

# **Tableau Server Administrator Guide**



# Table of Contents

Before you install...	1
Installing and Configuring	3
Upgrading	25
Distributed Environments	37
Working with the Server	61
Security	173
Embedding Views	181
Proxy Servers	207
Trusted Authentication	209
Run As User	221
SQL Server Impersonation	229
TCP/IP Ports	237
tabcmd	245
tabadmin	267
Database Maintenance	275
Troubleshooting	279
Copyright	289
Contact	291



# Before you install...

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

- **Supported operating systems**—You can install Tableau Server on Windows Server 2003 (SP1 or higher), Windows Server 2008, Windows Server 2008 R2, Windows Vista, or Windows 7. Although Tableau Server performs well on 32-bit operating systems, 64-bit editions are recommended.
- **Memory, cores, and disk space**—Tableau Server system requirements vary based on the number of users you plan to support:

<i>Deployment Type</i>	<i># Server Users</i>	<i>CPU</i>	<i>RAM</i>
Evaluation	1-2	2-core	4 GB
Small	<25	4-core	8 GB
Medium	<100	8-core	32 GB
Enterprise	>100	16-core	32 GB or more

- **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](#) and [SQL Server Impersonation](#).
- **IIS and port 80**—If you want to use the default port, port 80, the computer can't be running Internet Information Services (IIS). You can modify the gateway port number to avoid conflict with IIS. See [TCP/IP Ports](#).

## Configuration Information

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

Option	Description	Your Information
Server Account	The server must have a user account that the service 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.	Username: 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 <a href="#">fully-qualified domain name</a> .	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.	___ - Yes ___ - No

## Ports

By default Tableau Server requires several TCP/IP ports to be available to the server. See the topic [TCP/IP Ports](#) 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 [Editing the Default Ports](#) to learn how.

## Drivers

You may need to install additional database drivers. Download drivers from [www.tableausoftware.com/support/drivers](http://www.tableausoftware.com/support/drivers).

# Installing and Configuring

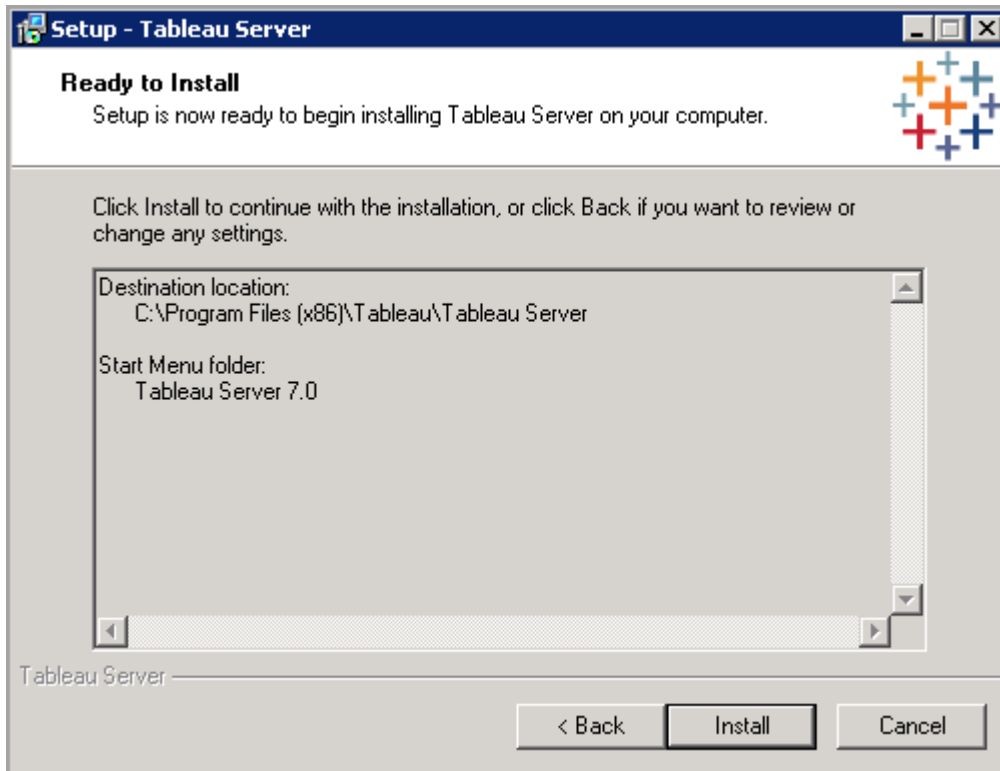
Here are the main steps you need to take to install and configure Tableau Server:

- [Running Server Setup](#)
- [Activating Tableau](#)
- [Configuring the Server](#)
- [SSL](#)
- [Domains](#)
- [Reconfiguring](#)

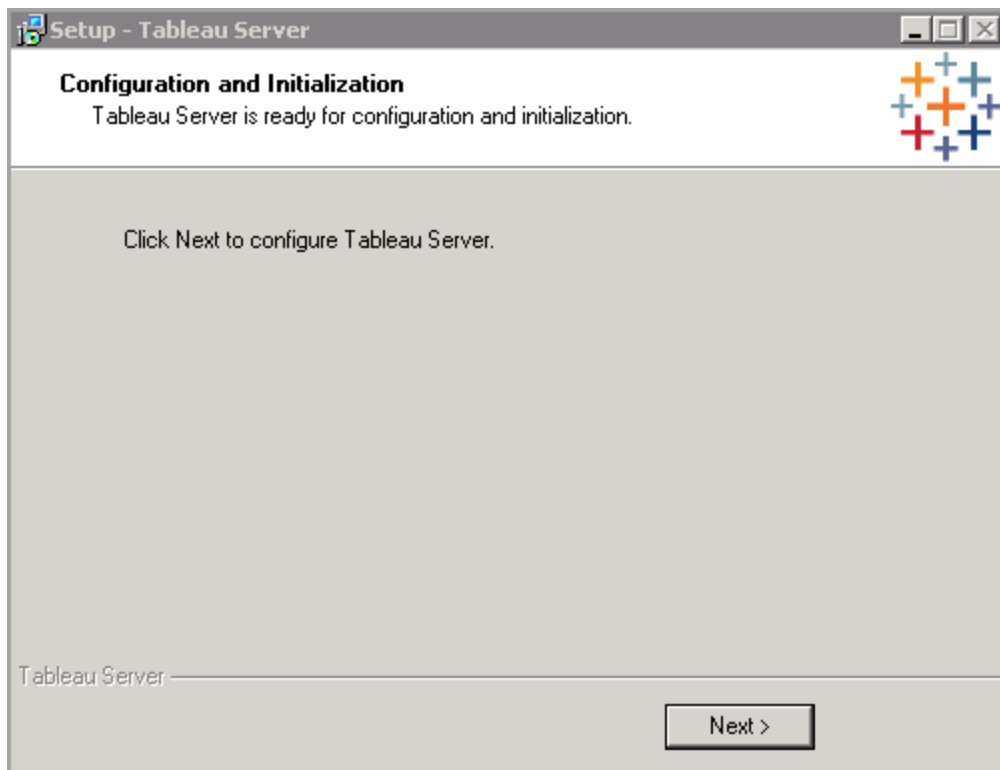
## Running Server Setup

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.



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



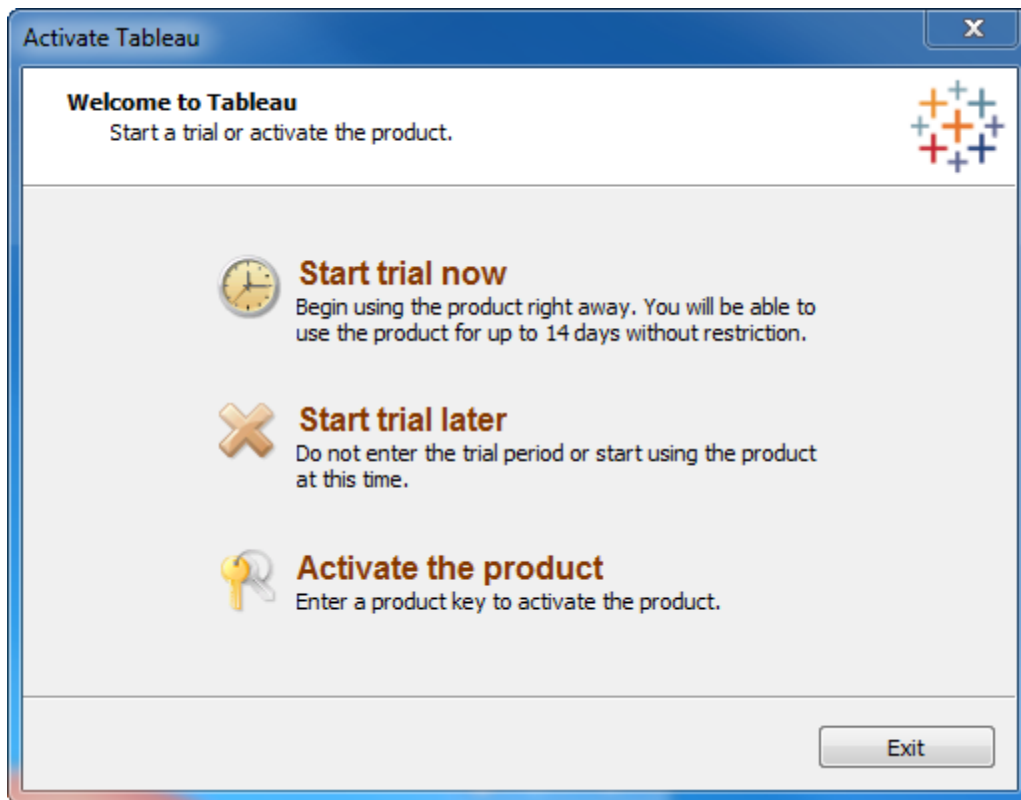
If you need to support characters that are not the Latin-1 set, install the Windows Language Packs via Control Panel > Regional and Language Options. The language packs will need to be installed on the primary server as well as any worker machines.



## Activating 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 [Activating Tableau Offline](#).

1. Select Activate and paste in your product key.

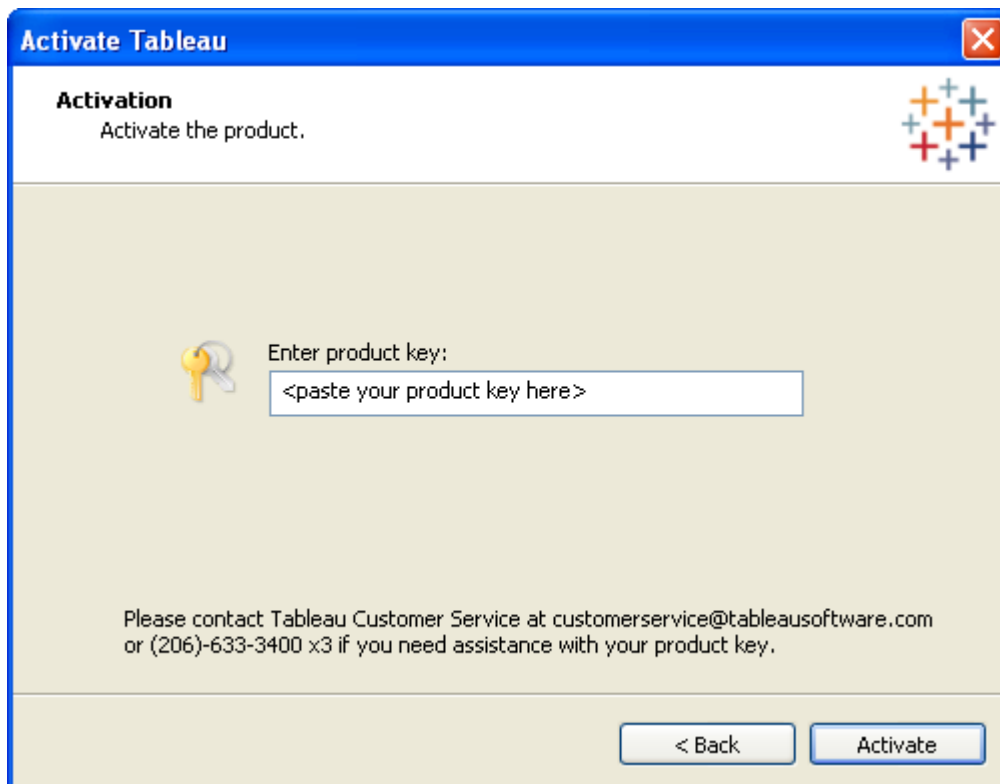


2. Refer to the [download help page](#) on the web site for step-by-step instructions.
- [Activating Tableau Offline](#)

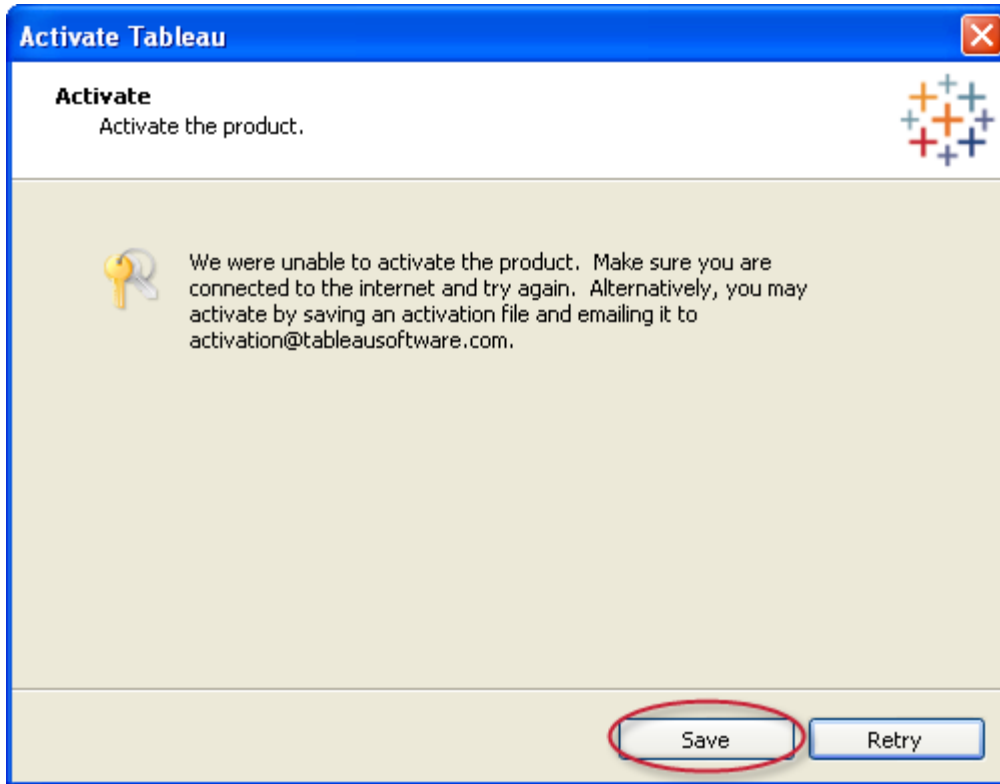
## Activating Tableau Offline

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

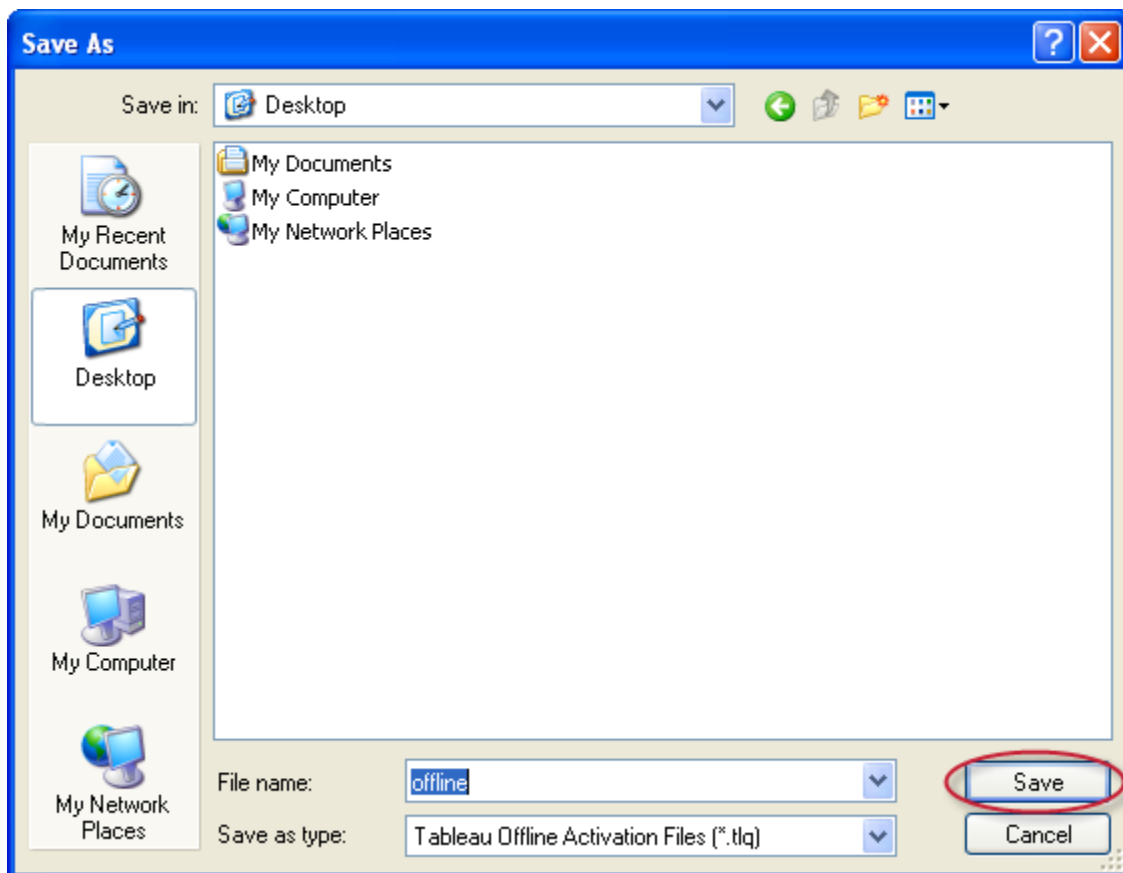
1. When the product key manager opens click Activate the product.
2. Paste your server product key into the corresponding text box and click Activate. You can get your product key from the [Customer Account Center](#) on Tableau's web site.



3. 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.

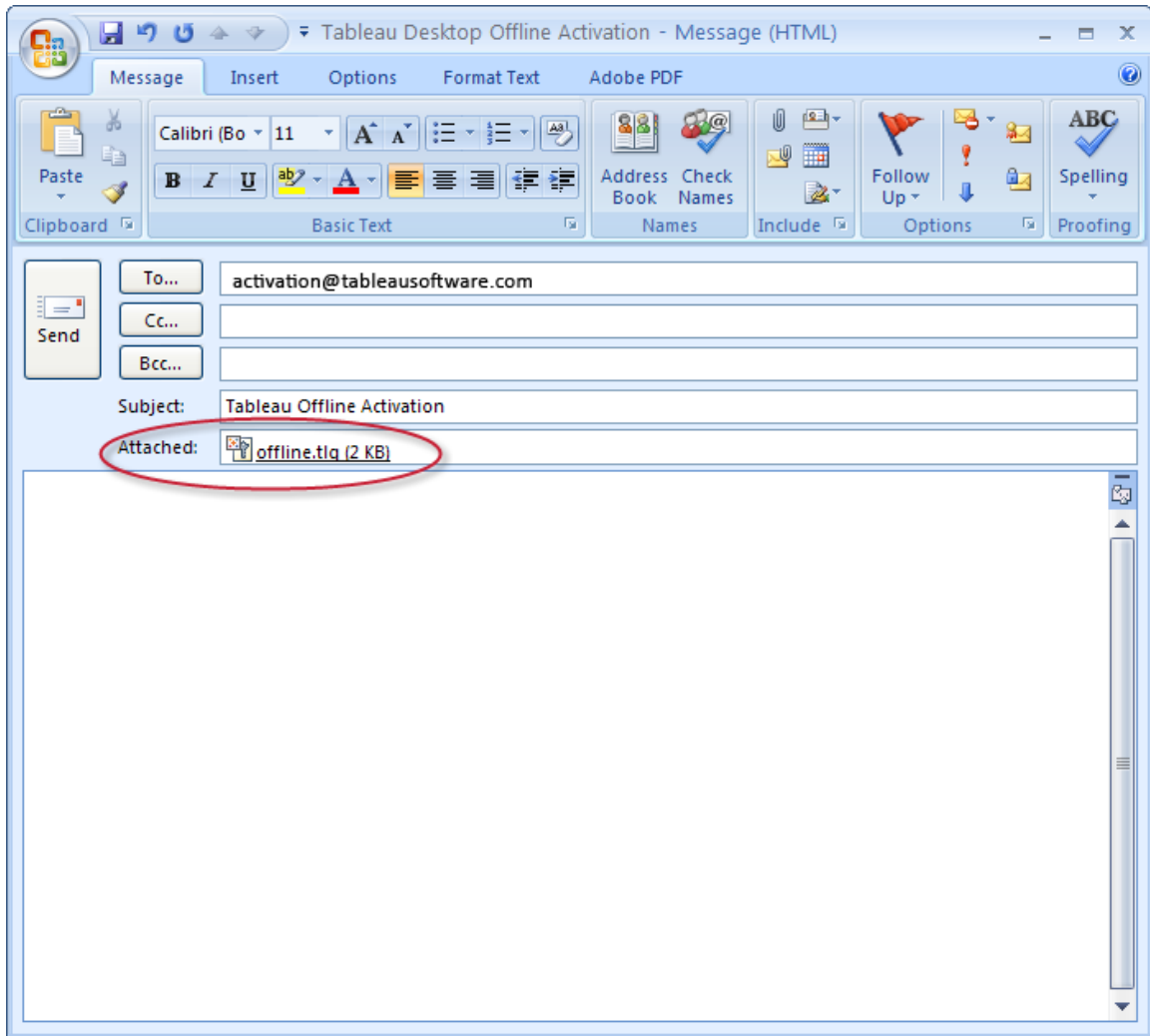


4. Select a location for the file and click Save. The file is saved as *offline.tlq*.



5. Back in Tableau click Exit to close the Activation dialog box.

6. Move the file to a computer that is online and open an email editor. Create a new email to *activation@tableausoftware.com*. Attach the file to the email and click Send.



7. Tableau will email you a file called activation.tlf. Move this file to the computer where you are installing Tableau Server. If you have Tableau Desktop installed on the computer you can then double-click the new file to complete activation. If you do not have Tableau Desktop installed continue to steps 8 and 9.
8. 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 (x86)\Tableau\Tableau Server\7.0\bin"
```

9. Next, type `tabadmin activate --tlf <path>\activation.tlf`, where `<path>` is the location of the response file Tableau emailed to you. For example:

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

If you need additional assistance, contact Tableau at [activation@tableausoftware.com](mailto:activation@tableausoftware.com).

## Configuring the Server

When you configure Tableau Server, your main tasks are to enter the account you want the server to run under, choose how users will be authenticated when they log into the server, specify a few universal aspects of data connections, and set up the Tableau Server administrator account. For procedures on how to do this, see the topics below:

- [Configuring General Settings](#)
- [Configuring Caching](#)
- [Disabling Initial SQL Usage](#)
- [Adding an Administrator Account](#)

### Configuring General Settings

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](#) 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:  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.

Use Active Directory

Use Local Authentication

Active Directory

Domain:

Nickname:

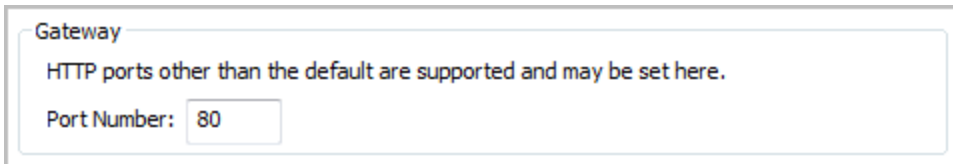
Enable Automatic Login

**If you use Active Directory:**

- You can optionally Enable Automatic Login, which uses Microsoft SSPI to automatically log in your users based on their Windows username and password. If you select this setting, you cannot also select [Enable Guest](#) later. Also, do not select Enable Automatic Login if you plan to configure Tableau Server for [trusted authentication](#).
- 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 log in, they will need to use the fully qualified domain name (for example, `myco.lan\jsmith`). On subsequent logins, they can use the nickname (`myco\jsmith`).

3. 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.

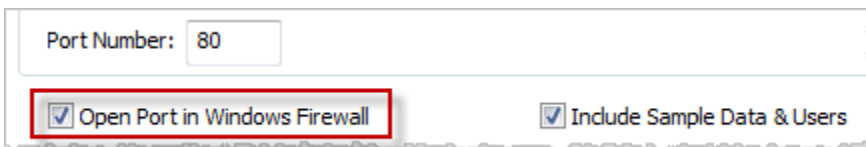


Gateway

HTTP ports other than the default are supported and may be set here.

Port Number: 80

4. 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.



Port Number: 80

Open Port in Windows Firewall  Include Sample Data & Users



5. Select whether to include sample data and users. The sample data can help you get familiar with Tableau Server, especially if you are installing a trial version of the product. Initially the sample user uses one interactor license. You can change this user to unlicensed in order to reclaim the license levels. See [Licenses](#) to learn how. If you select to include the sample user, a single user is installed. The username and password are shown below:

Username	Password
Tableau Software	test

6. Optionally continue to the next page to configure Caching and Initial SQL options. If you do not want to configure these options click OK.
- [About Enable Guest & Enable Automatic Login](#)

## About Enable Guest & Enable Automatic Login

This topic provides some background on the Enable Automatic Login and Enable Guest settings and why they must not be used together.

Enable Automatic Login is an option you can select during Setup. It uses Active Directory and NTLM to authenticate Tableau Server users and automatically logs them into the server when they click a link for a view.

Enable Guest is a setting on the Maintenance page that can be selected if you have a core-based server license. It has the same result as Enable Automatic Login—users click a link and they go directly to the view with no login—but unlike Enable Automatic Login, no authentication is performed. The Tableau Server Guest User account is used to access the server, but as long as Enable Guest is selected, anyone can use it. Administrators often limit the capabilities of the Guest User account. For example, they might edit the permissions of certain views so that Guest User is denied access.

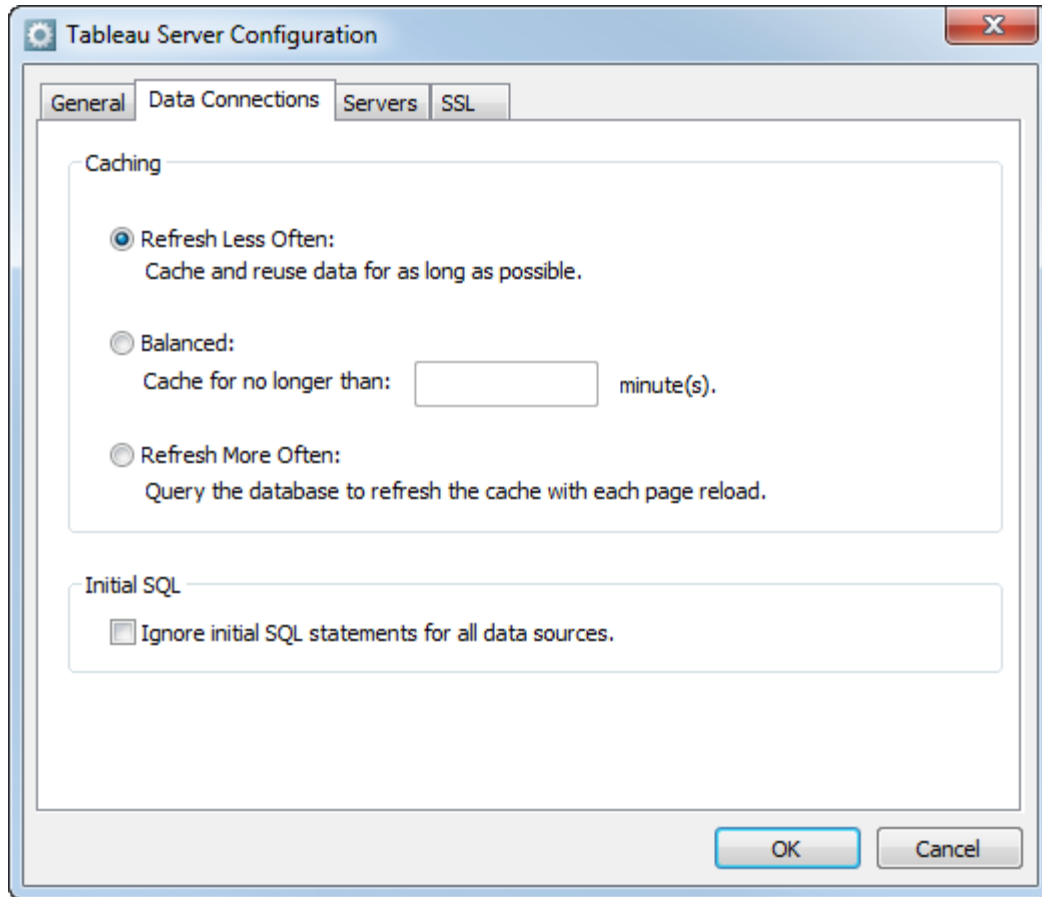
Enable Automatic Login and Enable Guest are not supported or recommended as a combination. Selecting the first during Setup causes the second to become grayed out. In rare situations, however, both settings can be enabled. For example, if you don't select Enable Automatic Login during Setup, you can later select Enable Guest, then return to the Configuration utility and select Enable Automatic Login.

If the above happens, one symptom you may notice is that server users may have full access to a view, then after their sessions time out, they are denied access. This happens because the first automatic login and access level are based on the individual server user's Active Directory credentials, but the second (post-timeout) is based on Guest User's—and that account has a lower access level.

For more information on this topic, including workarounds for the above situation, see the [Tableau Knowledge Base](#).

## Configuring 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.

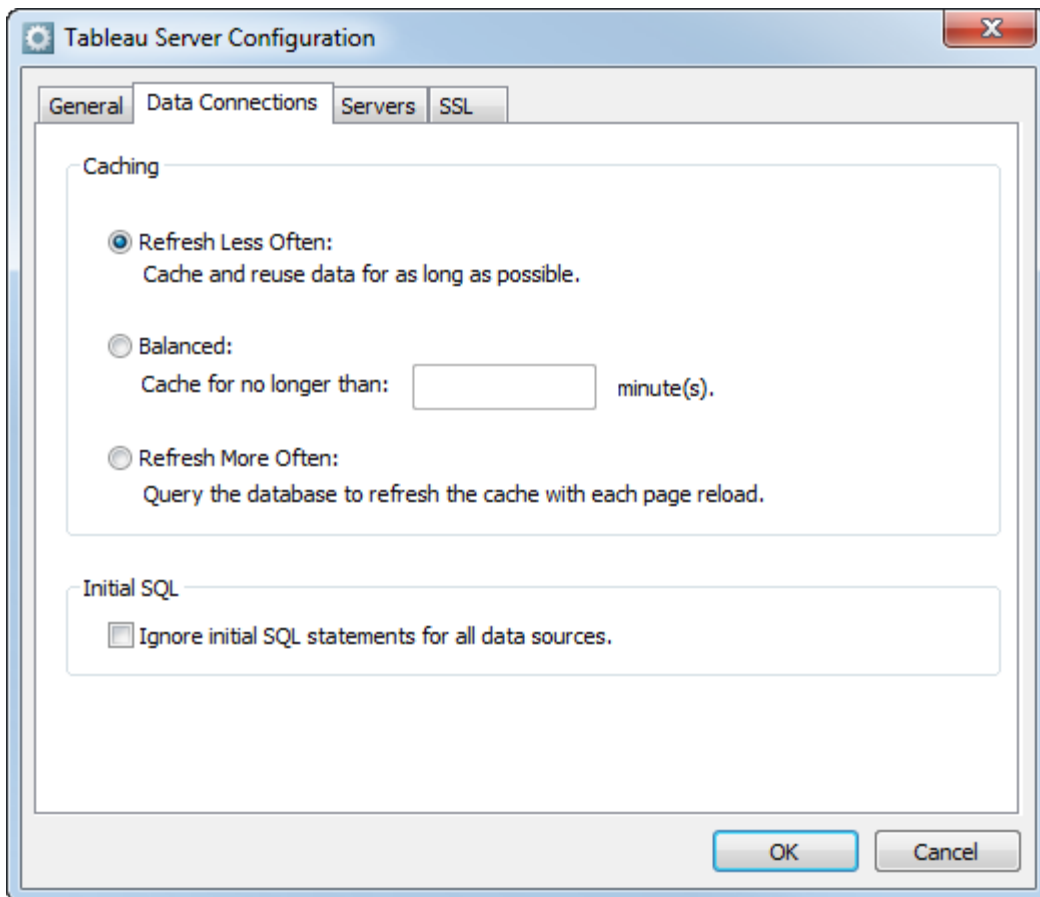
- **Refresh More Often**—The database is queried each time the page is loaded. The data is still cached and will be reused until the user reloads the page. This option will ensure users see the most up to date data; however, it may decrease performance.

**Note:**

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.

## Disabling Initial SQL Usage

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:



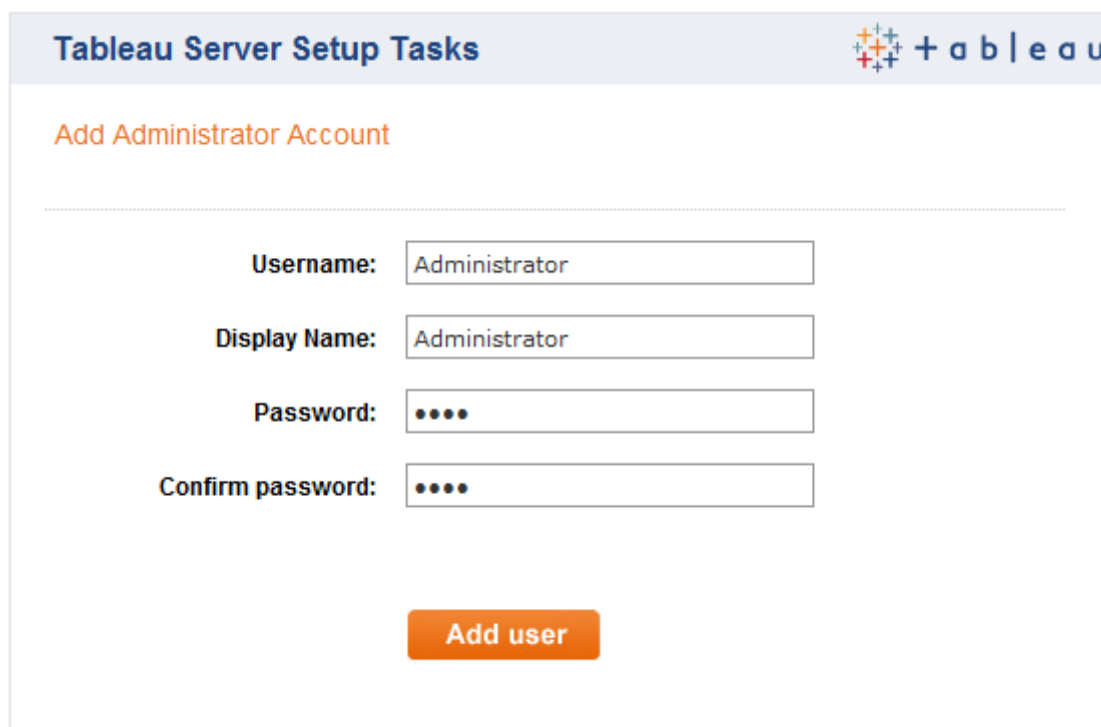
To disable initial SQL functionality, select the Ignore initial SQL statements for all data sources checkbox. Workbooks created with initial SQL statements will still open but the initial SQL commands will not be sent.

## Adding 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 the 'Tableau Server Setup Tasks' interface. At the top right is the Tableau logo. Below the header, the title 'Add Administrator Account' is displayed in orange. A horizontal dotted line separates the title from the form fields. The form contains four input fields: 'Username' with the value 'Administrator', 'Display Name' with the value 'Administrator', 'Password' with four black dots, and 'Confirm password' with four black dots. Below the fields is an 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.

## SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all HTTP traffic. Setting up SSL ensures that access to the web application 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.

- [Configuring SSL](#)

### Configuring SSL

Follow the steps below to configure Tableau Server to use SSL.

1. Acquire an Apache SSL certificate from a trusted authority (e.g., Verisign, Thawte, Comodo, GoDaddy, etc.). 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.

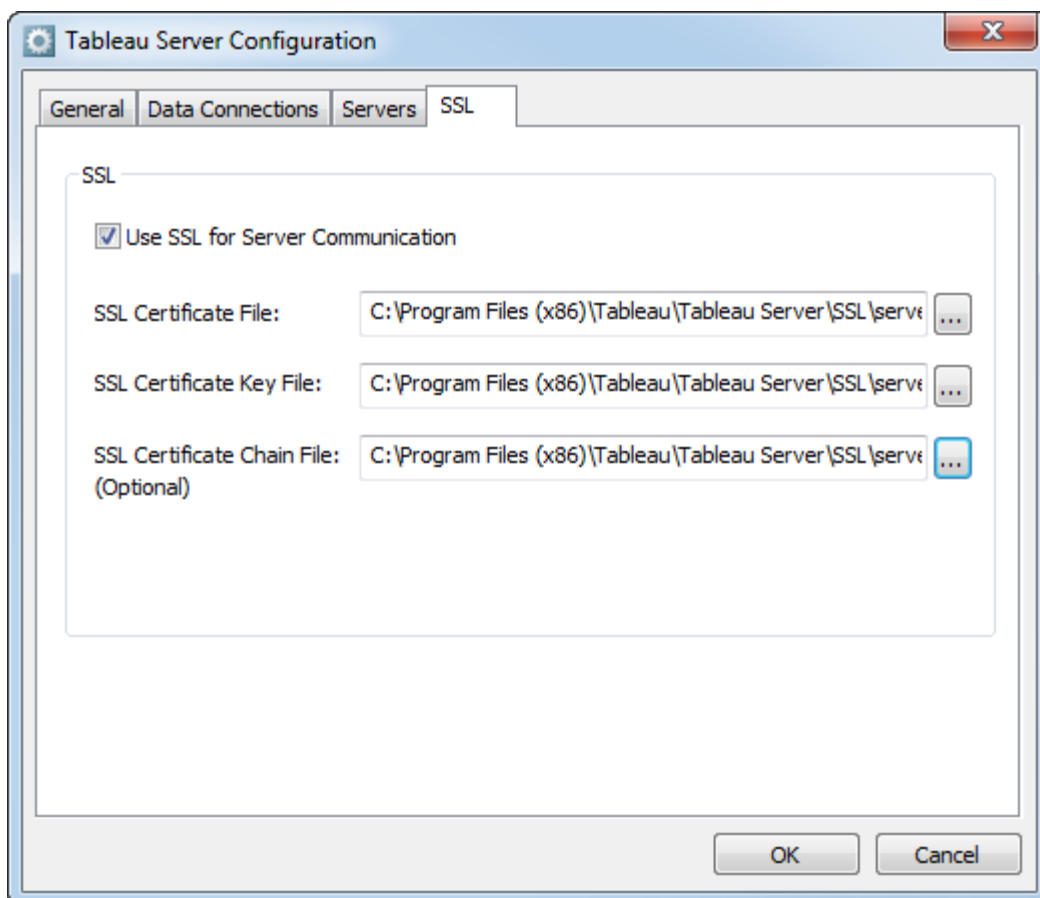
2. Place the certificate files in a folder named SSL, parallel to the Tableau Server 7.0 folder. For example:

```
C:\Program Files (x86)\Tableau\Tableau Server\SSL
```

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

3. Open the Tableau Server Configuration Utility by selecting Start > All Programs > Tableau Server 7.0 > 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 is not password protected with the file extension .key

- SSL Certificate Chain File (Optional)—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. 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. 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
```

**Note:**

Tableau Server only supports port 443 as the secure port. It cannot run on a machine where any other application is using port 443.

## Domains

When you are using Active Directory authentication for the server you can view a list of the domains that are being used and edit their domain names and nicknames. You may need to do this, for example, to ensure that Tableau Server is using the correct nickname for SSPI authentication, or the correct domain name.

- [Modifying Domain Names](#)

### Modifying Domain Names

To modify a domain name:

1. Select the Users link in the Administration area on the left side of the page.
2. Click the Domains link at the bottom of the list of users. The list of domains shows the number of users and groups that have been added to the server from each domain.
3. To display a list of users who are part of a domain, click the domain name.
4. To modify the domain name or nickname, click the Edit link, type a new, fully qualified domain name or a nickname, then click Modify.

Change?	Attribute	New Value
<input checked="" type="checkbox"/>	Fully Qualified Domain Name	domain.lan
<input checked="" type="checkbox"/>	Nickname	domain

Modify Cancel

**Note:**

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 log in. However, if the user name that you are currently logged in with exists in both the current domain and the new domain you can modify the full name for the current domain.

## Reconfiguring

Entering your Tableau Server configuration settings is part of Setup, but you can open the Configuration dialog box after Setup to make changes. See the steps below for details. You can also use the [tabadmin](#) command line tool to make configuration changes. Regardless of how you make the change, the new settings are written to the configuration file `tabsvc.yml`, which is located in the `config` directory.

**Note:** *You cannot switch between Active Directory and Local Authentication. These options can only be configured during Setup.*

To change a setting in the Tableau Server Configuration dialog box, do the following:

1. Stop the server by selecting All Programs > Tableau Server 7.0 > Stop Tableau Server on the Windows Start menu.
2. Next, 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 7.0 > Start Tableau Server on the Windows Start menu.



# Upgrading

Use the following topics to upgrade your Tableau Server software to version 7.0.x. If you are upgrading from a version earlier than 6.x, please contact Tableau Customer Support at [support@tableausoftware.com](mailto:support@tableausoftware.com).

- [Pre-Upgrade Checklist](#)
- [Upgrading to 7.0](#)
- [Migrating to New Hardware](#)

## Pre-Upgrade Checklist

Here are items you should locate and steps you should perform before you upgrade Tableau Server to version 7.0.x.

### Credentials, Setup Files, and Customizations

Before you upgrade, make sure you have the following:

- **User account credentials:** For each machine 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.

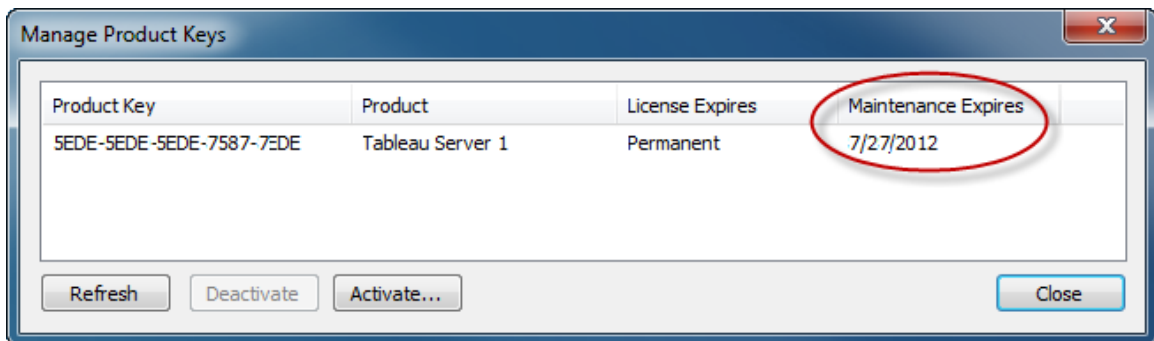
While Tableau retains configuration settings during an upgrade, it's a best practice to also note any customizations you've made so that you can verify them later. These include [configuring SSL](#), changing Tableau's default [port](#) and [time out](#) values, as well as using [custom logos](#).

## 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, contact [Tableau Customer Support](#). Reactivating the product key will be part of Setup. See [Activating Tableau](#) for details. If your server doesn’t have internet access, refer to [Activating Tableau Offline](#).

## Create a “Clean” Backup

In addition to your regular Tableau Server backups, it’s a best practice to create a backup just prior to upgrading. Before you create the backup, run the tabadmin cleanup command to remove non-essential files from your backup. See [Running Cleanup](#) and [Backing Up the Database](#) for steps.

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

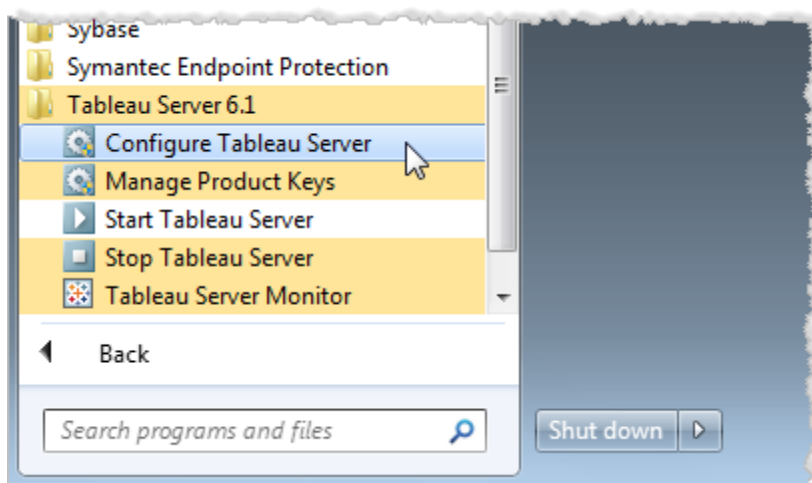
The Tableau backup file (.tsbak) includes configuration information as well as data. Therefore, a backup of a distributed installation of Tableau Server will include configuration information about the workers, including their IP addresses. If you don’t want this information as part of your backup (for example, because you are migrating workers 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, or plan on using the [--no-config](#) option when you

restore the backup file to your new installation. Note that with the latter option, no configuration information is restored—including the primary Tableau Server's.

If you are running a distributed installation of Tableau Server and have a worker running Windows XP, you must remove it from the configuration before upgrading. Windows XP is not a supported platform in version 7.0.

To delete a worker from your Tableau Server configuration:

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



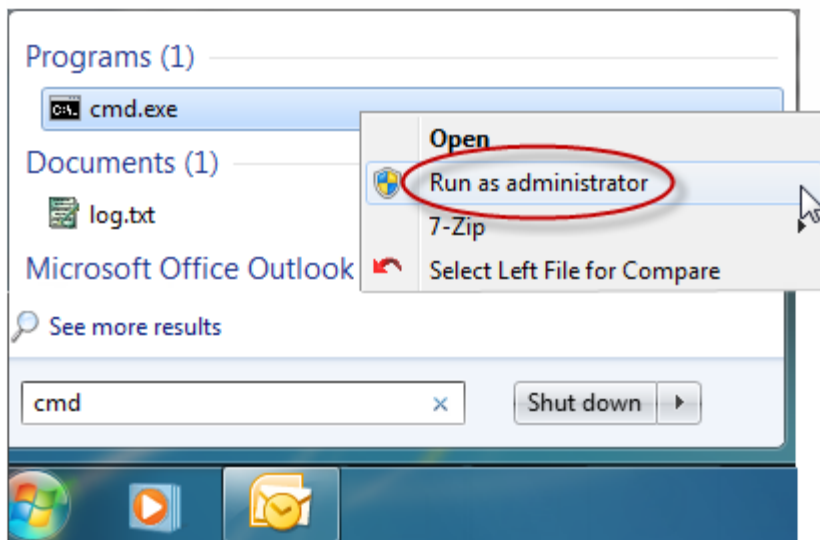
3. In the Configuration dialog box, select the Servers tab.
4. If the worker is hosting extracts and/or the repository, move those services onto another machine. See [Moving the Extract and Repository Services](#) for steps.
5. Next, highlight the worker and click Delete.
6. Click OK.
7. Start the server.

## 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.

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 (x86)\Tableau\Tableau Server\6.1\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 [Backing Up the Database](#) for steps.

**Distributed installations only:** If you removed workers from your server configuration prior to creating your backup and you are upgrading from 7.0.x to 7.0.x, you can now add the workers back to your configuration. Upgrading the primary Tableau Server will push out updates to the workers. Otherwise, if you are upgrading from version 6.x to 7.0.x, leave the workers off the configuration. See [Upgrading to 7.0](#) for details.

## Upgrading to 7.0

After you've completed the [Pre-Upgrade Checklist](#), upgrade your existing Tableau Server to version 7.0 by following the procedure below. If you are migrating to new hardware as part of your upgrade, refer to [Migrating to New Hardware](#) instead.

1. Use Add/Remove Programs on your Tableau Server (or primary Tableau Server, if you have a distributed installation), to uninstall the earlier version.

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

2. **Version 6.x distributed installations**—If you are upgrading a version 6.x worker that is hosting data extracts, refer to [Upgrading a Version 6.x Data Extract Worker to Version 7.0](#) then return to step 3, below. For version 6.x worker servers that are not hosting data extracts, do the following:
  - Use Add/Remove Programs to uninstall the worker software.
  - Delete the Tableau folders under Program Files and ProgramData.
  - Run version 7.0 of the Tableau Server Worker Software installer on all machines that you want to add to the Tableau Server cluster. See [Installing Worker Servers](#) for steps.

**Version 7.0 distributed installations**—If you are upgrading from 7.0.x to 7.0.x, there is no need to uninstall your worker software. Installing the new version on your primary Tableau Server (step 3, below) will push updates out to the workers.

3. [Install version 7.0](#) on your Tableau Server. If you have a distributed installation, this step is on your primary Tableau Server.

Tableau Server Setup will handle importing the data and configuration settings from your earlier version. Repository migration is part of this process. If an error occurs, you are prompted to run a tabadmin command. See [Repository Migration Error](#) for steps.

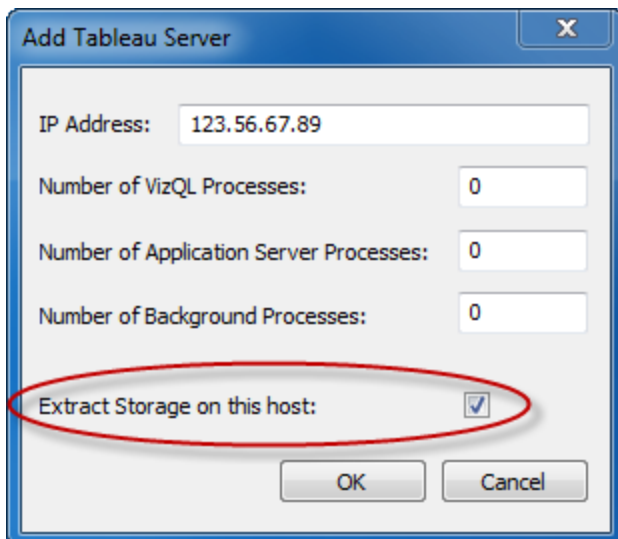
**Note:**

If you configured a 6.x server (primary or worker server) to run more than two instances of the web application (wgserver), VizQL (vizqlserver) or background (backgrounder) server process, after the upgrade you will have two instances of that process. You can change this—see [The Server Process Limit](#) for more information.

- [Upgrading a Version 6.x Data Extract Worker to Version 7.0](#)
- [Moving the Extract and Repository Services](#)
- [The Server Process Limit](#)
- [Repository Migration Error](#)

## Upgrading a Version 6.x Data Extract Worker to Version 7.0

As part of setting up a distributed server environment, you can configure a worker to host data extracts by selecting the Extract Storage on this host check box. Here's what this option looked like in version 6.x:



If you configured a worker to host data extracts in version 6.x, there are some extra steps you need to take to upgrade that worker to version 7.0.x. The steps you take depend on the worker's operating system. See the topics below for details. If you mistakenly deleted a worker's extract folder, refer to [Recovering Extracts from a Backup](#).

## Windows Server 2008, Windows Vista, and Windows 7

If a worker hosting extracts is running Windows Server 2008, Windows Vista, or Windows 7, do the following to upgrade it:

1. Use Add/Remove Programs to uninstall the 6.x worker software.
2. Delete the Program Files\Tableau and ProgramData\Tableau folders on the worker—*except* for the dataengine folder (ProgramData\Tableau\Tableau Server\data\tabsvc\dataengine).
3. Run version 7.0 of the Tableau Server Worker Software installer.
4. After you complete the above steps, return to step 3 in [Upgrading to 7.0](#).

## Windows Server 2003

If a worker hosting extracts is running Windows Server 2003, do the following to upgrade it:

1. Move the following folder to a location outside the Tableau Server folder so that the Uninstall program does not delete it (this is a known issue):
  - o **32-bit Windows Server 2003:** Program Files\Tableau\Tableau Server\data\tabsvc\dataengine
  - o **64-bit Windows Server 2003:** Program Files (x86)\Tableau\Tableau Server\data\tabsvc\dataengine
2. Use Add/Remove Programs to uninstall the 6.x worker software.
3. Delete any remaining Program Files\Tableau and ProgramData\Tableau folders on the worker.
4. Return the folder you moved in step 1 to its original location.
5. Run version 7.0 of the Tableau Server Worker Software installer.
6. After you complete the above steps, return to step 3 in [Upgrading to 7.0](#).

## Windows XP

Because Windows XP is not a supported platform in Tableau Server version 7.0, you cannot upgrade a Windows XP worker from version 6.x to 7.0. In addition, if you have any Windows XP workers, you should remove them from your Tableau Server configuration before you begin the upgrade process (see the [Pre-Upgrade Checklist](#) for steps).

If you had a Windows XP worker that was hosting extract data, the data it was hosting can be migrated to a new machine by doing the following:

1. Install version 7.0 on your primary Tableau Server (step 3 in [Upgrading to 7.0](#)).
2. Install version 7.0 of the Tableau Server Worker Software installer on a new machine.
3. Add the new machine to your Tableau Server cluster with Extract storage on this host selected. See [Installing Worker Servers](#) for details.

## Moving the Extract and Repository Services

If you need to delete a worker from the Tableau Server configuration and that worker is hosting the only instance of the repository or the extract service, you must first move the service onto another machine. This is because there must always be one active instance of the repository and extract services.

To move the extract or repository service:

1. If you haven't done so already, [stop the primary Tableau server](#) and open the Tableau Server Configuration dialog box (**Start > Tableau Server 7.0 > Configure Tableau Server**) on the primary Tableau Server.
2. On the Servers tab, highlight the IP address of the machine onto which you want to move the service. It can be another worker or the primary (This Machine).
3. Click Edit.
4. In the Edit Tableau Server dialog box, select the check box for the service you are moving: either Extract Storage on this host, Repository on this host or both, and click OK.
5. Click OK in the Tableau Server Configuration dialog box.
6. [Start the primary Tableau server](#) so that the changes can take effect.
7. [Stop the server](#) and open the Tableau Server Configuration dialog box.
8. On the Servers tab, highlight the IP address of the worker from which you are removing the service and click Edit.
9. Clear the check box for the service you moved and click OK.

10. Click OK again and [start the primary server](#) so that the changes can take effect.

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

## The Server Process Limit

In Tableau Server version 7.0, the `wgserver`, `vizqlserver`, and `backgrounder` server processes were re-engineered to be multi- instead of single-threaded. Any performance improvements you made in earlier versions by configuring multiple instances of these processes can now be achieved with fewer instances.

As a result of this change, when you upgrade to Tableau Server version 7.0.x, anything over two instances of a process is reset to two. Tableau recommends you try using this default configuration to assess server performance before making any adjustments.

If 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 Configuration dialog box. Eight instances of a process is the default upper limit. If your machine has enough RAM and CPU cores, you can change the upper 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. Still in the Tableau Server bin directory, enter 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 16
```

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

## Repository Migration Error

Starting with version 7.0, Tableau Server uses a new repository type. When you upgrade to version 7.0.x, the migration to the new repository type is handled for you. However, if something unexpected occurs during this process, an error message titled “Repository Migration Error” will direct you to run the `tabadmin migrate_to_new_repository` command. To run this command:

1. [Stop the server.](#)
2. From Tableau Server’s 7.0 bin directory, type the following to migrate the repository:

```
tabadmin migrate_to_new_repository
```

If the migration is successful, no error message will display. The migration does not delete your old repository. If you want to remove it, enter the following command:

```
tabadmin migrate_to_new_repository --remove-old-repository
```

The repository won’t be re-migrated, but the old repository will be removed.

3. [Start the server.](#)

## Migrating to New Hardware

Use the following procedure to migrate Tableau Server from one machine to another. Specifically, these steps describe how to move Tableau Server data and configuration settings from your in-production machine to a new machine where Tableau Server version 7.0.x is installed. Before you start, make sure you have followed the steps in the [Pre-Upgrade Checklist](#), including creating a [.tsbak file](#).

1. [Install version 7.0](#) of Tableau Server on the new machine.
2. Copy your *.tsbak* file to the bin folder on your new Tableau Server (for example, C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin).
3. Next, [stop Tableau Server](#).
4. Restore your in-production data and configuration information to your new Tableau Server installation by typing `tabadmin restore <filename>`, where `<filename>` is the name of the *.tsbak* file. For example:

```
tabadmin restore mybackup.tsbak
```

Or, to restore only the data from your in-production Tableau Server and no configuration information, type the following:

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

5. [Start the server](#).





# Distributed Environments

Use the topics below to learn more about running a distributed installation of Tableau Server:

- [About Distributed Environments](#)
- [Installing Worker Servers](#)
- [Maintaining a Distributed Environment](#)
- [High Availability](#)

## About Distributed Environments

After you complete the initial Tableau Server configuration, you can run Tableau Server on additional computers. This is called a distributed installation, or cluster. In a distributed installation there is a primary Tableau Server (installed using Tableau Server Setup) and one or more worker servers (installed using Tableau Worker Setup).

One reason to run Tableau Server in a distributed mode is so that you can assign more hardware to running certain server processes, thereby optimizing server performance in various ways. For example, if you expect your users to spend more time interacting with views than browsing and searching, you could set up an environment with several workers dedicated to running the VizQL server process, which handles view rendering. On the other hand, if you expect more browsing and searching you could dedicate a number of workers to the server's web application processes (wgserver). If your site handles a large number of tasks, such as extract refreshes, you can dedicate a machine to run several background task processes. (Even if you are running a single Tableau Server, you can change how many processes it's running.)

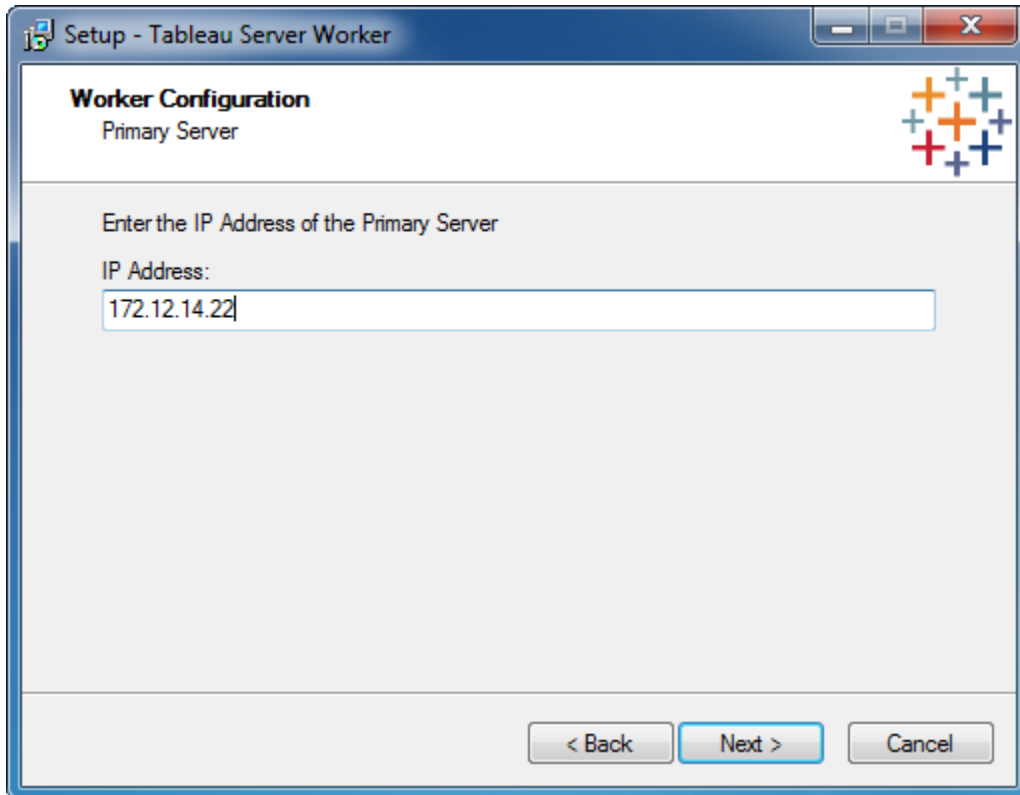
In addition to the above goals, you might also want to minimize the server system's potential for downtime—for example, in the event of a network, hardware, or software failure. In this case, you can configure for failover and high availability.

Because extracts and repository data can change rapidly over the course of a day, they can be a vulnerable area, even if you perform regular backups. For this reason, in a failover-ready configuration, two worker servers are dedicated to running the active and standby processes for data extracts and the repository. If the active worker fails, the standby worker takes over. The primary Tableau Server (called the gateway in this type of configuration) can be running all the remaining processes, or the workers can be handling everything, with the gateway acting only as the entry point to the cluster. You can build on this to improve system availability by also configuring a backup for the gateway. If the primary Tableau Server fails, it takes just a few steps to turn your backup gateway into the primary one.

## Installing Worker Servers

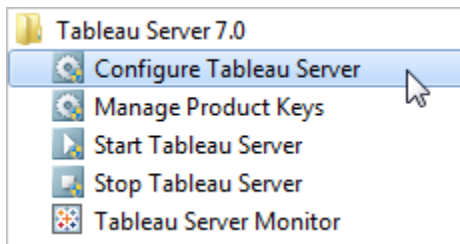
After you complete the initial configuration, you can set up Tableau Server to run on multiple machines. This is called a distributed installation. Running a distributed installation uses additional ports on the primary Tableau Server and requires that certain ports be available for binding during Setup on the Tableau worker server. See [TCP/IP Ports](#) for more information.

1. Make sure you've installed Tableau Server on the primary machine.
2. Stop the server on the primary machine (see [Tableau Server Monitor](#) to learn how).
3. Download the Tableau Server Worker software from the [Tableau Customer Account Center](#).
4. Run the Tableau Server Worker Software installer on all additional machines that you want to add to the Tableau Server cluster. During installation you will be asked to provide the IP address of the primary server.

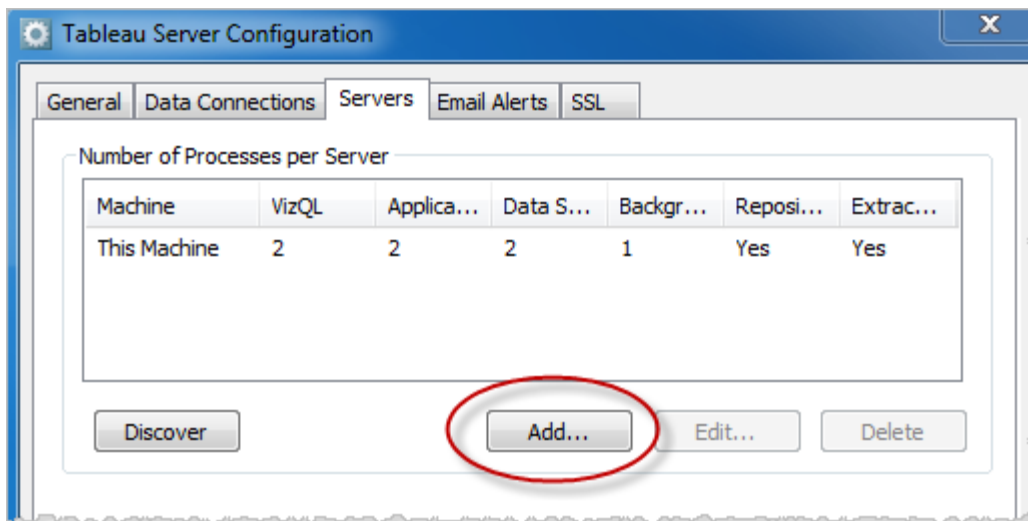


If you have a worker running Windows 7 with Windows Firewall enabled, refer to the [Tableau Knowledge Base](#) before proceeding.

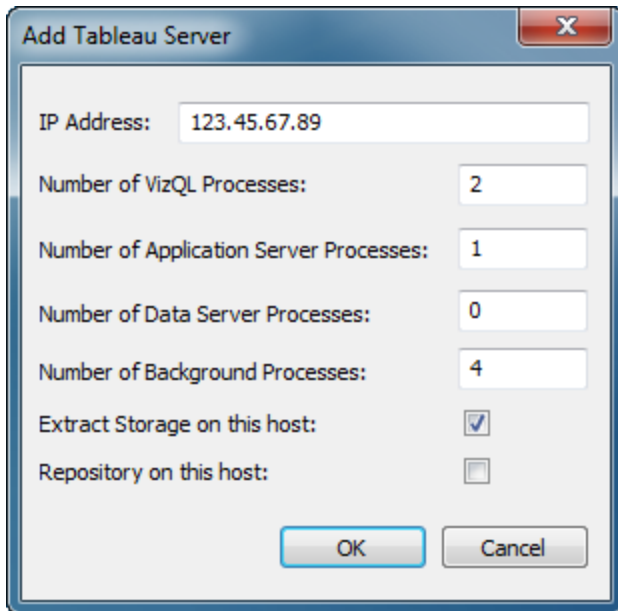
- Once the Worker software is installed on worker machines, and with the primary Tableau Server still stopped, return to the primary server and open the configuration utility by selecting Tableau Server 7.0 > Configure Tableau Server on the Start menu.



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



7. In the next dialog box, type the IP Address for one of the worker machines and specify the number of VizQL, Application Server, Data Server, and Background processes to allocate to the machine. You can assign up to [eight instances](#) of a process to a worker (or primary) server.



8. By default, the repository and extract storage are hosted on the primary server; however, you can select the Extract Storage on this host and Repository on this host checkboxes to use this server for extract storage and the repository—or as the standby server for same. See [High Availability](#) for more information.
9. Click OK.
10. Repeat these steps for each machine you want to add to the distributed environment. When you're finished adding workers, click OK again to save the changes, then start the server on the primary machine.

- [Database Drivers](#)

## 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 machines 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 machine dedicated as a VizQL server and another machine dedicated to extract storage, you only need to install drivers on the VizQL server machine.

























<b>Server process</b>	<b>Requires database driver?</b>
VizQL	yes
Application server	yes
Data server	yes
Backgrounder	yes
Extract storage (data engine)	no
Repository	no

## Maintaining a Distributed Environment

After you set up a primary and one or more worker machines 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 machine. Updates will be pushed to the workers automatically.

If the primary server changes its IP address, you will need to re-install all of the worker machines.

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

Maintenance							
 Service waiting for request  Service standing by  Service handling request  Service unlicensed  Service down							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
123.16.16.47			 	 	 		Gateway
12.36.30.0							
12.36.30.79							

## High Availability

Use the links below to learn more about Tableau Server's support for high availability:

- [Understanding High Availability](#)
- [High Availability Requirements](#)
- [Configuring for Failover](#)
- [Configuring a Highly Available Gateway](#)

### Understanding High Availability

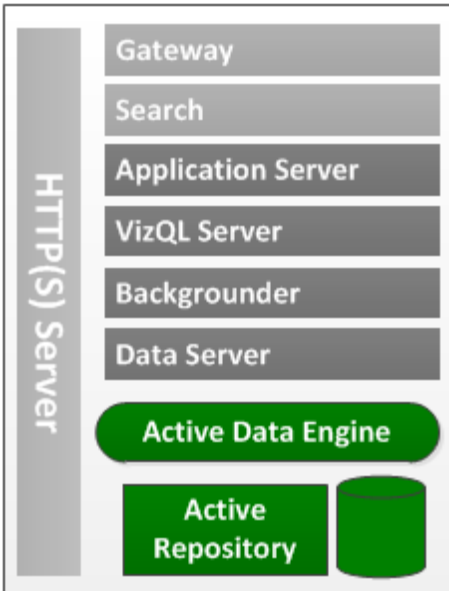
If you're configuring a Tableau Server system for high availability, the steps you perform are all about building in redundancy, thus reducing your potential downtime. The three areas that require redundancy are the data engine process, the repository process, and the primary Tableau Server (gateway). Because there must always be one active instance of each of these, 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 [Configuring for Failover](#) and [Configuring a Highly Available Gateway](#). See [High Availability Requirements](#) as well.

The topics below summarize how your server system topology evolves as you configure it for high availability. The minimum configuration for high availability is a three-node system. This includes a lightweight gateway that routes requests and two workers that host the main processes. You can increase reliability of the system by adding a fourth computer to serve as a backup gateway.

## 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 Status table on the Maintenance page typically looks like for a single-server system:

Maintenance							
<input checked="" type="checkbox"/> Service waiting for request <input checked="" type="checkbox"/> Service standing by <input checked="" type="checkbox"/> Service handling request <input type="checkbox"/> Service unlicensed <input checked="" type="checkbox"/> Service down							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
123.45.67.89	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

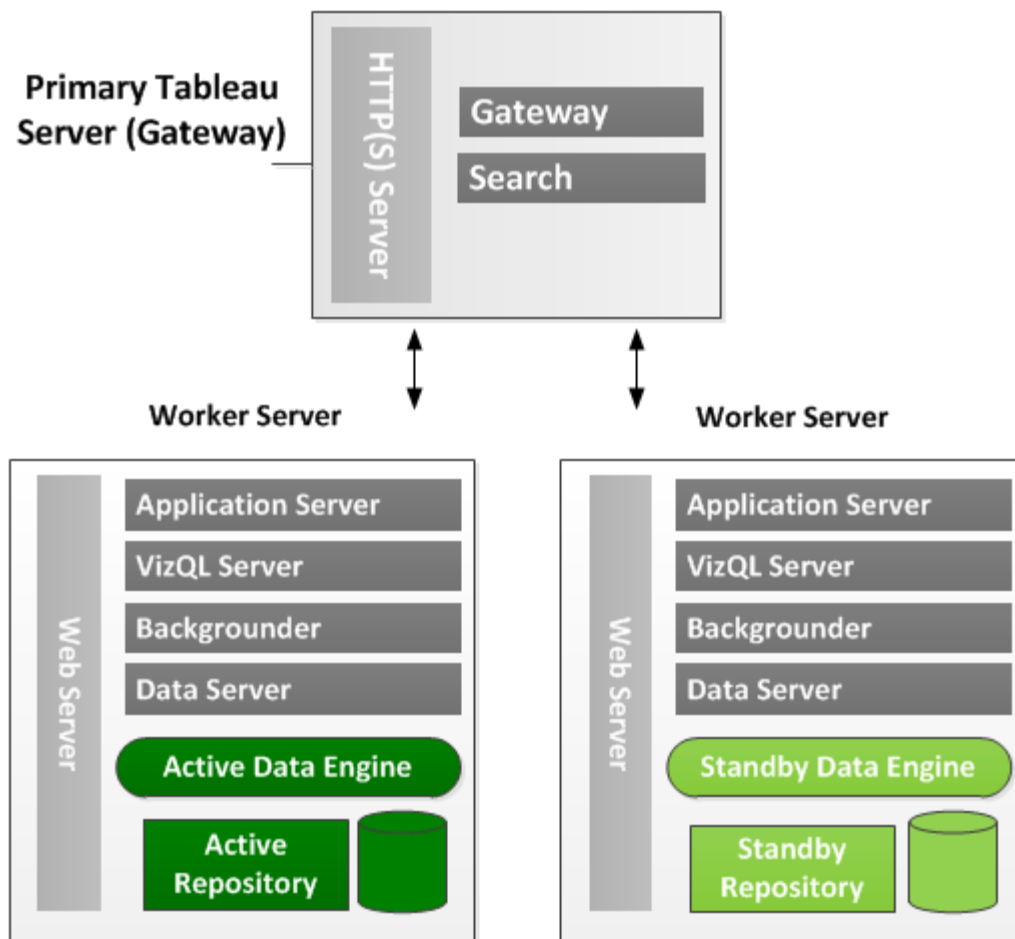
To build in redundancy, you need to add additional servers to host the active and standby data engine and repository processes. In addition, to reduce the system's vulnerability, the gateway should be isolated on its own node, ideally running as few of the server processes as possible.



### A Three-Node System

The next step in configuring for failover support and high availability is to install Tableau Server on two workers and add them to the primary's configuration one at a time. Assigning the data engine and the repository processes takes two steps because: 1) there must always be one active instance of each process, and 2) there cannot be more than two total. Processes also need to be removed from the primary Tableau Server.

Exactly how to add the workers and remove the processes from the primary is described in [Configuring for Failover](#). After you go through those steps, you have a three-node system:



The Status table on the Maintenance page looks similar to the following:

Maintenance							
<span>✔ Service waiting for request</span> <span>✔ Service standing by</span> <span>✔ Service handling request</span> <span>⬜ Service unlicensed</span> <span>✘ Service down</span>							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
123.45.67.89							Gateway
123.45.67.88	✔	✔	✔	✔	✔	✔	✔
123.45.67.87	✔	✔	✔	✔	✔	✔	✔

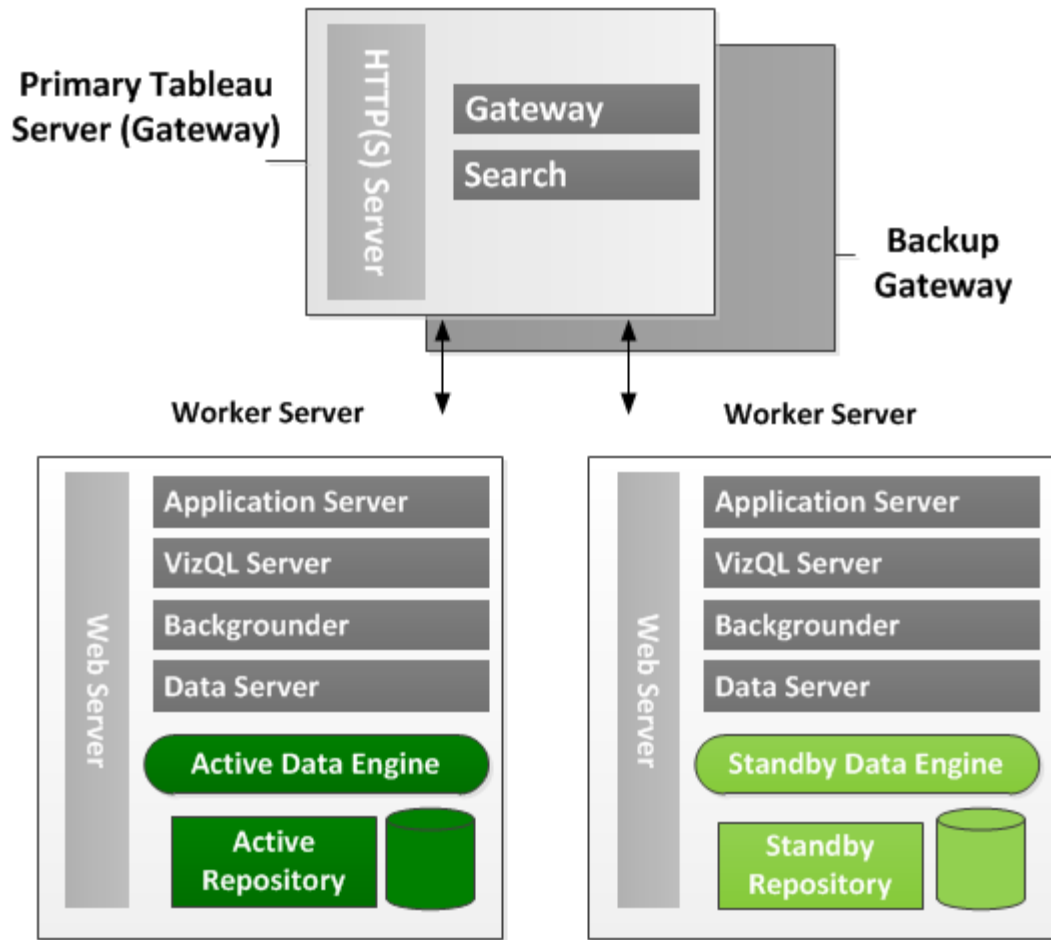
As you can see, the node running the gateway has no processes on it. Its purpose is to route requests to the two workers. In this configuration, if your active worker fails, the standby worker automatically becomes active and handles all requests from the gateway.

However, in a three-node system there is still a single point of failure: the gateway. You can mitigate risks in this area by creating a backup gateway. With a few manual steps, you can activate the backup gateway if the primary gateway fails.

### Adding a Backup Gateway

Adding a backup gateway provides a safeguard for your system. The backup gateway is an additional server added to the system to be ready if your primary gateway fails. While it is not an active server, after you complete the first set of steps in [Configuring a Highly Available Gateway](#), it is ready to be activated.

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



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

**Maintenance**

Service waiting for request 
  Service standing by 
  Service handling request 
  Service unlicensed 
  Service down

**Status**

Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
123.45.67.01							Gateway
123.45.67.88	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
123.45.67.87	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## High Availability Requirements

Before you start to configure a cluster for failover and high availability, make sure you meet the following requirements.

### Hardware

With the exception of the gateway, the systems you use for failover and high availability must meet the requirements described in [Before you install...](#) but do not need to be identical:

- **Failover—three computers:** To configure a cluster that provides failover support for the data engine and repository services, you need three computers or VMs: one for the primary Tableau Server and the other two for the Tableau workers.
- **High availability—four computers:** To configure for high availability, you need the three computers or VMs described above plus an additional one to be the backup gateway for your primary Tableau Server gateway.
- **Gateway computers:** If you configure for high availability, the primary Tableau Server gateway and the backup gateway may be running few or no Tableau Server processes. Therefore, the computers that run the gateway do not need as many cores as the ones running your worker servers. You will, however, need adequate disk space for backups. The gateway is temporarily used during the database backup and restore processes.

### Networking and Ports

As with any distributed system, the computers or VMs you use need to be able to communicate with one another. See [TCP/IP Ports](#) for a list of ports that must be available on the gateways and workers.

## Best Practices

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

- **IP addresses:** Note the IP addresses of each computer or VM you'll be working with. You will need to provide them during Tableau Worker Setup and configuration.
- **CNAME record:** If you're configuring for high availability, make sure your primary Tableau Server (gateway) and backup gateway have the same CNAME record so that your Tableau Server users have a smooth experience if one gateway fails and you configure the other to take over.
- **User account credentials:** For each machine you're upgrading, 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 gateway must be the same as the one you use for your backup Tableau Server gateway.
- **Backup:** It's a best practice to create a backup prior to making significant system changes. See [Backing Up the Database](#) for steps.

## Configuring for Failover

Do the following to configure a three-computer cluster that provides failover support:

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

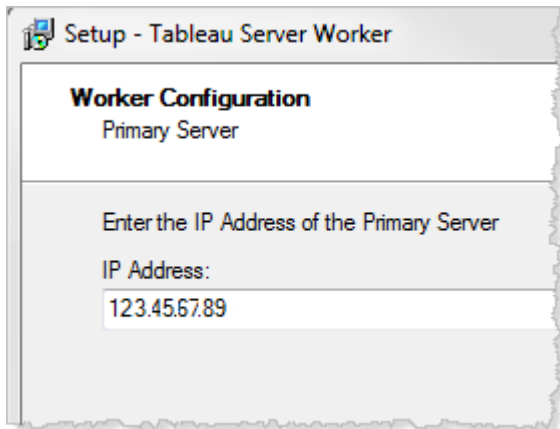
**Maintenance**

Service waiting for request
  Service standing by
  Service handling request
  Service unlicensed
  Service down

Status

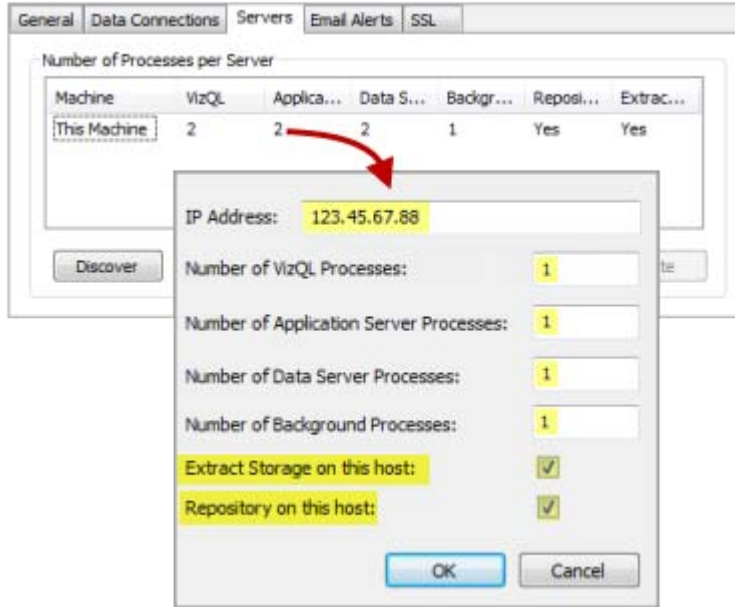
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
123.45.67.89	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

3. [Stop the server](#) on the primary.
4. Next, run [Tableau Worker Setup](#) on the two additional computers or VMs that will provide failover support. During Worker Setup, you will need to provide the IP address of the primary Tableau Server:



5. With the primary server still stopped, open its Configuration dialog box: **Start > All Programs > Tableau Server**. On the General tab enter the Run As account password.
6. Select the Servers tab then click Add to add a worker.

7. Enter the IP address of the worker, and select the **Extract Storage on this host** and **Repository on this host** check boxes:



If you want the worker to run other server processes, enter the number of instances you want to run, such as 1 or 2. The maximum number is eight of each.

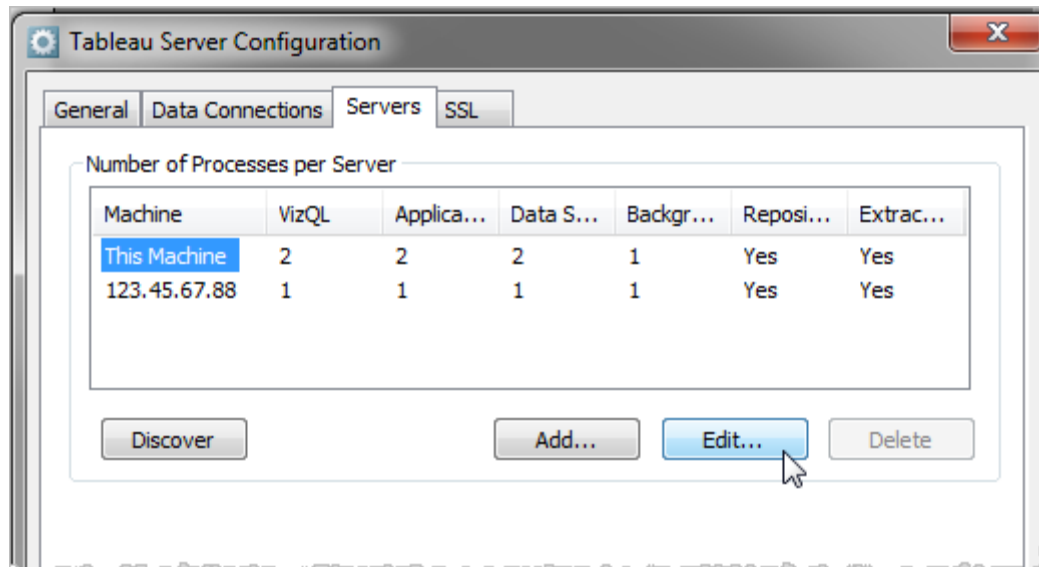
8. Click OK to close the Add Worker dialog box then click OK again to close the Configuration dialog box.
9. [Start the server](#) on the primary. It will take a few minutes for your changes to take effect and the repository and data engine services will initially have the red “service down” icon. After the data has been replicated, the Status table on the Maintenance page should look similar to the following:

Maintenance							
<input checked="" type="checkbox"/> Service waiting for request <input checked="" type="checkbox"/> Service standing by <input checked="" type="checkbox"/> Service handling request <input type="checkbox"/> Service unlicensed <input checked="" type="checkbox"/> Service down							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server Gateway
123.45.67.89	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Gateway
123.45.67.88	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

The worker you just added is running standby instances of the repository and data engine processes. Note that the primary now has a Web Server status of Gateway.

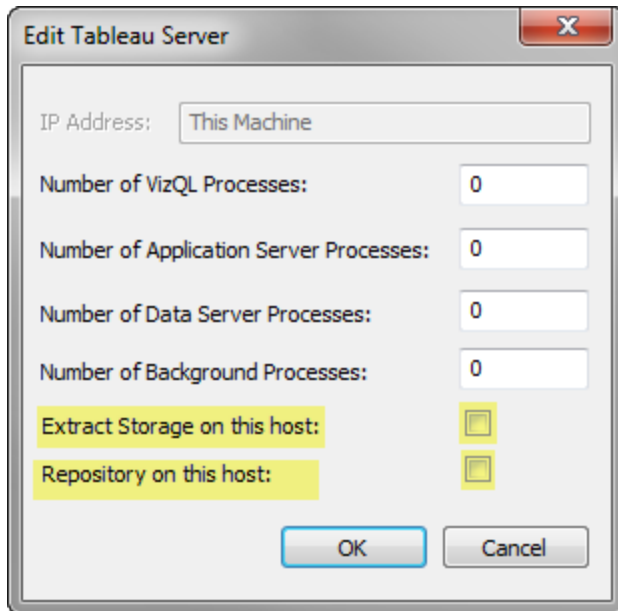
Next, you will remove processes from the primary and add a second worker to run them.

10. [Stop the server](#) on the primary and open its Configuration dialog box again. On the General tab enter your password.
11. Select the Servers tab, highlight This Machine (which is the primary Tableau Server), and click Edit



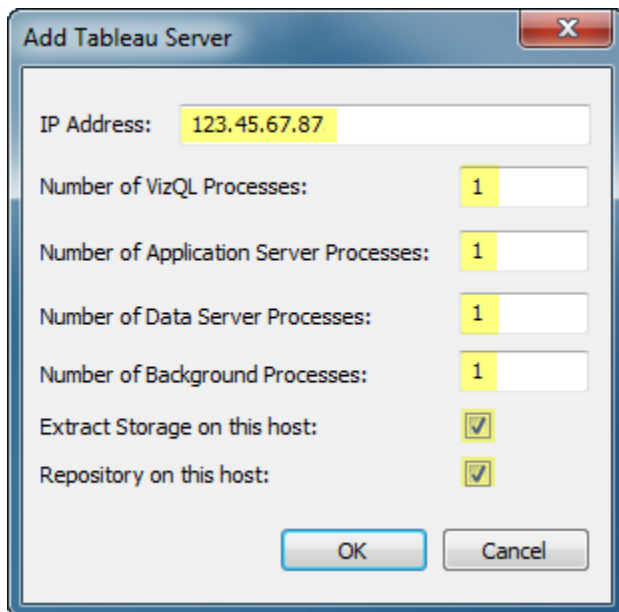


12. In the Edit Tableau Server dialog box, clear the Extract Storage on this host and Repository on this host check boxes. If you want the primary Tableau Server to run nothing but Apache (so, no Tableau Server processes), you can remove the remaining processes from it by entering 0 in each text box:



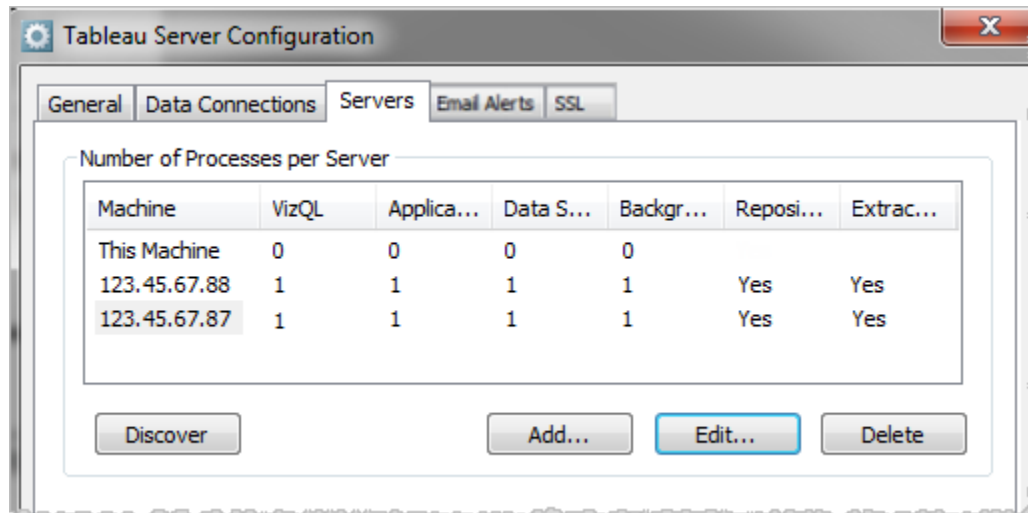
13. Click OK.
14. In the Tableau Server Configuration dialog box click Add to add a second worker.

15. In the Add Tableau Server dialog box, enter the IP address of the second worker, select the Extract Storage on this host and Repository on this host check boxes, and enter the number of instances you want to run, such as 1 or 2. The maximum number is eight of each.

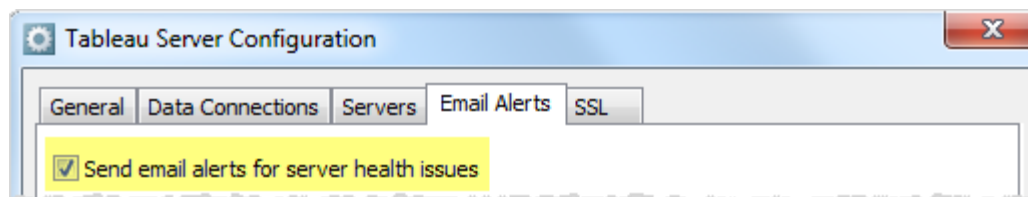


You do not need to specify which worker is active and which is standby for the extracts and repository.

16. Click OK. The Tableau Server Configuration dialog should now look similar to the following:

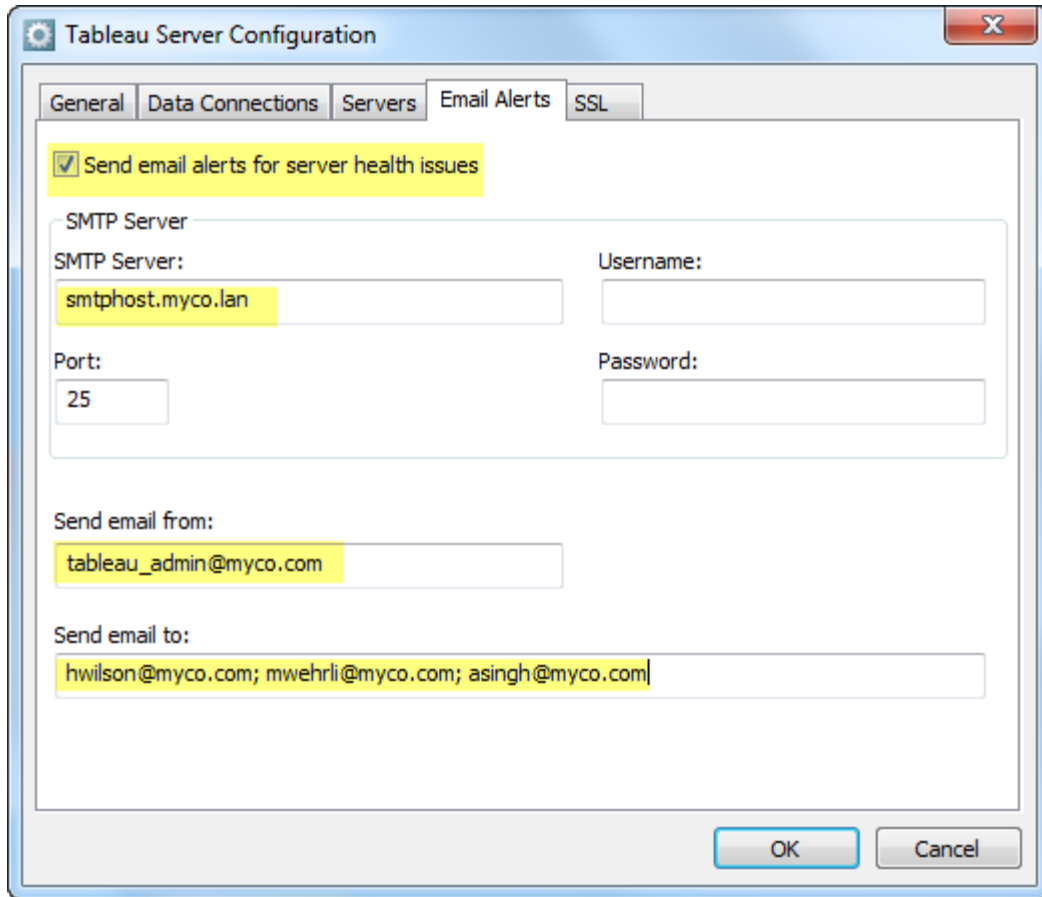


17. You can also set up email alerts so that you're notified of server failures or changes in status for your extract and repository services. To do this, click the Email Alerts tab in the Tableau Server Configuration dialog box.
18. Select the Send email alerts for server health issues check box:



19. 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 an alert if there's a system failure. Under Send email to enter at least one email address that will receive the alerts. If you enter multiple email addresses, separate them with commas or semicolons:



Click OK.

20. [Start the server](#) on the primary (it may take a few minutes for your changes to take effect). Your system is now configured to provide failover support for the data engine and repository services. The Status table on the Maintenance page should look similar to the following:

Maintenance							
<span>✔ Service waiting for request</span> <span>✔ Service standing by</span> <span>✔ Service handling request</span> <span>✘ Service unlicensed</span> <span>✘ Service down</span>							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
123.45.67.89	✔	✔	✔	✔	✔	✔	Gateway
123.45.67.88	✔	✔	✔	✔	✔	✔	✔
123.45.67.87	✔	✔	✔	✔	✔	✔	✔

A light green check mark means a service is standing by, ready to take over if the active service (dark green check mark) should fail.

## Configuring a Highly Available Gateway

Before you follow the procedures in this topic, follow the steps in [Configuring for Failover](#). After going through those steps, you will have two worker servers that are providing failover support and a gateway (your primary Tableau Server). Your gateway can be running nothing but Apache (no server processes) or it can be running any server process except for the data engine and repository processes. Those must be run by the workers.

The first procedure below describes how to create a backup of your gateway. The second procedure walks you through what to do if your current gateway fails.

## Creating a Backup Gateway

Do the following to create a backup gateway:

1. [Stop the server](#) on your primary Tableau Server (referred to as the gateway going forward).
2. On the gateway computer, open a command prompt as an administrator and navigate to the Tableau Server bin directory:
  - o **32-bit:** C:\Program Files\Tableau\Tableau Server\7.0\bin
  - o **64-bit:** C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin
3. Enter the following command, where `IP_address1` is the current gateway's IP address and `IP_address2` is the backup gateway's IP address:

```
tabadmin failovergateway --primary IP_address1 --secondary  
IP_address2
```

4. Next, create a copy of the gateway's `tabsvc.yml` file (located in `ProgramData\Tableau\Tableau Server\config`) and put the copy in a temporary location on your backup gateway computer.

The `tabsvc.yml` file contains server configuration settings. It gets written to when you change your configuration settings in the Tableau Server Configuration dialog box or via `tabadmin`. If `tabsvc.yml` changes, you will need to update the copy of `tabsvc.yml` on your backup gateway.

- On your backup gateway computer, open the `tabsvc.yml` file and replace the IP address for the gateway on the `worker.hosts` line with the IP address for the backup gateway (the computer you're currently on):

```

tabsvc_copy.yml - Notepad
File Edit Format View Help
worker0.wgserver.procs: 0
worker1.backgrounder.procs: 1
dataengine1.host: 123.45.67.88
pgsql.host: 123.45.67.87
service.runas.username: MyCo
\tableauadmin
worker0.vizqlserver.procs: 0
worker1.dataserver.procs: 1
worker0.dataserver.procs: 0
worker0.backgrounder.procs: 0
worker1.vizqlserver.procs: 1
worker2.backgrounder.procs: 1
worker1.wgserver.procs: 1
dataengine.host: 123.45.67.87
worker2.vizqlserver.procs: 1
worker2.wgserver.procs: 1
worker2.dataserver.procs: 1
dataengine0.host: 123.45.67.87
pgsql1.host: 123.45.67.88
pgsql0.host: 123.45.67.87
worker.hosts: 123.45.67.89, 123.45.67.01
123.45.67.87, 123.45.67.88
config.version: 9
service.init.state:
start

```

- On your backup gateway computer, install Tableau Server. Use the same Run As account and configuration settings that you used when you ran Tableau Server Setup on your gateway.
- After Setup completes, [stop the server](#) on the backup gateway computer.
- Still on your backup gateway computer, enter the following command to disable its Tableau Server service:

```
sc config tabsvc start= disabled
```

You've finished creating a backup gateway. See the next set of steps for what to do if your current gateway fails. If you are working in a test environment, this would be a good time to power down your current gateway to simulate a system failure.

## Configuring the Backup Gateway

Follow this second set of steps in the event of a gateway failure. All steps should be performed on the backup gateway computer.

1. On your backup gateway computer, use the tabsvc.yml file you edited in step 5 of the previous procedure to overwrite the locally installed one in ProgramData\Tableau\Tableau Server\config.
2. Open a command prompt as an administrator and navigate to the Tableau Server bin directory:
  - o **32-bit:** C:\Program Files\Tableau\Tableau Server\7.0\bin
  - o **64-bit:** C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin
3. Enter the following command, where IP\_address2 is the IP address of your backup gateway (soon to be your new gateway) and IP\_address1 is the IP address of your former gateway (soon to be your backup):

```
tabadmin failovergateway --primary IP_address2 --secondary IP_address1
```

4. Enter the following command:

```
sc config tabsvc start= auto
```

5. [Start the server](#). Your backup gateway is now your primary gateway. When you look at the Status table on the Maintenance page, you should notice that the IP address for the gateway has changed:

Maintenance									
<input checked="" type="checkbox"/> Service waiting for request		<input checked="" type="checkbox"/> Service standing by		<input checked="" type="checkbox"/> Service handling request		<input type="checkbox"/> Service unlicensed		<input type="checkbox"/> Service down	
Status									
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server	Gateway	
123.45.67.01	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
123.45.67.88	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
123.45.67.87	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

6. For your former primary gateway to now act as your backup gateway, you will need to do the following:
  - o Use Add/Remove Programs to remove Tableau Server from your former primary gateway. At the end of the Uninstall program you will receive a backup error, which you can ignore.
  - o Delete the Tableau folders under Program Files (x86) and ProgramData on your former primary gateway.
  - o Repeat the steps in this topic starting with step 4 under “Creating a Backup Gateway.”



# Working with the Server

Refer to the following topics as you use the Tableau Server user interface to administer your installation:

- [Users and Licenses](#)
- [Permissions](#)
- [Groups and Projects](#)
- [Scheduling Tasks](#)
- [Sites](#)
- [Server Maintenance](#)
- [Data Sources](#)
- [Data Connections](#)
- [Customizing the Server](#)

## Users and Licenses

Everyone who needs to access Tableau Server, whether it's to publish, browse, or administer, must be added as a user. In addition, users must be assigned a license level.

- [Users](#)
- [Licenses](#)

### Users

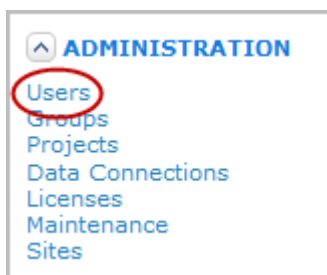
Everyone who needs to access Tableau Server—whether it's to publish, browse, or administer—must be added as a user. If Tableau Server is running multiple sites, the All Users page is where system administrators do this. Otherwise, if Tableau Server is in single site mode, system and content administrators can add users on the Users page.

Once users have been added, you can edit and delete them, add or remove them from sites, and assign them license levels and user rights. See the topics below for more information.

- [Adding Users](#)
- [Adding Users to a Site](#)
- [CSV Guidelines](#)
- [Adding Users to a Group](#)
- [Viewing, Editing & Deleting Users](#)

### Adding Users

Both system administrators and content administrators with the correct permissions can add users from the Users page:



There are two ways you can add users from this page: one at a time (described below) or in batches using a CSV file (described in [CSV Guidelines](#)).

**To add a single user:**

1. From the Users page, click the Add User link at the bottom of the list of users:



2. Enter a Username.

- *Local authentication:* If the server is configured for local authentication, using an email address for the Username is the best way to avoid username collisions (for example, *jsmith@myemail.com* instead of *jsmith*). After you enter a Username, click Add User.
- *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 Username 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:** Be sure not to enter the user's Full Name in this field as it can cause errors during the importing process.

3. *Local authentication only*, provide the following:
  - Full Name—Type a display name for the user (e.g., *John Smith*).
  - Password—Type a password for the user.
  - Confirm—Retype the password.
4. License Level: Select a license level. See [Licenses](#) and Permissions to learn more.
5. User Rights: Select whether the user can publish workbooks and assign administrator rights. Refer to [About User Rights](#) to learn more.
6. Click Add User.

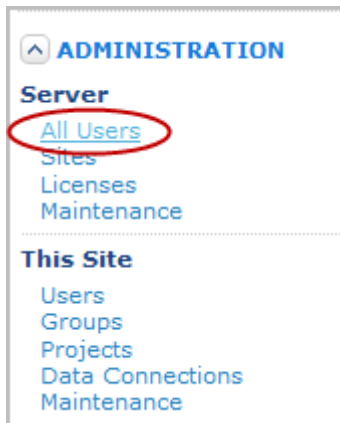
## Adding Users to a Site

See the topics below for more information:

- [Adding Users to a Site from All Users](#)
- [Adding Users to a Site from Users](#)

## Adding Users to a Site from All Users

As the system administrator of a multi-site system, only you can access the All Users page. It's the only place where you can add users to multiple sites all at once, remove users, and if the server is using local authentication, reset user passwords.



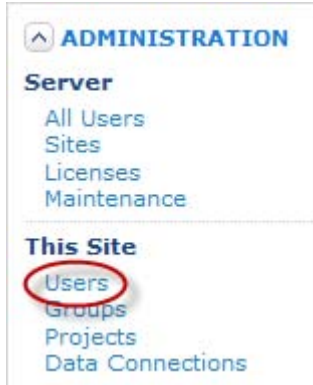
There are two ways you can add users from here: one at a time (described below) or in batches using a CSV file (described in [CSV Guidelines](#)).

To add a single user:

1. From the All Users page, click the Add User link at the bottom of the list of users.
2. Enter a Username:
  - *Local authentication*—If the server is using local authentication, using an email address for the username is the best way to avoid username collisions (for example, `jsmith@myco.com` instead of `jsmith`).
  - *Active Directory*—If the server is using Active Directory for user authentication and the domain is different than the server domain, include the fully qualified domain name for the first user you add. Subsequent users can use the domain nickname. Refer to [Modifying Domain Names](#) to learn how to modify the nickname.
3. If the server is using local authentication, provide the following:
  - Full Name—Type a display name for the user (e.g., John Smith).
  - Password—Type a password for the user.
  - Confirm—Retype the password.
4. Site Membership—Select which site(s) the user should be a member of. The site you are logged into is selected by default.
5. License Level and User Rights—Select a license level, Admin role, and whether the user can publish workbooks and data sources. A user who belongs to multiple sites can have different license levels and user rights on each site. Refer to [About License Levels](#), [Permissions](#), and [About User Rights](#) to learn more.
6. Click OK.

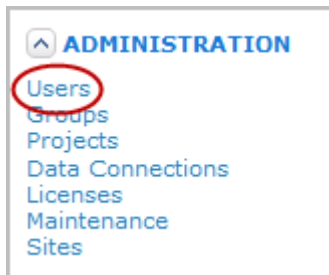
## Adding Users to a Site from Users

Both system administrators and content administrators with the correct permissions can add users from the Users page. For a system administrator, Users is located under This Site:



When you add users from here, it only adds them to the site you're currently logged into. If you want to assign users to multiple sites or reset passwords, go to the All Users page under Administration.

For content administrators, Users is located under Administration:



There are two ways you can add users from this page: one at a time (described below) or in batches using a CSV file (described in [CSV Guidelines](#)).

To add a single user:

1. From the Users page, click the Add User link at the bottom of the list of users.
2. Enter a Username:
  - *Local authentication*—If the server is using local authentication, using an email address for the username is the best way to avoid username collisions (for example, `jsmith@myco.com` instead of `jsmith`).

If the username already exists on the server system, but not on the site, a dialog appears that lets you either enter a different username or add the user to the site. If you add the user, skip to step 4.

- *Active Directory*—If the server is using Active Directory for user authentication and the domain is different than the server domain, include the fully qualified domain name for the first user you add. Subsequent users can use the domain nickname. Refer to [Modifying Domain Names](#) to learn how to modify the nickname.

If the username already exists on the server system, but not on the site, a dialog appears that lets you either enter a different username or add the user to the site. If you add the user, skip to step 4.

3. If the server is using local authentication and you're adding a user who's new to the server as well as to the site, provide the following:
  - Full Name—Type a display name for the user (e.g., John Smith).
  - Password—Type a password for the user.
  - Confirm—Retype the password.
4. License Level and User Rights—Select a license level, Admin role, and whether the user can publish workbooks and data sources. A user who belongs to multiple sites can have different license levels and user rights on each site. Refer to [About License Levels](#), [Permissions](#), and [About User Rights](#) to learn more.
5. Click Add User.

**Note:**

Content administrators can edit an existing user's account as long as the user is only a member of sites that the content administrator also controls. For example, if User Joe is a member of Site A and Site B and the content administrator is only an administrator of Site B, the content administrator can't edit Joe's Full Name or reset his password.

### CSV Guidelines

If you are importing users through a CSV file, see the following topics for format requirements, multi-site server considerations, and how to do the import.

### Requirements

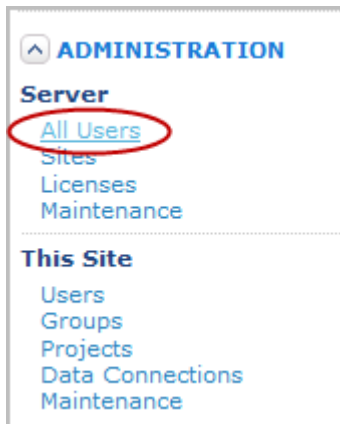
- The CSV file must be saved in UTF-8 or UTF-16 format.
- Character encodings other than UTF-8 and UTF-16, such as BIG-5, must be converted. You can do this via a “Save As”.
- The following two column headers are always required:
  - Username
  - Password: If Tableau Server is configured to use Active Directory user authentication, there must be a Password column header, but the column itself should be empty. If the server is using local authentication, you must provide passwords for new users. See also “Multi-Site Mode and Where you Import From” for other information.
- The CSV file can also have the following additional columns, in the order shown below (after the Username and Password columns):
  - Full Name
  - License Level (Interactor, Viewer, or Unlicensed)
  - Administrator (System, Content, or None)
  - Publisher (yes/true/1 or no/false/0)



## Multi-Site Mode and Where You Import From

If the server is running multiple sites and you are a system administrator, there are two pages you can do a CSV user import from. Each has different capabilities where existing server user accounts are concerned.

- All Users page: This page displays if a server is running multiple sites and only system administrators can access it.



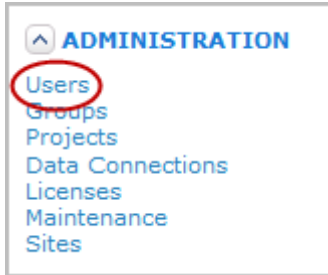
CSV imports from here allow you to update existing server user accounts, in addition to adding new ones. For example, if you do a CSV import that has a new password for each existing user, their passwords will be reset.

- The Users page under This Site:



When a system administrator is working from here, they have the same capabilities as a content administrator. This means they can add new user accounts with CSV imports and, if existing users are part of the import, the Password and Full Name fields must either match or be left blank. If new passwords or full names are used, the import will fail.

If you are a content administrator on a server with multiple sites, you perform CSV user imports from the Users page under Administration.

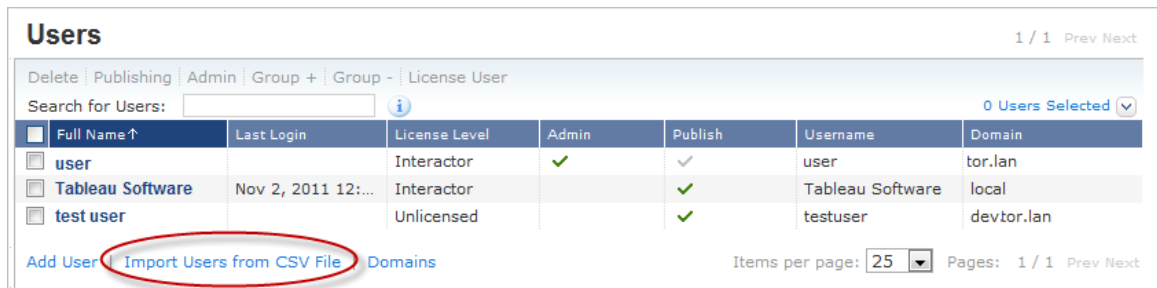


A user can belong to more than one site on the same server system, but they must use the same credentials for each site. This becomes important when you are adding users who are new to your site but perhaps not to the server (in other words, they're a member of a different site on the server). If you think this may be the case, try leaving the Password column blank (but keep the required Password column header). If the server is configured for local authentication and a new site user is also new to the server system, you'll see a message in the CSV import window telling you to provide a password for the user.

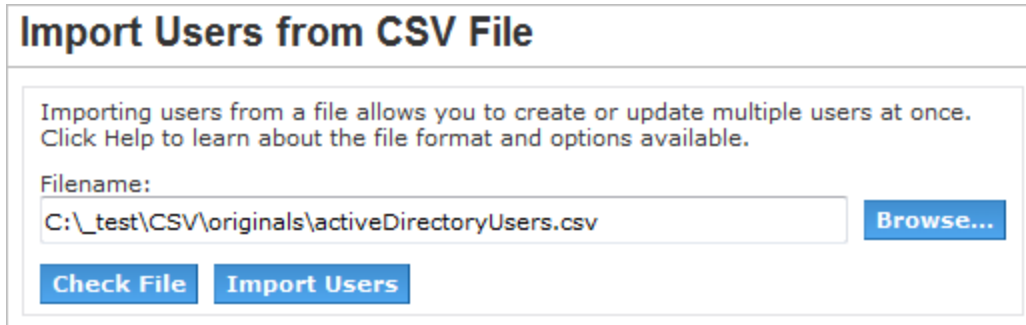
## Adding Users from a CSV File

To add users from a CSV file:

1. From the Users or All Users page, click the Add Users from CSV File link:



2. Click Browse, select the file, then click Check File:



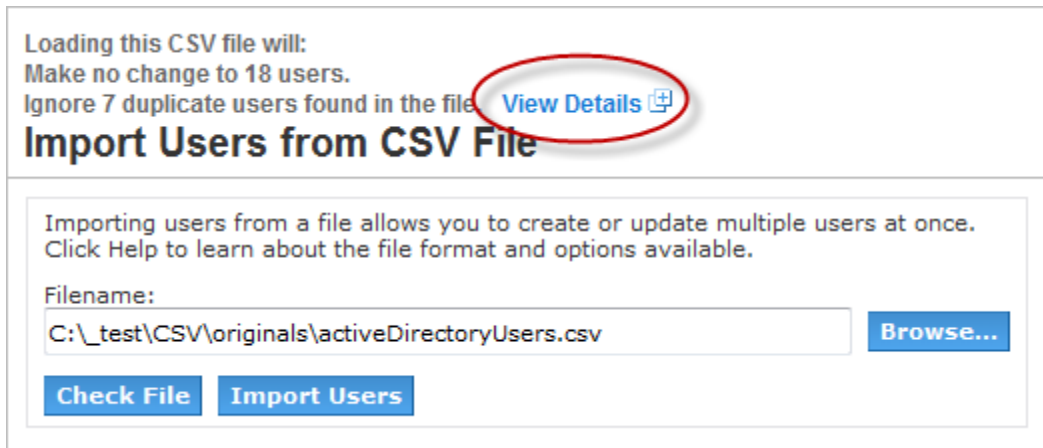
**Import Users from CSV File**

Importing users from a file allows you to create or update multiple users at once. Click Help to learn about the file format and options available.

Filename:  
C:\\_test\CSV\originals\activeDirectoryUsers.csv [Browse...](#)

[Check File](#) [Import Users](#)

3. Preliminary results display. To see account-specific information, select View Details:



Loading this CSV file will:  
Make no change to 18 users.  
Ignore 7 duplicate users found in the file. [View Details](#)

**Import Users from CSV File**

Importing users from a file allows you to create or update multiple users at once. Click Help to learn about the file format and options available.

Filename:  
C:\\_test\CSV\originals\activeDirectoryUsers.csv [Browse...](#)

[Check File](#) [Import Users](#)

4. To continue, click Import Users then click Exit in the final dialog box. To reimport the file or select a different file, click Browse again.

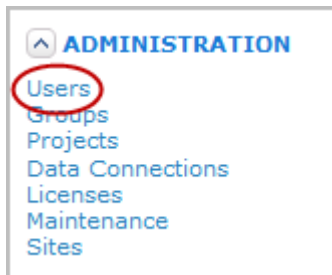
After you import the CSV file, assign the users to a site.

## Adding Users to a Group

One way to make it easier to manage users is to group them. That way you can assign permissions to an entire group rather than each individual user. To add a user to a group, the group must already exist. See [Creating Groups](#) for more information.

To add a user to a group:

1. In the Administration section, open the Users page:

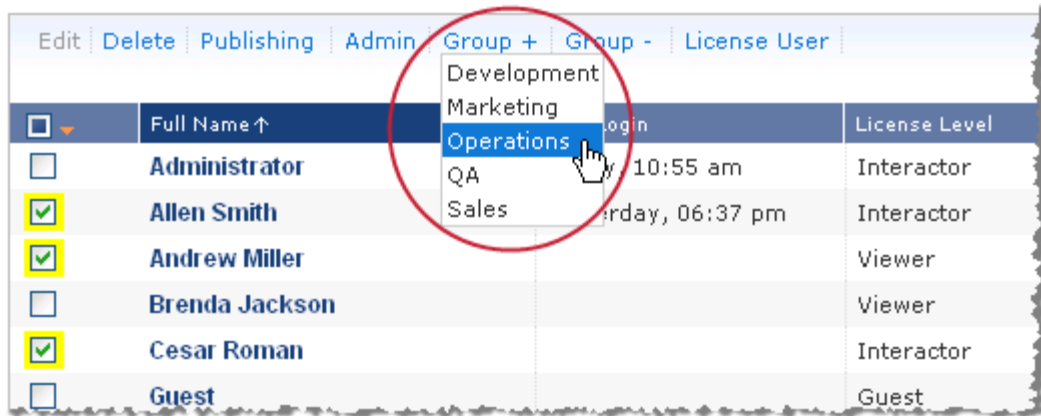


If Tableau Server is running multiple sites, the Users link is under This Site.



2. Select one or more users in the users list.
3. Click the Group + link in the Actions toolbar along the top of the list.

4. Select a group to add the users to.



<input type="checkbox"/>	Full Name ↑	Group	Login	License Level
<input type="checkbox"/>	Administrator			Interactor
<input checked="" type="checkbox"/>	Allen Smith			Interactor
<input checked="" type="checkbox"/>	Andrew Miller			Viewer
<input type="checkbox"/>	Brenda Jackson			Viewer
<input checked="" type="checkbox"/>	Cesar Roman			Interactor
<input type="checkbox"/>	Guest			Guest

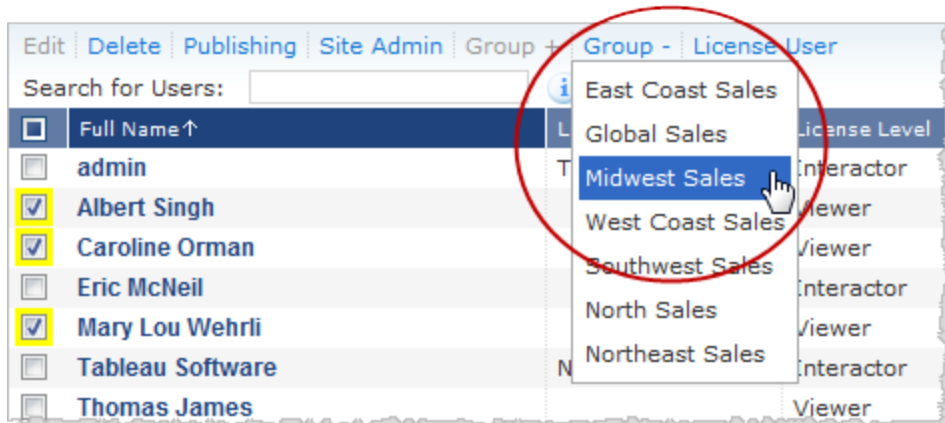
The screenshot shows a user management interface. At the top, there are buttons for 'Edit', 'Delete', 'Publishing', 'Admin', 'Group +', 'Group -', and 'License User'. Below these buttons is a table with columns for checkboxes, Full Name, Group, Login, and License Level. A dropdown menu is open over the 'Group +' button, showing options: Development, Marketing, Operations (highlighted), QA, and Sales. A red circle highlights the dropdown menu. The table contains several users, with checkboxes for Allen Smith, Andrew Miller, and Cesar Roman checked.

- [Deleting Users from a Group](#)

## Deleting Users from a Group

You can remove users from a Tableau Server group using the Users page.

1. Select one or more users in the users list.
2. Click the Group - link in the Actions toolbar along the top of the list.
3. Select the group to remove the users from:

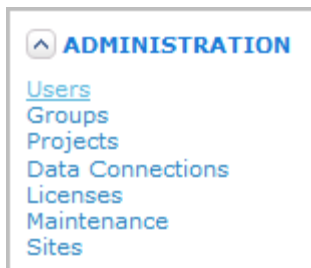


## Viewing, Editing & Deleting Users

Use this topic to learn how to view, edit, and delete Tableau Server users.

### Viewing Users

To view a list of all users on the server, click the Users link in the Administration section on the left side of the page:



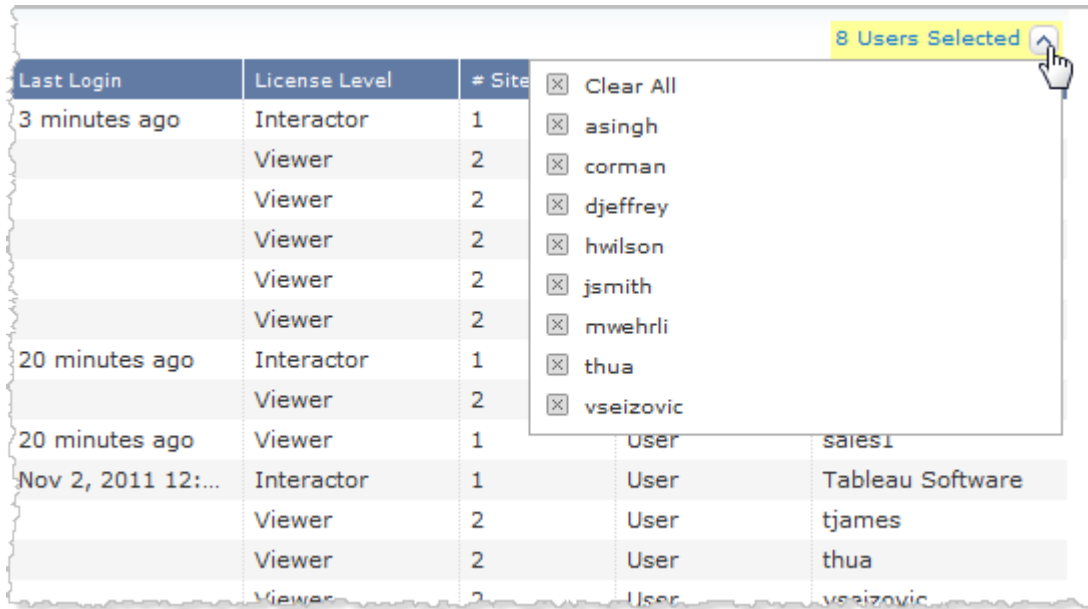
If Tableau Server is running multiple sites, the All Users link under Server lists all users on the server system, and the Users link under This Site displays all users for the site you're currently logged into:



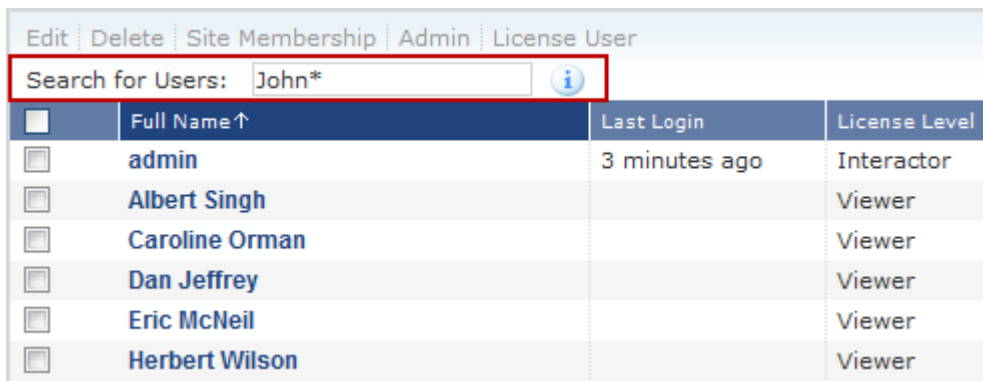
#### Note:

By default, this list of users is private and can only be seen by administrators. You can make the list of users public by enabling Public User List, in the Settings area of the Maintenance page. If the server is running multiple sites, enabling this setting will only show users the names of users on their site.

Users may be listed across multiple pages. As you select users in the list they are added to a quick list in the upper right corner. The quick list lets you see how many users you have selected and easily remove users from the selection. Click the "x" button next to the username in the quick list to remove someone from the selection.



You can also use the Search box at the top of the user list to quickly find a specific user in the list. Type all or part of the user's name and press Enter on your keyboard. You can use an asterisk (\*) character as a wildcard in the search. For example, searching for `John*` will return all names that start with *John*.

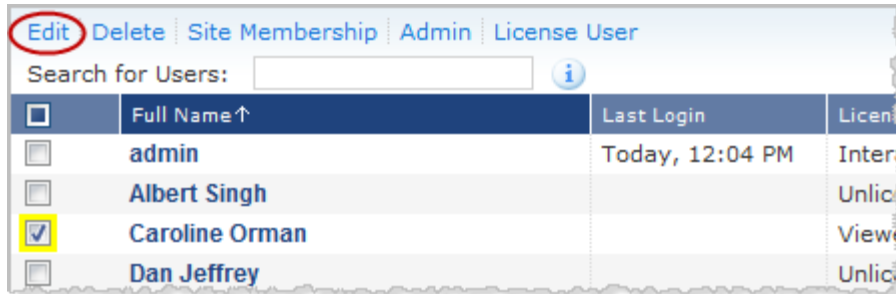




## Editing Users

If the server is configured to use the internal user management system, you can edit the Display Name and Password for users after they have been added. If you are making a lot of changes it is easiest to import from a CSV file, see [Adding Users](#). To edit user information:

1. Select a single user in the user list.
2. Click the Edit link in the Actions toolbar at the top of the list.



3. Type a new Display Name and Password into the corresponding text boxes.
4. Click Update.

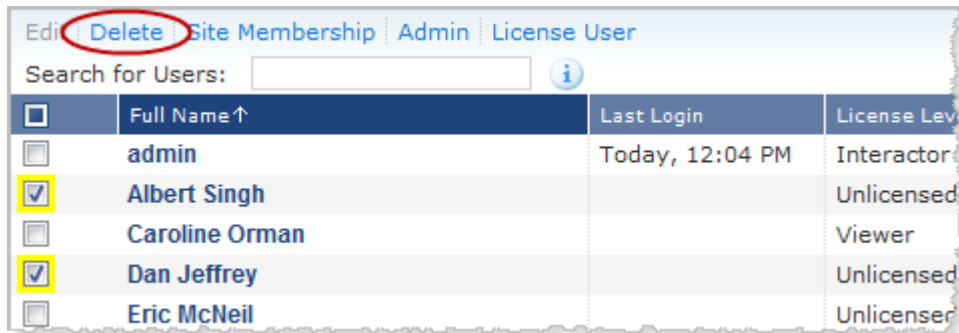
The screenshot shows the user edit form. It has a table with columns 'Change?' and 'Attribute'. The 'Username' is 'corman'. The 'Display Name' is 'Caroline Orman'. The 'Password' and 'Confirm' fields are masked with dots. The 'Submit' button is circled in red.

Change?	Attribute
	Username: corman
<input checked="" type="checkbox"/>	Display Name: Caroline Orman
<input checked="" type="checkbox"/>	Password: .....
	Confirm: .....
	<input type="button" value="Submit"/>

## Deleting Users

You can remove users from Tableau Server using the Users page, if you're running a single-site system, or the All Users page, if you're running a multi-site system.

1. From the Users or All Users page, select one or more users to delete.
2. Click Delete in the Actions toolbar at the top of the list.



3. Click Yes in the dialog that appears.

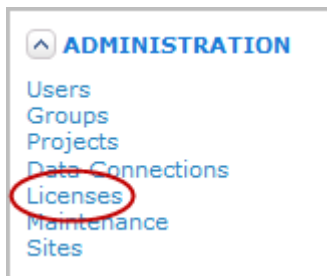
## Licenses

Use the topics below to learn more about licensing:

- [About License Levels](#)
- [Assigning License Levels](#)
- [About User Rights](#)
- [Allowing or Denying User Rights](#)

### About License Levels

To open the Licenses page, click the Licenses link in the Administration section on the left side of the page:



Your product key gives you a set of license levels which, as an administrator, you can distribute to your users. You can assign the following license levels:

License Level	Description
Unlicensed	The user cannot log in to the server. All users are added as unlicensed by default.
Viewer	The user can log in and see published views on the server but cannot interact with the views. Users with this level can only be given permission to view, add comments, and view comments.
Interactor	The user can log in, browse the server, and interact with the published views.
Guest	The guest license level is available 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 logging in. This option is only available with a core-based server.

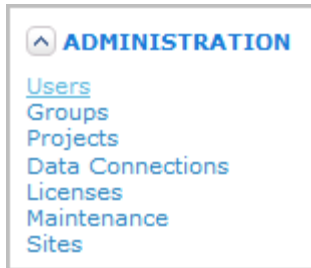
You can review how these levels have been distributed on the Licenses page:

License Levels	
Name ↑	Total Users
Guest	1
Interactor	7
Unlicensed	1
Viewer	4

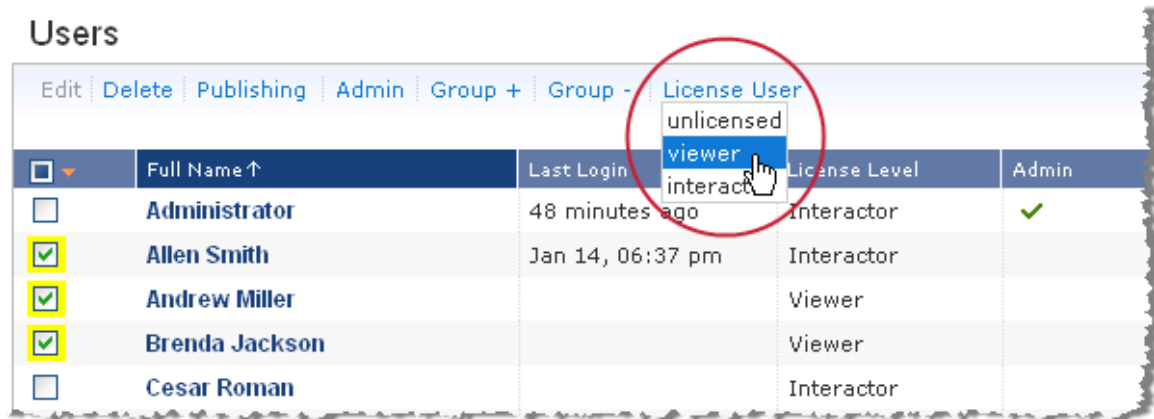
## Assigning License Levels

To assign license levels to users:

1. Log into Tableau Server using your administrator user name and password.
2. Click Users in the Administration area on the left side of the page.



3. Select one or more users you want to assign license levels to.
4. Click the License User link in the Actions toolbar along the top of the list.
5. Select the license level to assign to the selected users.



The Licensed Level column in the list of users is the updated to reflect the changes. See About License Levels to learn more about each level.

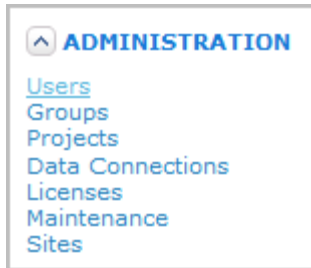
## About User Rights

In addition to the license levels, what you can do on Tableau Server is also affected by your user rights:

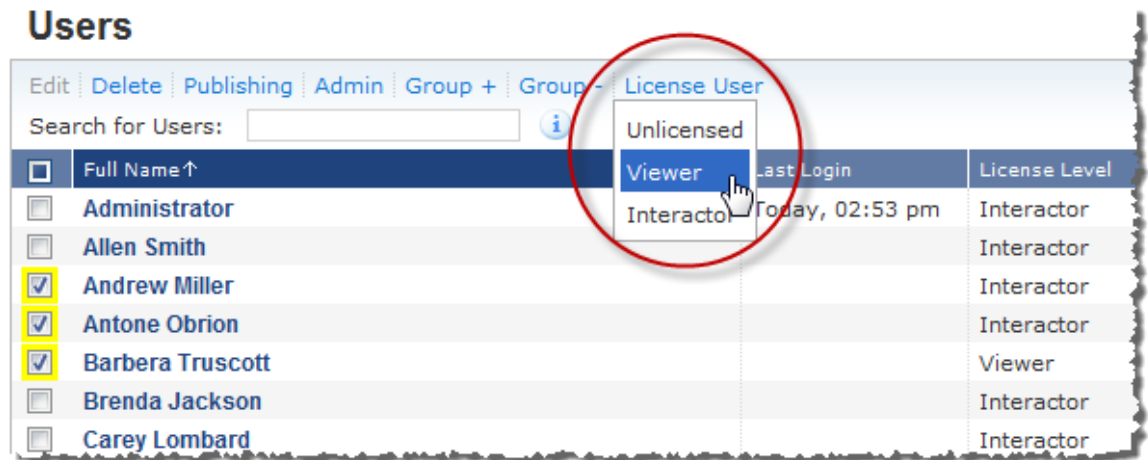
User Right	Description
Publish	Allows the user to connect to Tableau Server from Tableau Professional so that she can publish and download workbooks and data sources.
Admin	<p>Makes the user an administrator. There are two types of administrators: Content administrators and system administrators.</p> <ul style="list-style-type: none"> <li>• <b>Content administrators</b> can manage groups, projects, workbooks, and data connections. By default, content administrators can also add users and assign user rights and license levels but a system administrator can disable that (see <a href="#">Editing Sites</a>). A content administrator cannot license an unlicensed user (change their level from Unlicensed to, for example, Interactor).</li> <li>• <b>System administrators</b> have all the rights of a content administrator, plus they can license unlicensed users, control whether content administrators can add users, create additional system administrators, and they can administer the server itself. This includes handling maintenance, settings, schedules, and the search index.</li> </ul> <p>The Admin right can only be assigned to users with the Interactor license level and the Publish right.</p>

## Allowing or Denying User Rights

1. Log into Tableau Server using your administrator user name and password.
2. Click Users in the Administration area on the left side of the page.

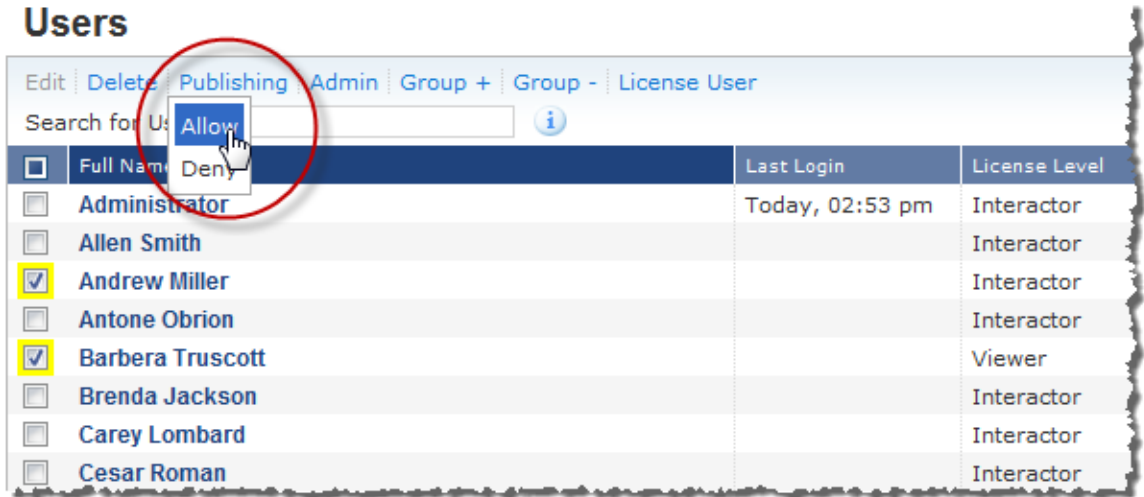


3. Select one or more users you want to assign user rights to.



4. Click the Publishing or Admin links in the Actions toolbar along the top of the list.

5. Select Allow or Deny to change the Publishing right for the select user(s).



6. Select System, Content, or None to change the Admin right for the selected users. The Admin and Publish columns in the list of users are updated to reflect the changes.



## Permissions

What you can do with views, workbooks, projects, and data sources on Tableau Server is controlled by both your license level (specified by an administrator) and the permissions set by the author of the view or data source.

You can change the permissions if you have an Interactor license level and at least one of the following is true:

- You are the owner of the workbook or data source (you published it to the server).
- You have been given permission to Set Permissions.
- You have been given Project Leader permission on the project that contains the view or data source.
- You have been granted the Admin right.

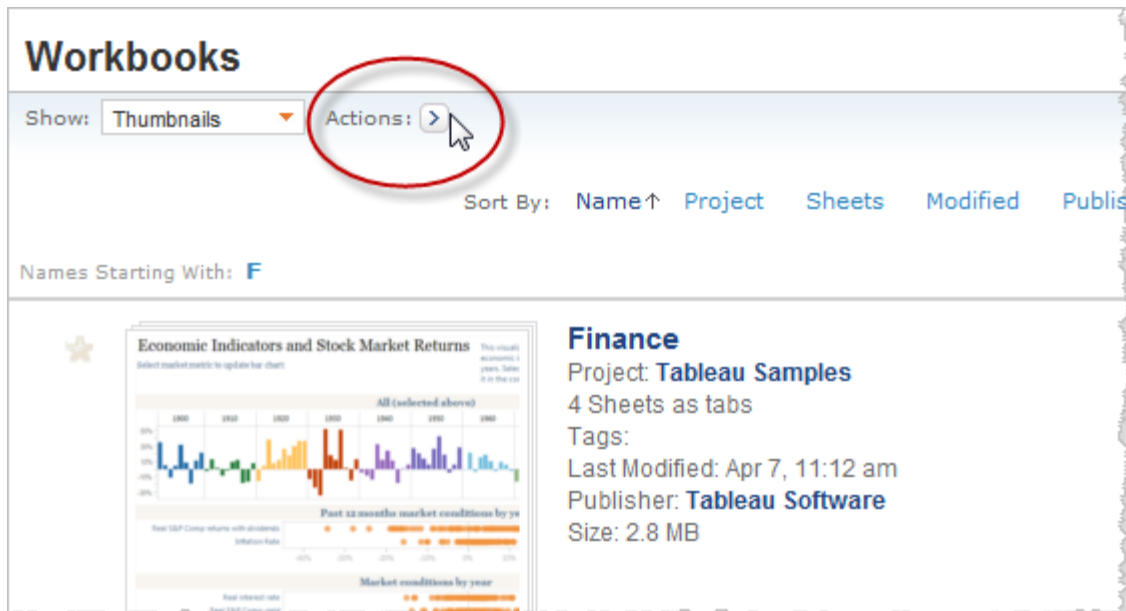
See the following topics for more information:

- [Setting Permissions for Workbooks and Views](#)
- [Setting Permissions for a Data Source](#)
- [Viewing Current Permissions](#)

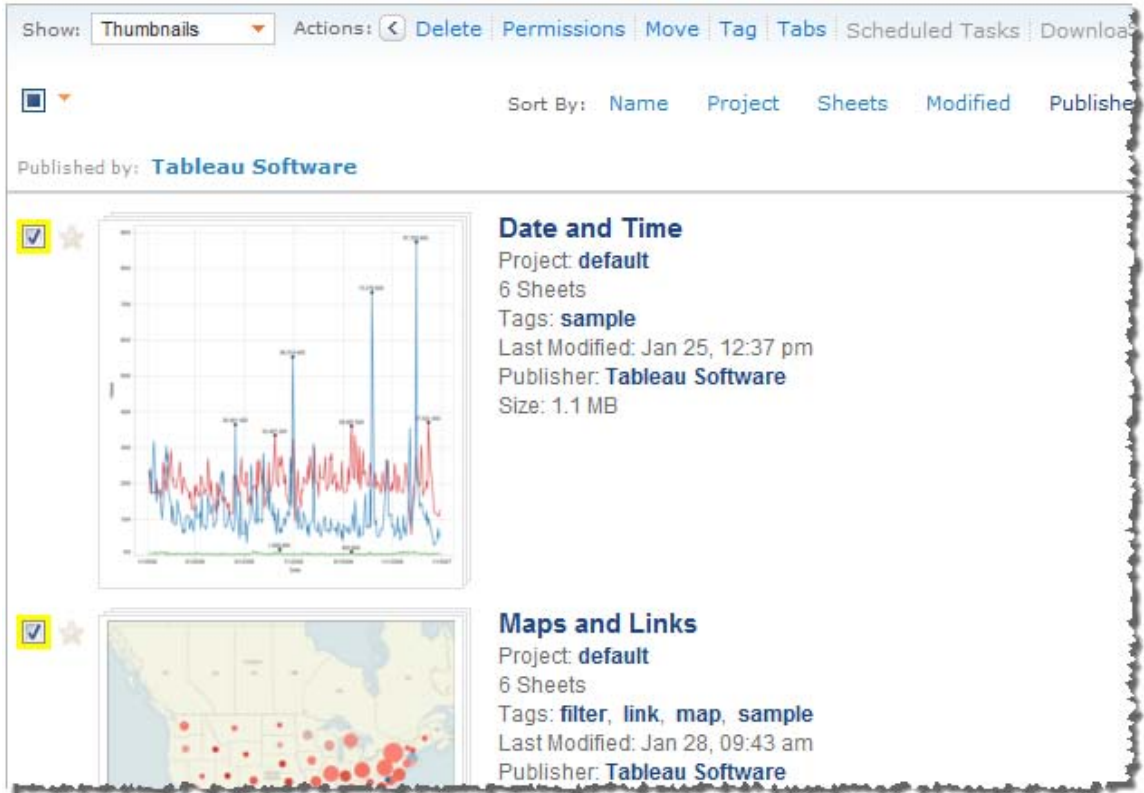
## Setting Permissions for Workbooks and Views

Follow the steps below to set permissions for a workbook or a view.

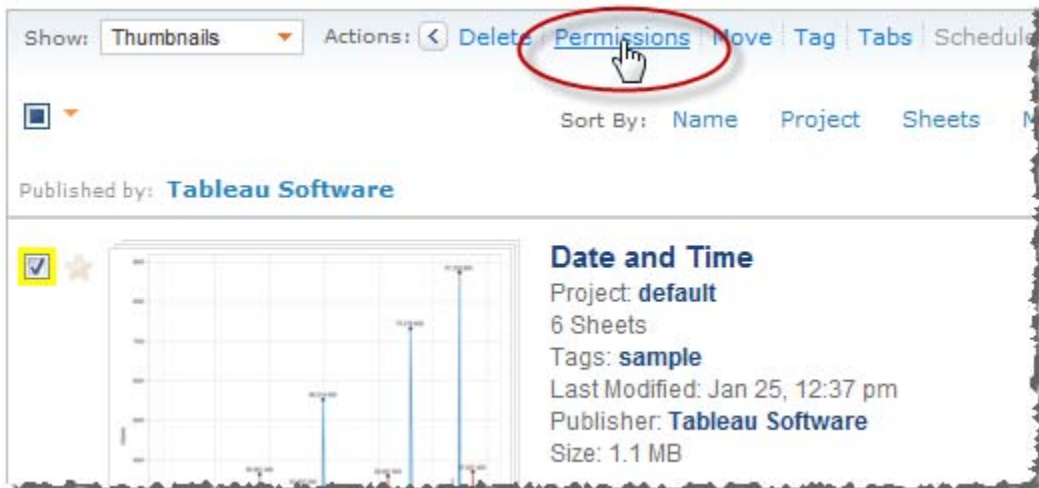
1. Click the Actions link at the top of a list of views or workbooks.



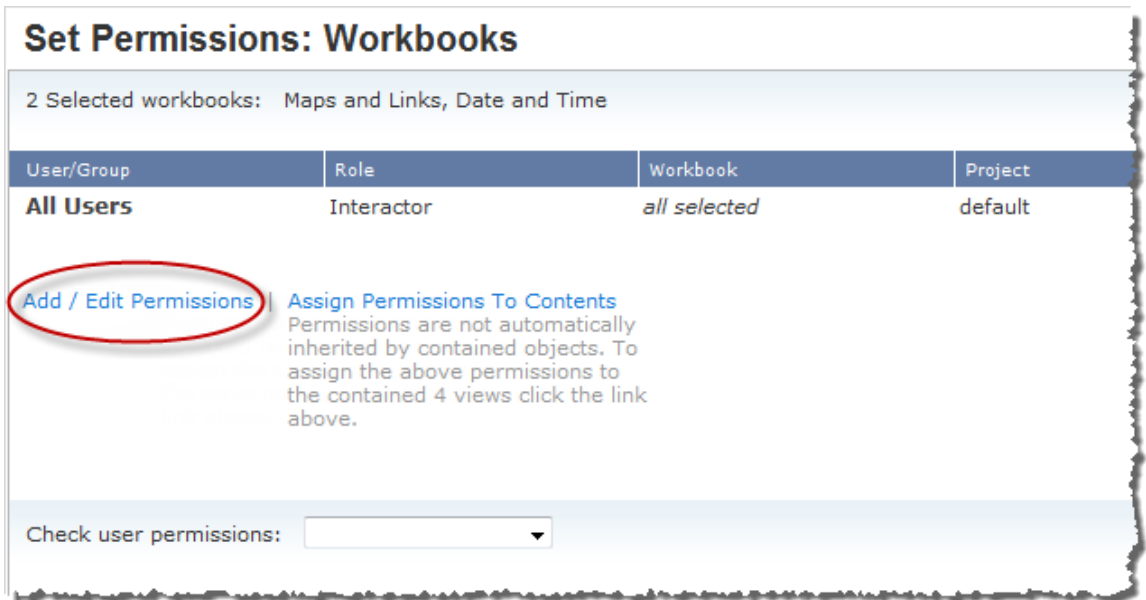
2. Select one or more views or workbooks in the list.



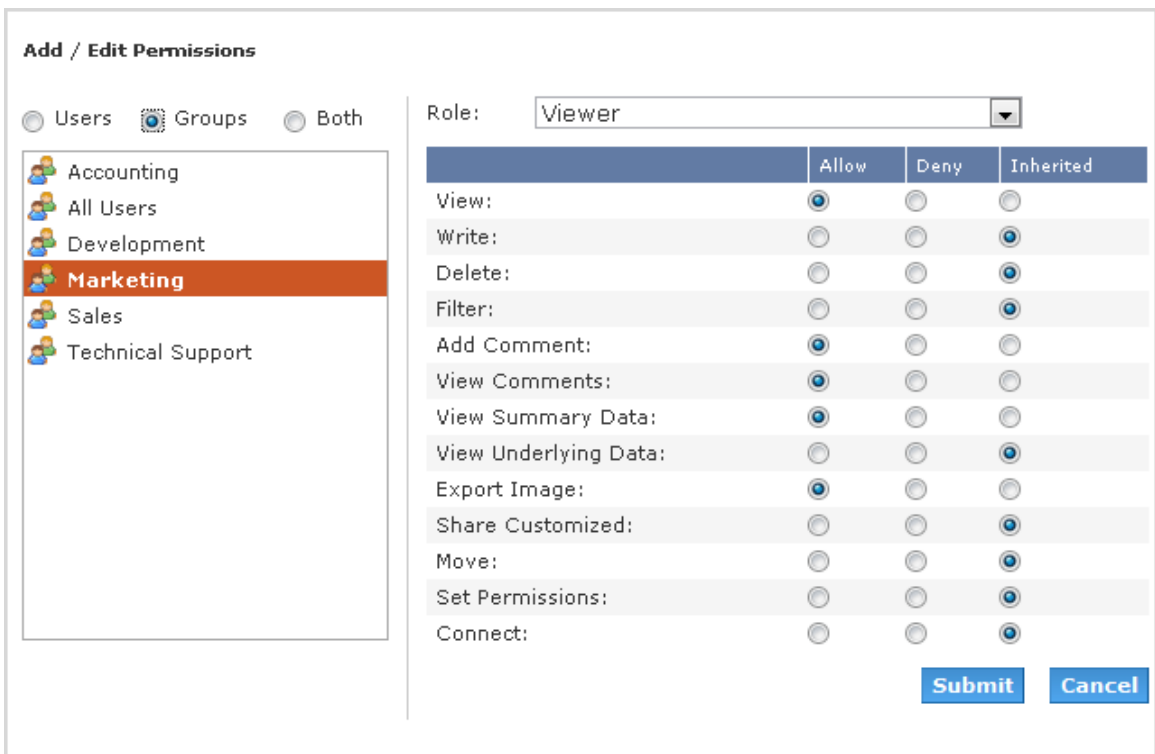
3. Select Permissions on the toolbar at the top of the list.



- Click the Add/Edit Permissions link at the bottom of the permissions list to add permissions that apply to the workbook.



- Select for whom you want to set permissions. You can select an entire group or an individual user. Use the radio buttons along the top to filter the list to show only Users, Groups, or Both.



6. Select the capabilities that you would like to Allow or Deny. If you leave the permission set to Inherited, the user or group's access to the view will be determined by the group or project permissions. To make it easy, you can select a pre-defined set of common capabilities called Roles. You can assign one of the following roles:

Option	Description
Viewer	See the workbook on the server, add and view comments, and export the view as an image or export the data.
Interactor	See and interact with the views (filter, comment, export, etc.).
Editor	See, interact with, manage, and republish views.
Custom	A role that you design.

7. Click OK.

**Note:**

When you set permissions on a workbook or project, the permissions are not automatically inherited by its contained objects. To assign the permissions to each of the objects contained in a workbook or project click the Assign Permissions to Contents link.

### Set Permissions: Workbooks

2 Selected workbooks: Maps and Links, Date and Time

User/Group	Role	Workbook
<b>All Users</b>	Interactor	<i>all selected</i>
<b>Andrew Miller</b>	Viewer	<i>all selected</i>

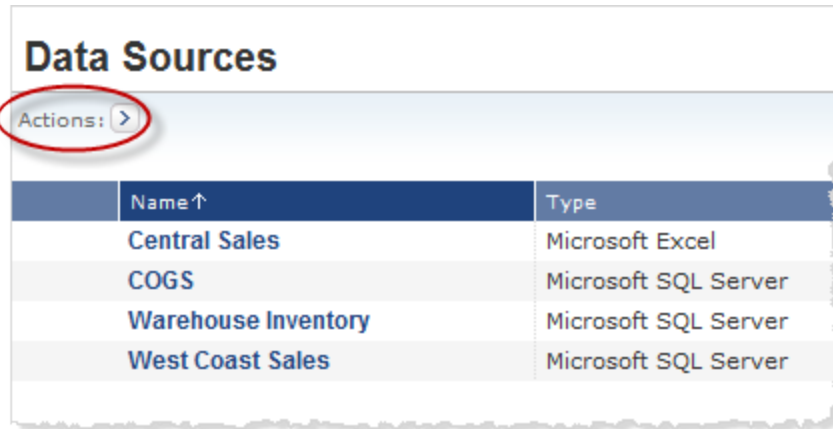
[Add / Edit Permissions](#) [Assign Permissions To Contents](#)

Permissions are not automatically inherited by contained objects. To assign the above permissions to the contained 4 views click the link above.

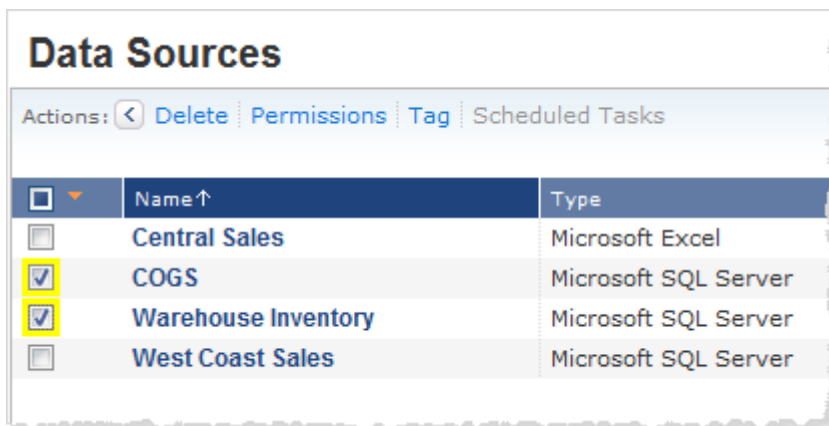
## Setting Permissions for a Data Source

Follow the steps below to set permissions for a data source.

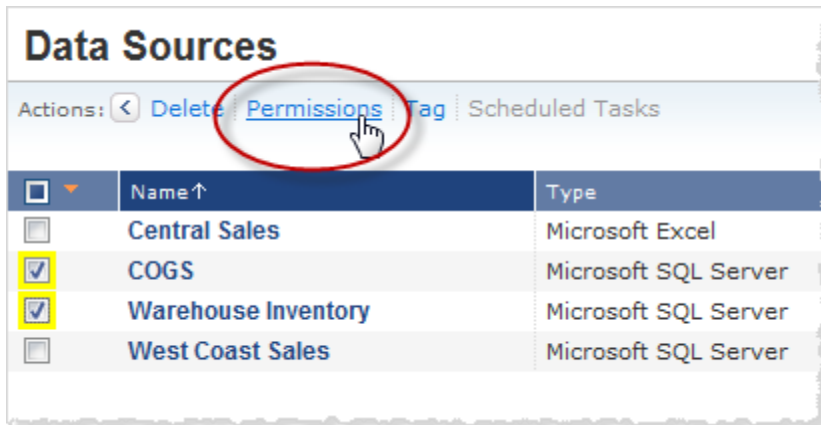
1. On the Data Sources page, click the Actions link at the top of the list of data sources.



2. Select one or more data sources in the list.



3. Select Permissions on the toolbar at the top of the list.



**Data Sources**

Actions: < Delete Permissions Tag Scheduled Tasks

<input type="checkbox"/>	Name ↑	Type
<input type="checkbox"/>	Central Sales	Microsoft Excel
<input checked="" type="checkbox"/>	COGS	Microsoft SQL Server
<input checked="" type="checkbox"/>	Warehouse Inventory	Microsoft SQL Server
<input type="checkbox"/>	West Coast Sales	Microsoft SQL Server

4. Click the Add/Edit Permissions link at the bottom of the permissions list to add permissions that apply to the data source.



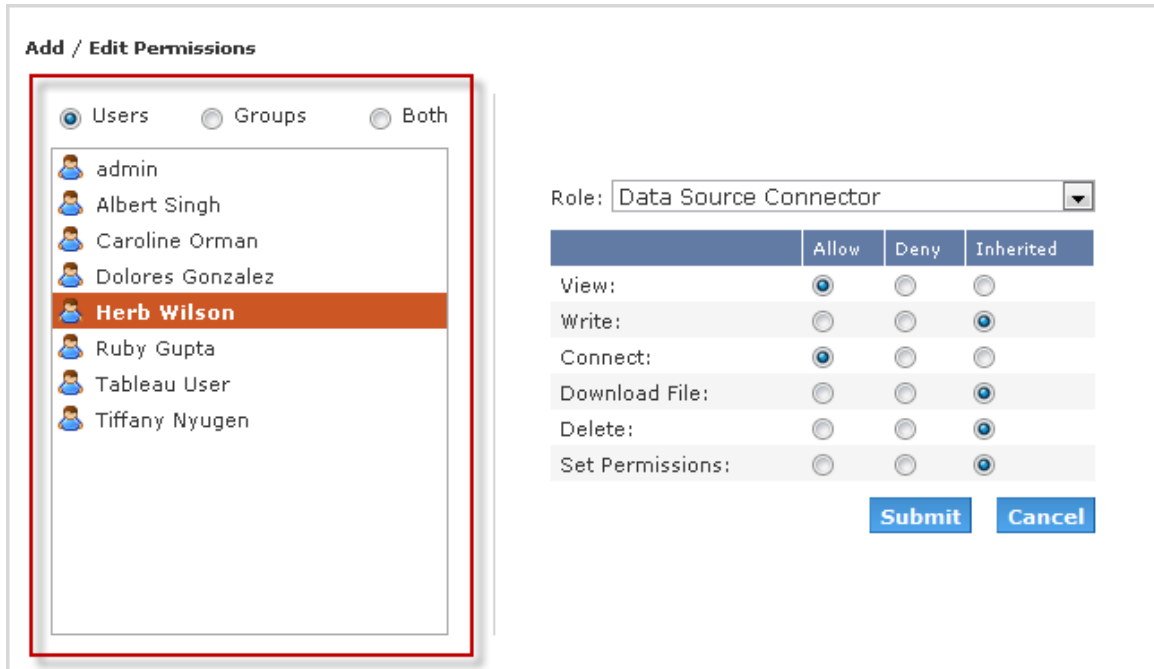
**Permissions: Data Sources**

2 Selected data sources: Warehouse Inventory, COGS

User/Group	Role	Data Source
<b>All Users</b>	Data Source Connector	<i>all selected</i>

[Add / Edit Permissions](#)

5. Select for whom you want to set permissions. You can select an entire group or an individual user. Use the radio buttons along the top to filter the list to show only Users, Groups, or Both.



6. Select the capabilities that you would like to Allow or Deny. If you leave the permission set to Inherited, the user or group's access to the data source will be determined by the group or project permissions. To help make it easier, you can select a pre-defined set of common capabilities called Roles. You can assign one of the following roles:

Option	Description
Data Source Connector	Connect to the data source on the server.
Data Source Editor	Connect to data sources on the server. Publish, edit, download, delete, and set permissions for a data source. Schedule refreshes for data sources you publish.
Custom	A role that you design.

7. A user who accesses a view that connects to a data source must have both View and Connect permission to see the view.
8. Click OK.



## Viewing Current Permissions

At any time, you can see a user's permissions for a particular view, workbook, project, or data source. Additionally, you can see the specific capabilities allowed to a user. Select a user on the Check user permissions drop-down list. Click on a specific capability to see more details.

Check user permissions: <input type="text" value="Henry Wilson"/>			
Capability	Allow	Deny	Reason
<a href="#">View</a>	✓		
<a href="#">Write</a>	✓		
<a href="#">Delete</a>		✓	Not granted by any permission for Area Sales Performance.
<a href="#">Filter</a>	✓		
<a href="#">Add Comment</a>	✓		
<a href="#">View Comments</a>	✓		
<a href="#">View Summary Data</a>	✓		
<a href="#">View Underlying Data</a>		✓	
<a href="#">Export Image</a>	✓		
<a href="#">Share Customized</a>	✓		
<a href="#">Move</a>		✓	Not granted by any permission for Area Sales Performance.
<a href="#">Set Permissions</a>		✓	Not granted by any permission for Area Sales Performance.
<a href="#">Connect</a>	✓		

Click on a capability above to highlight the associated permission.

## Groups and Projects

Groups and Projects are available on Tableau Server to help you organize your workbooks and the people using the server.

- [Groups](#)
- [Projects](#)

### Groups

You can organize users on the Tableau Server into groups to make it easier to assign permissions to multiple people at once. You can either create groups locally on the server or import from Active Directory. You can create and delete groups on the Groups page which shows a list of all groups on the server or site, if the server hosts multiple sites.

- [Viewing Groups](#)
- [Creating Groups](#)
- [Deleting Groups](#)

### Viewing Groups

To see a list of all groups on the server, Click Groups in the Administration section on the left side of the page:



For servers running multiple sites, groups are managed per-site. System administrators can find the Groups link is under This Site:



## Creating Groups

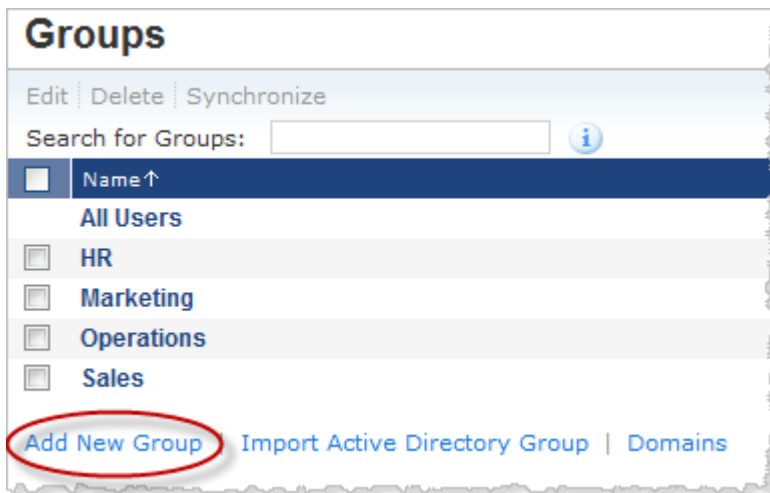
Depending on how the server has been configured you can add groups using the internal user management system (local authentication) or you can import from Active Directory.

- [Creating Local Groups](#)
- [Creating Groups via Active Directory](#)

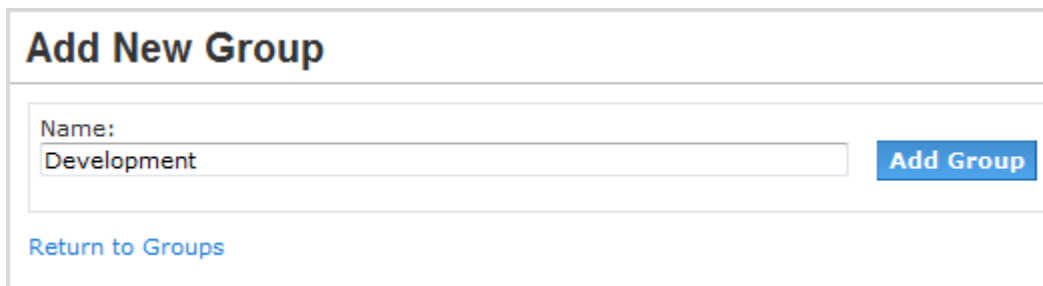
## Creating Local Groups

A local group is one that's created on Tableau Server using the internal user management system. After you create a group you can add and remove users. To create a local group:

1. Click Add New Group at the bottom of the list of groups.



2. Type a name for the group and click Add Group.

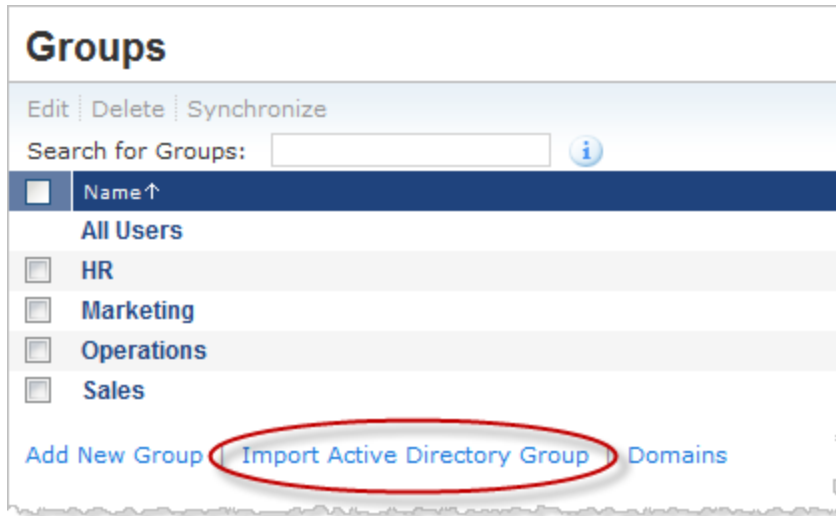


3. Click Return to Groups to return to the list of groups.

## Creating Groups via Active Directory

Groups can also be imported from 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. Each user is unlicensed and does not have permission to publish. If the user already exists on the server, he or she is added to the new group and his or her permissions are not changed. See [Licenses](#) to learn more about license levels and user rights.

1. Click Import Active Directory Group at the bottom of the list of groups.



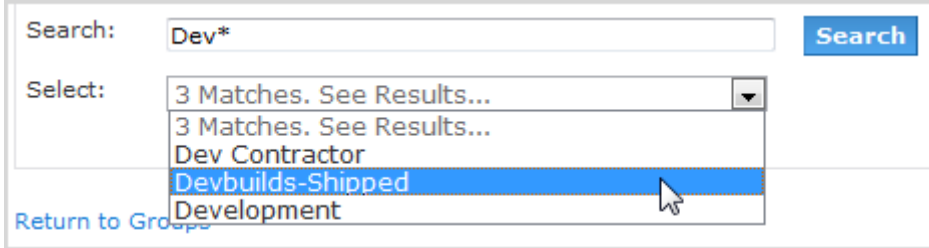
2. Type the name of the Active Directory group you want to import and click Import.



3. If you don't know the exact name of the group you can find it by typing all or part of the group name into the Search text box. Then click Search. You can use the asterisk ( \* ) as a wildcard.

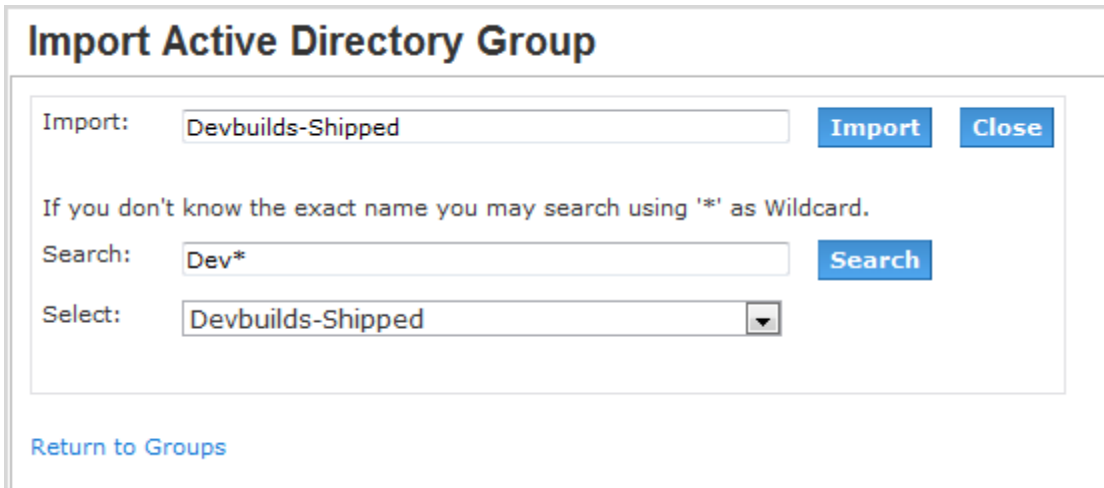


4. Select the group from the list of search results.



The screenshot shows a search interface with a search box containing 'Dev\*' and a 'Search' button. Below the search box is a 'Select:' dropdown menu. The dropdown menu is open, showing three options: '3 Matches. See Results...', '3 Matches. See Results...', and 'Devbuilds-Shipped'. The 'Devbuilds-Shipped' option is highlighted in blue, and a mouse cursor is pointing at it. Below the dropdown menu is a 'Return to Groups' link.

5. The group name is automatically added to the Import text box. Click Import to add the group to Tableau Server.



The screenshot shows the 'Import Active Directory Group' dialog box. The title is 'Import Active Directory Group'. Below the title is an 'Import:' text box containing 'Devbuilds-Shipped'. To the right of the text box are 'Import' and 'Close' buttons. Below the text box is a message: 'If you don't know the exact name you may search using '\*' as Wildcard.' Below the message is a 'Search:' text box containing 'Dev\*' and a 'Search' button. Below the search box is a 'Select:' dropdown menu containing 'Devbuilds-Shipped'. Below the dropdown menu is a 'Return to Groups' link.

**Note:**

You cannot change the name of groups imported from Active Directory. The group name can only be changed in Active Directory.

- [Synchronizing an Active Directory Group](#)

## Synchronizing an Active Directory Group

At anytime, you can synchronize an Active Directory group with Tableau Server so that any new users in Active Directory are also added to the server. You can synchronize individual groups or multiple groups at once.

1. On the Groups page, select one or more groups.
2. Click Synchronize.

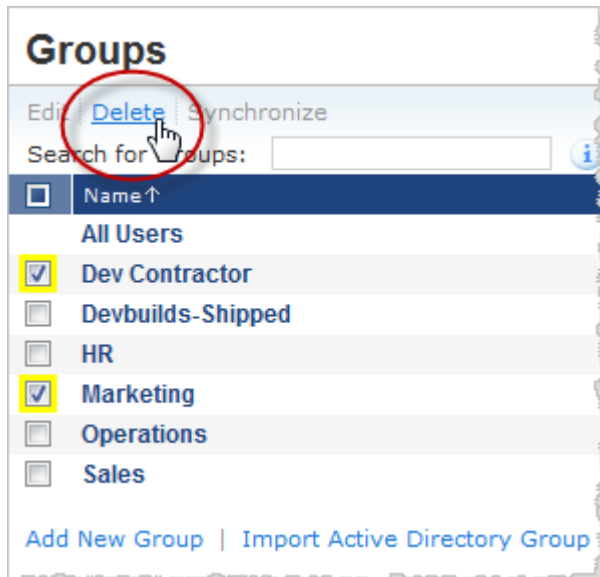


If you are adding a group that is from the same Active Directory domain that the server is running on you can simply type the group name. In addition, if there is a two way trust set up between the domain the server is using and another domain, you can add groups from both domains. The first time you 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 [Modifying Domain Names](#) to learn more about managing domain names.

## Deleting Groups

You can delete any group from the server. When you delete a group, the users are removed from the group but are not deleted from the server. See [Deleting Users](#) for steps on how to delete users.

1. On the Groups page, select one or more groups to delete.
2. Select Delete on the Actions toolbar:





## Projects

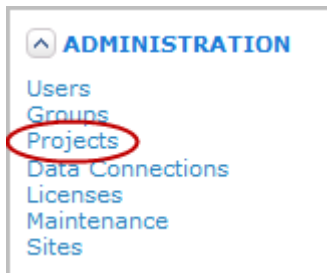
A project is a collection of related workbooks. People browsing the server can find projects by clicking the **Projects** link in the navigation area on the left side of the page. They can also search for a specific project or filter and sort lists by project. Projects are useful for collectively setting permissions on all of the workbooks in the project. While only administrators can create new projects, users and groups can be assigned the Project Leader permission role.

The Project Leader role allows a user or group administrative access to all of the workbooks contained in the project, including the ability to specify project permissions. See the topics below for procedures and more information on working with projects:

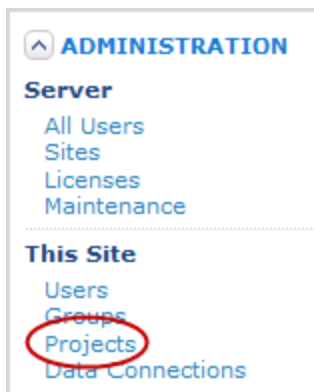
- [Viewing Projects](#)
- [Adding Projects](#)
- [Specifying Project Permissions](#)
- [Moving Workbooks into Projects](#)
- [Deleting Projects](#)

### Viewing Projects

To see a list of all projects on the server, Click Projects in the Administration section on the left side of the page.



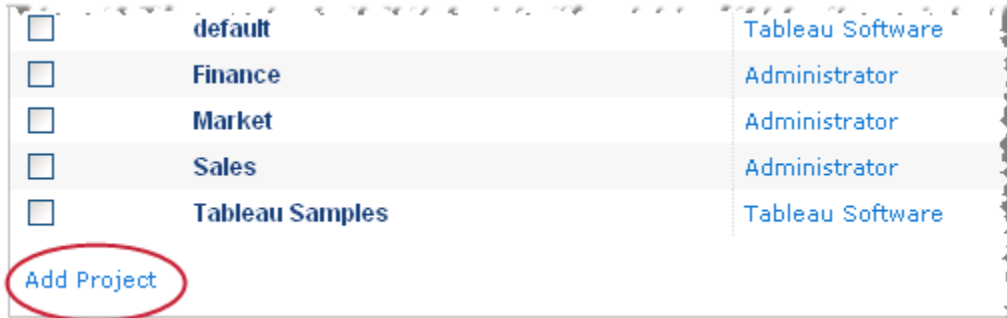
For servers running multiple sites, projects are managed per-site. System administrators can find the Projects link under This Site:



## Adding Projects

To add a project:

1. Click the Add Project link at the bottom of the project list.



2. Type a name and description for the project and click Add Project. You can include formatting and hyperlinks in the project description.

Name:

Description: (852 characters remaining)

Format sample: "MyLink":http://www.example.com \*bold\* \_italics\_ +underline+

Displays as: [MyLink](#) **bold** *italics* underline

[Add Project](#)

3. Click the Return to Projects link to return to the list of projects.

## Specifying Project Permissions

Administrators and Project Leaders can specify project permissions. When you create a new project, it is given the same permissions as the default project. You can specify permissions for the project to allow or deny individual users and groups permission to access the project. To specify project permissions:

1. Select the project in the project list.
2. Click the Permissions link in the Actions toolbar along the top of the list.
3. Add or edit the permissions for the project. To learn more, see [Setting Permissions for Workbooks and Views](#).



The permissions you specify apply to the project itself. Any permissions set on the workbooks and views contained in the project are not affected.

You have, however, the option to assign the project permissions to all of the workbooks and views contained in the project. In that case, the new permissions override the existing permissions on the workbooks and views. For example, say there are several workbooks that have each been published with custom permissions and you group the workbooks into a new project with a new permission set. You can apply the new permissions to each of the workbooks by clicking the Assign Permissions to Contents link on the Permission page.

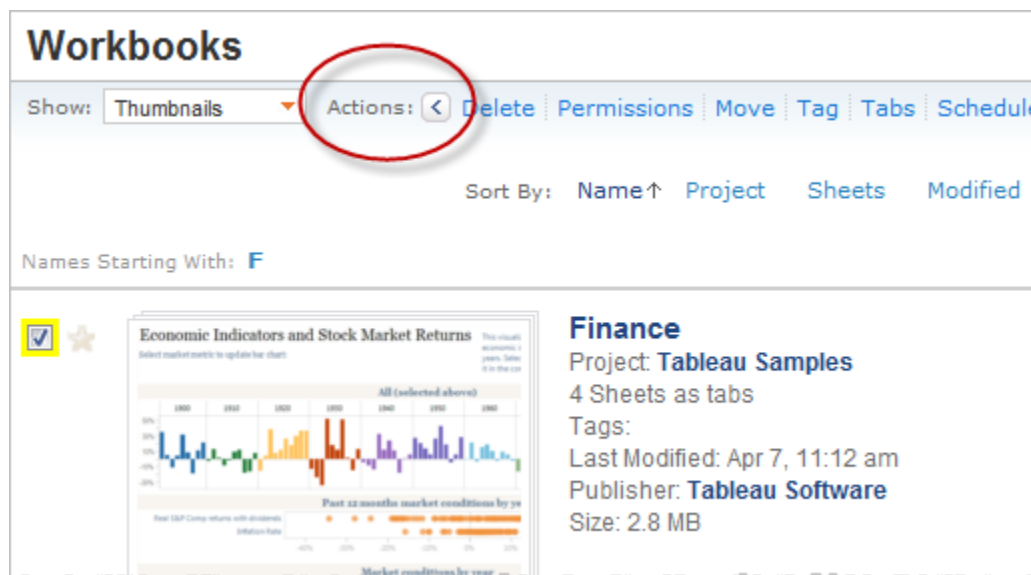
## Moving Workbooks into Projects

All workbooks must be part of a project. By default, workbooks are added to an automatic project called default. After you've created your own projects you can move workbooks into them from any list of workbooks. You can move workbooks into projects if you have an Interactor license level and at least one of the following is true:

- You have been given permission to Write to the project.
- You have been given Project Leader permission for the project.
- You have been granted the Admin right.

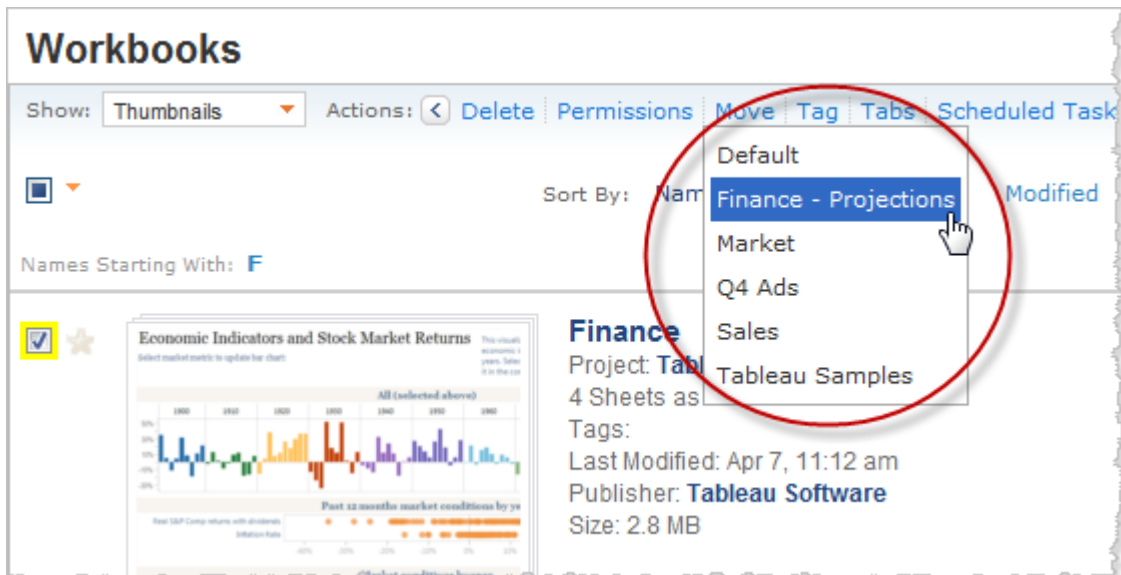
To move a workbook into a project:

1. Click the Actions link at the top of the list of workbooks.



2. Select one or more workbooks to move.
3. Click Move in the Actions toolbar along the top of the list.

4. Select a project to move the workbook into.



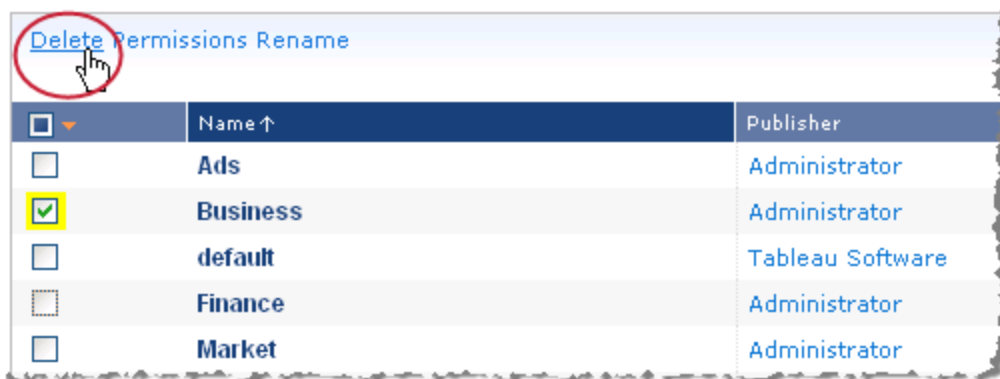
**Note:**

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 part of a single project.

## Deleting 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. To delete a project:

1. Select the project in the project list.
2. Click Delete on the Actions toolbar along the top of the list.
3. Click OK in the warning dialog box.



### Note:

The default project cannot be deleted, even if you have been granted permission to delete this project.

## Scheduling Tasks

Tasks are jobs performed by Tableau Server and schedules control when many of the tasks are run.

As the server administrator you have the highest level of control over server tasks and schedules, however, extract refreshes are one server task that workbook authors can also schedule when they publish a workbook or a data source with an extract. As the administrator, you can adjust an extract's schedule, create new schedules and refresh tasks, and delete them. You also control whether workbook authors are allowed to schedule (see [Enabling Scheduling](#)). Any changes you make to an extract's schedule are reflected in the Tableau Desktop Schedule dialog box the next time the author publishes.

See the topics below for more information:

- [Managing Schedules](#)
- [Managing Tasks](#)

## Managing Schedules

The schedules in Tableau Server help you control when and how certain server tasks are performed. For example, you may want to schedule extract refreshes for a time when database usage is low, such as on Saturday nights. To further minimize the database impact, you can specify that the schedule's tasks should occur sequentially instead of concurrently.

Tableau Server comes with three schedules. You can use the schedules as they are, modify them, or create your own. The Schedules page shows a list of schedules including their name, type, number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

- [About Extracts and Schedules](#)
- [Enabling Scheduling](#)
- [Adding Workbooks to Schedules](#)
- [Adding Data Sources to Schedules](#)
- [Opening the Schedules Page](#)
- [Creating a New Schedule](#)
- [Modifying a Schedule](#)



## About Extracts and Schedules

Tableau Desktop allows authors to create a data extract, which is a copy or a subset of data from the original data source. Workbooks that use data extracts are generally faster than those that use live database connections because the extracted data is imported into Tableau's built-in, fast data engine. Extracts can also increase functionality.

When an author publishes a workbook or data source that uses an extract, he or she can assign it to a recurring refresh schedule. The author chooses the schedule from a drop-down list in the Schedule dialog box in Tableau Desktop. At the interval the author chooses, Tableau Server refreshes all the data in the extract.

Authors can also optionally define an incremental update for an extract, where they identify a column in the extract that has a numerical value, such as a timestamp. (Specifically, the value must be an integer, up to 18 digits long. Dates and datetime are both valid.) Tableau uses this column to identify new rows that need to be added to your extract. This is called an incremental refresh and it can also be scheduled.

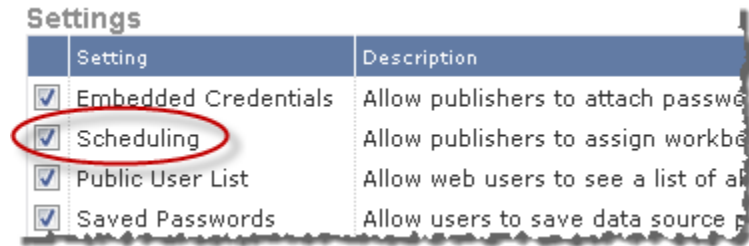
As the administrator, you can change or reassign refresh schedules on Tableau Server, regardless of whether a workbook or data source with an extract was given a refresh schedule at the time it was published. Any changes you make in Tableau Server are reflected in the Schedule dialog box in Tableau Desktop when the workbook or data source is published again.

Before you can create refresh schedules you must enable scheduling on the server. See [Server Maintenance](#) to learn more.

## Enabling Scheduling

Before you can schedule an extract refresh, scheduling must be enabled on the server. After you enable scheduling, you can add workbooks and data sources to schedules, create and edit schedules, manage scheduled tasks, and change schedule settings to allow publishers to assign workbooks to schedules.

To enable scheduling, select the Scheduling checkbox under Settings on the Server Maintenance page:



Settings	
Setting	Description
<input checked="" type="checkbox"/> Embedded Credentials	Allow publishers to attach passwords to data sources.
<input checked="" type="checkbox"/> Scheduling	Allow publishers to assign workbooks to schedules.
<input checked="" type="checkbox"/> Public User List	Allow web users to see a list of all users.
<input checked="" type="checkbox"/> Saved Passwords	Allow users to save data source passwords.

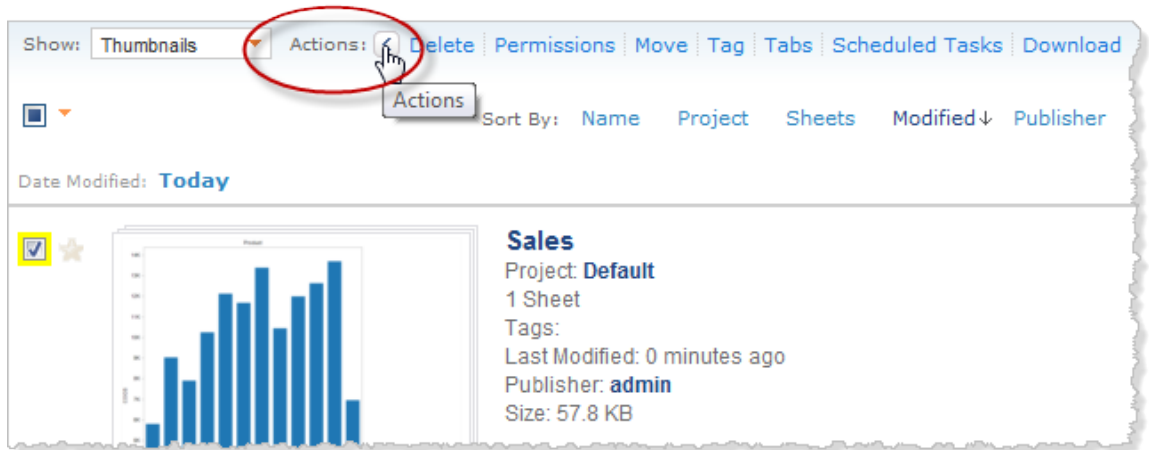
**Note:**

Because database passwords may be required to refresh the extract, you must enable Embedded Credentials in order to allow scheduling.

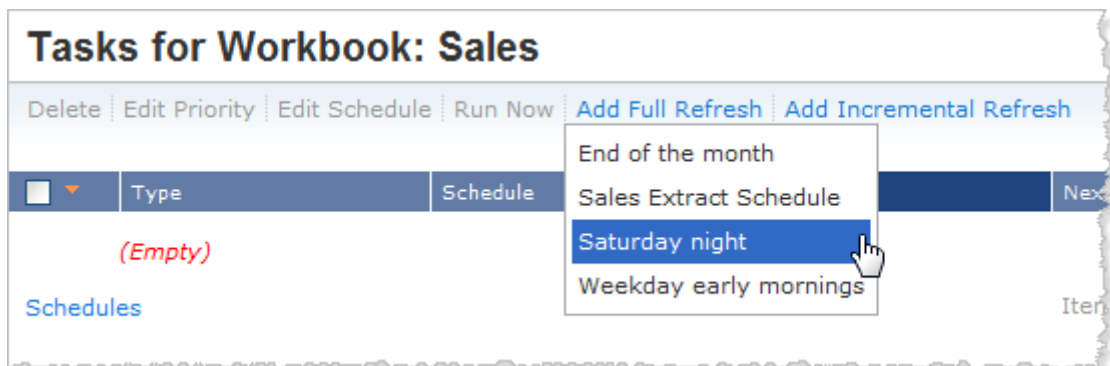
## Adding Workbooks to Schedules

Once you've enabled schedules you can add workbooks to schedules from the Workbooks list. By default Tableau Server has three built in schedules. To learn how to create your own, see [Creating a New Schedule](#).

1. On the workbooks page click the Actions button to show the toolbar.



2. Click Scheduled Tasks in the toolbar.
3. Select one or more workbooks you want to schedule for refresh.
4. Select a refresh task, then a schedule from the list:

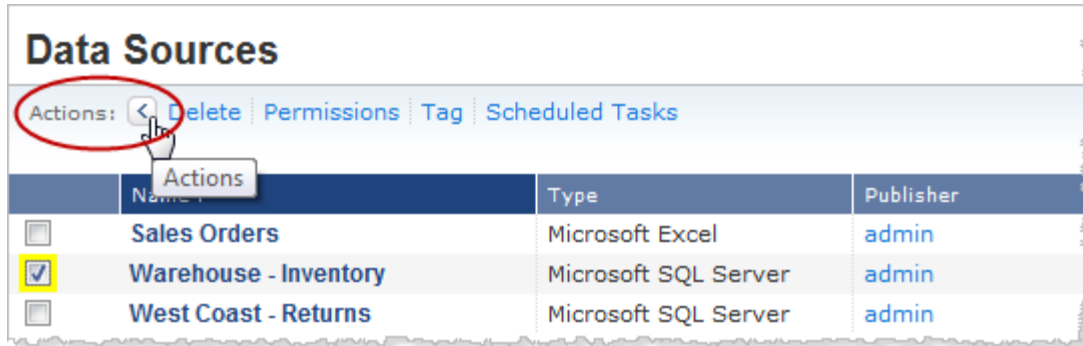


Add Full Refresh is only available if the selected workbooks connect to an extract data source. Add Incremental Refresh is only available if the selected workbooks connect to an extract data source for which you've defined an incremental refresh. Tableau Server cannot refresh workbooks that connect to local file data sources on a mapped drive. Update the connection to use the full path to the data source to allow scheduled refresh.

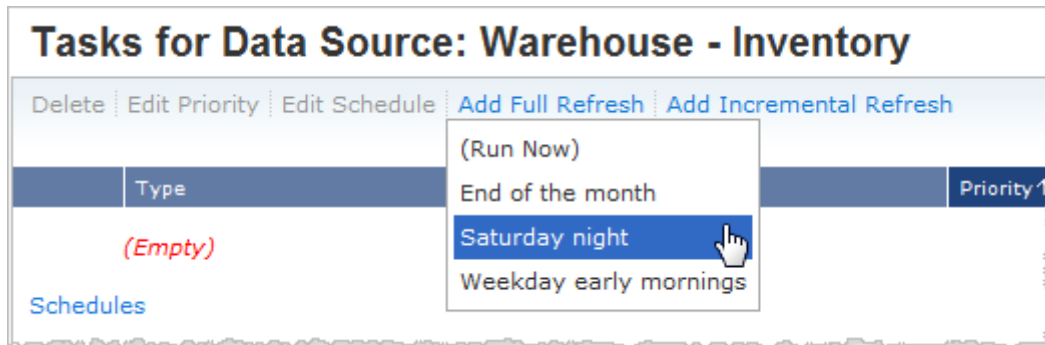
## Adding Data Sources to Schedules

Once you've enabled schedules you can add a data source to a schedules from the Data Sources list. By default Tableau Server has three built in schedules. To learn how to create your own, see [Creating a New Schedule](#).

1. On the Data Sources page click the Actions button to show the toolbar.



2. Select one or more data sources you want to schedule for refresh.
3. Click Scheduled Tasks in the toolbar.
4. Select either Add Full Refresh or Add Incremental Refresh, then select a schedule from the list:



Add Full Refresh is only available if the selected data source connects to an extract. Add Incremental Refresh is only available if the selected data source connects to an extract data source for which you've defined an incremental refresh. Tableau Server cannot refresh data sources that connect to local file data sources on a mapped drive. Update the connection to use the full path to the data source.

## Opening the Schedules Page

The Schedules page shows a list of schedules, including their name, type, number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

Open the Schedules page by clicking the Maintenance link in the Administration section on the left side of the page, then click Schedules:

**ADMINISTRATION**

- Server
  - All Users
  - Sites
  - Licenses
  - Maintenance

**Maintenance**

Service waiting for req

**Status**

Machine	Repository	Data Engine
127.0.0.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Analysis**

View	Analysis
Server Activity	A da
User Activity	A vie
View Performance History	A vie
Background Tasks	A vie
Space Usage	A da
Customized Views	A da

**Schedules**

- Schedules
- Tasks

If you are a content administrator in a multi-site system, the Maintenance page displays fewer settings:

**ADMINISTRATION**

- Users
- Groups
- Projects
- Data Connections
- Maintenance

**Site Maintenance**

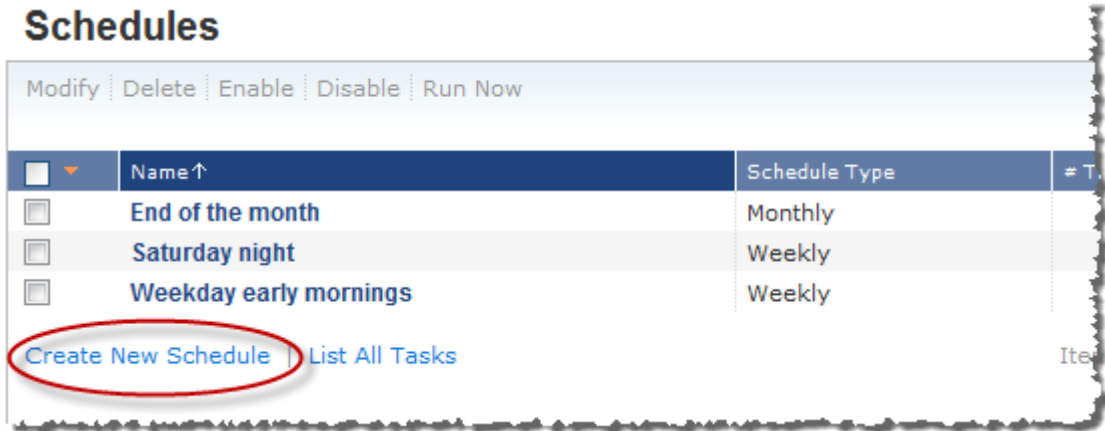
**Schedules**

- Schedules
- Tasks

## Creating a New Schedule

The Schedules page shows a list of schedules, including their name, type, number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run. To create a new schedule:

1. Click Create New Schedule at the bottom of the Schedules list.



2. Specify a descriptive Name for the schedule (e.g., Every Saturday Morning, End of the Month).
3. 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 highest priority runs first. See [Managing Tasks](#) to learn more about modifying a task's priority.
4. Choose whether the jobs in the schedule will run at the same time (concurrently, the default) or one after the other (sequentially).

5. Finish defining the schedule. You can define an hourly, daily, weekly, or monthly schedule.

## Create New Schedule

**Schedule Properties:**

Schedule name:

Default priority:  (from 1-100, 1 is highest priority, 100 is lowest)

Serialization:  Jobs in schedule can run concurrently  
 Run jobs sequentially

**Schedule Definition:**

Hourly every  from  :  AM  to  :  AM

Daily at  :  AM

Weekly  Sunday at  :  AM   
 Monday  
 Tuesday  
 Wednesday  
 Thursday  
 Friday  
 Saturday

Monthly on the  day of the month at  :  AM

[Create Schedule](#)

[Return to Schedules](#)

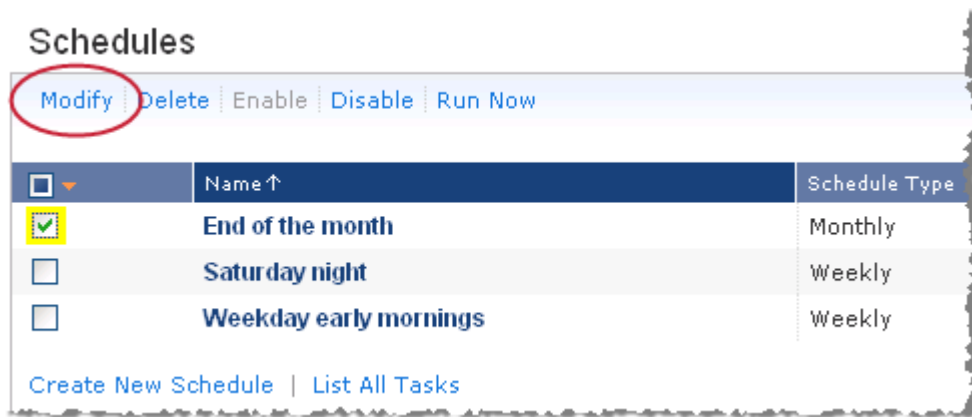
6. Click Create Schedule.

## Modifying a Schedule

The Schedules page shows a list of schedules, including their name, type, number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run. System administrators can modify, delete, enable, disable, or run schedules. Content administrators can run schedules.

To modify a schedule:

1. On the Schedules page, select the schedule you want to modify. You can only modify one schedule at a time.
2. Click Modify in the toolbar.



3. Change the schedule as needed.
4. Click **Save Schedule**.

You can also perform actions other than modifying a schedule (such as enabling or deleting) to a group of schedules. Just select one or more schedules in the list and choose an option on the toolbar.



## Managing Tasks

You can see a list of all scheduled tasks on the Tasks page. There you can change a task's priority, move it to different schedule, run it, or delete it.

You can open the Tasks page by clicking the Maintenance link in the Administration section, then click Tasks:

The screenshot shows the Administration section of a server management interface. The 'ADMINISTRATION' section is expanded to show 'Server' options: 'All Users', 'Sites', 'Licenses', and 'Maintenance'. The 'Maintenance' page is open, displaying a 'Status' section with a green checkmark and the text 'Service waiting'. Below this is a table with columns 'Machine', 'Repository', and 'Date'. The first row shows '127.0.0.1' in the 'Machine' column and a green checkmark in the 'Repository' column. Underneath the table is an 'Analysis' section with a 'View' button and several links: 'Server Activity', 'User Activity', 'View Performance History', 'Background Tasks', 'Space Usage', and 'Customized Views'. At the bottom, there is a 'Schedules' section with a 'Schedules' link and a 'Tasks' link, which is circled in red. A red arrow points from the 'Maintenance' link in the left sidebar to the 'Tasks' link in the 'Schedules' section.

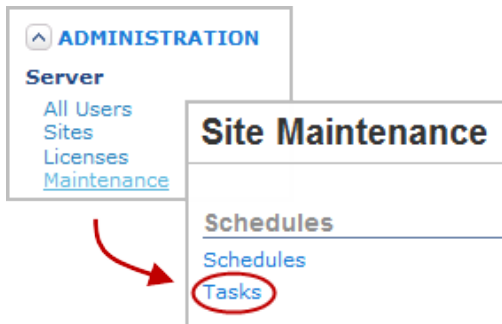
Machine	Repository	Date
127.0.0.1	✓	

**Schedules**

[Schedules](#)

[Tasks](#)

If you are a content administrator in a multi-site system, the Maintenance page displays fewer settings:

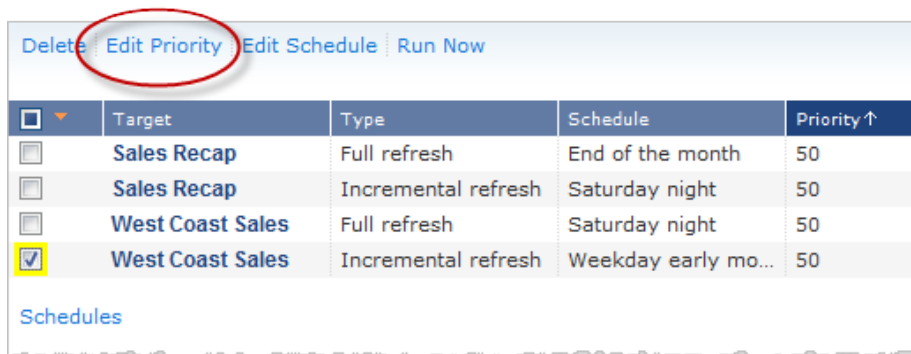


- [Changing a Task's Priority](#)
- [Moving a Task](#)

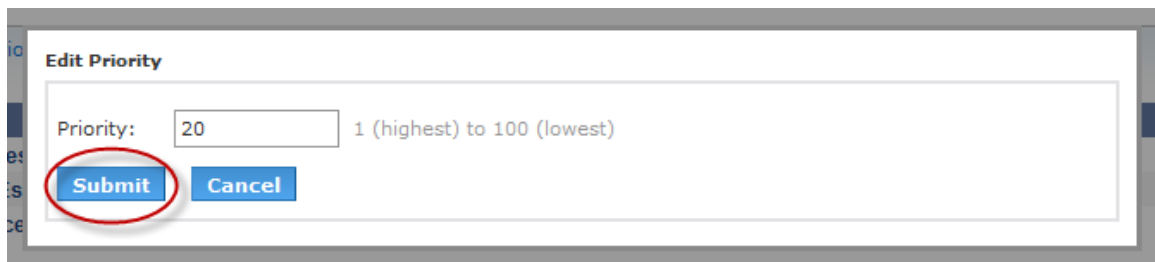
### Changing a Task's Priority

Follow the procedure below to change a task's priority:

1. On the Tasks page select one or more tasks to modify.
2. Click Edit Priority on the toolbar.



3. Type a new priority from 0 to 100 and click Submit.



## Moving a Task

Follow the procedure below to move a task to a new schedule:

1. On the Tasks page select one or more tasks to modify.
2. Click Edit Schedule on the toolbar.
3. Select a new schedule from the list of schedules.

The screenshot shows a 'Tasks' interface with a toolbar and a table. The toolbar includes 'Delete', 'Edit Priority', 'Edit Schedule', and 'Run Now'. A dropdown menu is open under 'Edit Schedule', showing 'End of the month' and 'Weekday early mornings'. The table below has columns for 'Target', 'Full refresh', 'Schedule', and 'Priority'. The 'West Coast Sales' task with 'Full refresh' is selected, and its current schedule is 'Saturday night'. Below the table is a 'Schedules' link.

<input type="checkbox"/>	Target	Full refresh	Schedule	Priority ↑
<input type="checkbox"/>	Sales Recap	Full refresh	End of the month	50
<input type="checkbox"/>	Sales Recap	Incremental refresh	Saturday night	50
<input checked="" type="checkbox"/>	West Coast Sales	Full refresh	Saturday night	50
<input type="checkbox"/>	West Coast Sales	Incremental refresh	Weekday early mo...	50

[Schedules](#)

You can also delete and run tasks by selecting one or more tasks in the list and selecting an option on the toolbar.

## Sites

Use the Sites page to create independent sites for different organizations or groups on a single server system. Each site's workbooks, data, and user lists are isolated from those of other sites. As the system administrator, only you can see every site and perform actions such as creating sites and making system-wide changes. See the topics below for more information:

- [Working with Sites](#)
- [Adding Sites](#)
- [Adding Users to a Site](#)
- [Editing Sites](#)
- [Deleting Sites](#)
- [Multi-Site Navigation](#)

## Working 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 Login Credentials

All sites on a server use the same Server Run As account and user authentication mode. You choose both of these settings when you install Tableau Server. See [Configuring General Settings](#) 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 username of jsmith and a password of “MyPassword” on Site A, she uses those same credentials on Site B. When she logs into 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 log into when you log into 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 has no corresponding web folder named “default”. For example, the URL for a view named Profits on a site named Sales is `http://localhost/t/sales/views/profits`. The URL for this same view on the Default site would be `http://localhost/views/profits`.

## The Content and System Administrator Roles

There are two types of administrators in Tableau Server, system administrators and content administrators. System administrators can control whether content administrators can add and remove users in the Add New Site (or Edit Site) dialog box:

**Add New Site**

Name:

Site ID:

Site URL:

Add Users:

- System Administrators only
- System and Content Administrators
  - Up to server capacity
  - Maximum site users:

If System Administrators only is selected, content administrators cannot add or remove site users. However, they can still manage groups, projects, workbooks, and data connections within their site. If System and Content Administrators is selected (the default), content administrators can do all of the above, and add or remove users.

## Licensing and User Limits

Users can belong to multiple sites, with different user rights and license levels 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.

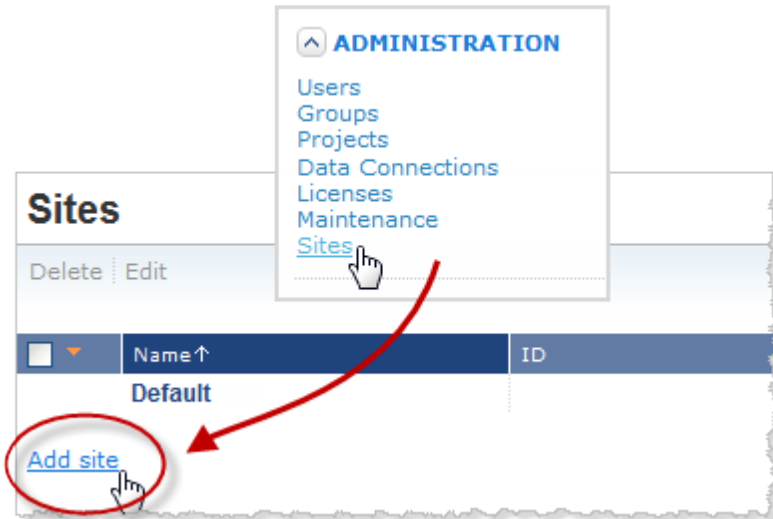
System administrators can use the Maximum site users <n> setting to specify a user limit for a site. Only licensed users are counted; system administrators are excluded. For example, if a site has 90 licensed users, 20 unlicensed users, and one system administrator, the user count is 90. If Maximum site users is set to 100, 10 more licensed users can be added.

System administrators can change a server user's license level from unlicensed to licensed (for example, to Interactor). Content administrators cannot do this.

## Adding Sites

If you are a system administrator, you can add a site to Tableau Server by doing the following:

1. Open the Sites page by clicking Sites under Administration, then click **Add site**:



2. Enter a **Name** and **Site ID** name for the new site:

**Add New Site**

Name:

Site ID:

Site URL:

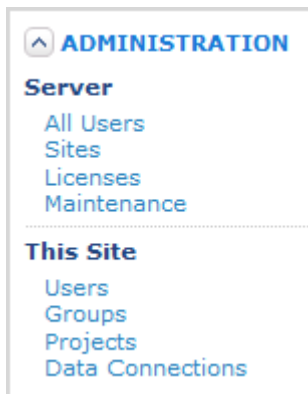
Add Users:

- System Administrators only
- System and Content Administrators
  - Up to server capacity
  - Maximum site users:

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

3. Select whether only you, the system administrator, can add and remove users (System Administrators only) or if it can be done by both types of administrators (System and Content Administrators).
4. If you are allowing content administrators to add users, specify how many users they can add to the site by selecting one of the following:
  - Up to server capacity: 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.
  - Maximum site users <n>: Allows a content administrator to add users up to a limit you specify. See [Working with Sites](#) for information on licensing and user limits.
5. Click OK.

Now, the Administration category on the left is divided into two sections:



The page links under Server lead to server-wide settings. As the system administrator, only you can access this section. Settings under This Site pertain to the site you're currently logged into. System and content administrators can access this section.



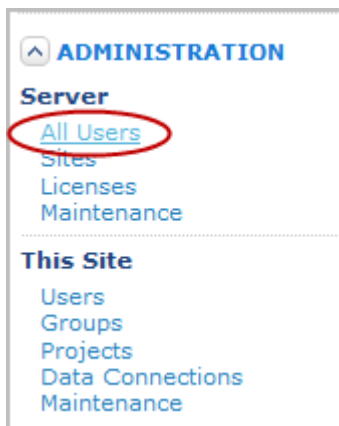
## Adding Users to a Site

See the topics below for more information:

- [Adding Users to a Site from All Users](#)
- [Adding Users to a Site from Users](#)

### Adding Users to a Site from All Users

As the system administrator of a multi-site system, only you can access the All Users page. It's the only place where you can add users to multiple sites all at once, remove users, and if the server is using local authentication, reset user passwords.



There are two ways you can add users from here: one at a time (described below) or in batches using a CSV file (described in [CSV Guidelines](#)).

To add a single user:

1. From the All Users page, click the Add User link at the bottom of the list of users.
2. Enter a Username:
  - *Local authentication*—If the server is using local authentication, using an email address for the username is the best way to avoid username collisions (for example, jsmith@myco.com instead of jsmith).
  - *Active Directory*—If the server is using Active Directory for user authentication and the domain is different than the server domain, include the fully qualified domain name for the first user you add. Subsequent users can use the domain nickname. Refer to [Modifying Domain Names](#) to learn how to modify the nickname.
3. If the server is using local authentication, provide the following:
  - Full Name—Type a display name for the user (e.g., John Smith).

- Password—Type a password for the user.
- Confirm—Retype the password.
- 4. Site Membership—Select which site(s) the user should be a member of. The site you are logged into is selected by default.
- 5. License Level and User Rights—Select a license level, Admin role, and whether the user can publish workbooks and data sources. A user who belongs to multiple sites can have different license levels and user rights on each site. Refer to [About License Levels](#), [Permissions](#), and [About User Rights](#) to learn more.
- 6. Click OK.

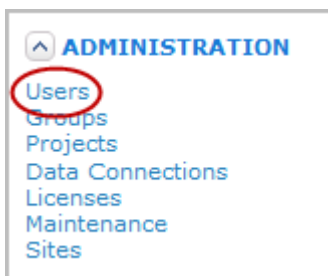
### Adding Users to a Site from Users

Both system administrators and content administrators with the correct permissions can add users from the Users page. For a system administrator, Users is located under This Site:



When you add users from here, it only adds them to the site you're currently logged into. If you want to assign users to multiple sites or reset passwords, go to the All Users page under Administration.

For content administrators, Users is located under Administration:



There are two ways you can add users from this page: one at a time (described below) or in batches using a CSV file (described in [CSV Guidelines](#)).

To add a single user:

1. From the Users page, click the Add User link at the bottom of the list of users.
2. Enter a Username:
  - *Local authentication*—If the server is using local authentication, using an email address for the username is the best way to avoid username collisions (for example, `jsmith@myco.com` instead of `jsmith`).  
  
If the username already exists on the server system, but not on the site, a dialog appears that lets you either enter a different username or add the user to the site. If you add the user, skip to step 4.
  - *Active Directory*—If the server is using Active Directory for user authentication and the domain is different than the server domain, include the fully qualified domain name for the first user you add. Subsequent users can use the domain nickname. Refer to [Modifying Domain Names](#) to learn how to modify the nickname.  
  
If the username already exists on the server system, but not on the site, a dialog appears that lets you either enter a different username or add the user to the site. If you add the user, skip to step 4.
3. If the server is using local authentication and you're adding a user who's new to the server as well as to the site, provide the following:
  - Full Name—Type a display name for the user (e.g., John Smith).
  - Password—Type a password for the user.
  - Confirm—Retype the password.
4. License Level and User Rights—Select a license level, Admin role, and whether the user can publish workbooks and data sources. A user who belongs to multiple sites can have different license levels and user rights on each site. Refer to [About License Levels](#), [Permissions](#), and [About User Rights](#) to learn more.
5. Click Add User.

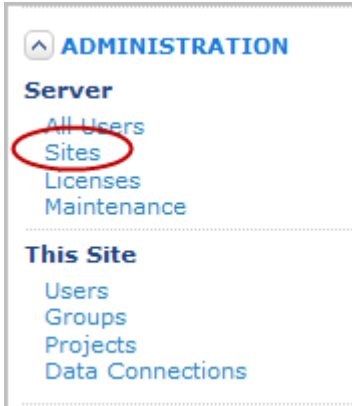
**Note:**

Content administrators can edit an existing user's account as long as the user is only a member of sites that the content administrator also controls. For example, if User Joe is a member of Site A and Site B and the content administrator is only an administrator of Site B, the content administrator can't edit Joe's Full Name or reset his password.

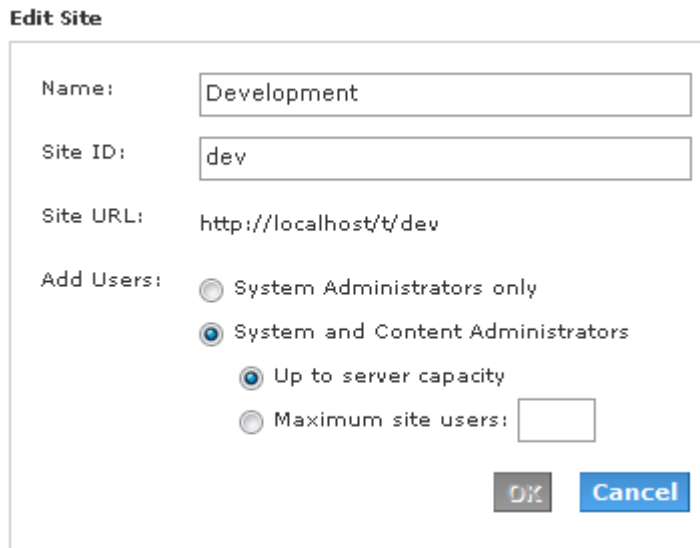
## Editing Sites

The Edit Site dialog allows system administrators to rename a site or change a content administrator's ability to add users to a site. To edit a site:

1. Open the Sites page under Server:



2. Select the site you want to change and click Edit.
3. You can edit the site's Name or the Site ID that's used for URLs. The URLs will point to the site's contents, such as views and workbooks:

A screenshot of the 'Edit Site' dialog box. It contains the following fields and options:

- Name: Development
- Site ID: dev
- Site URL: http://localhost/t/dev
- Add Users:
  - System Administrators only
  - System and Content Administrators
    - Up to server capacity
    - Maximum site users:

At the bottom right, there are 'OK' and 'Cancel' buttons.

4. If you are allowing content administrators to add users (the default), you can specify how many users they can add to the site by selecting one of the following:

- Up to server capacity: 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.
- Maximum site users <n>: Allows a content administrator to add users up to a limit you specify. The number you enter cannot be lower than your current number of licensed users.

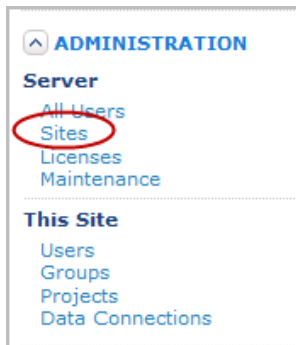
5. Click OK.

## Deleting Sites

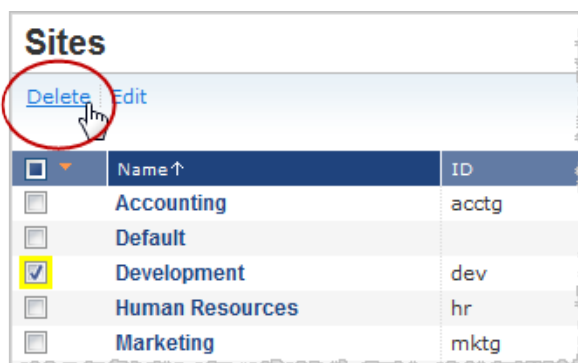
System administrators can delete sites that have been added to Tableau Server. Deleting a site also removes workbooks and data sources that were published to the site, as well as users. If a user belongs to additional sites, they will not be removed. To permanently remove a user, you need to use the All Users page. See [Deleting Users](#).

To delete a site:

1. Open the Sites page under Server:



2. Select the site you want to remove and click Delete:



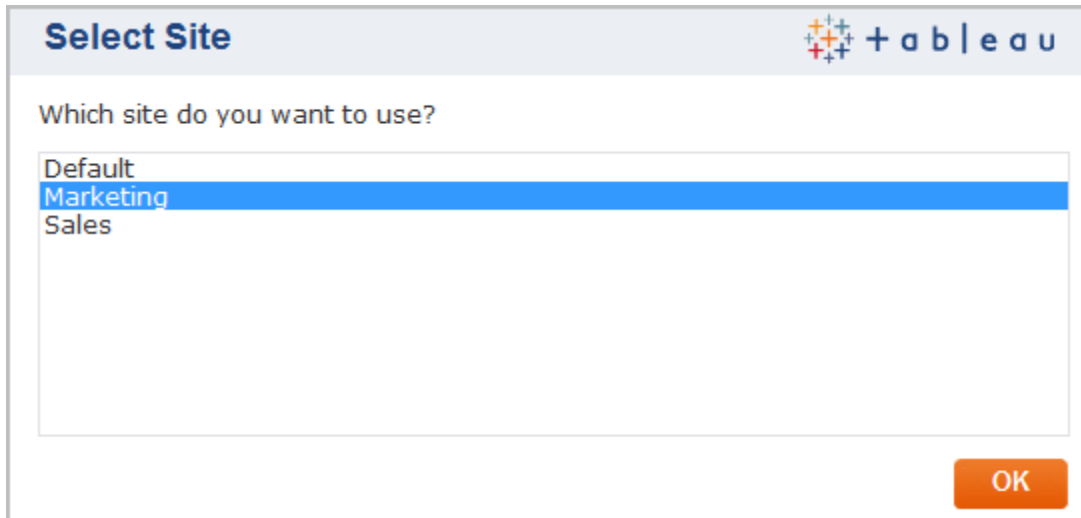
3. Click OK.


## Multi-Site Navigation

Here are some tips on how to identify which site you're logged into and how to navigate from site to site if you belong to multiple sites.

### Site Login

If you are a member of multiple sites, at server login you are prompted to choose a site:



**Select Site** 

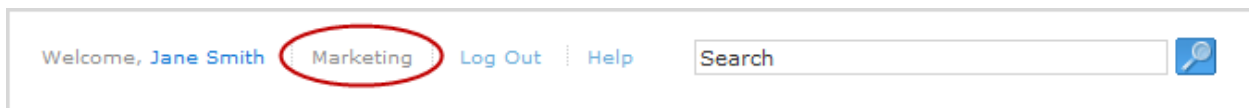
Which site do you want to use?


- Default
- Marketing**
- Sales

**OK**

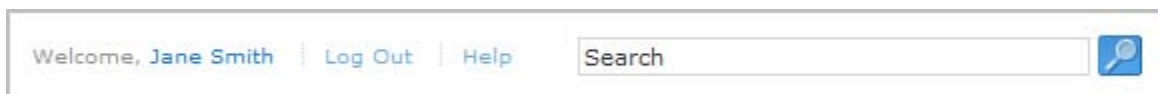
### Identifying Your Logged In Site


If you only belong to one site, you are not prompted to choose your site at server login. After you login, the name of your site is displayed at the top of the page:



Welcome, [Jane Smith](#) **Marketing** | [Log Out](#) | [Help](#)  

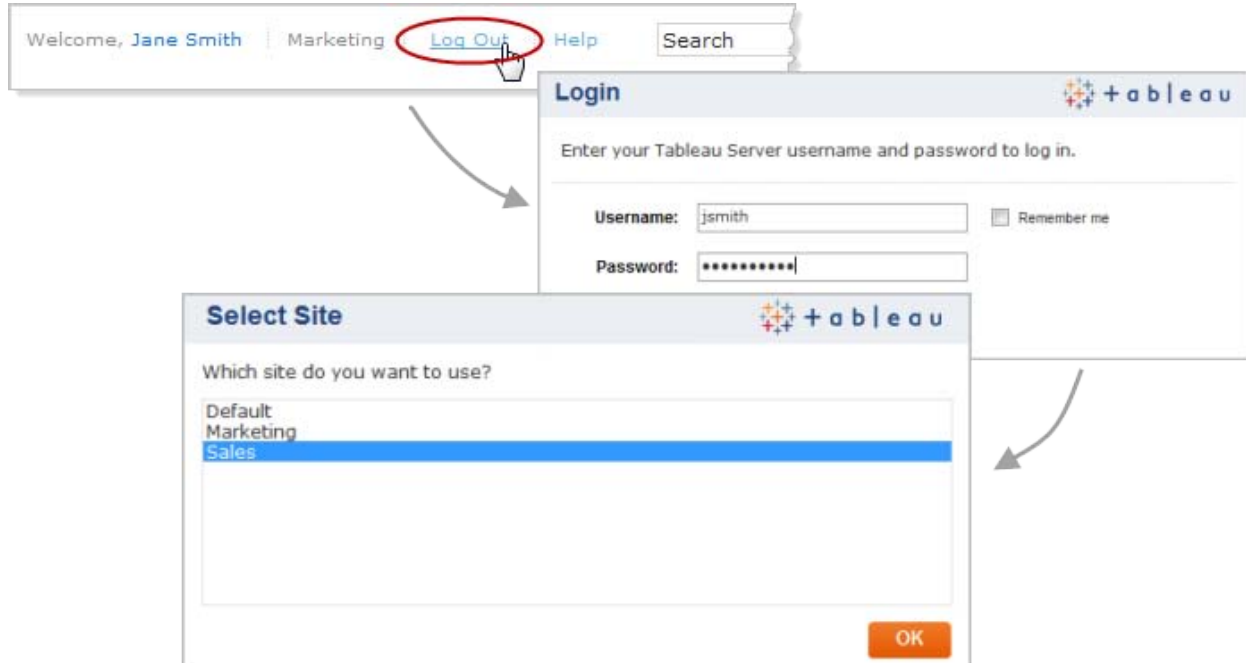
If Tableau Server isn't running multiple sites, no site name is displayed at the top of the page:



Welcome, [Jane Smith](#) | [Log Out](#) | [Help](#)  

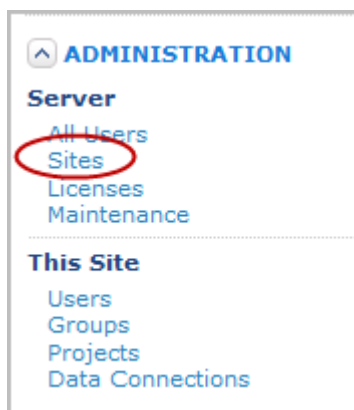
## Navigating to Other Sites

If you belong to multiple sites, you navigate from one site to another by clicking Log Out at the top of the page, logging back into Tableau Server, then choosing your site:

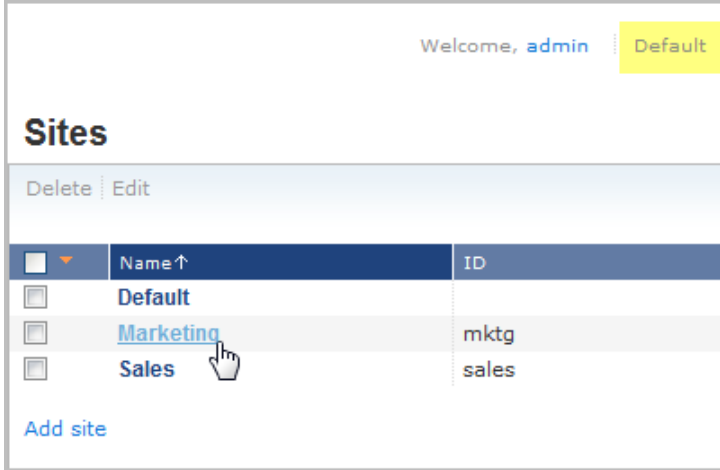


If you are a system administrator, you can also navigate to other sites without logging out of the server.

1. Click the **Sites** link under Administration:



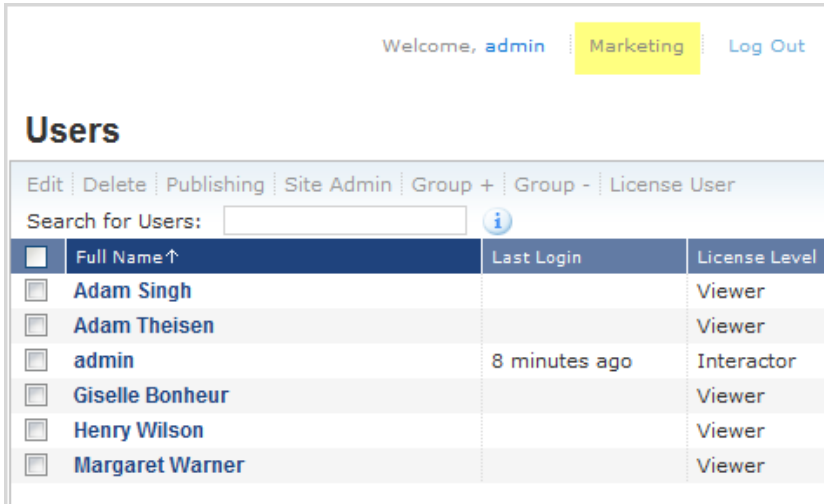
2. On the Sites page, click the name of the site you want to log into:



3. Next, click one of the items you see listed, such as Users:



You are now logged into the other site:





## Server Maintenance

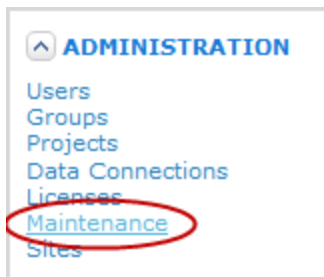
As a system administrator, you will want to check the status of the server, analyze and monitor the activity on the server, manage scheduled tasks, or perform certain maintenance activities such as rebuilding the search index. In addition, there are several settings that you may want to specify to customize the user experience for people using the server. You can do all of these tasks from the Maintenance page.

- [Accessing the Maintenance Tools](#)
- [Viewing Service Status](#)
- [Accessing Status Remotely](#)
- [Rebuilding the Search Index](#)
- [Clearing Saved Passwords](#)
- [Maintenance Settings](#)
- [Tableau Server Monitor](#)
- [Administrative Views](#)

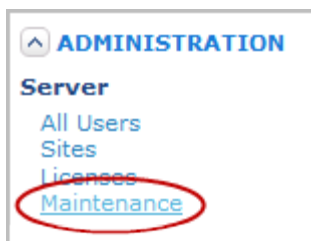
## Accessing the Maintenance Tools

In addition to Tableau Server Monitor, you also have access to maintenance tools on Tableau Server. There you can see detailed status for each service and process, monitor server and user activity, perform administrative tasks, and specify settings to customize the server. To access the maintenance tools:

1. Log into Tableau Server using your administrator username and password.
2. Click Maintenance in the Administration area on the left side of the page.



If you're running a multi-site server system, click Maintenance under the Server heading:



## Viewing Service Status

You can use the Status table on the Maintenance page to view the state of Tableau services on each Tableau server:

<span>✔ Service waiting for request</span> <span>✔ Service standing by</span> <span>✔ Service handling request</span> <span>🟡 Service unlicensed</span> <span>✘ Service down</span>							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
172.16.16.35	✔	✔	✔✔✔✔✔✔✔✔	✔✔✔✔✔✔✔✔✔✔	✔✔	✔	

### Note:

For information about unlicensed status for a VizQL Server process, see [Handling an Unlicensed VizQL Server Service](#).

To display a machine-readable version of the above information, from the Maintenance page, replace the word status in your URL with systeminfo (for example, <http://jsmith/admin/systeminfo>). A web page similar to the following appears:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <systeminfo>
- <machines>
- <machine name="100.0.0.1">
  <repository worker="100.0.0.1:80" status="OK" />
  <dataengine worker="100.0.0.1:80" status="OK" />
  <serverwebapplication worker="100.0.0.1:8000" status="OK" />
  <serverwebapplication worker="100.0.0.1:8001" status="OK" />
  <serverwebapplication worker="100.0.0.1:8002" status="OK" />
  <serverwebapplication worker="100.0.0.1:8003" status="OK" />
  <serverwebapplication worker="100.0.0.1:8004" status="OK" />
  <serverwebapplication worker="100.0.0.1:8005" status="OK" />
  <serverwebapplication worker="100.0.0.1:8006" status="OK" />
  <serverwebapplication worker="100.0.0.1:8007" status="OK" />
  <vizqlserver worker="100.0.0.1:9100" status="OK" />
  <vizqlserver worker="100.0.0.1:9101" status="OK" />
  <vizqlserver worker="100.0.0.1:9102" status="OK" />
  <vizqlserver worker="100.0.0.1:9103" status="OK" />
  <vizqlserver worker="100.0.0.1:9104" status="OK" />
  <vizqlserver worker="100.0.0.1:9105" status="OK" />
  <vizqlserver worker="100.0.0.1:9106" status="OK" />
  <vizqlserver worker="100.0.0.1:9107" status="OK" />
  <backgroundtasks worker="100.0.0.1:0" status="Busy" />
</machine>
</machines>
</systeminfo>
```

The three types of status for a Tableau service are OK, Busy, and Down.

## Accessing Status Remotely

As the Tableau administrator, only you can see the tools on the Maintenance page, including the Status table. You can, however, 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—for example, as part of a remote monitoring process. To grant remote access to Tableau Service status:

1. On the computer running the primary Tableau Server, open the Tableau Server config file:

```
ProgramData\Tableau\Tableau Server\config\tabsvc.yml
```

2. Add the line `wgserver.systeminfo.allow_referrer_ips: <IP address>` to `tabsvc.yml`, where `<IP address>` is the IP address of the computer you'd like to add. If you are granting service status access to multiple computers, use commas (no spaces) to separate each IP address. For example:

```
wgserver.systeminfo.allow_referrer_ips: 123.45.67.89,123.45.67.88
vizqlserver.extract.connection.class: dataengine
worker0.vizqlserver.procs: 4
service.runas.username: MYCO\jsmith
vizqlserver.extract.type: internal
config.version: 4
wgserver.authenticate: activedirectory
worker0.wgserver.procs: 4
wgserver.sspi.ntlm: true
service.init.state: start
```

3. Save and close `tabsvc.yml`.
4. Open a command prompt as an administrator and type:
  - o **32-bit:** `cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"`
  - o **64-bit:** `cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"`

5. Then use the following command to restart the Tableau Services:

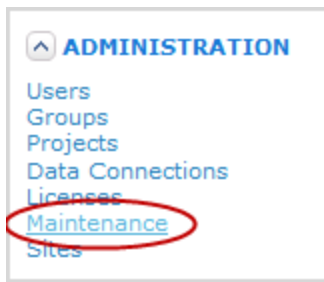
```
tabadmin restart
```

Now, users at computers whose IP addresses are added to `tabsvc.yml` can view Tableau service status by entering the URL `http://<server>/admin/systeminfo` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo`). This functionality can also be used as part of an automated remote monitoring process.

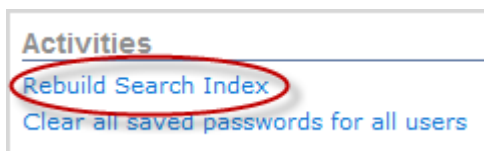
## Rebuilding the Search Index

If for any reason the search index stops returning the correct results or is missing results, you may need to rebuild the search index. Additionally, you should rebuild the search index if the indexer goes down for an extended period of time.

1. To rebuild the search index, click the Maintenance link in the Administration section on the left side of the page:



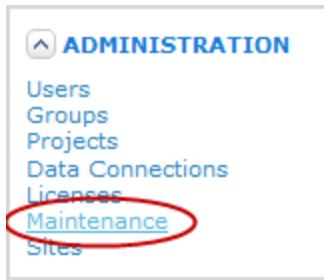
2. Click Rebuild Search Index to begin.



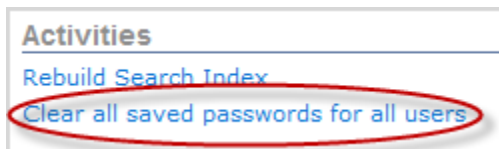
## Clearing Saved Passwords

If you enable the [Saved Passwords](#) setting, users can save data source passwords across multiple visits and browsers. As an administrator you can reset all of the passwords for all users, which forces users to log into the data sources the next time they visit a view that requires database authentication. To make this happen:

1. Click the Maintenance link in the Administration section on the left side of the page:



2. Under Activities, click Clear all saved passwords for all users.



## Maintenance Settings

The following settings are available in the Settings section of the Maintenance page on the server:

**Embedded Credentials** - 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. That way if you turn the setting back on, users don't have to re-embed the passwords.

**Scheduling** - Allows publishers to assign workbooks to schedules. This option is only available if Embedded Credentials is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.

**Public User List** - Allows web users to see a list of all users on the system. When this setting is enabled, a link to a list of all users is added to the left navigation bar. This is useful if your user list is not private and you want to let web users browse by user. When you browse by user, you can see all workbooks, tags, and comments associated with a selected user.

**Saved Passwords** - Allows users to save data source passwords across multiple visits and browsers. By default users can choose to "Remember my password until I log 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.

**Enable Guest** - Allows users to view and interact with embedded views without having to log into 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. Also, it cannot be used with Enable Automatic Login, an option you can select during Setup ([learn why](#)). If Enable Automatic Login is selected, Enable Guest is grayed out.

**Set default start page for all users** - Set the page all server users will be taken to after they log in to Tableau Server. Individual users can override this setting on their User Account page.

**Set default language and locale for all users** - Controls the language used for the server user interface and the locale used for views. Individual users can override this setting on their User Account page. Also, web browser settings are the first thing that's used to determine which language and locale are used. See [Language and Locale](#) for more information.

**Reset all settings to their default values** - Any server settings that have been changed since Setup are returned to their original state.



## 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 services, launch 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 aborted. 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, make 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 tools](#) Tableau Server.

### 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. This command does not stop Tableau Server. You can re-open the application by selecting All Programs > Tableau Server 7.0 > Tableau Server Monitor on the Windows Start menu.

## Administrative Views

Tableau Server comes with several views for administrators, to help monitor activity on Tableau Server. The views are located in the Analysis table on the server's Maintenance page:

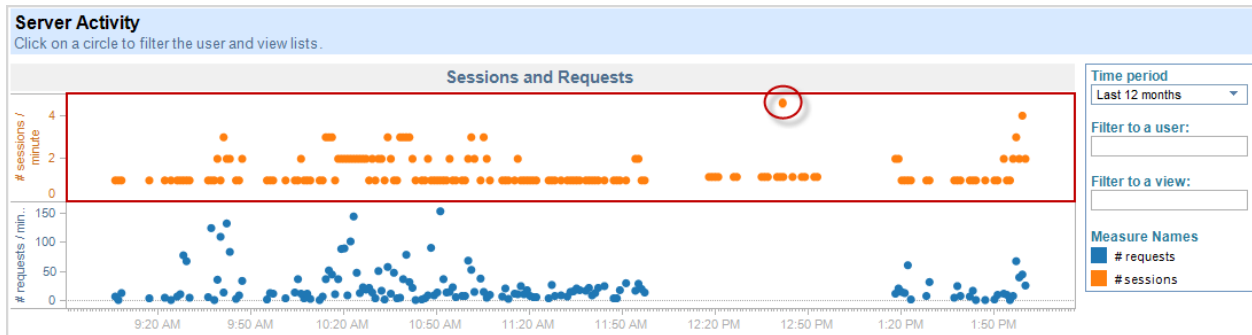
Analysis		Click on the views below to display the corresponding analysis.
View	Analysis	
<a href="#">Server Activity</a>	A dashboard view showing recent activity on the server	
<a href="#">User Activity</a>	A view describing user activity, including login time, hostname, and idle time	
<a href="#">View Performance History</a>	A view describing server activity broken down by view	
<a href="#">Background Tasks</a>	A view showing completed and pending task details	
<a href="#">Space Usage</a>	A dashboard view showing the space used by published workbooks and data sources	
<a href="#">Customized Views</a>	A dashboard view showing utilization of customized views	

See the following topics for more information:

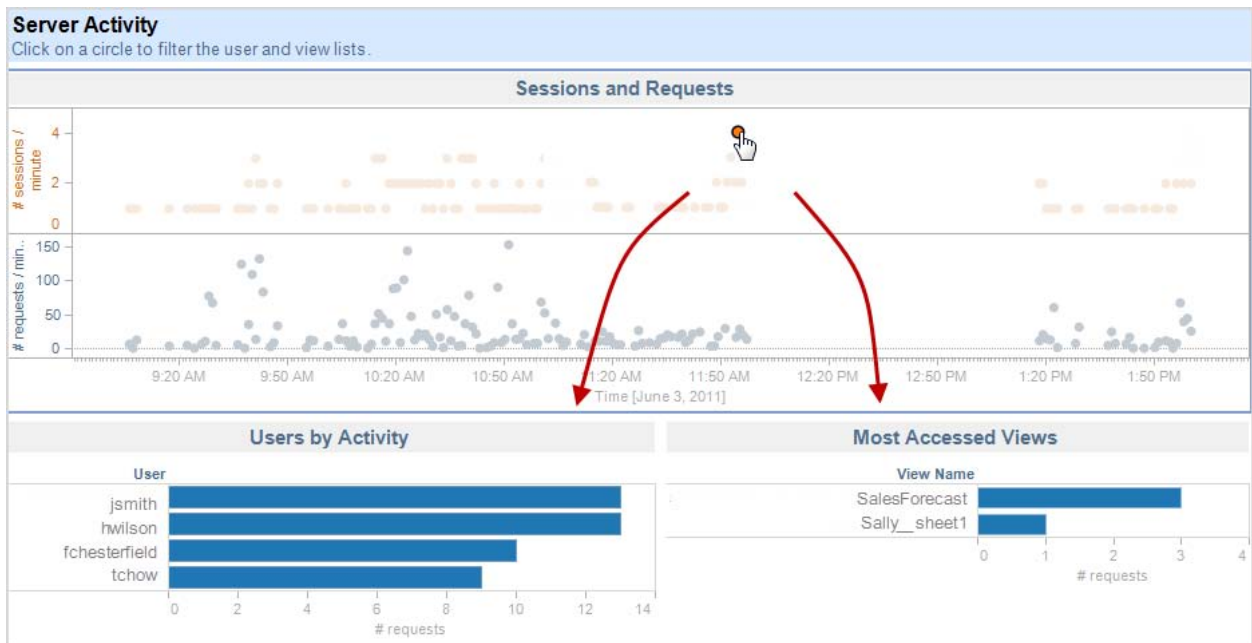
- [Server Activity](#)
- [User Activity](#)
- [Performance History](#)
- [Background Tasks](#)
- [Space Usage](#)
- [Customized Views](#)
- [Creating Your own Administrative Views](#)

## Server Activity

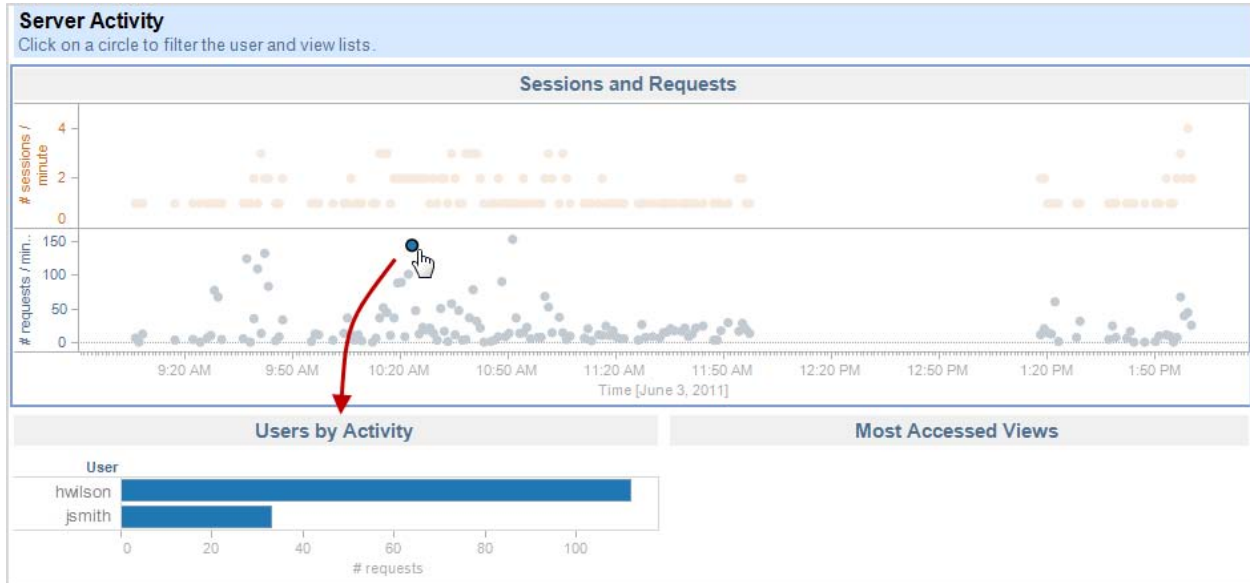
The Server Activity administrative view gives you a snapshot of Tableau Server activity for every minute within the selected time period. In Sessions and Requests, each orange circle represents a minute during which there was at least one server session (logged in user). A highly placed orange circle indicates a large number of Tableau Server users for that particular minute. The activity could be a variety of things, from browsing views to modifying settings on Tableau Server:



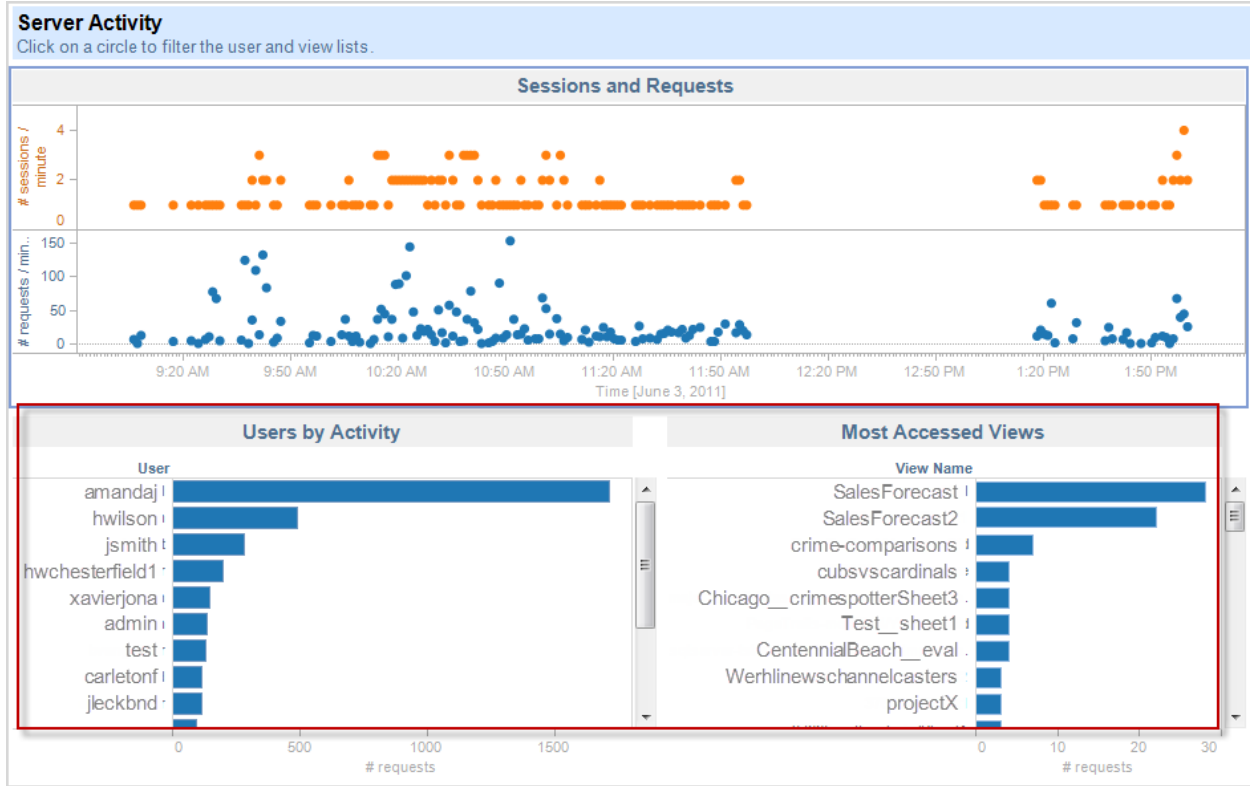
Click a circle to see who was active during that minute and which views they were accessing, if any:



Each blue circle in Sessions and Requests shows the number of server requests during any given minute. A highly placed circle indicates a high amount of activity. Just click a circle for details:



Without selecting an individual session or request circle you can see which server users have been most active during the selected time period, as well as which views are being accessed the most:



## User Activity

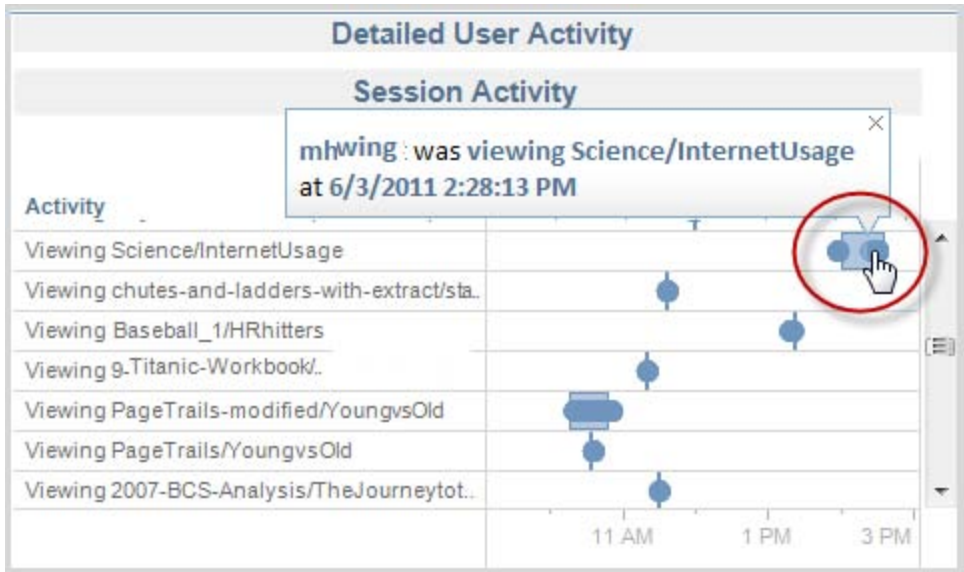
The User Activity view can help you gauge how heavily your Tableau Server installation is being used and whether you may need to buy additional licenses. Specifically, this view shows you who is logged into Tableau Server, from where, and when they last interacted with the server.

If a user is logged in from multiple browsers, that will be displayed as well. For example, if a user logs in once from Internet Explorer and once from Mozilla Firefox, their name appears twice. If a user logs in twice from Mozilla Firefox, their name appears once.

Currently Logged In Users				
User	IP Address	Last Activity Time		
elosin	123.45.66.201	06/03/2011	02:45:43 PM	Currently Active
jgartin	123.45.66.158	06/03/2011	02:40:19 PM	Recently Active
mgroupuser	16.1.109.136	06/03/2011	02:00:25 PM	Recently Active
avolosin	204.16.136.48	06/03/2011	02:30:39 PM	Recently Active
tmartin	204.16.104.204	06/03/2011	01:58:05 PM	Recently Active
tgroupuser	204.16.104.204	06/03/2011	02:25:52 PM	Recently Active
bvanga	172.16.104.229	06/03/2011	02:11:58 PM	Recently Active
ddun	123.45.66.243	06/03/2011	02:07:30 PM	Recently Active
msiu	16.1.109.53	06/03/2011	01:59:26 PM	Recently Active
mhiggins	16.1.89.103	06/03/2011	02:00:03 PM	Recently Active
tma	123.45.604.204	06/03/2011	01:53:45 PM	Recently Active
amend	172.16.136.41	06/03/2011	01:29:51 PM	Idle
bvanga	172.16.104.229	06/03/2011	01:19:12 PM	Idle
dgroupuser	16.1.109.16	06/03/2011	11:47:14 AM	Idle

Currently Active means that the user interacted with the server during the past five minutes. Recently Active indicates that the user was active between the last five to 15 minutes, and Idle means there's been no activity from the user for the last 15 minutes. By default, after four hours of inactivity, users are logged off of Tableau Server. You can change this setting by using the `tabadmin wgserver.session.idle_limit` option. See [Reconfiguring](#) for more information.

Circles indicate an action, such as logging into the server or filtering a view. Bars span the total time period over which there was activity. To learn more, just hover over an area and a tooltip appears:

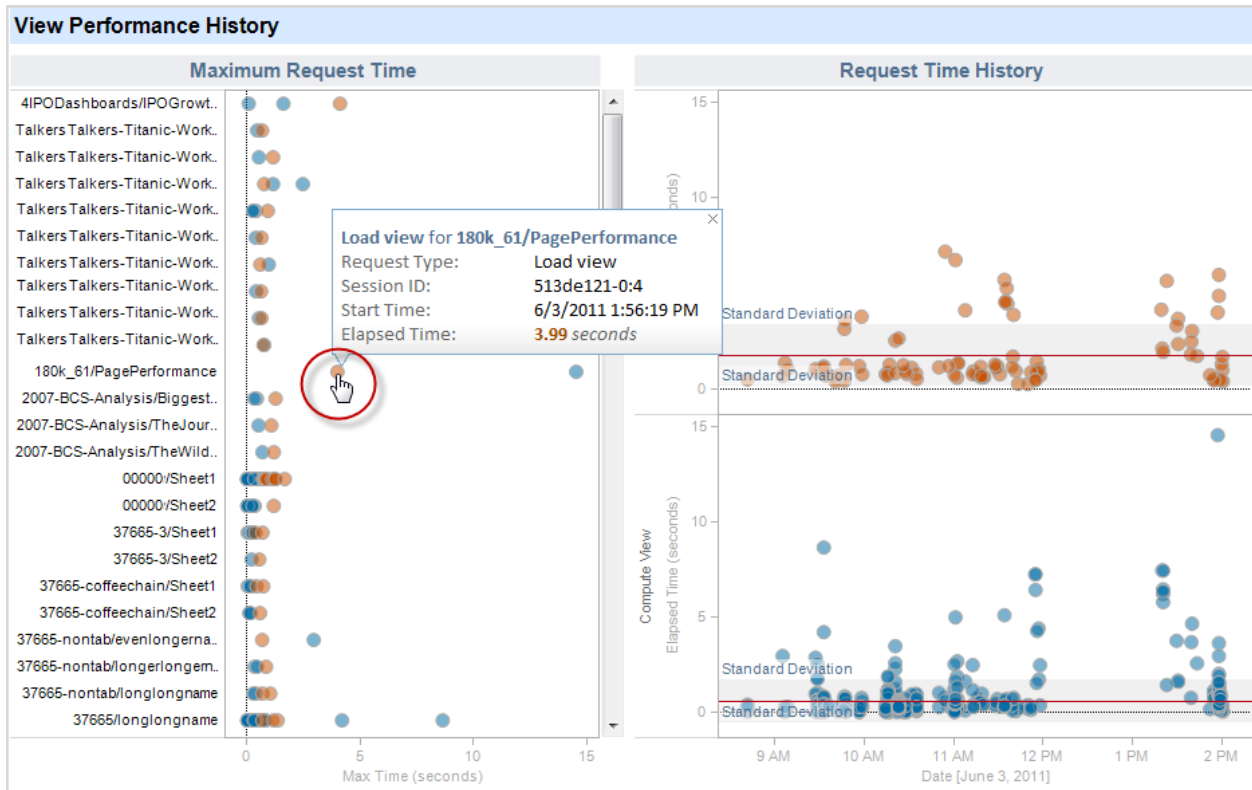




## Performance History

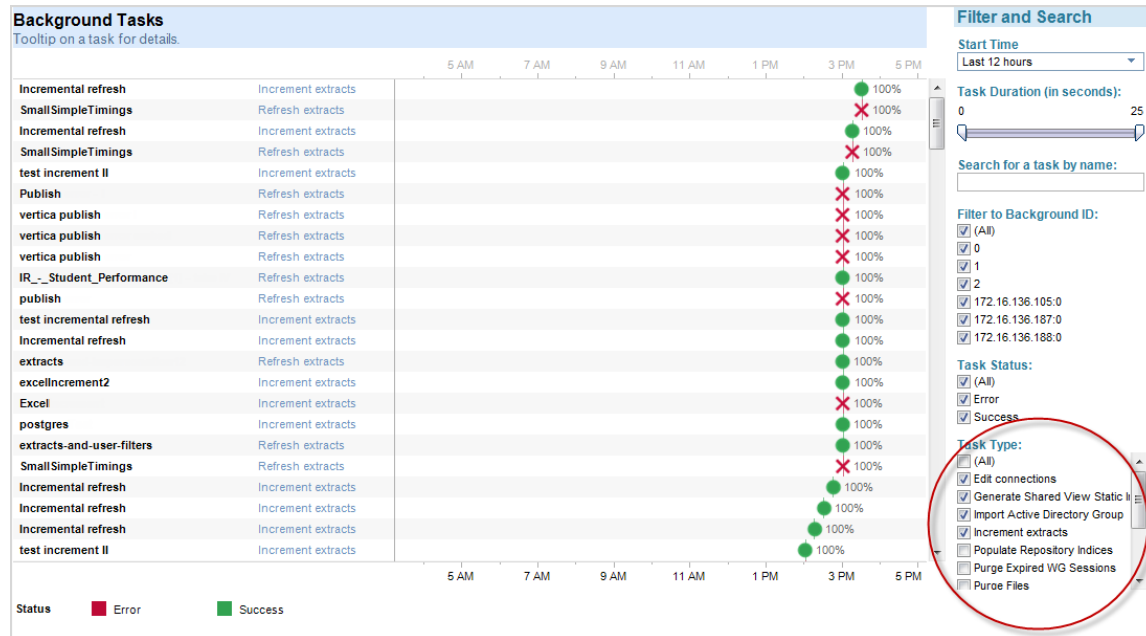
Use View Performance History to see which views are the most expensive in terms of server performance.

There are two different requests associated with views: Initial load requests, in orange, and compute requests, in blue. The latter are anything that causes Tableau Server to recompute what the user is seeing. This includes reload requests, and selecting and filtering items within a view. Outlier marks represent requests with the biggest impact on server performance:







## Background Tasks

The Background Tasks view displays tasks that the server runs. The most common tasks are those associated with user actions. These are selected by default under Task Type:



Tasks can have a status of successful completion, error, in process, or pending:

Icon	Description
	<b>Error</b> —Server was unable to complete the task.
	<b>Success</b> —Server completed the task.
	<b>In process</b> —Server is currently completing the task.
	<b>Pending</b> —A task that the server has not yet started.

For details on a task, hover over its icon:



Tableau Server can run multiple background processes in parallel. The IP addresses under Filter to Background ID in the Background Tasks view show you which machines are assigned to run background processes:

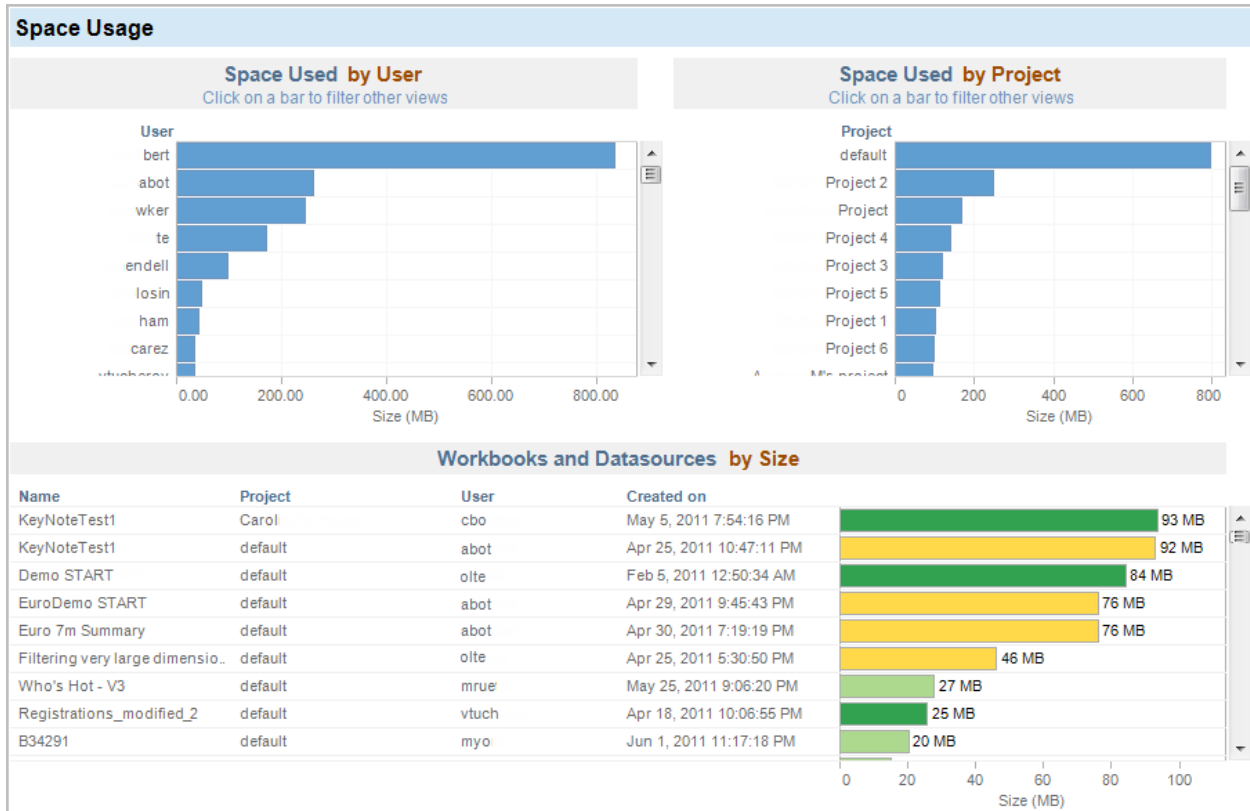
**Filter to Background ID:**

- (All)
- 116.16.136.187:0
- 116.16.136.187:1
- 116.16.136.188:0
- 136.16.136.105:0
- 136.16.136.187:0
- 136.16.136.188:0

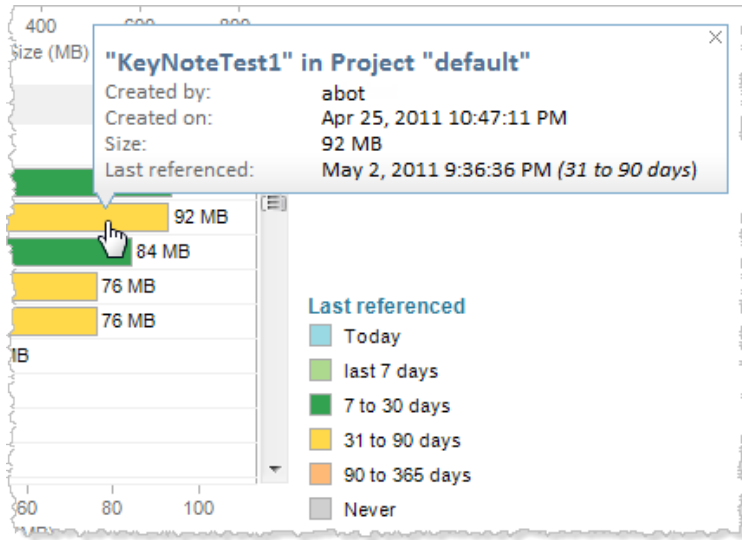
A multi-core machine running more than one background process will be listed with <IP address> : 0 for the first process, <IP address> : 1 for the second, and so on.

## Space Usage

The Space Usage view can help you identify which workbooks and data sources are taking up the most disk space on your 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:



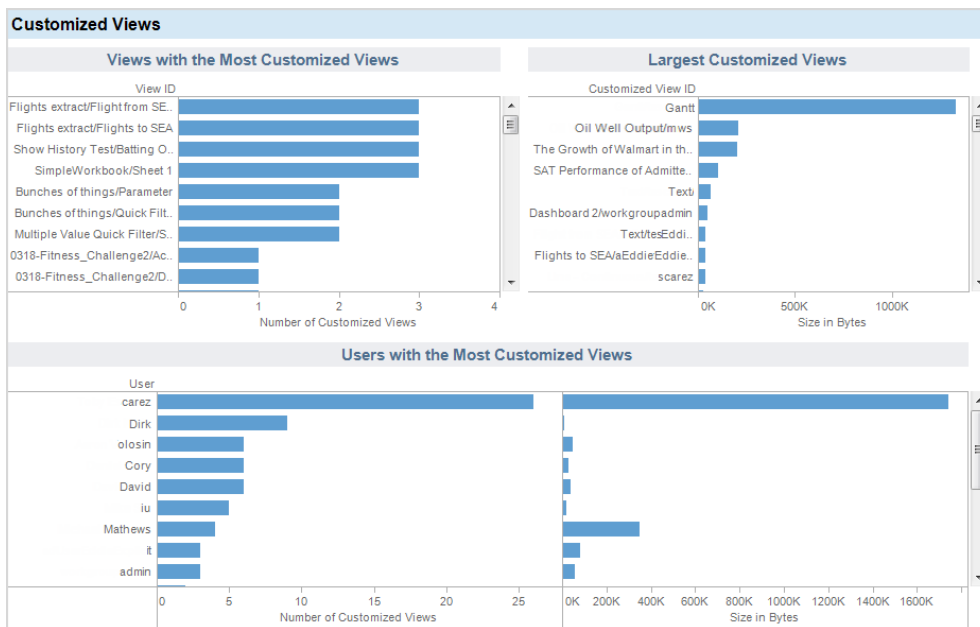
Move your cursor over any size bar to display usage details:



## Customized Views

People working with views can use the Remember my changes option to save their customized views and publishers can allow or prevent the [sharing of customized views](#).

The Customized Views administrative view lists all the views on the server that have been customized with Remember my changes. It can be used as one indicator of a view's popularity or importance:



## Creating Your own 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. The Tableau Server repository has several database views set up that you can connect to and query.

To access these views you must first use the command line tool to enable external access to the Tableau Server database. Next, if you are accessing the database from a computer that is not running Tableau Server, you will need to add your computer's IP address to Tableau Server's `pg_hba.conf` file. Finally, you need to connect to and query the Tableau Server database.

- [Enabling Access to the Tableau Server database](#)
- [Adding Your IP Address to pg\\_hba.conf](#)
- [Connecting to the Tableau Server Database](#)

## Enabling Access to the Tableau Server database

The Tableau Server repository has several database views set up that you can connect to and query as part of building your own analyses of Tableau Server activity. To access these views you must first use the `tabadmin` command line utility to enable external access to the database.

1. Open a command prompt as an administrator and type:

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau  
Server\7.0\bin"
```

2. Next, use the following command to enable external access to the database for the user "tableau" with a password that you specify.

```
tabadmin dbpass [password]
```

Substitute the `[password]` option with your own password.

3. Restart the server.

You can disable external access by running `tabadmin dbpass --disable` then restarting the server.

After you've enabled external access to the database you can connect to and query the database. Follow the steps in [Connecting to the Tableau Server Database](#) to connect.

## Adding Your IP Address to pg\_hba.conf.templ

If you are accessing the Tableau Server database from a computer that is not running Tableau Server, you will need to add your computer's IP address to the `pg_hba.conf.templ` file on the Tableau Server machine. The following procedure will allow you to make persisted configuration changes to Tableau Server.

1. On the machine running Tableau Server, navigate to the following location:

64-bit: `C:\Program Files (x86)\Tableau\Tableau Server\7.0\templates`

32-bit: `C:\Program Files\Tableau\Tableau Server\7.0\templates`

2. Open the file `pg_hba.conf.templ` in a text editor.
3. In `pg_hba.conf.templ`, add your computer's IP address to the list of local connections. For example:

#	TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
#	IPv4	local	connections:		
host	all	all	all	100.0.0.1/32	trust
host	all	all	all	100.16.16.47/32	trust
host	all	all	all	100.0.0.1/32	trust
host	all	all	all	123.4.5.6/32	trust

4. Save the file and restart Tableau Server for your change to take effect.



## 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.

1. 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.tableausoftware.com/drivers](http://www.tableausoftware.com/drivers).
2. In the PostgreSQL Connection dialog box, type the name or URL for Tableau Server. If you have a distributed server installation and a worker is hosting the repository, enter the name of the worker instead.

You should connect using the port you have set up for the pgsqll.port, which is 8060 by default. For more information about ports, see [TCP/IP Ports](#).

3. Type `workgroup` as the database to connect to.
4. Connect using the following username and password:
  - o Username: `tableau`
  - o Password: The password you specified when you enabled access to the Tableau Server database.
5. Select a table to connect to and click OK. The "tableau" user has access to all of the tables that start with an underscore. For example, you can connect to "`_background_tasks`" and "`_datasources`".

## Data Sources

A Tableau Server data source is a reusable connection to data. It can include a data extract or information for a pass-through connection to a live, relational database. It can also include a layer of customizations, such as calculations, groups, or sets. As an administrator, there are two main tasks you'll perform on the Data Sources page:

- **Edit and view data source permissions:** These include which users or groups can connect to data sources, modify them, and download them. See [Setting permissions for a data source](#) for more information.
- **Schedule data source extracts for refresh:** If a data source includes an extract, you can assign the extract to a refresh schedule. See [Scheduling Tasks](#) for more information.

Although both of the above tasks can be performed in Tableau Desktop by the person who published the data source, as an administrator, you can change the settings. You can also use the Data Sources page to remove a data source or add tags to it. See the topics below for more information.

- [Managing Data Sources](#)
- [About Tableau Data Server](#)
- [Troubleshooting Data Sources](#)

## Managing Data Sources

For users to work with Tableau Server data sources, they need to have the appropriate permissions for the data source. For data sources that are proxy connections, you should also be aware of how users will be authenticating to the database, and whether you have the appropriate drivers installed on Tableau Server. For more information, see the topics below.

- [Setting Permissions for a Data Source](#)
- [Database Drivers](#)
- [Data Security](#)

## About Tableau Data Server

Tableau's 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 (cubes are not supported). In the latter case, the information stored in the data source is for a pass-through connection. 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 extract can be used by many workbooks, 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.

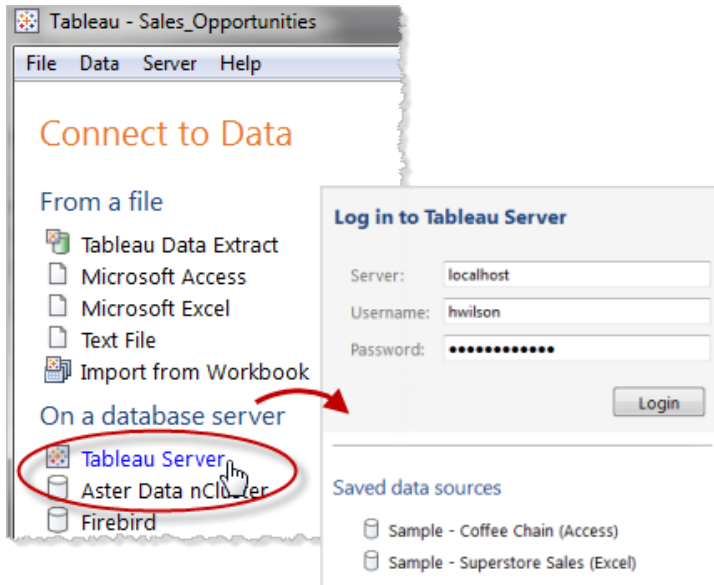
To use the data server, all authors have to do is connect to data in Tableau Desktop, either by creating an extract or a connection to a live relational database, and publish it to Tableau Server. Once published, these reusable data sources and the server contain everything workbook authors need to quickly connect to data and start authoring.

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](#) for more information.

- [Using Data Sources](#)

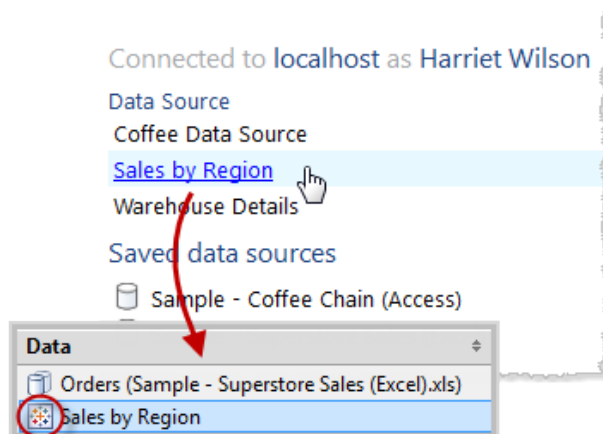
## Using Data Sources

If you're a workbook author, using a Tableau Server data source is simply a matter of connecting to it from Tableau Desktop. On the Connect to Data page in Tableau Desktop, click **Tableau Server**, then provide your server credentials:



After you log in to Tableau Server, data sources available to you are listed on the right. To see a data source, the person who published it had to set the Connect permission to **Allow** for you as a user. By default, all Tableau Server users have this permission.

Select a data source and it will load in the Data window in the workbook. Tableau Server data sources have a Tableau icon instead of a database icon:



For more information about creating and using data sources, see the Tableau Desktop online help.

## Troubleshooting Data Sources

For users to work with Tableau Server data sources, up to three things need to be in place:

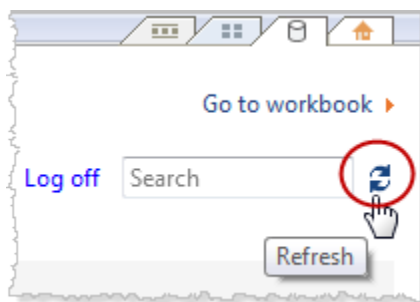
- **Permissions for the data source:** Anyone connecting to a data source must have the **Connect** and **View** permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be licensed to **Publish** and also have the **Write** and **Download File** permissions. See [Permissions](#) and [Setting Permissions for a Data Source](#) for more information.
- **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](#) 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](#) 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 [Permissions](#) and [Setting Permissions for a Data Source](#) 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:



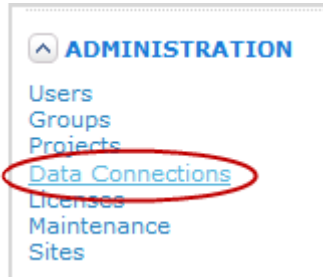
**Unable to connect to this Tableau Server data source:** This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct 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:** Connections to cube data sources (such as MSAS) are not supported. The data must be either an extract or a live connection to a relational database.

## Data Connections

Every workbook that is published to the server contains one or more connections. These connections are listed on the Data Connections page in the Administration area on the server.



Data connections are different from data sources in that each connection 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 connections listed on the connections page.

The Data Connections page allows you to manage the connection information for all of the workbooks published to the server. For example, you may have a large number of workbooks that connect to a database on a specific 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 data source using a specific user name and password. You can quickly update all of the workbooks to use a different set of credentials.

 A screenshot of a table displaying data connections. The table has columns for 'Name' and 'Connection Type'. The 'Modify' button in the 'Actions' column is circled in red.
 

Select: All Page None		Actions	
	Name		Connection Type
<input type="checkbox"/>	coffee		Microsoft SQL Serve
<input type="checkbox"/>	Cache [Staples (TESTV1.Staples)] Connection		InterSystems Cach
<input checked="" type="checkbox"/>	sql 2005 OnStar [ONSTAR] Connection	<a href="#">Modify</a>	Microsoft SQL Serve
<input checked="" type="checkbox"/>	14846		Microsoft SQL Serv
<input type="checkbox"/>	ETDERTID		Firebird

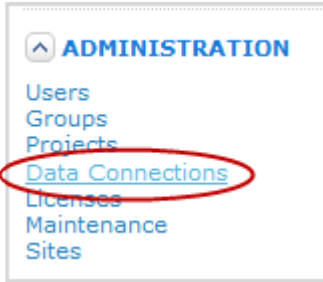
See the following topic for more information:

- [Modifying Data Connections](#)

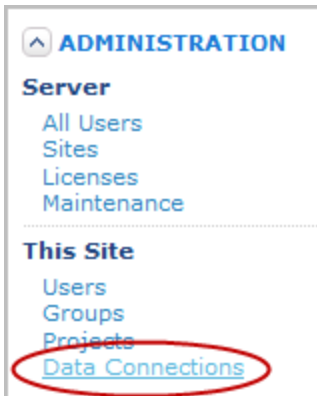
## Modifying Data Connections

Use the Data Connections page to manage the connection information for all of the workbooks published to the server or a site. To modify connection attributes:

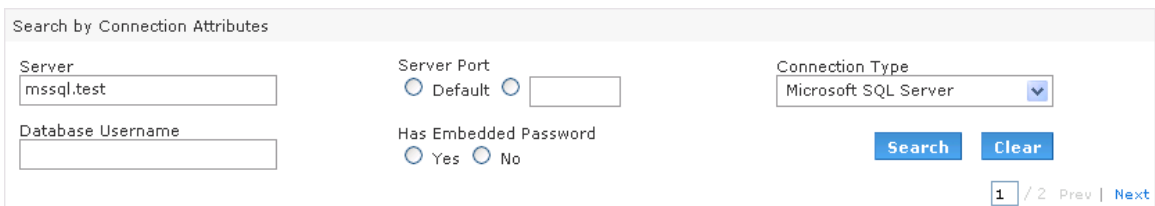
1. If you are running multiple sites on the server, log into the site that has the data connections you want to modify.
2. Navigate to the Data Connections page.



If you're running multiple sites, access the Data Connections page from under This Site:



3. Use the search box at the top of the list of connections to find the connections to modify. You can search by Server, Connection Type, the Server Port, Database User Name, and whether it Has Embedded Password.

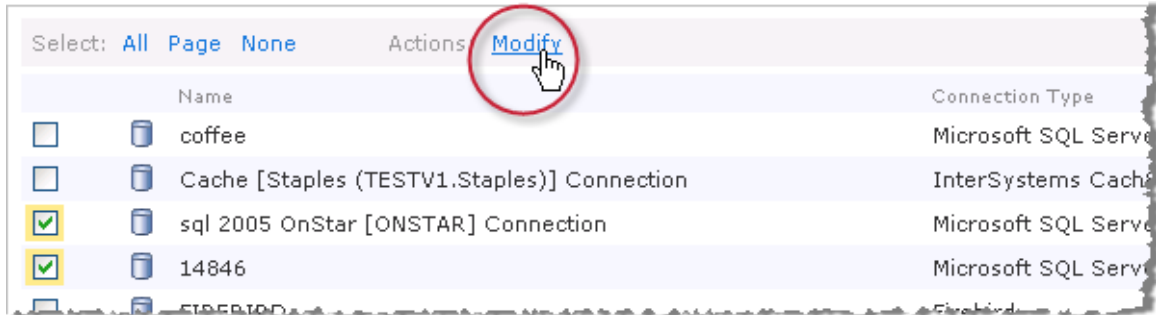
A screenshot of the search interface for Data Connections. The interface is titled 'Search by Connection Attributes' and contains several search criteria: 'Server' (text input with 'mssql.test'), 'Database Username' (text input), 'Server Port' (radio buttons for 'Default' and an empty input field), and 'Has Embedded Password' (radio buttons for 'Yes' and 'No'). There is also a 'Connection Type' dropdown menu set to 'Microsoft SQL Server'. At the bottom right, there are 'Search' and 'Clear' buttons, and a pagination indicator showing '1 / 2 Prev | Next'.



**Note:**

The values you type into the Server and Database User Name fields are treated as regular expressions.

4. Select the connections to modify in the list of search results and then select Modify on the Actions toolbar.



5. Type in a new value for one or more of the connection attributes. If a database or database driver doesn't support connecting via an IP address, the value entered for Server must be the database name. All attributes selected under the Change? column will be updated. If you select the Change? checkbox and leave the New Value field blank, the attribute will be set to blank as well.

Change?	Attribute	New Value
<input checked="" type="checkbox"/>	Server	<input type="text" value="mssql2005.test"/>
<input type="checkbox"/>	Server Port	<input type="text"/>
<input checked="" type="checkbox"/>	Database Username	<input type="text" value="myuser"/>
<input checked="" type="checkbox"/>	Password	<input type="password" value="....."/>
	Confirm Password	<input type="password" value="....."/>

6. Click Modify.
7. Refresh the server page (press F5) for your changes to take effect.

## Monitoring Progress

A monitor dialog opens automatically where you can watch the progress of the changes. If you close the monitoring dialog box, the modifications will run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped but will not impede other changes from being made. 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, but because you cannot add a password to a data source, the passwords for the data sources will not be changed.

There is an administrative view that allows you review details for completed and pending tasks. See [Background Tasks](#) to learn more.

## Customizing 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:

- [Look and Feel](#)
- [User Interaction](#)
- [Language and Locale](#)

## Look and Feel

You can customize the following aspects of Tableau Server's look and feel:

- Name: The name that appears in screen tips, warning messages, and error messages. The default name is Tableau Server.
- Logo: The image that appears on login page and in the left column of most pages. The logo that's shown when you open an individual view cannot be changed.
- [Changing the Name](#)
- [Changing the Logo](#)
- [Restoring the Default Name or Logo](#)

### Changing the Name

You can customize Tableau Server's look and feel by customizing the name that appears in screen tips and messages. To change the name:

1. Open a command prompt as an administrator and type the following:

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"
```

2. Change the name by typing the following:

```
tabadmin customize name "new_name"
```

In the above line, replace "new\_name" with the text that you want to appear as the name on the server. Example: `tabadmin customize name "Company Server"`

3. Restart the server for the change to take effect by typing:

```
tabadmin restart
```

## Changing the Logo

You can customize Tableau Server's look and feel by customizing the logo that appears on the Tableau Server login page and in the left column of most pages. To change the logo:

1. Open a command prompt as an administrator and type the following:

*32-bit:* `cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"`

*64-bit:* `cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"`

2. Change the logo by typing the following:

```
tabadmin customize logo "C:\My Pictures\logo.png"
```

In the above line, replace "C:\My Pictures\logo.png" with the path and file name of the image that you want to appear as the logo on the server. For best results, use an image that is 125 pixels x 35 pixels in size. The image can be a .png, jpg, or .gif file.

3. Restart the server for the change to take effect by typing:

```
tabadmin restart
```

## Restoring 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:

*32-bit:* `cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"`

*64-bit:* `cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"`

2. Change the logo by typing the following:

```
tabadmin customize <parameter> -d
```

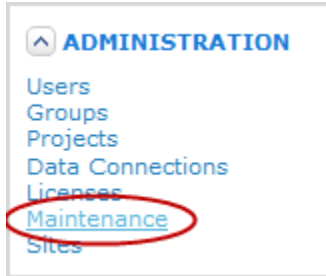
In the above line, replace <parameter> with what you want to restore, either name or logo.

3. Restart the server for the change to take effect by typing:

```
tabadmin restart
```

## User Interaction

In addition to customizing the look and feel of Tableau Server, you can customize specific ways that users can interact with the server. For example, you can allow publishers to attach database passwords to workbooks so that web users will automatically be logged in when viewing the workbook. You can customize the user interaction on the Maintenance page on the server.



See [Maintenance Settings](#) for more information.

## Language and Locale

Tableau Server is localized into several languages and has language and locale settings that you can configure on a per-user (set on the User Account page) and system-wide basis (set on the [Maintenance page](#)). 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.

### Default Settings

Tableau Server obtains its default language setting during Setup. If the host computer is set to 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 logs 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.

Here are the settings Tableau uses to determine language and locale, in the following order of precedence:

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





# Security

There are four main components to security in Tableau Server:

- [Authentication](#)
- [Authorization](#)
- [Data Security](#)
- [Network Security](#)

## Authentication

Authentication establishes a user's identity. This is done to prevent unauthorized access to Tableau Server and allow for a personalized user experience. Tableau Server supports three types of authentication:

- **Active Directory:** Authenticates Tableau Server users based on their Windows credentials.
- **Local Authentication:** Uses the internal authentication mechanism provided with Tableau Server.
- **Trusted Authentication:** Handles authentication through a trust relationship between Tableau Server and one or more web servers.

Whether to use Active Directory or Local Authentication is a choice you make during [Tableau Server Setup](#). After Setup, you can't switch between the two. To switch authentication types, uninstall Tableau Server (your data will be preserved) and rerun Setup.

### Active Directory

When Active Directory is used for user authentication, all usernames and passwords are managed by Active Directory. When a user enters their credentials at the Tableau Server login, Tableau passes them to the Active Directory server. It does not participate in the authentication process—although it does store usernames (but not passwords) in its repository.

With Active Directory user authentication, administrators can also automatically log in users based their current Windows credentials ([Enable Automatic Login](#)). This means that the user's credentials are being passed from their local computer, not from another system or portal that they may have logged in to.

For example, if a user logs into their local computer as 'MSmith' and then logs into a SharePoint portal as 'Mary', the credentials passed to Tableau Server will be for 'MSmith'. To use the credentials from the SharePoint site ('Mary') for automatic login, the SharePoint portal must use the [Tableau web part with trusted authentication](#).

Administrators can synchronize groups with Active Directory either manually or programmatically, using `tabcmd`. See [Synchronizing an Active Directory Group](#) and [syncgroup group-name](#) for more information.

### Local Authentication

When Local Authentication is used for user authentication, Tableau Server manages users, groups, passwords and the entire authentication process. User lists can easily be imported to the Tableau Server and most user management functions can be performed programmatically

through [tabcmd](#). Users can either manually login by entering their credentials when prompted or, when accessing content in a portal, via transparent trusted authentication.

## Trusted Authentication

Trusted authentication means that you have set up a trusted relationship between Tableau Server and one or more web servers. For example, you may have your corporate wiki use trusted authentication to show dashboards to employees who are already signed onto the wiki, without requiring another login.

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.

See [Trusted Authentication](#) for information on how to set up trusted authentication at your site.

## Authorization

Authorization is what a user can access and do once he or she is authenticated. In Tableau, authorization is handled by the following:

- **Roles and permissions:** Define specific capabilities that users can or cannot perform on certain objects in Tableau. A role is a set of permissions that administrators can use as-is or customize. See [Permissions](#) for details.
- **Licensing and user rights:** Control the maximum set of permissions that a user can have. See [Licenses](#) and [Allowing or Denying User Rights](#).

While the above items control which actions a user can perform and on what, they do not control which data will appear inside a view. The data a user sees is controlled by your [data security choices](#).

### Initial Permissions

The initial permissions for a project are copied from the Default project. The initial permissions for a workbook are copied from the permissions for its project. The initial permissions for a view are copied from the permissions of its workbook. This is a one-time copy of the parent's permissions. Changes to the parent's permissions are not automatically applied to the children unless the new permissions are actively assigned to the contents.

Any item can have permissions that differ from the parent. For example, a group might not have permission to see Project X, but it may have permission to see a view that's published to Project X. Tableau Server does not support hierarchical object permissions; however, it does provide an inheritance model for users and groups. If a user does not have a permission explicitly set to Allow or Deny, the setting will be inherited from the groups the user belongs to.

### Permissions and the Default Project

If Tableau Server is deployed in an open environment where knowledge and information sharing is key, then you should consider setting the permissions for the Default project to include the All Users group, with its role set to Interactor. Users will be able to automatically publish to and consume content from new projects.

If Tableau Server is deployed in a restrictive environment where data security and access control is key, then consider emptying the permissions for the Default project: Delete the permissions for all users and groups. Users and groups will need to be explicitly granted permission to publish and consume content in new projects.

## 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:** Workbook authors can create filters that control which data a person sees in a published view, based on their Tableau Server login account. User filters can also be included in a Tableau Server data source (authors still need to apply them to the view).

The table below outlines some dependencies with the above options:

<i>Database Connection Options</i>		<i>Data Security Questions</i>		
<b>Database login account uses...</b>	<b>Authentication mode</b>	<b>Is database security possible per Tableau Server user?</b>	<b>Are user filters the only way to restrict which data each user sees?</b>	<b>Are web caches shared among users?</b>
<i>Window NT Integrated Security (Windows Authentication)</i>	<i>Server Run As account</i>	No	Yes	Yes
	<i>Impersonate via server Run As account</i>	Yes	No*	No
<i>Username and Password</i>	<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 password:</i> The workbook or data	No	Yes	Yes

<i>Database Connection Options</i>		<i>Data Security Questions</i>		
<b>Database login account uses...</b>	<b>Authentication mode</b>	<b>Is database security possible per Tableau Server user?</b>	<b>Are user filters the only way to restrict which data each user sees?</b>	<b>Are web caches shared among users?</b>
	source publisher can embed their database credentials.			
	<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 password 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:

- [Run As User](#)
- [SQL Server Impersonation](#)
- [Embedded credentials](#)
- [Saved passwords](#)

## 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` 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`. Communications between Tableau Server and its clients use standard HTTP requests and responses. Tableau Server can also be configured for HTTPS (see [Configuring SSL](#)). When Tableau Server is configured for SSL, all content and communications between clients are encrypted and use the HTTPS protocol.

Passwords are communicated from browsers and `tabcmd` to Tableau Server using 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.

### 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.

### Server Component Communication

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. The primary server is the only machine in the cluster that accepts requests from 3rd parties (clients), all other machines in the cluster 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. With the exception of license validation and accessing the repository, transmission of all internal communication is performed via HTTP.

When passwords are transmitted within the cluster, a key is used to encrypt the passwords transmitted between Tableau Server components (for example, between the application server and the VizQL server processes). Each encrypted transmission uses a key one time before it is discarded.



# Embedding Views

You can embed views from Tableau Server into webpages, blogs, wikis, web applications, and intranet portals. The embedded views blend seamlessly into your webpages and are interactive. The views update as the underlying data changes or the workbooks are updated on the server. Embedded views follow the same licensing and permission restrictions used on the server. Generally, people loading a webpage with an embedded view must also have an account on Tableau Server. If you have a core-based license you can alternatively select [Enable Guest](#), which allows users to load the view without logging in. There are three ways you can embed views:

- Use the Share embed code as-is: The Share link in the upper left corner of each view provides automatically-generated embed code. All you have to do is copy the code and paste it into your webpage.
- Write your own embed code: You can enhance the default embed code Tableau provides or you can [build your own code](#). Either way you can add 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](#).

See the following topics for details:

- [Writing Embed Code](#)
- [List of Embed Parameters](#)
- [Using the Tableau JavaScript API](#)
- [Examples](#)

## 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. Just use the Share embed code 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:** As with earlier versions of Tableau, you can embed a view using an `iframe` or `Image` tag, where the source is the raw URL for the view. You may want to do this if you can't use JavaScript at your web site. There may also be situations where all you can specify is an URL—such as if you're embedding a view using [SharePoint's Page Viewer Web Part](#).

Regardless of the approach you take, you must define a width and height if you are embedding a view.

### Tableau JavaScript

Here's an example of the embed code you get by default when you click Share:

```
<script type="text/javascript"
src="http://myserver/javascripts/api/viz_v1.js"></script>
<div class="tableauPlaceholder" style="width:800; height:600;">
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="site_root" value="/t/Sales" />
  <param name="name"
value="MyCoSales/SalesScoreCard/jsmith@myco.com/EastCoastSales" />
  <param name="tabs" value="yes" />
  <param name="toolbar" value="yes" /></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. For examples, see the [List of Embed Parameters](#) and the "Script Tag Examples" in the [Examples](#) section.

## View URL as the Source

Here's an example of embedding the same view using an IFrame, where the source is the URL for the view:

```
<iframe  
src="http://myserver/t/Sales/MyCoSales/SalesScoreCard?:embed=yes&:tabs=yes&:t  
oolbar=yes" width="800" height="600"></iframe>
```

You must specify the `embed` URL parameter and 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. For examples, see the [List of Embed Parameters](#) and the "Iframe Tag Examples" in the [Examples](#) 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.	<code>&lt;param name="customViews" value="no" /&gt;</code> <a href="http://tabserver/views/Date-Time/DateCalcs?embed=yes&amp;customViews=no">http://tabserver/views/Date-Time/DateCalcs?embed=yes&amp;customViews=no</a>
-	:embed	yes	Required for URL parameter. Hides the top navigation area, making the view blend into your web page better.	<a href="http://tabserver/views/Date-Time/DateCalcs?embed=yes">http://tabserver/views/Date-Time/DateCalcs?embed=yes</a>
filter	-	string	Customizes what is displayed when the view opens.	<code>&lt;param name="filter" value="Team=Blue" /&gt;</code>
	:format	pdf ; png	Displays a view as a PDF or .png file.	<a href="http://tabserver/views/Sales/Q2?format=pdf">http://tabserver/views/Sales/Q2?format=pdf</a>
host_url	-	string	The server name as it appears in the URL.	<code>&lt;param name="host_url" value="http://myserver/" /&gt;</code>
linktarget	:linktarget	string	The target window name for external hyperlinks.	<code>&lt;param name="linktarget" value="_blank" /&gt;</code> <a href="http://tabserver/views/Date-Time/DateCalcs?embed=yes&amp;linktarget=_blank">http://tabserver/views/Date-Time/DateCalcs?embed=yes&amp;linktarget=_blank</a>
load-order	-	number	When multiple views are embedded, the default load order	<code>&lt;param name="load-order" value="2" /&gt;</code>

Object Parameter	URL Parameter	Values	Description	Examples
			is the order in which the views are listed. Use this setting to override that order. Negative numbers are allowed.	
name	-	string	Required for object parameter. Workbook and sheet name and optionally, a custom view (username@domain/[custom view name]).	<pre>&lt;param name="name" value="MyCoSales/Sales" /&gt;</pre> <pre>&lt;param name="name" value="MyCoSales/Sales/jsmith@myco.com/EastCoastSales" /&gt;</pre>
path	-	string	For trusted authentication only, cannot be used with the "ticket" parameter. Overrides value of the "name" parameter and is used as the URL. See the <a href="#">Trusted Authentication examples</a> .	<pre>&lt;param name="path" value="trusted/123456789/views/workbookQ4/SalesQ4" /&gt;</pre> <pre>http://tableauserver/trusted/123456789/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes</pre>
-	:refresh	yes; no	Rerenders the page.	<pre>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:refresh=yes</pre>
-	:revert	all; filters; sorts; axes; ;	Returns the item to its original state.	<pre>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:revert=all</pre>

Object Parameter	URL Parameter	Values	Description	Examples
		shelves		
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 <a href="#">Trusted Authentication examples</a> .	<pre>&lt;param name="site_root" value="/t/Sales"/&gt;</pre> <pre>&lt;param name="site_root" value=""/&gt;</pre>
tabs	:tabs	yes; no	Displays or hides tabs.	<pre>&lt;param name="tabs" value="yes"/&gt;</pre>
ticket	-	number	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 <a href="#">Trusted Authentication examples</a> .	<pre>&lt;param name="ticket" value="123456789"/&gt;</pre> <pre>http://tableauserver/trusted/123456789/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes</pre>
toolbar	:toolbar	yes; no	When yes the toolbar is included with the embedded view. The toolbar is shown by default if this parameter is not set.	<pre>&lt;param name="toolbar" value="yes"/&gt;</pre> <pre>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:toolbar=yes</pre>

## Using the Tableau JavaScript API

If you're a web application developer, you can control the Tableau JavaScript library using your own external web logic. For example, you may have dynamic, server-side logic outside of Tableau that determines the identity of the currently logged in user. You can use that logic to control the Tableau JavaScript and, in particular, the filter parameter—so that the current user only sees data that pertains to him or her. Here's an example to get you started:

```
<html>
<head></head>
<body>
<script type="text/javascript"
src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="site_root" value="/t/Site" />
  <param name="name"
value="Workbook/ViewName/<username>@domain/<custom_view_name>" />
  <param name="tabs" value="yes" />
  <param name="toolbar" value="yes" />
</object>
<form>
</form>
<script type="text/javascript">
  function useVizAPI() {
    var viz = window.tableau.vizs[0];
    viz.hide();
    viz.show();
    //viz.refresh();
    //viz.revert();
    //viz.filter({Product: ['Mint', 'Green Tea']});
  }
  if (document.addEventListener) {
    window.addEventListener("load", useVizAPI, false);
  } else if (document.attachEvent) {
    window.attachEvent("onload", useVizAPI);
  }
</script>
</body>
</html>
```

### Note:

The above functions are available at or after the browser's 'onload' event. Also, the argument to the filter function must be a JavaScript hash where the keys are dimension names and the value is either a string or an array of strings. The example above uses an array. Here's an example that uses a single string value: `viz.filter({Product: ['Mint']})`

## Example Code Location

Web application example code for SharePoint, PHP, Ruby, and Java applications is installed with Tableau Server and is located in:

- *32-bit*: C:\Program Files\Tableau\Tableau Server\7.0\extras\embedding
- *64-bit*: C:\Program Files (x86)\Tableau\Tableau Server\7.0\extras\embedding

## Examples

Here are some examples of ways you can customize or work with your embed code:

- [Example: Adding Filters](#)
- [Example: Filtering on Multiple Fields](#)
- [Example: Filtering Dates and Times](#)
- [Example: Filtering Measures](#)
- [Example: Controlling the Load Order of Multiple Views](#)
- [Example: Embedding Views into SharePoint \(Microsoft SSPI\)](#)
- [Example: Embedding Views into Wikis](#)
- [Example: Embedding Images](#)
- [Example: Embedding Views into SharePoint \(Trusted Authentication\)](#)



## Example: Adding 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

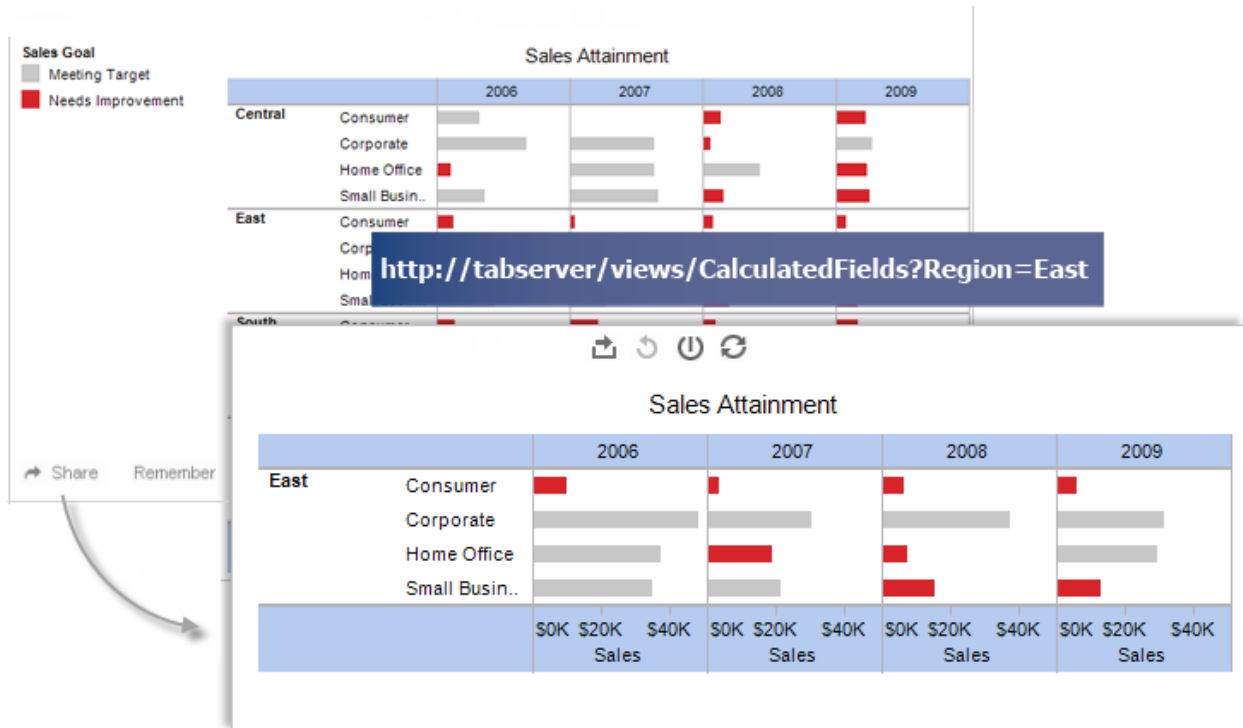
```
<script type="text/javascript"
src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="Sales/Sales-Performance" />
  <param name="filter" value="Region=East" />
</object>
```

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="http://myserver/views/CalculatedFields?:embed=yes&Region=East"width="800
" height="600"></iframe>
<iframe src="http://myserver/views/Sales/Sales-
Performance?:embed=yes&Region=East,West" width="900px"
height="700px"></iframe>
```



## Example: Filtering on Multiple Fields

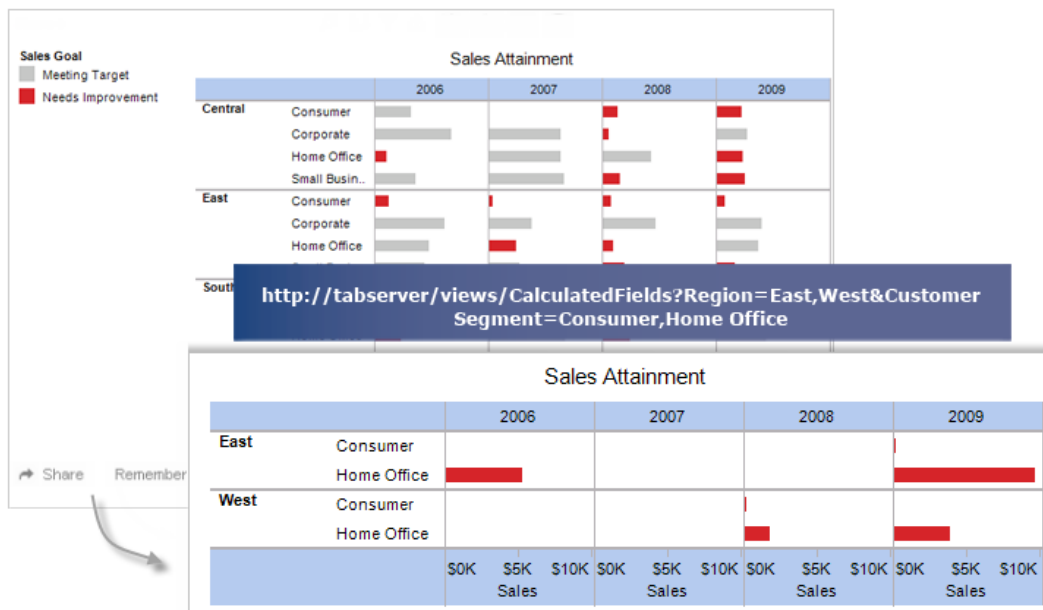
You can pass filters on as many fields as you want, including fields that are not in the original view.

### Script Tag Example

```
<script type="text/javascript"
src="http://myserver/javascrpts/api/viz_v1.js">
</script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="Sales/Sales-Performance" />
  <param name="filter" value="Region=East,West&Customer
Segment=Consumer,HomeOffice" />
</object>
```

### Iframe Tag Example

```
<iframe
src="http://myserver/views/CalculatedFields?:embed=yes&Region=East,West&Custo
mer Segment=Consumer,Home Office" width="800" height="600"></iframe>
```



### Note:

If a filter value contains a comma, replace the comma with %5c%2c. This is the URL encoding sequence for /, (forward slash, comma). The forward slash is needed to escape the comma.

## Example: Filtering 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="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <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/SalesPerformance?:embed=yes&Date=2008-12-01%2022:18:00" width="800" height="600"></iframe>
```

To filter multiple dates, separate each date with a comma.

## Example: Filtering 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="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="Sales/Sales-Performance" />
  <param name="filter" value="Profit=100, 200" />
</object>
```

### Iframe Tag Example

```
<iframe src="http://myserver/Sales/Sales-
Performance?:embed=yes&Profit=100,200" width="800" height="600"></iframe>
```

## Example: Controlling 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

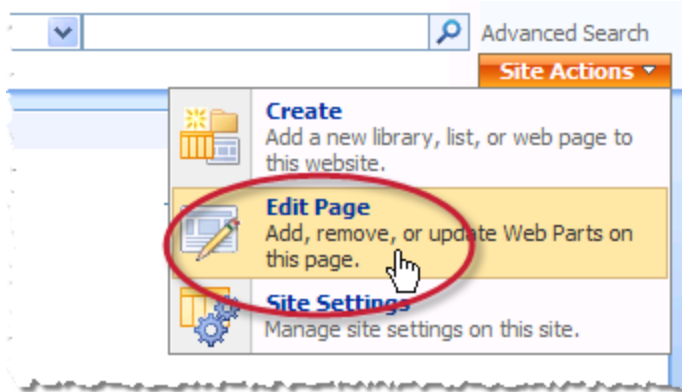
```
<script type="text/javascript"
src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="600" height="400" style="display:none;">
  <param name="name" value="MyCoSales/TopPerformers" />
  <param name="tabs" value="yes" />
  <param name="toolbar" value="yes" />
  <param name="filter" value="Salesperson=Top 5" />
  <param name="load-order" value="0" />
</object>

<script type="text/javascript"
src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="600" height="400" style="display:none;">
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="tabs" value="yes" />
  <param name="toolbar" value="yes" />
  <param name="load-order" value="-1" />
</object>
```

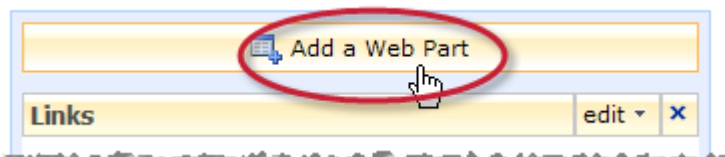
## Example: Embedding Views into SharePoint (Microsoft SSPI)

If both Tableau Server and SharePoint are using Microsoft SSPI, you can embed views using the Page Viewer Web Part. Follow the steps below to embed a view into a SharePoint page.

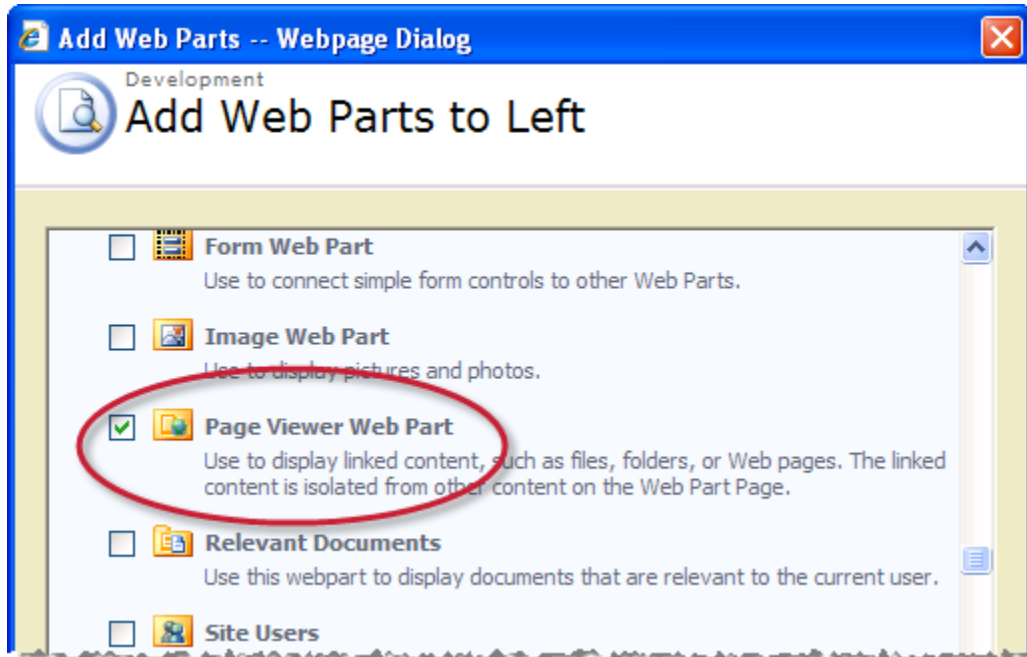
1. Navigate to the SharePoint page that you want to embed a view into.
2. On the Site Actions menu in the upper right corner of the page select Edit Page.



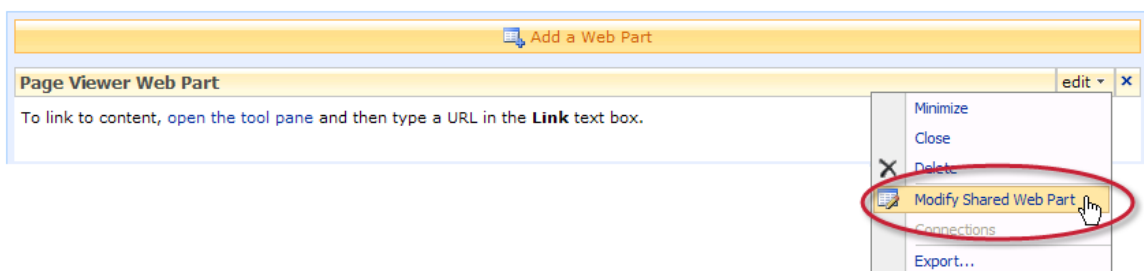
3. Click the Add a Web Part button in the section of the page where you want to embed the view.



4. On the page that opens, select the Page Viewer Web Part located in the Miscellaneous section and click Add.



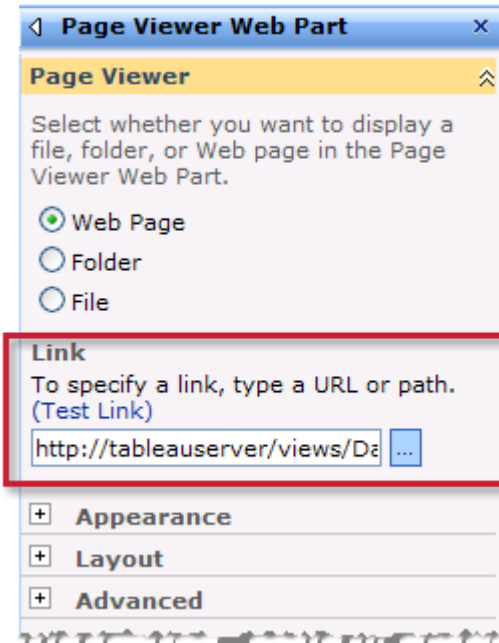
5. Back on the SharePoint page select Modify Shared Web Part on the Edit menu for the new web part.





6. On the right side of the page, you can specify the attributes of the Page View Web Part. Type the URL for the view you want to embed. Use the format specified in [Embedding Views](#). For example you may type:

`http://tableauserver/views/Date-Time/DateCalcs?:embed=yes&:toolbar=no`



7. Then in the Appearance section you can specify a Title of the web part, the Height, and Width. In general you should specify a fixed height (e.g., 700 Pixels) and adjust the width to fit to the zone.

The screenshot shows a configuration panel titled "Appearance" with a collapse icon. It contains three sections: "Title" with a text input field containing "My Embedded View"; "Height" with a question "Should the Web Part have a fixed height?", a radio button for "Yes" (selected) with a text input field containing "700" and a dropdown menu set to "Pixels", and a radio button for "No. Adjust height to fit zone."; and "Width" with a question "Should the Web Part have a fixed width?", a radio button for "Yes" (unselected) with an empty text input field and a dropdown menu set to "Pixels", and a radio button for "No. Adjust width to fit zone." (selected).

8. 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.

## Example: Embedding 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.

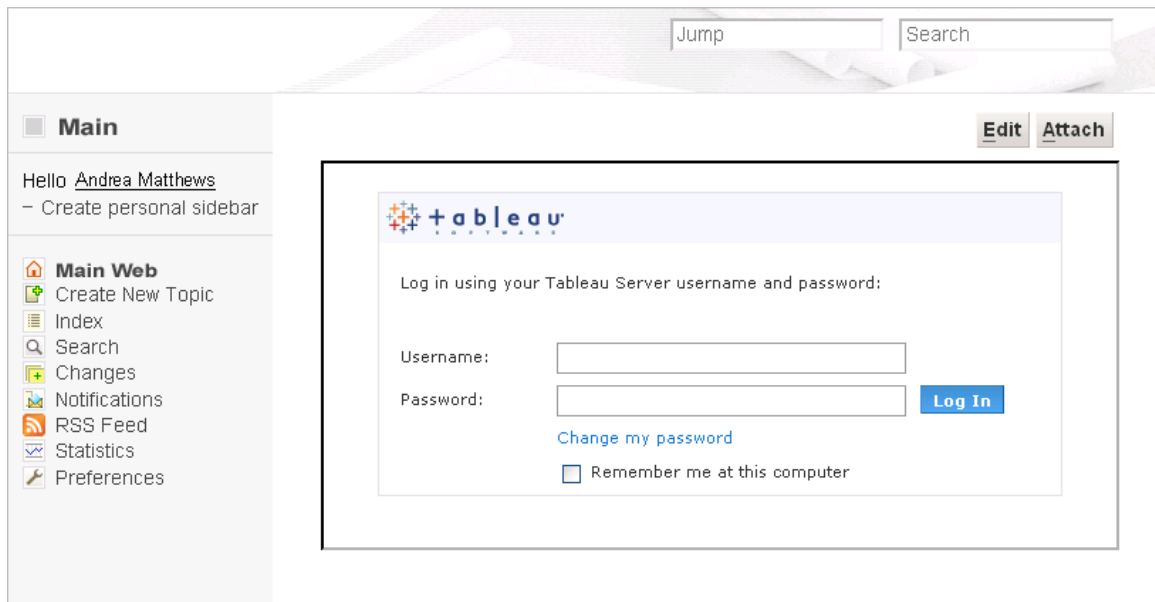
1. Navigate to the wiki page you want to embed a view into.
2. Edit the page and add an `<iframe>` where the source is the URL for the view. For example:

```
<iframe src="http://tableauserver/views/Date-
Time/DateCalcs?: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 logged 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 log into the server before they can see the view.



## Example: Embedding 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 `<img>` tag where the source is the URL for the view plus the `.png` file extension. For example:

```

```

### Note:

If both the web page and Tableau Server are using Microsoft SSPI for authentication, then anyone accessing the embedded image will be automatically logged into Tableau Server and be able to see the view. However, if the server and the web page are not using the same authentication method, the image will not display.

## Example: Embedding Views into SharePoint (Trusted Authentication)

If you are embedding a view into SharePoint but you don't use Microsoft SSPI for authentication, you can set up trusted authentication using the extra web part .dll installed with Tableau Server. Follow the instructions below to install the Tableau Web Part dll and embed a view into a SharePoint page.

1. Locate the TableauEmbeddedView.dll file that is installed with Tableau Server. The file is usually located in:

```
C:\Program Files\Tableau\Tableau
Server\7.0\extras\embedding\sharepoint\
```

2. Copy the .dll file into the root directory of your SharePoint server. The root directory is usually located at:

```
C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin
```

3. In a text editor, open the web.config file located at:

```
C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin
```

4. Add the following text to the bottom of the SafeControl section:

```
<SafeControl Assembly="TableauEmbeddedView, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=9f4da00116c38ec5"
Namespace="TableauEmbeddedView" TypeName="*" Safe="True" />
```

5. You also need to allow the webpart access to your SharePoint server. You can do this one of the following three ways:

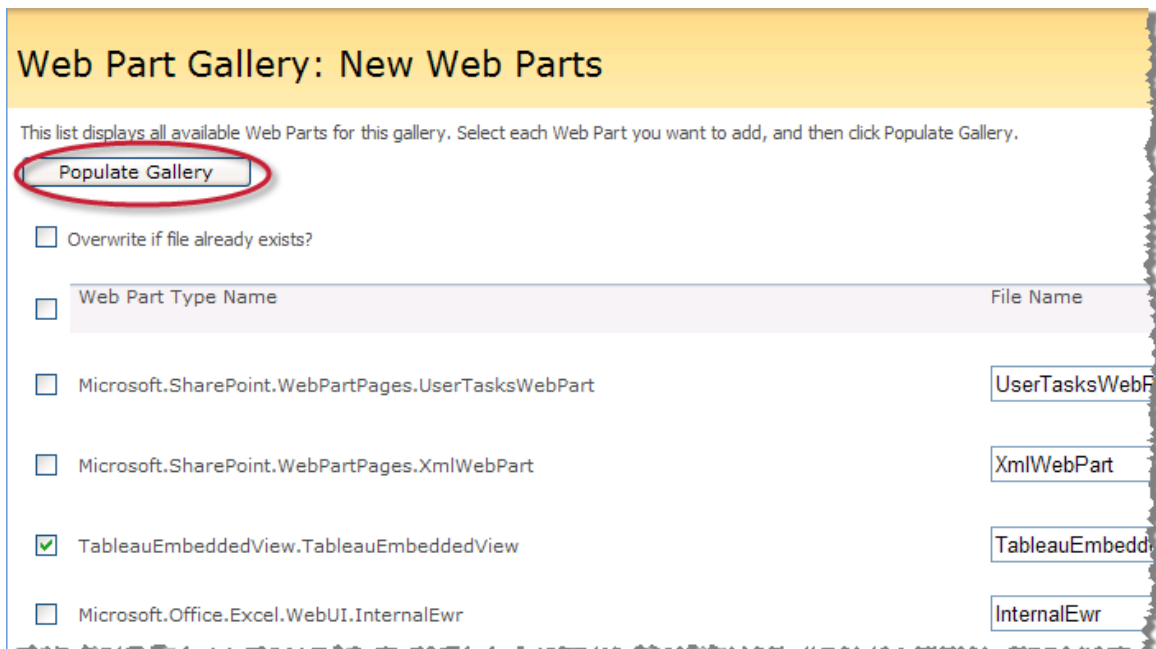
- o Copy the TableauEmbeddedView.dll file into your C:\Windows\assembly folder and delete it from the bin file you copied it into in step 2 above.
- o Reopen the web.config file you opened in step 3 above and find the following line:

```
<trust level="WSS_minimal" originUrl="" />
```

Change the line above to the following:

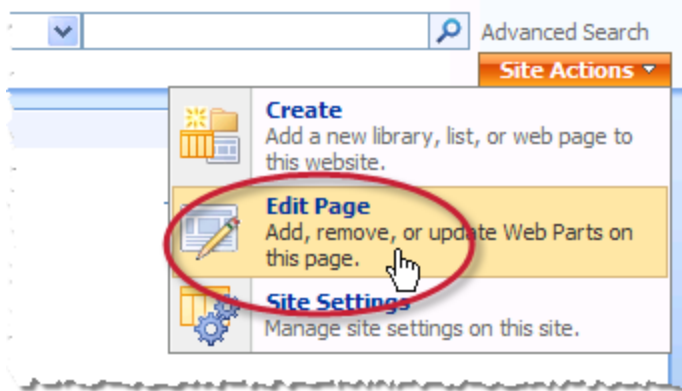
```
<trust level="Full" originUrl="" />
```

- o Create a custom trust policy, which will grant full access to the TableauEmbeddedView.dll only. Refer to the [Microsoft Technical Article](#) to learn more about how to do this.
6. Open a browser and navigate to:  
`http://<your_sharepoint_server>/_layouts/newdwp.aspx`.
  7. Select the entry titled TableauEmbeddedView.TableauEmbeddedView and click the Populate Gallery button.

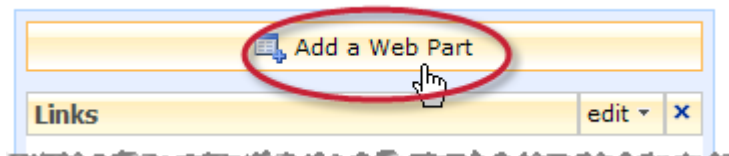


8. Navigate to the SharePoint page that you want to embed a view into.

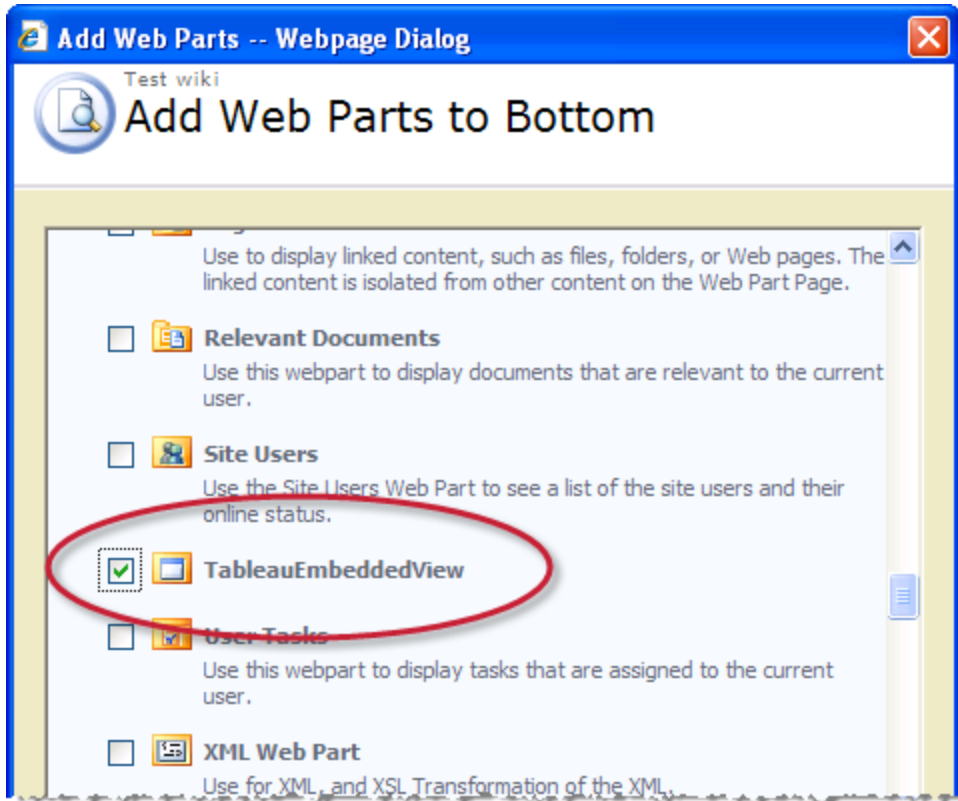
9. On the Site Actions menu in the upper right corner of the page select Edit Page.



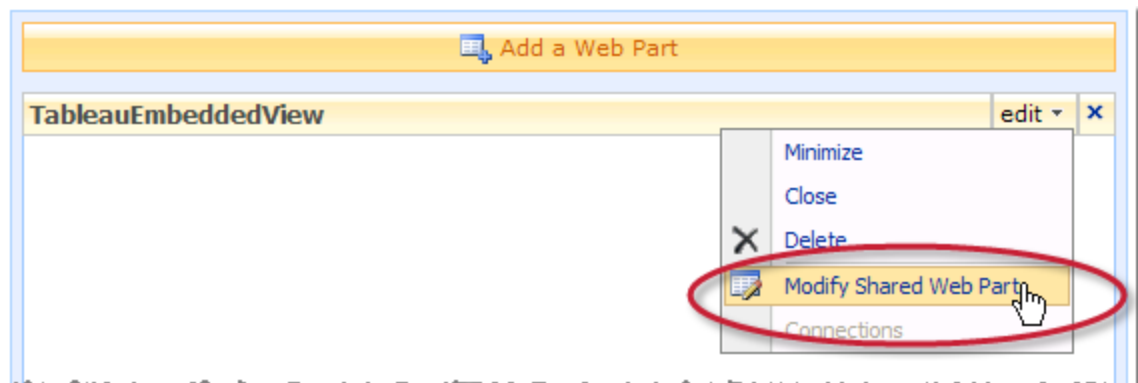
10. Click the Add a Web Part button in the section of the page where you want to embed the view.



11. On the page that opens, select TableauEmbeddedView located in the Miscellaneous section and click Add.



12. Back on the SharePoint page select Modify Shared Web Part on the Edit menu for the new web part.





13. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part. Type the name of your Tableau Server.
14. Then type the path to the view you want to embed. For example you may type /views/Date-Time/DateCalcs.

**Tableau View Settings**

Tableau Server Name

View Path

15. Specify other attributes such as whether you want to show the toolbar, or even if you want embed the view as an image instead of as an interactive view.

Show Comments

▼

Show Toolbar

---

Embed view as an image

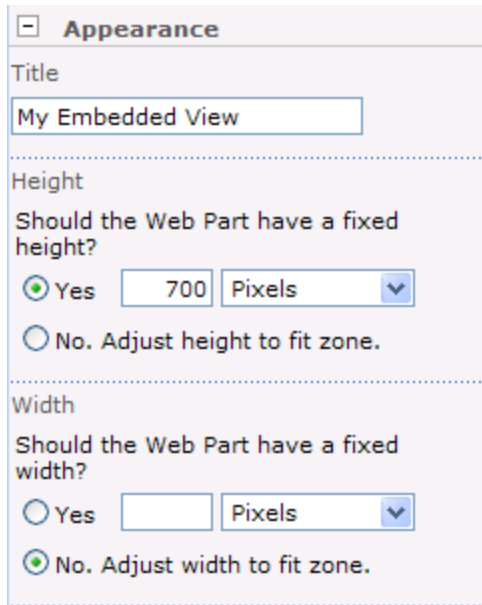
Width

 pixels

Height

 pixels

16. Then in the Appearance section you can specify a Title of the web part, the Height, and Width. In general you should specify a fixed height (e.g., 700 Pixels) and adjust the width to fit to the zone.



**Appearance**

Title  
My Embedded View

Height  
Should the Web Part have a fixed height?  
 Yes 700 Pixels  
 No. Adjust height to fit zone.

Width  
Should the Web Part have a fixed width?  
 Yes  Pixels  
 No. Adjust width to fit zone.

17. 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 logged in based on their user name and password for SharePoint. Anyone who accesses an embedded view needs to be a licensed user on Tableau Server and their user name on SharePoint must be the same as their user name on Tableau Server.

**Note:**

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 and even build your own .dll file. See [Using the Tableau JavaScript API](#) for more information.

# 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.

Use the topics below for more information:

- [Preparing to Configure for a Proxy Environment](#)
- [Configuring Tableau to Work with a Proxy Server](#)

## Preparing 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 machine. The address must be in IPv4 format, for example, *123.45.67.89*.
- **FQDN:** The fully-qualified domain name of the proxy server. For example, *bigbox.myco.com*.
- **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.myco.com* and aliases of *ftp.myco.com* and *www.myco.com*.

## Configuring Tableau to Work with a Proxy Server

After you collect the information described in [Preparing to Configure for a Proxy Environment](#), 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](#).

1. [Stop the server](#).
2. Still in the Tableau Server bin directory, enter the following command, where `name` is the canonical (externally-visible) name of the proxy server:

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

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 `IP_address` is the IP address of the proxy server:

```
tabadmin set gateway.trusted "IP_address"
```

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

```
tabadmin set gateway.trusted "123.45.67.89, 123.45.67.88,  
123.45.67.87"
```

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 "bigbox.myco.com, bigbox,  
ftp.myco.com, www.myco.com"
```

6. [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 log into 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 log 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.

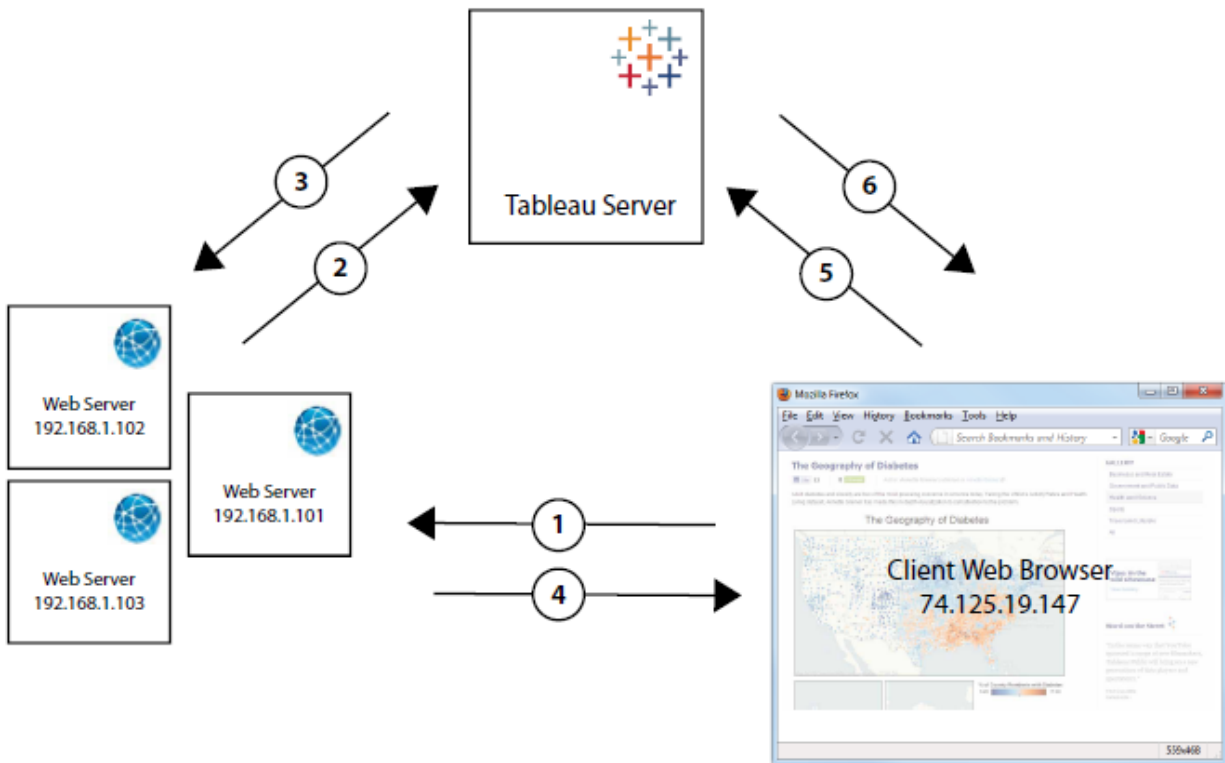
If your web server uses SSPI (Security Support Provider Interface), you do not need to set up trusted authentication. You can embed views and your users will have access to them as long as they are licensed Tableau Server users and members of your Active Directory. Using both [Enable Automatic Login](#) (an option configured during Setup that uses Microsoft SSPI) and trusted authentication is not supported.

If you are not using SSPI with Active Directory, you will need to set up trusted authentication to allow your users to avoid logging into Tableau Server.

- [How Trusted Authentication Works](#)
- [Adding Trusted IP Addresses to Tableau Server](#)
- [Getting a Ticket from Tableau Server](#)
- [Displaying the View with the Ticket](#)
- [Optional: Configuring Client IP Matching](#)
- [Troubleshooting Trusted Authentication](#)

## 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 Tableau Server. That POST request must have a `username`

**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/requiredviewname`). 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.

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 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 nine-digit 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.

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.

## Adding Trusted IP Addresses 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 (x86)\Tableau\Tableau Server\7.0\bin).
2. Next, type the following command:

```
tabadmin set wgserver.trusted_hosts "<Trusted IP Addresses>"
```

In the command above, <Trusted IP Addresses> should be a comma-separated list of the IP addresses of your web server(s). For example:

```
tabadmin set wgserver.trusted_hosts "192.168.1.101, 192.168.1.102,  
192.168.1.103"
```

**Note:**

The comma separated list should be within quotes with one space after each comma. Host names are not allowed.

3. If you have one or more proxy servers between the machine that is creating the trusted ticket (step 2, above) and Tableau Server, you also need to add them as trusted gateways. See [Configuring Tableau to Work with a Proxy Server](#) for steps.
4. Finally, type the following command to restart the server:

```
tabadmin restart
```

Next, you need to [configure your web server to receive tickets from Tableau Server](#).



## Getting 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](#)).

For code examples that you can use to create the POST request in Java, Ruby, and PHP, see the following:

- **32-bit:** C:\Program Files\Tableau\Tableau Server\7.0\extras\embedding
- **64-bit:** C:\Program Files (x86)\Tableau\Tableau Server\7.0\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's [Web Folder name](#).
- **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: Configuring Client IP Matching](#) for more information.

Tableau Server's response to the POST request will be a unique nine-digit 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](#) for tips on how to correct this.

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.

## Displaying 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 name>` to the path. For example:

```
http://tabserver/trusted/<ticket>/t/Sales/views/<workbook>/<view>
```

### 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="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value="123456789" />
</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="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="site_root" value="/t/Sales" />
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value="123456789" />
</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="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="path"
value="trusted/123456789/views/MyCoSales/SalesScoreCard" />
</object>
```

Here's the same example, but for a multi-site server. Note that `/t/<site name>` is used here:

```
<script type="text/javascript"
src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="path"
value="trusted/123456789/t/Sales/views/MyCoSales/SalesScoreCard" />
</object>
```

### Iframe Tag Example

```
<iframe
src="http://tabserver/trusted/123456789/views/workbookQ4/SalesQ4?:embed=yes"
width="800" height="600"></iframe>
```

## Optional: Configuring 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 (x86)\Tableau\Tableau Server\7.0\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
```

## Troubleshooting 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](#)
- [HTTP 401 - Not Authorized](#)
- [HTTP 404 - File Not Found](#)
- [Invalid User \(SharePoint or C#\)](#)
- [Attempting to Retrieve the Ticket from the Wrong IP Address](#)

## 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
```

Here are some things to confirm:

- **All web server IP addresses are added to trusted hosts**

The IP address for the machine sending the POST request must be in the list of trusted hosts on Tableau Server. See [Adding Trusted IP Addresses to Tableau Server](#) to learn how to add IP addresses to this list.

- **Trusted hosts list is properly formatted**

The list of trusted hosts on Tableau Server 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, and so on.

- **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 logging into 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`

## 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.

Concurrent use of Enable Automatic Login and trusted authentication is not supported.

## 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: Configuring Client IP Matching](#) to learn how to configure Tableau Server to enforce client IP address matching.





# 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 NT authentication, the AD account can also automatically handle the authentication process, 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. The steps under [Run As Account Settings to Confirm](#) may vary from site to site.

## 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 [Verifying Tableau Service Settings](#) for more information.

- [Identifying the Account](#)
- [Confirming Domain Two-Way Trust](#)
- [Verifying Tableau Service Settings](#)
- [Configuring Local Security Policy](#)
- [Configuring Data Source Connection Settings](#)
- [Run As Account Settings to Confirm](#)

## Identifying 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.

## Confirming 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.

## Verifying Tableau Service Settings

Confirm that Tableau services are assigned the correct Log On and Startup values:

1. Log on as administrator to the computer running Tableau Server.
2. On the Tableau Server computer, select Start > Control Panel > Administrative Tools > 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 Login	Local System	Automatic
Tableau Server (tabsvc)	<domain>\<username> This is the Run As user account. See below.	Automatic
Tablicsrv	Local System	Automatic

## Changing the Log On Value

To change the Log On value for Tableau Server (tabsvc) to the Run As User account:

1. In the Services window, stop the Tableau Server service by right-clicking Tableau Server (tabsvc) and selecting Stop.
2. Select Start > All Programs > Tableau Server > Configure Tableau Server.
3. On the General tab, enter the domain, username, and password for Tableau Server's Run As User account.
4. Click OK, then restart Tableau Server (tabsvc).

## Configuring Local Security Policy

If your Run As User account isn't an administrator on the Tableau Server machine, you must configure 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:

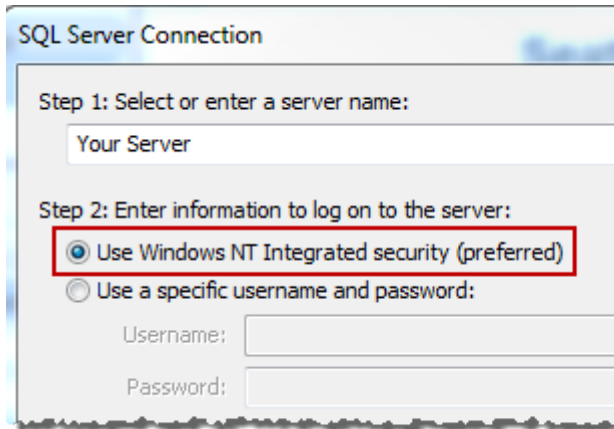
1. Select Start > Control Panel > Administrative Tools > Local Security Policy.
2. In the Local Security Settings window, open Local Policies, highlight User Rights Assignments, then right-click Log on as a service and select Properties.
3. In the Log on as a service Properties window, click Add User 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. Click OK to close the Local Security Settings windows.

## Configuring 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

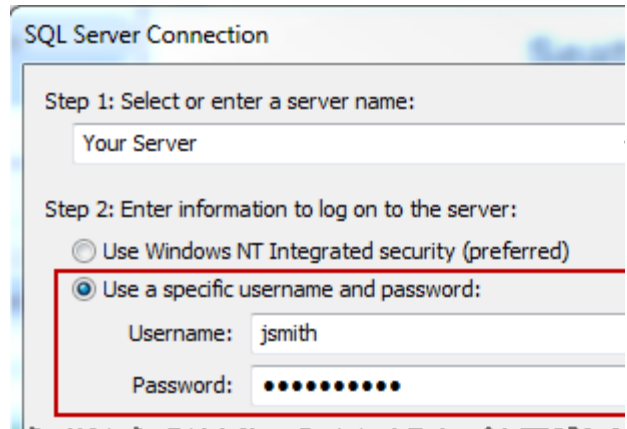
Authenticates with the server's Run As User account



The screenshot shows the 'SQL Server Connection' dialog box. Step 1 is 'Select or enter a server name:' with a text box containing 'Your Server'. Step 2 is 'Enter information to log on to the server:'. There are two radio button options: 'Use Windows NT Integrated security (preferred)' which is selected and highlighted with a red box, and 'Use a specific username and password:'. Below the second option are text boxes for 'Username:' and 'Password:'.

### Username and Password

Each Tableau Server user is prompted for database credentials



The screenshot shows the 'SQL Server Connection' dialog box. Step 1 is 'Select or enter a server name:' with a text box containing 'Your Server'. Step 2 is 'Enter information to log on to the server:'. There are two radio button options: 'Use Windows NT Integrated security (preferred)' and 'Use a specific username and password:' which is selected and highlighted with a red box. Below the second option are text boxes for 'Username:' containing 'jsmith' and 'Password:' containing ten black dots.

## Run As Account Settings to Confirm

The Run As User account needs permissions that allow it to read, execute, and sometimes modify files. Depending on the account you used as a starting point, it may already have the correct permissions. Any time you change the server's Run As account you should confirm that it meets the following requirements.

- [Granting Read and Execute Permissions](#)
- [Granting Modify Permissions](#)

### Granting Read and Execute Permissions

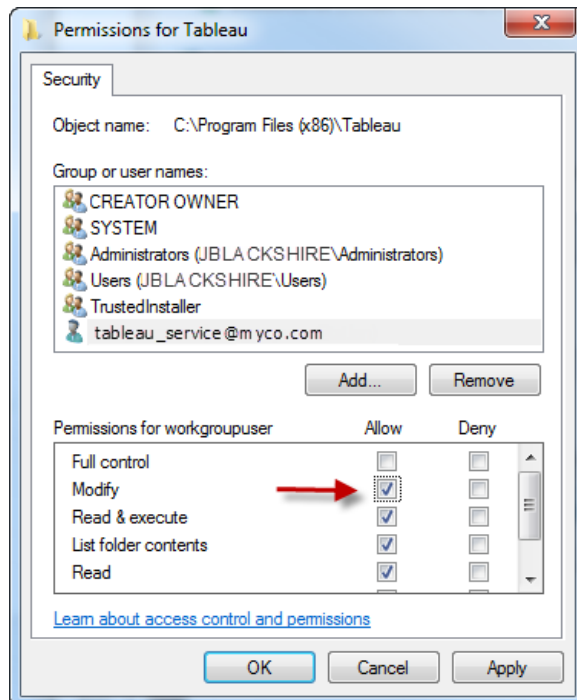
The account the Tableau Server service runs under needs permission to read and execute files. Any time the server's Run As User account is changed, confirm or configure the following:

1. On the machine hosting Tableau Server, use Windows Explorer to right-click on Local Disk (C:) and select Properties.
2. In the Local Disk (C:) Properties Window, select the Security tab.
3. Click Edit, then Add.
4. Type the <domain>\<username> for the Tableau Server Run As User 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.

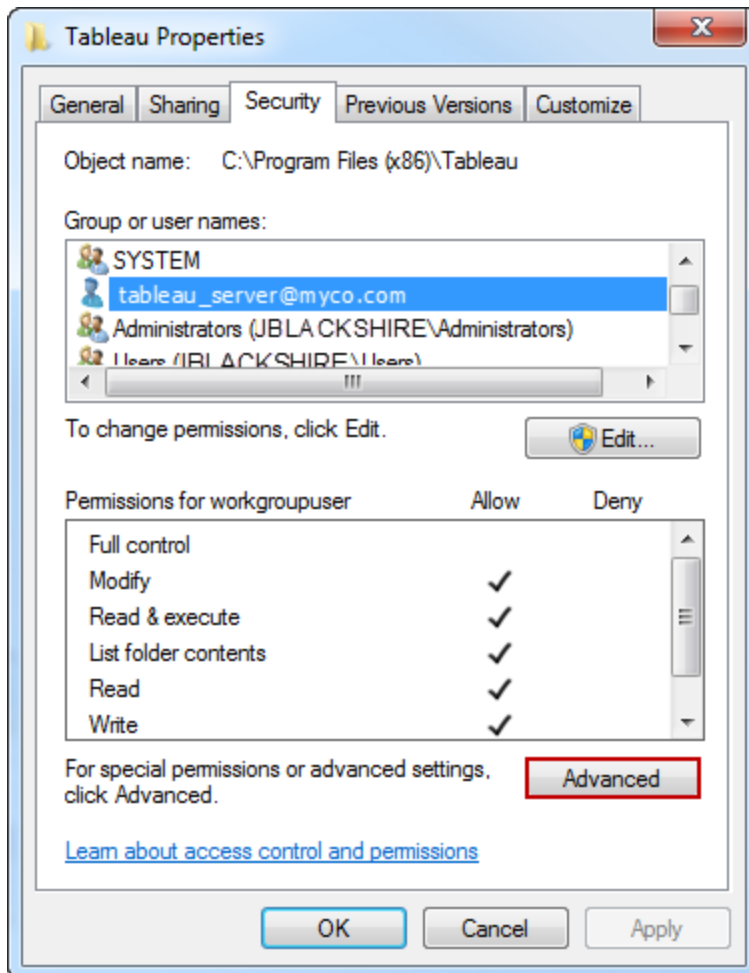
## Granting Modify Permissions

The account also needs the ability to do things like create log files. Confirm or configure the following:

1. Navigate to the following folders:
  - o 32-bit: C:\Program Files\Tableau
  - o 64-bit: C:\Program Files (x86)\Tableau\
  - o Windows Server 2008, Windows Vista, Windows 7:  
C:\ProgramData\Tableau\
2. Right-click the folder, select Properties, and click the Security tab:
  - o Click Edit, then Add.
  - o Type the <domain>\<username> for the Tableau Server Run As User account.
  - o Click Check Names to resolve the account, then OK to confirm.
  - o 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:

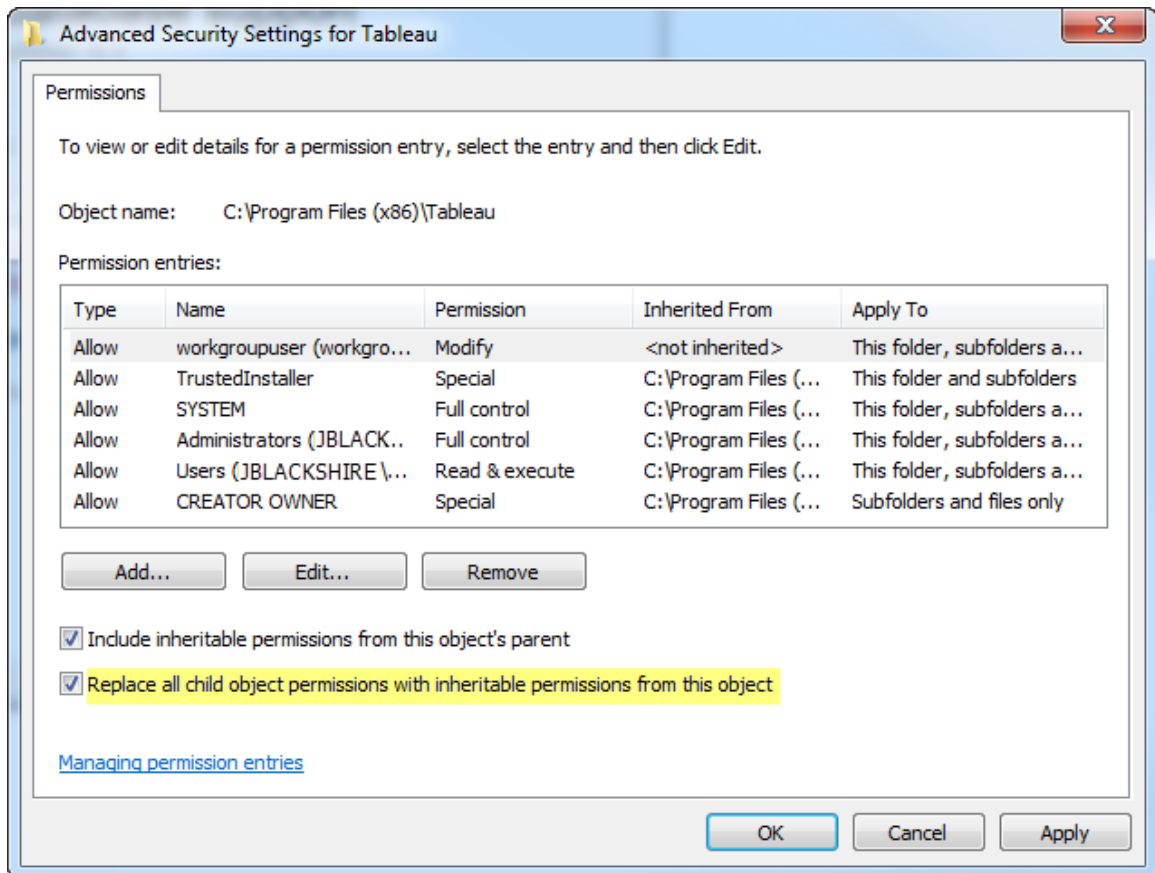


3. For each folder, on the Tableau Properties Security tab, click Advanced:



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.
7. Click OK to confirm changes.



# SQL Server Impersonation

Impersonation is when one user account acts 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 login prompt when they access 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](#)
- [How Impersonation Works](#)
- [Impersonating with a Run As User Account](#)
- [Impersonating with Embedded SQL Credentials](#)

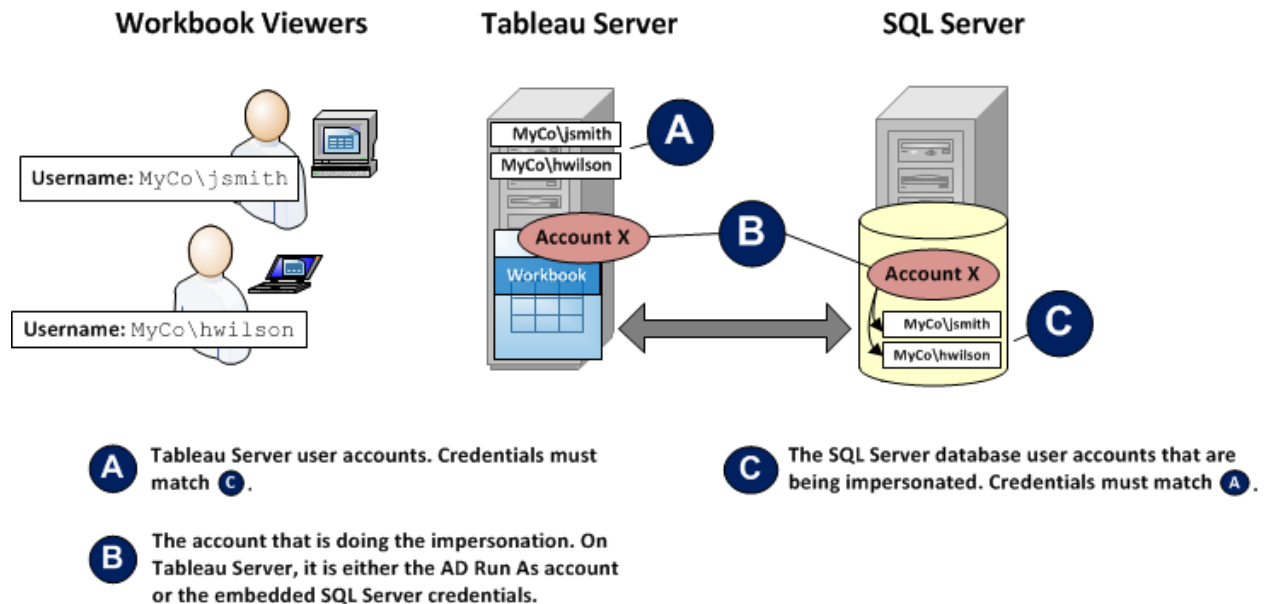
## Impersonation Requirements

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 [Impersonating with a Run As User Account](#)).
  - The workbook publisher's account (see [Impersonating with Embedded SQL Credentials](#)).

## 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 log into 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.

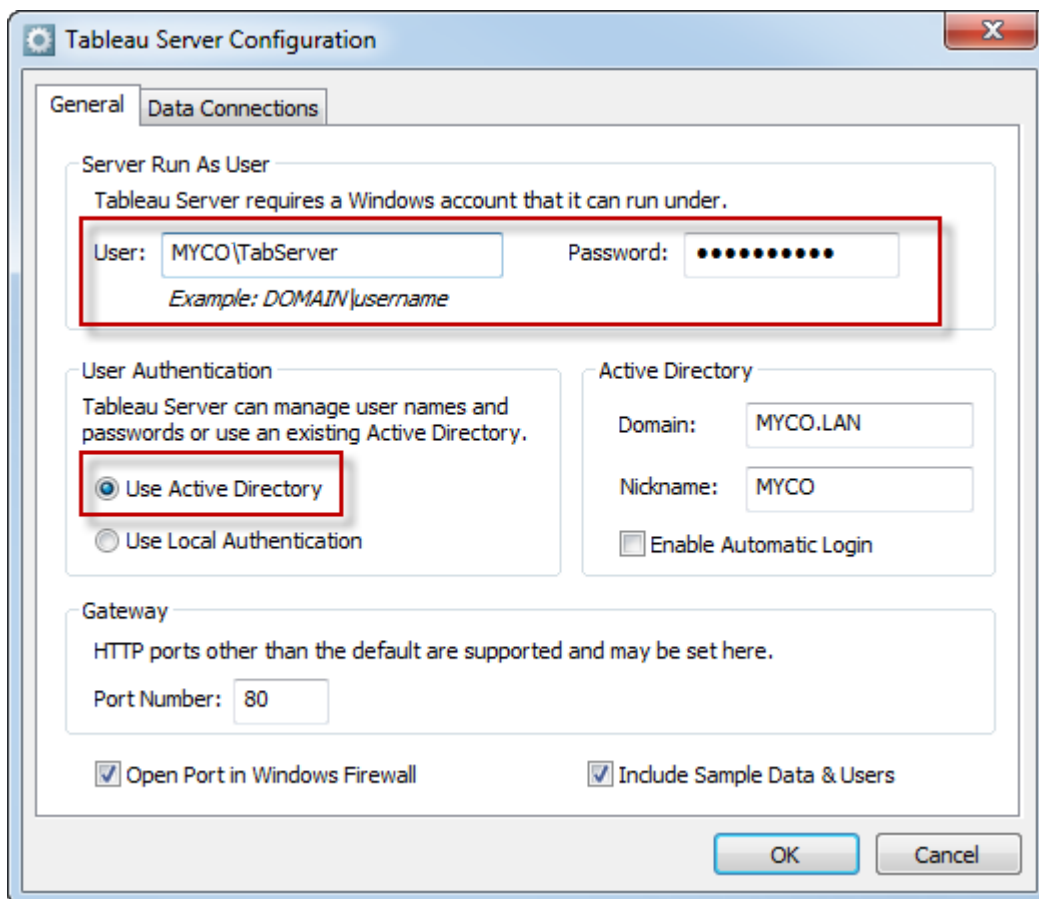
When the view displays, it is restricted by each user's individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.

## Impersonating 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 AD account the Tableau Server service can run under on the machine hosting Tableau Server (see [Run As User](#)). 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.

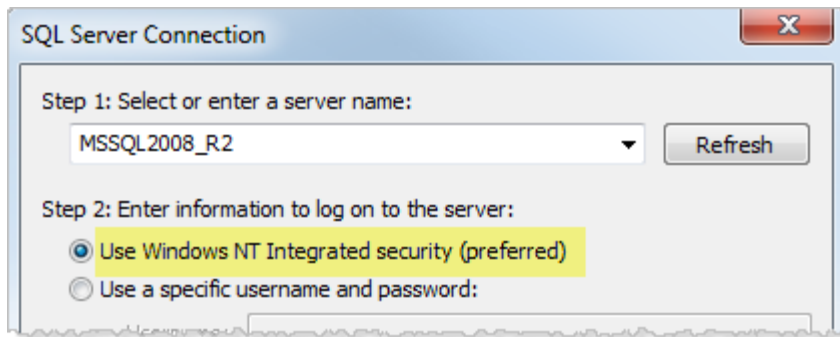
To set up impersonation with a Run As User account:

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:



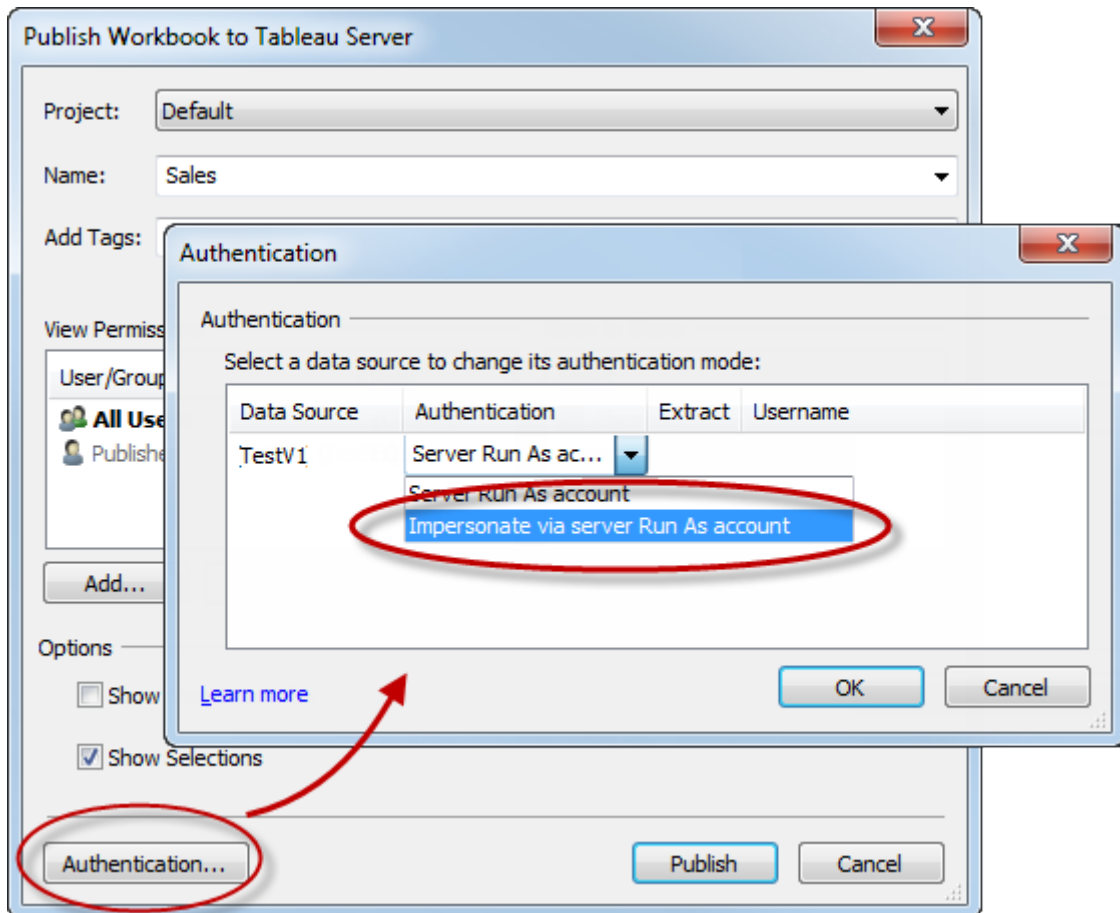
2. Click OK to finish configuration.

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 logging 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.

## Impersonating 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 AD account and if you're comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

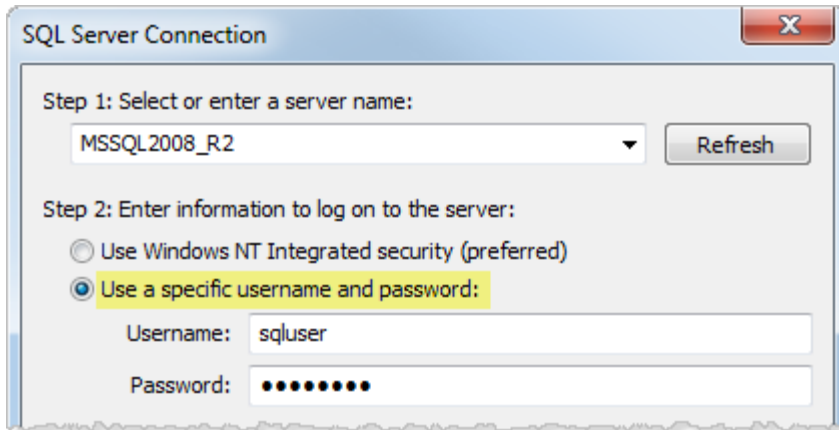
### Note:

To use this approach, [Embedded Credentials](#) must be enabled on Tableau Server:



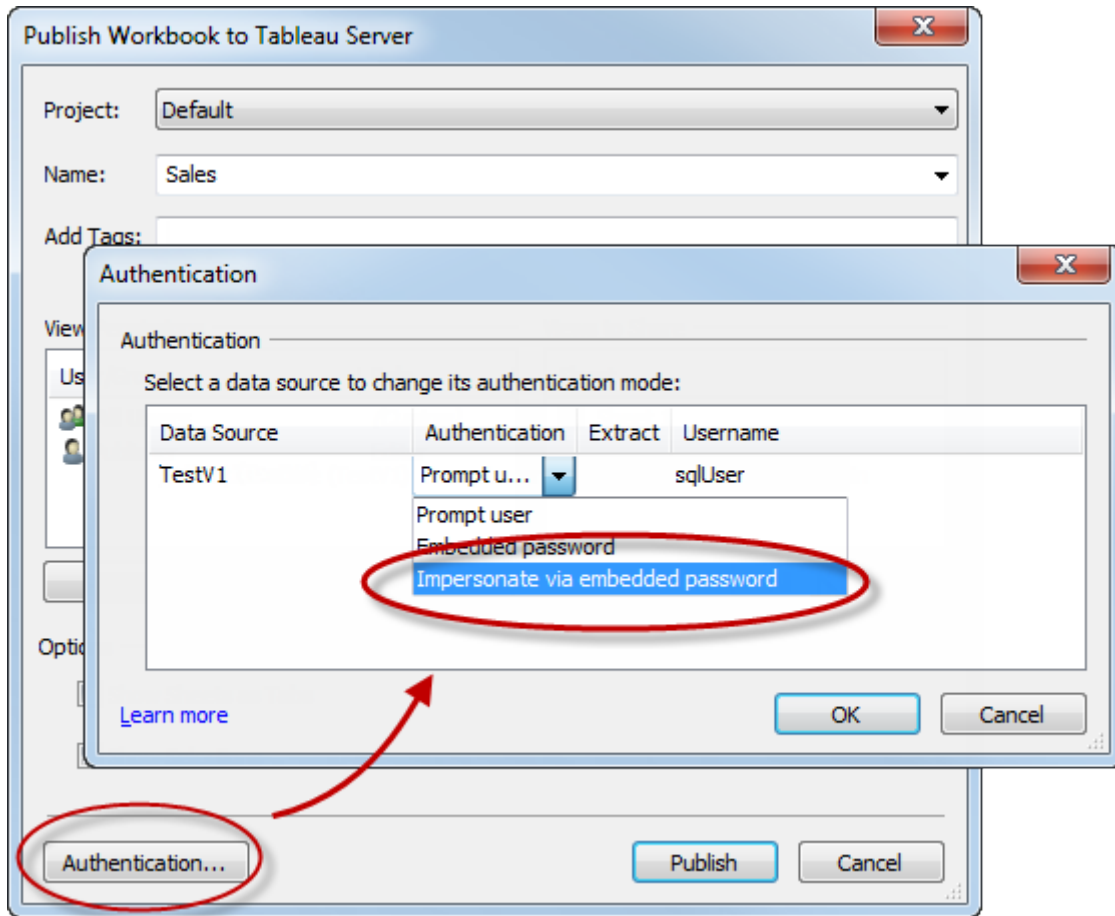
To impersonate with the workbook publisher's SQL account:

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 logging 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.



# TCP/IP Ports

The following table lists the ports that Tableau Server uses by default, and which must be available for binding. If Windows Firewall is enabled, Tableau Server will open the ports it needs—you do not need to do it yourself (for distributed installations with a worker running Windows 7, refer to the [Tableau Knowledge Base](#)).

<b>Port</b>	<b><u>Used by this server process...</u></b>	<b><u>TYPE OF INSTALLATION</u></b>			<b><u>Parameter</u></b>
		<b>All</b>	<b>Distributed</b>	<b>High Availability</b>	
80	Application server.	<b>X</b>			gateway.public.port, worker0.gateway.port
443	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port.	<b>X</b>			--
3729	Tableau Server Setup.	<b>X</b>			--
3730-3739	Tableau worker servers in <a href="#">distributed</a> and <a href="#">highly available</a> environments (the primary Tableau Server does not listen on these ports). Because the ports are determined dynamically during Setup, this port range is approximate.		<b>X</b>	<b>X</b>	--
8000 - 8059	Application server (base port 8000). Consecutive ports after 8000 are used, up to the number of processes. By default, Tableau Server installs with two application server processes (ports 8000 and 8001).	<b>X</b>			wgserver.port
8060	PostgreSQL database.	<b>X</b>			pgsql.port
8061	Firebird.	<b>X</b>			firebird.port
8062	Process that performs discovery in a distributed environment that's been			<b>X</b>	pgsql.initport

<b>Port</b>	<b>Used by this server process...</b>	<b><u>TYPE OF INSTALLATION</u></b>			<b><u>Parameter</u></b>
		<b>All</b>	<b>Distributed</b>	<b>High Availability</b>	
	configured for <a href="#">high availability</a> .				
8080	Solr and Tomcat HTTP.	<b>X</b>			solr.port, tomcat.http.port <sup>1</sup>
8300 - 8359	Application server <a href="#">JMX</a> . Determined by the application server port(s) + 300.	<b>X</b>			--
8550	Backgrounder monitor <a href="#">JMX</a> . Determined by the unused backgrounder port of 8250 + 300.	<b>X</b>			--
9090	Process that performs replication in a distributed environment that's been configured for <a href="#">high availability</a> .			<b>X</b>	rsync.port
9095	Service monitor <a href="#">JMX</a> .	<b>X</b>			svcmonitor.jmx.port
9100 - 9199	VizQL server (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).	<b>X</b>			vizqlserver.port
9400 - 9499	VizQL server <a href="#">JMX</a> . Determined by the VizQL server port(s) + 300.	<b>X</b>			--
9700 - 9899	Data server (base port 9700). Consecutive ports after 9700, up to the number of processes, are also used. By default, Tableau Server installs with two data server processes	<b>X</b>			dataserver.port

<u>Port</u>	<u>Used by this server process...</u>	<u>TYPE OF INSTALLATION</u>			<u>Parameter</u>
		All	Distributed	High Availability	
	(ports 9700 and 9701).				
10000 - 10299	Data server <a href="#">JMX</a> . Determined by the data server port(s) + 300.	X			--
27000 - 27009	Workers and primary server to communicate licensing information in <a href="#">distributed</a> and <a href="#">highly available</a> environments.		X	X	--
27042	Data engine. Tableau Server installs with one data server process. There can only be one active data engine per deployment.	X			dataengine.port
27043	Data engine initialization in a distributed environment that's been configured for <a href="#">high availability</a> .			X	

- [Editing the Default Ports](#)
- [About the JMX Ports](#)
- [Restoring the Default Value for a Port](#)

<sup>1</sup> These parameters must be set to the same value.

## Editing the Default Ports

You can modify the default ports used by Tableau Server processes by using the command line administrative tool, [tabadmin](#). For example, the default port for the application server process (`wgserver`) is 8000. You can use the `tabadmin` parameter `workerX.wgserver.port` to change it to a different port. Follow the steps below to change the Tableau Server port configuration.

1. Open a command prompt as an administrator and type the following:

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"
```

2. Modify a port value by typing the following:

```
tabadmin set <workerX>.<parameter> <new port value>
```

In the above command, `<workerX>` refers to the machine whose port you want to change, `<parameter>` is one of the values in the table below (a server process' port, such as `wgserver.port`), and `<new port value>` is the new port number you want the server process to use. 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. In this last case, you would need to run the command (from a command prompt on the primary) once for each machine in the cluster.

Here's an example that sets the port on the primary or a standalone server to 8020 for the application server process (`wgserver`):

```
tabadmin set worker0.wgserver.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 [TCP/IP Ports](#) for a complete list of tabadmin parameters that can be set.

Port to Change	Parameter
80	gateway.public.port, worker0.gateway.port
8000	wgserver.port
8060	ppgsql.port
8080	solr.port, tomcat.http.port <sup>1</sup>
9100	vizqlserver.port
9700	dataserver.port

3. After you make the necessary port configuration changes, restart the server services by typing the following:

```
tabadmin restart
```

**Note:**

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.

<sup>1</sup> These parameters should be set to the same value.

## About the JMX Ports

The JMX ports are used for optional monitoring and troubleshooting with tools like JConsole. While Tableau Server does not actually use the JMX ports, they must still be available for binding.

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”. 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.

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 [Editing the Default Ports](#)) to change the data server base port from 9700 to 9800. This will move the data server JMX port to 11000.

The backgrounder process is unique in that it doesn’t use its base port of 8250, but that port number is used to determine its JMX port of 8550 (8250 + 300).

To reduce security risks, it’s a good practice to configure your firewall to block outside traffic to the JMX ports.

## Restoring 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 (x86)\Tableau\Tableau Server\7.0\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 the Tableau Server services by typing the following:

```
tabadmin restart
```





# tabcmd

tabcmd is one of the two command line tools that come with Tableau Server. It helps you automate common tasks including batch publishing workbooks and user and group administration. For information on Tableau Server's other command line tool, refer to [tabadmin](#).

- [How to Use tabcmd](#)
- [tabcmd Global Options](#)
- [tabcmd Commands](#)

## How to Use tabcmd

tabcmd takes a command, an argument, and options as shown in the format below:

```
tabcmd command command-argument [options option-arguments]
```

Using that format and the commands in this document you can run the tool. For example, you could use the following command to create a session on a server called sales-server logged in as Administrator and delete a workbook on the Sales site called Sales\_Analysis:

```
tabcmd delete "Sales_Analysis" -s sales-server -t Sales -u administrator -p p@ssw0rd!
```

Here's the same command for a workbook on the Default site, or for a server that is not running multiple sites:

```
tabcmd delete "Sales_Analysis" -s sales-server -u administrator -p p@ssw0rd!
```

When the command is successful, tabcmd will return 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:

- Windows 7, Windows Vista, and Windows Server 2008 R2:  
C:\Users\\AppData\Roaming\Tableau
- Windows Server 2003: C:\Documents and Settings\\Application Data\Tableau

## tabcmd Global Options

Some options are common to all commands. 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.

<b><u>Option (short)</u></b>	<b><u>Option (long)</u></b>	<b><u>Argument</u></b>	<b><u>Description</u></b>
-h	--help		Displays the help for the command.
-s	--server	Tableau Server URL	Required at least once to begin session.
-u	--user	Tableau Server username	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.
-P	--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 name	Use the specified Tableau Server site. If you do not specify a site, the Default site is assumed. Only applies to servers with multiple sites.
-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-]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.

<b><u>Option (short)</u></b>	<b><u>Option (long)</u></b>	<b><u>Argument</u></b>	<b><u>Description</u></b>
	--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.

## tabcmd Commands

Here are the commands that can be used with the tabcmd command line tool:

<a href="#">addusers group-name</a>	<a href="#">get url</a>
<a href="#">creategroup group-name</a>	<a href="#">listsites</a>
<a href="#">createproject project-name</a>	<a href="#">login</a>
<a href="#">createsite site-name</a>	<a href="#">logout</a>
<a href="#">createsiteusers filename.csv</a>	<a href="#">publish filename.twb(x) or filename.tds(x)</a>
<a href="#">createusers filename.csv</a>	<a href="#">refreshextracts workbook-name or datasource-name</a>
<a href="#">delete workbook-name or datasource-name</a>	<a href="#">removeusers group-name</a>
<a href="#">deletegroup group-name</a>	<a href="#">runcschedule schedule-name</a>
<a href="#">deletesite site-name</a>	<a href="#">set setting</a>
<a href="#">deleteusers filename.csv</a>	<a href="#">syncgroup group-name</a>
<a href="#">editsite site-name</a>	<a href="#">version</a>
<a href="#">export</a>	

### addusers group-name

Adds the users listed in the --users argument to the group with the given group-name.

#### 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 username per line. The users should already be created on Tableau Server. See also <a href="#">CSV</a>

Option (short)	Option (long)	Argument	Description
			<a href="#">Guidelines.</a>
	--[no- ]complete		When set 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 with the given group name. 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 with the given project name.

### 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 with the given site name.

### Example

```
tabcmd createsite "Sales"
```

## createsiteusers filename.csv

This command allows content administrators to add users to a site. It creates users on the current site, using the given comma separated values (csv) file. The file may have the following columns, in the order shown below:

1. Username
2. Password
3. Full Name
4. License Level (interactor/viewer/unlicensed)
5. Administrator (content/none)
6. Publisher (yes/true/1 or no/false/0)

The file can have fewer columns. For example it can be a simple list with one username per line. When the server is using Active Directory authentication, the `password` column is ignored. Quotes may be used if a value contains commas. See [CSV Guidelines](#) for other details.

### Example

```
tabcmd createsiteusers "users.csv" --license "Interactor" --publishers
```

Option (short)	Option (long)	Argument	Description
	<code>--nowait</code>		Do not wait for asynchronous jobs to complete.
	<code>--silent-progress</code>		Do not display progress messages for asynchronous jobs.

<b>Option (short)</b>	<b>Option (long)</b>	<b>Argument</b>	<b>Description</b>
	<code>--license</code>	Interactor, Viewer, or Unlicensed	Sets the default license level for all users. This setting may be overridden by the value in the CSV file.
	<code>--admin-type</code>	Content or None	Assigns or removes the content admin right to all users in the CSV file. This setting may be overridden by the value in the CSV file. The default is None for new users and unchanged for existing users. System administrators cannot be created or demoted using <code>createsiteusers</code> (use <code>createusers</code> instead).
	<code>--[no-]publishers</code>		Assigns or removes the Publish right to all users in the CSV file by default. This setting may be overridden by the value in the CSV file. The default is no for new users and unchanged for existing users.
	<code>--[no-]complete</code>		Require (or not require) that all rows be valid for any change to succeed. By default, <code>--complete</code> option is used.

**createusers filename.csv**

Creates the users listed in the given comma separated values (csv) file. This command can only be used by system administrators. The file may have the following columns, in the order shown below:

1. Username
2. Password
3. Full Name
4. License Level (interactor/viewer/unlicensed)
5. Administrator (system/content/none)
6. Publisher (yes/true/1 or no/false/0)

The file can have fewer columns. For example it can be a simple list with one username per line. When the server is using Active Directory authentication, the `password` column should be left blank. Quotes may be used if a value contains commas. See [CSV Guidelines](#) for other details.

**Example**

```
tabcmd createusers "users.csv" --license "Interactor" --publishers
```

Option (short)	Option (long)	Argument	Description
	<code>--nowait</code>		Do not wait for asynchronous jobs to complete.
	<code>--silent-progress</code>		Do not display progress messages for asynchronous jobs.
	<code>--license</code>	Interactor, Viewer, or Unlicensed	Sets the default license level for all users. This setting may be overridden by the value in the CSV file.
	<code>--admin-type</code>	System, Content, or None	Assigns or removes the Admin right to all users in the CSV file by default. This setting may be overridden by the value in the CSV file. The default is None for new users and unchanged for existing users.
	<code>--[no-]publishers</code>		Assigns the Publish right to all users in the CSV file by default. This setting



Option (short)	Option (long)	Argument	Description
			may be overridden by the value in the CSV file. The default is no for new users and unchanged for existing users.
	--[no- ]complete		Requires that all rows be valid for any change to succeed. By default, --complete option is used.

## delete workbook-name or datasource-name

Deletes the given 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.
	--datasource	Data source name	The name of the data source you want to delete.

## deletegroup group-name

Deletes the group with the given group-name from the server.

### Example

```
tabcmd deletegroup "Development"
```

## deletesite site-name

Deletes the site with the given site-name from the server.

### Example

```
tabcmd deletesite "Development"
```

## deleteusers filename.csv

Deletes the users listed in the given comma separated (csv) file. The file is a simple list of one username 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.

## editsite site-name

Allows you to change the name of a site or its web folder name. You can also use this command to allow or deny content administrators the ability to add and remove users. If content administrators have user management rights, you can specify how many users they can add to a site.

### Examples

```
tabcmd editsite wc_sales -n "West Coast Sales"
```

```
tabcmd editsite wc_sales --url "wsales"
```

```
tabcmd editsite wsales --content-mode
```

```
tabcmd editsite wsales --user-quota 50
```

Option (short)	Option (long)	Argument	Description
-n	--site-name	Name to change the site to	Change the name of the site.
-r	--url	URL namespace to change the site to	The URL namespace of the site. In the UI, this is called the Site ID.
-n	--[no-]content-mode		Allow or prevent content administrators (who are members of the site) from adding users to the site.
	--user-quota		The maximum number of users who can be members of the site.

## export

Exports a view or workbook from Tableau Server and saves it to a file. 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.
- **The view, workbook, or data being exported:** You specify this using the "workbook/view" string as it appears in the URL for the workbook or view, not using its "friendly name." For example, to export the Tableau sample view *Investment Growth* from the *Finance* workbook, you would use the string `Finance/InvestmentGrowth`. 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`), a PNG (`--png`), or you can export the view's data as a CSV file (`--csv`).
- **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.

## Examples

### Views

```
tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report"
```

```
tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f
"C:\Tableau_Workbooks\Weekly-Reports"
```

```
tabcmd export "Finance/InvestmentGrowth" --png
```

## Workbooks

```
tabcmd export "Q1Sales/Sales_Report" --fullpdf
```

```
tabcmd export -t Sales "Sales/Sales_Analysis" --fullpdf --pagesize tabloid -f
"C:\Tableau_Workbooks\Weekly-Reports"
```

Option (short)	Option (long)	Argument	Description
-f	--filename	Name to save the file as	Saves the file with the given filename.
	--csv		View only. Export the view's data in CSV format.
	--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, quatro	Sets the page size of the exported PDF. Default is letter.
	--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.

## get url

Using a URL string as one of its parameters, makes an HTTP “GET” request of Tableau Server. 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 File 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 string of the file you want to GET must include a file extension—such as `"/views/Finance/InvestmentGrowth.pdf"`. The extension (for example, `.pdf`) determines what's returned. A view can be returned in PDF, PNG, CSV (data only), or XML (information only) format. A workbook can be returned as a TWB or TWBX. To figure out the correct extension to use, you can use a web browser to navigate to the item you're interested in on Tableau Server and add the file extension to the end of 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 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.

### 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.csv"
```

#### *Workbooks*

```
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Workbooks\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 filename.

## listsites

Returns a list of sites to which the logged in user belongs.

### Example

```
tabcmd listsites -u corman -pw P@ssword!
```

## login

Logs into the server. Use the `--server`, `--site`, `--username`, `--password` global options to create a session. 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 username stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify it.

You only need the `--site` option 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 login, the session will continue until it expires on the server or the `logout` command is run.

### Example

```
tabcmd login -s http://sales-server -t Sales -u administrator -p p@ssw0rd!
```

```
tabcmd login -s https://sales-server -t Sales -u administrator -p p@ssw0rd!
```

```
tabcmd login -s http://sales-server:8000 -t Sales -u administrator -p  
p@ssw0rd!
```

## logout

Logs out of the server.

### Example

```
tabcmd logout
```



## publish filename.twb(x) or filename.tds(x)

Publishes the given workbook (.twb(x)) or data source (.tds(x)) to Tableau Server. By default, all sheets in the workbook are published without database usernames or passwords.

### Example

```
tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-user "jsmith" --db-
password "p@ssw0rd"
```

Option (short)	Option (long)	Argument	Description
-n	--name	Name of the workbook or data source on the server	If omitted, the workbook or data source will be named after filename.
-o	--overwrite		Overwrites the workbook or data source if it already exists on the server.
-r	--project	Name of a project	Publishes the workbook or data source into the specified project. Publishes to the "Default" project if not specified.
	--db-username		Use this option to publish a database username with the workbook or data source.
	--db-password		Use this option to publish a database password with the workbook or data source.
	--save-db-password		Stores the provided database password on the server.
	--thumbnail-username		If the workbook contains users filters, the thumbnails will be generated based on what the specified user can see. Cannot be specified when --thumbnail-group option is set.
	--thumbnail-group		If the workbook contains users filters the thumbnails will be generated based on what the specified group can see. Cannot be specified when --thumbnail-username option is set.
	--tabbed		When a workbook with tabbed views is published, each sheet becomes a tab

Option (short)	Option (long)	Argument	Description
			that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.

If the workbook contains user filters, one of the thumbnail options must be specified.

## 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.

### 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		Runs the full refresh operation immediately in the foreground.
	--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
	<code>--url</code>	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 the users listed in the `--users` argument from the group with the given group-name.

### Example

```
tabcmd removeusers "Development" --users "users.csv"
```

Option (short)	Option (long)	Argument	Description
	<code>--users</code>	filename.csv	Remove the users in the given file from the specified group. The file should be a simple list with one username per line.
	<code>--[no-] complete</code>		Requires that all rows be valid for any change to succeed. If not specified <code>--complete</code> is used.

## runschedule schedule-name

Runs the specified schedule. This command takes the name of the schedule as it is on the server.

### 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:

- `embedded_credentials`
- `public_users_list`
- `remember_passwords_forever`

### Example

```
tabcmd set embedded_credentials
```

## syncgroup group-name

Synchronizes the group with the given group-name with Active Directory. This command can also be used to create a new group on the server that is based on an existing Active Directory group.

### Example

```
tabcmd syncgroup "Development"
```

Option (short)	Option (long)	Argument	Description
	<code>--license</code>	<code>viewer</code> <code>interactor</code> <code>unlicensed</code>	Sets the license level for all users in the group.
	<code>--administrator</code>	<code>system</code> <code>content</code> <code>none</code>	Assigns or removes the Administrator right for all users in the group. The Administrator user type can be system, content, or none. Default is none (new users do not get the Administrator right) and existing users are unchanged.
	<code>--[no-]publisher</code>		Assigns or removes the Publish right for all users in the gorup. If

Option (short)	Option (long)	Argument	Description
			unspecified, new users are not assigned this right and existing users are unchanged.
	--[no- ]complete		Requires that all rows be valid for any change to succeed. If unspecified, --complete is used.
	--silent- progress		Suppresses progress messages.

## version

Prints 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. To access it, open a command prompt as an administrator and change directories using the command below:

- 32-bit: `cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"`
- 64-bit: `cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"`

To see a list of all available tabadmin commands, type the following:

```
tabadmin help commands
```

For more information, see the topics below:

- [tabadmin set](#)
- [tabadmin set options](#)
- [Restoring a Setting to its Default Value](#)
- [tabadmin stop](#)
- [tabadmin start](#)

## tabadmin set

One of the most commonly used `tabadmin` commands is `tabadmin set`, which allows you to change the value of [Tableau Server configuration options](#). The syntax is as follows:

```
tabadmin set option-name value
```

To use this command:

1. [Stop the server](#).
2. Type `tabadmin set` followed by the name of the option and the value.

For example, to change the default value of the `tabadmin` option [backgrounder.querylimit](#) from 7200 seconds (2 hours, the default) to 9000 seconds, type the following:

```
tabadmin set backgrounder.querylimit 9000
```

If you're setting an option whose value begins with a hyphen, use nested quotes:

```
tabadmin set option-name "'-value'"
```

```
tabadmin set option-name "'-value1 -value2'"
```

3. After you use `tabadmin set`, enter the `configure` command:

```
tabadmin configure
```

4. Finally, [start Tableau Server](#).



## tabadmin set options

Use the table below to learn more about Tableau Server options you can configure using the [tabadmin set](#) command. See [TCP/IP Ports](#) for a complete list of ports.

<b><u>Option</u></b>	<b><u>Default Value</u></b>	<b><u>Description</u></b>
backgrounder.querylimit	7200	Longest allowable time for completing an extract refresh, in seconds (7200 seconds = 2 hours).
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 canonical name of the server, used for external access to Tableau Server. If Tableau Server is configured to work with a proxy server, it is the canonical name of the proxy server (not Tableau Server).
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) 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.
java.heap.size	128m	Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.
pgsql.port	8060	Port that PostgreSQL listens on.
service.max_procs	# of processes	<a href="#">Maximum number of server processes.</a>
solr.port	8080	Port that solr listens on. This must be the same value as tomcat.http.port.
tomcat.http.port	8080	Port that Tomcat runs on.
tomcat.https.port	8443	SSL port for Tomcat (unused).
tomcat.server.port	8085	Port that tomcat listens on for shutdown

<b><u>Option</u></b>	<b><u>Default Value</u></b>	<b><u>Description</u></b>
		messages.
vizqlserver.port	9100	Base port for the VizQL servers.
vizqlserver.querylimit	1800	Longest allowable time for updating a view, in seconds.
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.trustedticket.log_level	info	The logging level for trusted authentication, written to ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\vizql-*.log. Set to debug for more information.
wgserver.domain.fqdn	value of %USERDOMAIN%	The fully qualified domain name of the Active Directory server to use.
wgserver.session.idle_limit	240	The number of minutes of idle time before a login to the web application times out.
wgserver.show_view_titles_not_names	true	You can only use this option if you upgraded to version 7.0 from 6.0 or earlier. A value of true keeps the earlier behavior and causes Tableau Server to display view titles as their identifiers (for ex., in search results); false causes the server to display view names as their identifiers.
wgserver.trusted_hosts		This option takes a comma separated list of trusted IP addresses (not host names) for the machine you want to accept trusted requests from. A common value is 127.0.0.1 if you want to put the webserver and Tableau Server on the same machine. This option is used when setting up a trusted relationship between the web server and Tableau Server when embedding views.
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

<b><u>Option</u></b>	<b><u>Default Value</u></b>	<b><u>Description</u></b>
		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.

## Restoring 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:

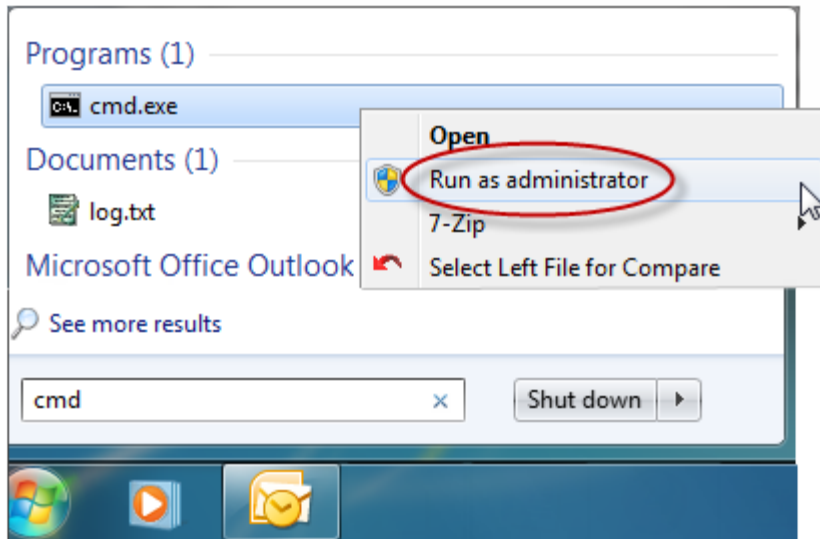
```
tabadmin configure
```

4. [Start the server.](#)

## tabadmin stop

To use tabadmin to stop Tableau Server:

1. Open a command prompt as an administrator:



2. Type the following:
  - o 64-bit: `cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"`
  - o 32-bit: `cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"`

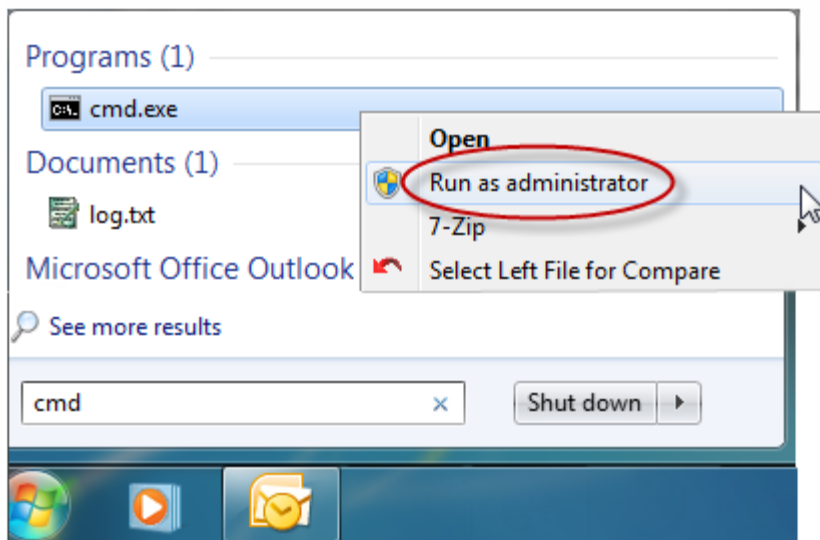
3. Type the following to stop the server:

```
tabadmin stop
```

## tabadmin start

To use tabadmin to start Tableau Server:

1. Open a command prompt as an administrator:



2. Type the following:
  - o 64-bit: `cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\bin"`
  - o 32-bit: `cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"`

3. Type the following to start the server:

```
tabadmin start
```

# Database Maintenance

Use the Tabadmin command line tool to back up and restore the database. To automate backups, you can use the commands described in the topics below, along with the built-in Windows task scheduler.

- [Backing Up the Database](#)
- [Restoring from a Backup](#)
- [Recovering Extracts from a Backup](#)

## Backing Up the Database

It is important to back up the database so you can restore the published views in the case of a system failure. When you back up the database a single file is created with the `.tsbak` file extension. This file contains the contents of the database and the configuration files. Be sure to store this file on a different computer. .

### 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](#).

1. Open a command prompt as an administrator and type the following:

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau  
Server\7.0\bin"
```

2. Create a backup file by typing:

```
tabadmin backup filename --stop-server -d
```

In the above line, replace `filename` with the name of your backup file. The `--stop-server` option stops the server for the backup and then restarts it when it is done. If you are doing a simple backup you must either have this option or the `--unsafe` option to backup while the server is running. Backups that are part of a batch operation do not require these options.

The `-d` is optional; if included the current date is appended to the file name.

If you are running a distributed installation of Tableau Server (primary and workers) and you want to create a `.tsbak` file that excludes the worker configuration details (such as the worker IP addresses), remove the workers from the server configuration before you create the `.tsbak` file.



## Restoring from a Backup

When you restore, the contents of the database as well as configuration files are overwritten with the content in the backup file. 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 services by typing:

```
tabadmin start
```

## Recovering Extracts from a Backup

The file *uninstall-<version>.tsbak* (for example, *uninstall-6.1.tsbak*) is created as part of the uninstall process. After you upgrade to version 7.0, 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 7.0 Tableau Server bin directory, type the following:

```
Windows Server 2008, Windows Vista, and Windows 7: tabadmin restore  
\ProgramData\Tableau\Tableau Server\uninstall-6.1.tsbak
```

```
64-bit Windows Server 2003: tabadmin restore \Program Files  
(x86)\Tableau\Tableau Server\uninstall-6.1.tsbak
```

```
32-bit Windows Server 2003: tabadmin restore \Program  
Files\Tableau\Tableau Server\uninstall-6.1.tsbak
```

# Troubleshooting

Use the following topics to troubleshoot issues you may be having with Tableau Server:

- [Logs and Temporary Files](#)
- [Handling an Unlicensed Server](#)
- [Handling an Unlicensed VizQL Server Service](#)
- [Repository Migration Error](#)
- [VizQL "Out of Memory" Error](#)

## Logs and Temporary Files

The Tableau Service generates several logs and temporary files that can help you understand and track recent activity as well as debug any problems that may arise. If you need to save space on the hard drive, you can occasionally delete these files.

Use the next topics to learn where the log files are located, their purpose, how to archive them, and how to save space by running the clean up command.

- [Log File Locations](#)
- [Archiving Log Files](#)
- [Removing Log Files](#)

## Log File Locations

Tableau Server log files can be found in the following folders:

### 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`

## Archiving Log Files

You can archive Tableau Server log files using the `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.

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

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\7.0\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\7.0\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 02/14/2012
```

The above command creates a zipped file named `logs.zip` that includes logs dated February 14, 2012 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.

## Removing Log Files

The `cleanup` command removes service logs and HTTP table entries older than seven days in order to save space. This will affect some of the information presented in the Tableau Server [Administrative Views](#).

At a command prompt type the following:

```
tabadmin cleanup --restart
```

**Note:**

In general you should shut down the server prior to running this command. However, if the server is running you should include the `--restart` option to ensure a successful clean up and restore.

## Handling 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 has a unique username assigned to him on the server and is required to identify himself when connecting to the server. The software may be installed on a single machine or distributed across any number of machines in a distributed server environment.

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. You may install the server on one or more machines to create a cluster, with the restrictions that the total number of cores in all the machines do not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

### 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 (x86)\Tableau\Tableau Server\7.0\bin`
2. Type the following: `tabadmin licenses`.



## Handling an Unlicensed VizQL Server Service

There are several status indicators on the Tableau Server Maintenance page that help you understand the state of Tableau Server services. An orange-color status box, "Service unlicensed", indicates that one of the VizQL server processes is unable to retrieve the Tableau Server license information.

Maintenance							
 Service waiting for request  Service standing by  Service handling request  Service unlicensed  Service down							
Status							
Machine	Repository	Data Engine	Server Web Application	VizQL Server	Data Server	Background Tasks	Web Server
101.10.101.217			  	  	  		Gateway
101.10.101.227							
101.10.101.218							

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.

## Repository Migration Error

Starting with version 7.0, Tableau Server uses a new repository type. When you upgrade to version 7.0.x, the migration to the new repository type is handled for you. However, if something unexpected occurs during this process, an error message titled “Repository Migration Error” will direct you to run the `tabadmin migrate_to_new_repository` command. To run this command:

1. [Stop the server.](#)
2. From Tableau Server’s 7.0 bin directory, type the following to migrate the repository:

```
tabadmin migrate_to_new_repository
```

If the migration is successful, no error message will display. The migration does not delete your old repository. If you want to remove it, enter the following command:

```
tabadmin migrate_to_new_repository --remove-old-repository
```

The repository won’t be re-migrated, but the old repository will be removed.

3. [Start the server.](#)

## VizQL "Out of Memory" Error

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\Log
```

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:

- **Increasing the number of VizQL processes:** This may mean that you need to add one or more workers. See [Installing Worker Servers](#) 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.

## Repository Migration Error

Starting with version 7.0, Tableau Server uses a new repository type. When you upgrade to version 7.0.x, the migration to the new repository type is handled for you. However, if something unexpected occurs during this process, an error message titled “Repository Migration Error” will direct you to run the `tabadmin migrate_to_new_repository` command. To run this command:

1. [Stop the server.](#)
2. From Tableau Server’s 7.0 bin directory, type the following to migrate the repository:

```
tabadmin migrate_to_new_repository
```

If the migration is successful, no error message will display. The migration does not delete your old repository. If you want to remove it, enter the following command:

```
tabadmin migrate_to_new_repository --remove-old-repository
```

The repository won’t be re-migrated, but the old repository will be removed.

3. [Start the server.](#)

# Copyright

©2012 Tableau Software, Incorporated and its licensors. All rights reserved.

Patent - <http://www.tableausoftware.com/ip>

Portions of the code copyright ©2002 The Board of Trustees of the Leland Stanford Junior University. All rights reserved.

For a listing of third party copyright notices please refer to the following file that is installed with Tableau Server:

- 32-bit: C:\Program Files\Tableau\Tableau Server\7.0\COPYRIGHTS.rtf
- 64-bit: C:\Program Files (x86)\Tableau\Tableau Server\7.0\COPYRIGHTS.rtf

This product includes software developed by Andy Clark.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).



# Contact

For sales inquiries please contact [sales@tableausoftware.com](mailto:sales@tableausoftware.com).

For customer support, please contact [support@tableausoftware.com](mailto:support@tableausoftware.com).

For other inquiries, please contact [info@tableausoftware.com](mailto:info@tableausoftware.com).