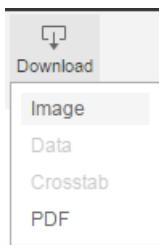
Toolbar Menus

Use the **Dashboard** menu to rename your dashboard or create a new worksheet in your workbook.

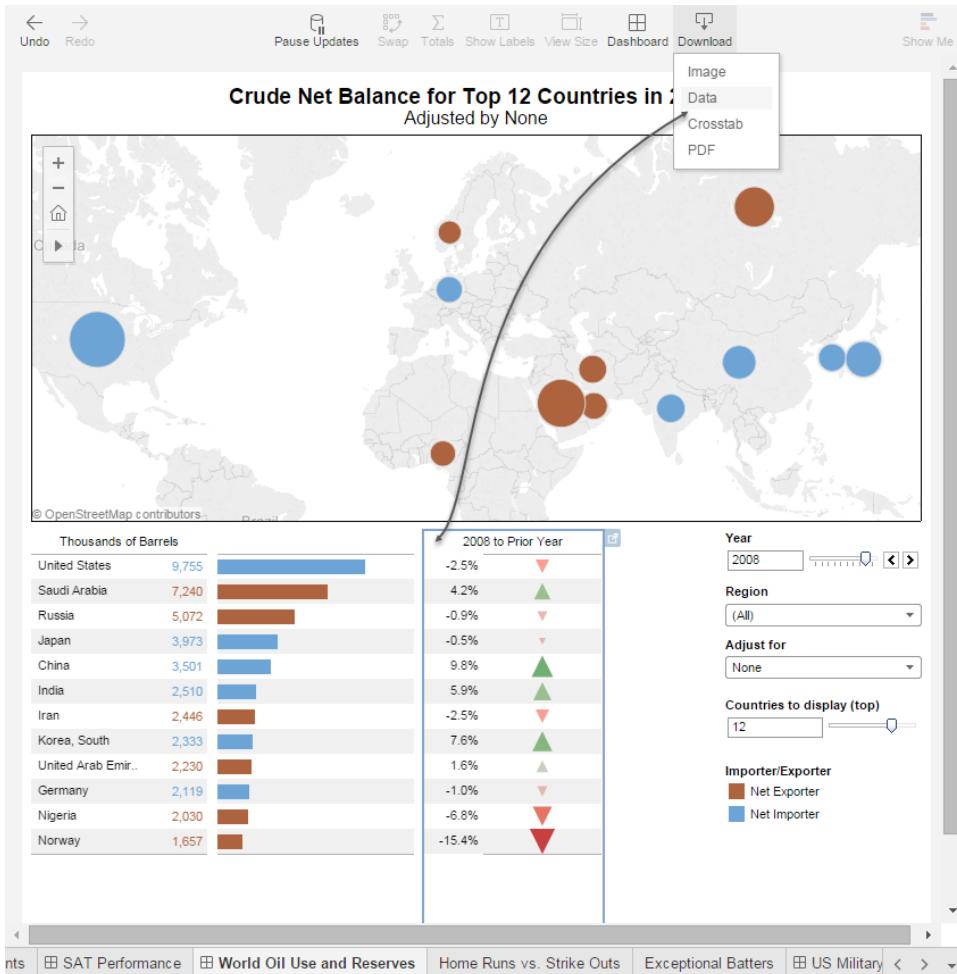


Use the **Download** menu to save your dashboard for offline viewing or to download underlying data.

Select **Download > Image** to save your dashboard as a PNG file. Select **Download > PDF** to save your dashboard as a PDF.



To download a view's underlying data, first select the view of interest on your dashboard. The context-sensitive **Download** menu updates to add options **Data** and **Crosstab**.

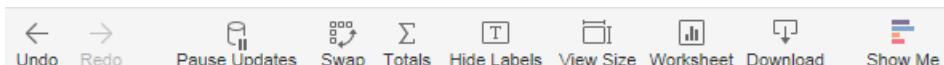


Selecting **Data** opens a new window where you can preview and download both summary and underlying data for the view you've selected.

Select **Crosstab** to download summary data for the view you've selected as a text table (CSV file).

Toolbar

When you are editing a view, you can use the toolbar at the top of the view to perform common actions.



Undo/Redo

Undo and redo an action or series of actions. You can undo or redo almost any type of change in the view by selecting these toolbar buttons.

[Pause Updates](#)

When you place a field on a shelf, Tableau generates the view by querying the data source. If updates seem slow when editing the view, you can pause updates while making a series of edits, then turn them on again.

[Swap](#)

This moves the fields on the Rows shelf to the Columns shelf and vice-versa. Most used with view types that are based on x- and y-axes.

[Totals](#)

You can automatically compute grand totals and subtotals for the data in a view. Select Totals to see four options:

- **Show Column Grand Totals:** Adds a row showing totals for all columns in the view.
- **Show Row Grand Totals:** Adds a column showing totals for all rows in the view.
- **Row Totals to Left:** Moves rows showing totals to the left of a crosstab or view.
- **Column Totals to Top:** Moves columns showing totals to the top of a crosstab or view.
- **Add All Subtotals:** Inserts subtotal rows and columns in the view, if you have multiple dimensions in a column or row.
- **Remove All Subtotals:** Removes subtotal rows or columns.

[Show/Hide Labels](#)

Select to show or hide marks labels in the view.

[View Size](#)

Use the options under **View Size** to change the proportions of your view within the browser window, and go back and forth between seeing details and seeing the whole picture. The Cell Size commands have different effects depending on the type of visualization.

[Worksheet](#)

Contains options for making changes at the worksheet level. Create worksheets, modify sheet names, clear sheet formatting, or clear the entire sheet.

[Download](#)

Use the options under Download to capture parts of your view for use in other applications.

- **Image:** Displays the view, dashboard, or story as an image in a new browser tab.
- **Data:** Displays the data from the view in a new browser window with two tabs:
Summary, showing aggregated data for the fields shown in the view, and **Underlying**, showing underlying data for the selected marks in the visualization. If the new window does not open, you may need to disable your browser's popup blocker.

- **Crosstab:** Saves the underlying data for the selected marks in the visualization to a CSV (comma-separated values) file which can then be opened in Microsoft Excel.
- **PDF:** Opens the current view as a PDF in a new browser window. From there you can save it to a file. If the new window does not open, you may need to disable your browser's popup blocker.

Show Me

Opens a control that shows a range of visualization types that you can use in Tableau. When you display the Show Me list, Tableau uses the data in the current view to determine which visualization types to make available for you to select. Among the available types, it draws a different color outline around the one that it determines is the best match for your data.

You can also hover over a visualization type to see what field types are required to make that visualization type available.

Data Pane

At the top of the Data pane is a list of available data sources for the workbook. If you are editing an existing workbook, there may be multiple data sources. Select a data source to see the dimensions and measures for that data source. If you are creating a new workbook, you see just the data source from which you created the workbook.

All data sources contain fields. These fields appear below the list of data sources in the Data pane. Dimensions and measures always appear, other field types appear if they are present in the data source:

- **Dimensions** are fields that contain discrete qualitative data. Examples of dimensions include dates, customer names, and customer segments.
- **Measures** are fields that contain numerical data that can be aggregated. Examples of measures include sales, profit, number of employees, temperature, frequency, and pressure.
- **Sets** are custom fields that define a subset of data based on some conditions. A set may be based on a computed condition, which updates as the data changes, or a constant list of values. Sets may be present in workbooks that you edit, but you cannot create sets.
- **Parameters** are dynamic values that can replace constant values in calculations, filters, and reference lines. Parameters may be present in workbooks that you edit, but you cannot create parameters.

By default, Tableau treats all relational fields containing numbers as measures. However, you might decide that some of these fields should be treated as dimensions. For example, a field containing ages may be categorized as a measure by default in Tableau because it contains numeric data. However, if you want to look at each individual age rather than an axis you can convert the **Age** field to a dimension. To do this, drag the **Age** measure and drop it into the Dimensions area in the Data pane. Now if you drag the **Age** field to the **Rows** or **Columns** shelf it will create column headers (1, 2, 3, etc.) instead of a continuous axis.

To build visualizations, you drag fields from the Data pane to the **Rows** and **Columns** shelves, the Marks card, or one of the other available shelves. For a demonstration, see [Build a View on page 673](#).

Manage Fields in the Data Pane

Right-click a field in the Data pane to see a set of options for modifying that field. You can:

- Convert a dimension to a measure or vice-versa.
- Convert a discrete field to continuous or vice-versa. This option is available for measures and date dimensions.
- Change the data type for a field.
- Change the geographic role for a field.
- Change the default aggregation for a measure.

Data Blending

If you upload a workbook that uses blended data sources, you can see a link next to the field or fields in the primary data source that are being used to link the two data sources:

The screenshot shows the Tableau Data pane. At the top, there are tabs for 'Data' and 'Analytics'. Below the tabs, there are two data sources listed: 'Sales Target' (highlighted with a gray background) and 'Sample - Superstore'. Under the 'Dimensions' section, there is a list of fields: 'Category' (with a linking icon), 'Order Date' (with a linking icon), 'Segment' (with a linking icon), and 'Measure Names' (without a linking icon). The linking icons are small orange circles with a double-headed arrow inside.

You can click on the link icon to activate or deactivate specific fields. When fields that *can* be used as linking fields are *not* being used, the link icon changes appearance:

The screenshot shows the 'Dimensions' and 'Measures' sections of the Tableau Data Source pane. In the Dimensions section, 'Order Date' is selected and highlighted with a blue border. Other dimensions listed are Category, Segment, and Measure Names. In the Measures section, three measures are listed: Number of Records, Sales Target, and Measure Values.

Dimensions
Abc Category
Order Date
Abc Segment
Abc Measure Names

Measures
Number of Records
Sales Target
Measure Values

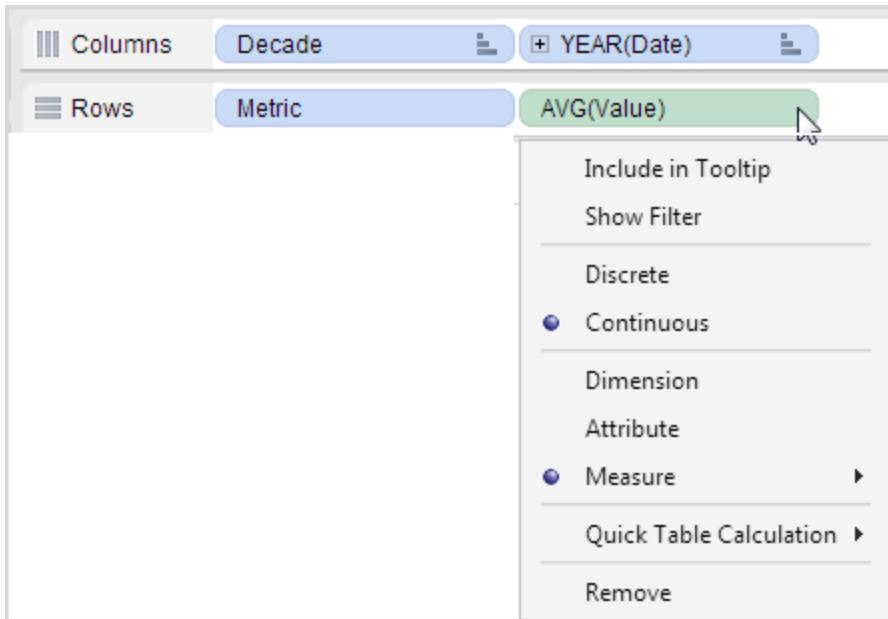
Data sources are linked using a left join, where the view is using all data rows from the primary data source but only those data rows from the secondary data source that have values for fields that are in the view or for fields that are designated as linking fields. So changing the linking field, or designating multiple linking fields, can actually pull in different or additional data rows from the secondary data source, thereby changing the values returned by aggregations.

Columns and Rows Shelves

Drag fields to the Columns shelf to create the columns of a table, or to the Rows shelf to create the rows of a table. You can drag multiple fields to either shelf.

Discrete values (typically, dimensions) are displayed in blue on the Columns and Row shelves; continuous values (typically, measures) are displayed in green.

At the right end of any field you place on the Columns or Rows shelf is a drop down menu that you can use to configure the dimension or measure:



The options that are available depend on the type of field. The complete list of options includes:

- **Include in Tooltip**

By default, all fields on the Columns and Rows shelf are included in the tooltips that appear when you move your mouse over one or more marks in the view. Un-check this option to remove a field from tooltips.

- **Show Filter**

Choose this option to add a filter for this field to the view. Users will then be able to specify which data to include and exclude for this dimension or measure.

- **Discrete/Continuous**

Use these options to convert a continuous range of values into a set of discrete values, or a discrete set into a continuous range.

- **Dimension/Attribute/Measure**

Use this range of options to convert a dimension to a measure or a measure to a dimension.

You can also define the option as an Attribute, which returns the value of the given expression if it only has a single value for all rows in the group, and otherwise displays an asterisk (*) character. Null values are ignored.

- **Quick Table Calculation**

Provides a set of options for redefining the meaning of the marks for the value.

- **Remove**

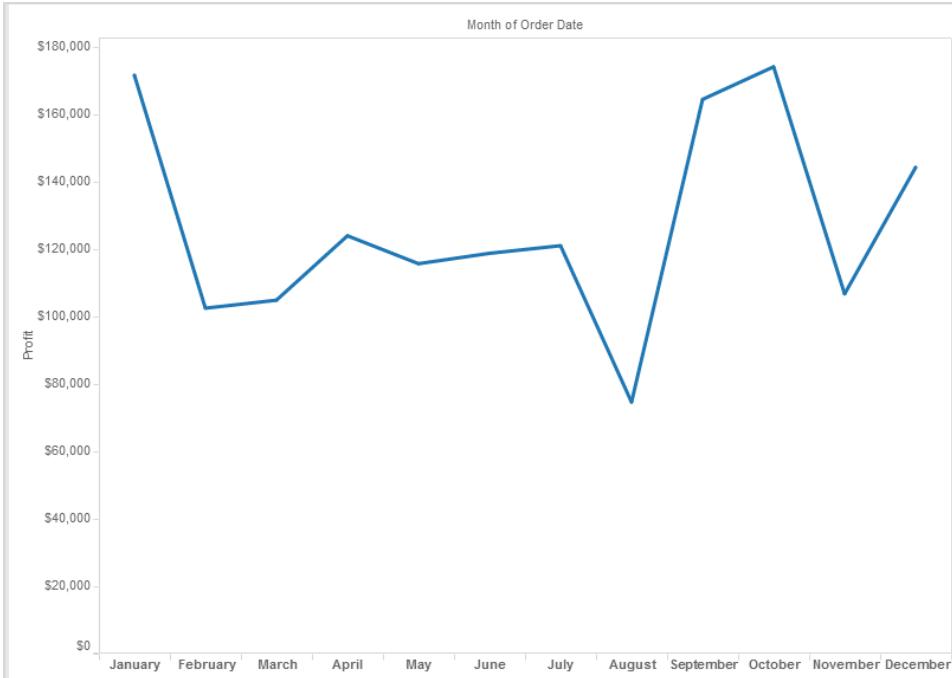
Removes the value from the Columns or Rows shelf.

Options for Date Dimension

An additional set of options is available for date dimensions:

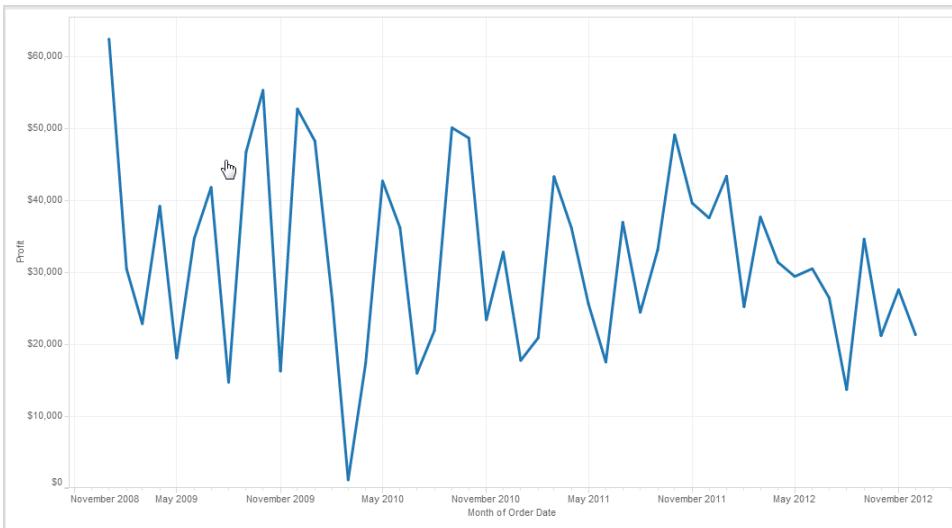
<input checked="" type="radio"/> Year	2011
Quarter	Q2
Month	May
Day	8
More	▶
Year	2011
Quarter	Q2 2011
Month	May 2011
Week Number	Week 5, 2011
Day	May 8, 2011
More	▶

Choose one of the options from the upper group to define the granularity of the data as discrete values. For example, if you choose **Month** your view will combine the data for each named month in your data across the full range of years:



There are exactly twelve marks in the data--one for each month. The November mark combines the data from November 2008, November 2009, etc.

Choose one of the options from the lower group to define the granularity of the data as continuous values. For example, if you choose Month your view will show your data sequentially, over the range of available months.



In this case there are 48 marks in the data--one for each month since November 2008.

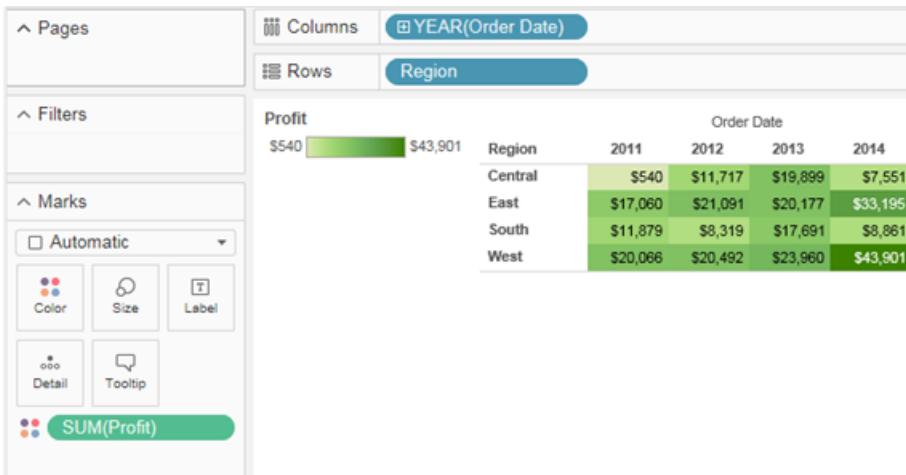
Marks

When you drag fields to the view, the data are displayed using marks. Each mark represents the intersection of all of the dimensions in the view. For example, in a view with **Region** and **Year** dimensions, there is a mark for every combination of those two dimensions (East 2011, East 2012, West 2011, West 2012, etc.).

The screenshot shows a Tableau view with the following structure:

		Order Date			
Region		2011	2012	2013	2014
Central		Abc	Abc	Abc	Abc
East		Abc	Abc	Abc	Abc
South		Abc	Abc	Abc	Abc
West		Abc	Abc	Abc	Abc

Marks can be displayed in many different ways including lines, shapes, bars, maps, and so on. You can show additional information about the data using mark properties such as color, size, shape, labels, etc. The type of mark you use and the mark properties are controlled by the Marks card. Drag fields to the Marks card to show more data. For example, the same view above is shown again below but this time with **Profit** on Color. With this additional information, it is clear that the West region had the highest profit in 2014.

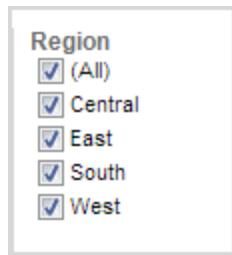


Control the marks in the view using the Marks card. Use the drop-down menu to specify the type of mark to show. Drag fields to the Marks card and use the drop-down controls to add more information to the view and control the color, shape, size, labels, and number of marks in the view.

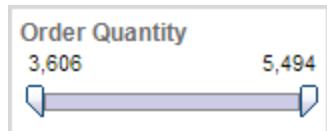
Filters Shelf

Use the Filters shelf to specify which data to include and exclude for a dimension or measure. For example, you might want to analyze the profit for each customer segment, but only for certain shipping containers and delivery times. By placing the Container dimension on the Filters shelf you can specify which containers to include. Similarly, you can put the Delivery Date field on the Filters shelf to define which delivery times to include.

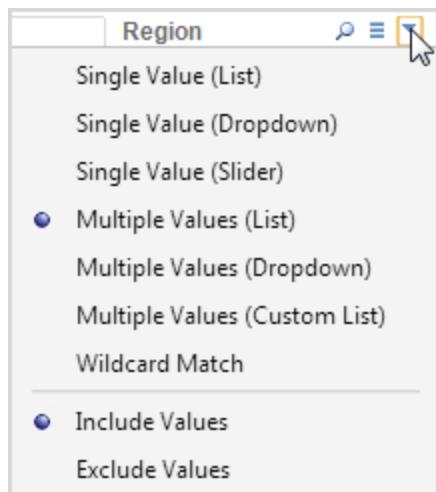
When you drag a dimension or measure to the Filters shelf, Tableau automatically inserts a filter control into the view for selecting the values to display. For example:



For dimensions, the filter control shows discrete values, as above. For measures, the control shows a continuous range:



Hover your mouse to the right of the title for the filter control to specify how values in the control are to be displayed:

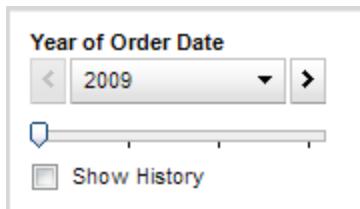


Pages Shelf

Drag a dimension or measure to the Pages Shelf to break a view into a series of pages so you can better analyze how a specific field affects the rest of the data in a view. Dragging a dimension to the Pages shelf is like adding a new row for each member in the dimension.

Dragging a measure to the Pages shelf automatically converts the measure into a discrete measure that can be broken into individual pages.

When you drag a dimension or measure to the Pages shelf, Tableau automatically inserts a control into the view to let you navigate the pages in your view. For example:



You can manually advance through the sequence of pages in any of the following ways:

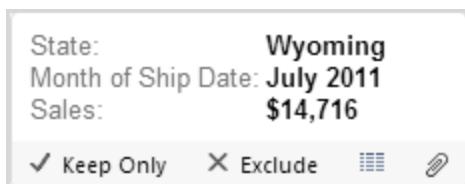
- Use the drop-down menu to select a value.
- Use the forward and back buttons on either side of the drop-down list to navigate through the pages one at a time.
- Use the Page slider to quickly scroll forward and backward in the sequence of pages.

Select **Show History** to show marks from previous pages in addition to marks for the current page.

Toolips

Place your cursor over a mark in the view to see the tooltip for that mark.

Toolips provide information on the values of dimensions and measures for the selected mark:



Toolips also provide these options:

- **Keep Only**
Exclude all marks from the view except this one.
- **Exclude**

Exclude this mark only.

- **Group Members**

Choose the paperclip icon to create a new group, which is a dimension, from the selected mark. Typically, you would select multiple marks and then create a group. For example, if you have a dimension Region with values North, South, East and West, you could select South and West and then create a group from them.

- **View Data**

Choose the table icon to open a new browser window to display two tabs: **Summary**, which shows only data for the current mark, and **Underlying**, which shows data for the entire view.